

October 20, 2023

Multiple Reference Frame 기법을 이용한 KP505 프로펠러의 수치 해석

김한준 ¹

¹서울대학교 조선해양공학과

목차

I. 연구 개요

II. 이론적 배경

III. 해석 결과 및 분석

IV. 결론 및 향후 연구

목차

I. 연구 개요

II. 이론적 배경

III. 해석 결과 및 분석

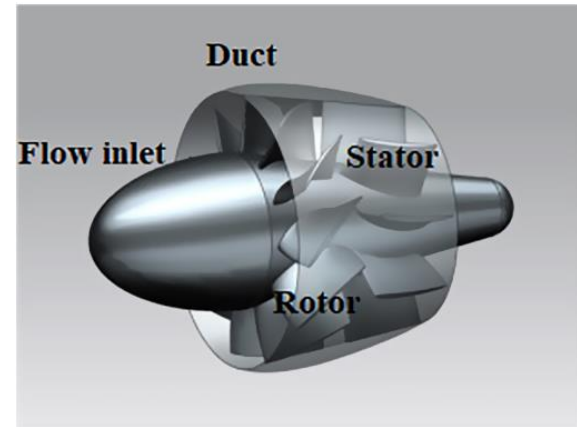
IV. 결론 및 향후 연구

연구 배경 및 목적

- 펌프젯 추진 방식
 - 고정부인 Stator와 회전부인 Rotor로 구성
 - 기존의 프로펠러 추진 방식에 비해 캐비테이션 및 소음 성능이 우수
 - 최근 잠수함에 펌프젯 추진 방식을 적용하여, 잠수함의 성능을 개선하려는 연구가 활발하게 수행 되고 있음
- 회전부의 운동 모사를 위해 프로펠러로 선행 연습 수행



< Installation for the self-propulsion test* >



< Pump-jet propeller model >

*Ahn, J. W., Seol, H. S., Jung, H. S., & Park, Y. H. (2022). Study of the Self-Propulsion Test and Analysis for a Pumpjet Propulsor in LCT .
Journal of the Society of Naval Architects of Korea, 59(5), 271-279.

선형연구

- 선체로 인한 프로펠러의 추진 성능 변화에 대한 연구
 - 모형선 scale과 실선 scale의 비교 수행
 - STAR-CCM+ 으로 해석 수행
 - 모형선 scale의 Propeller Open Water Test 수행 시 MRF(Multiple Reference Frame) 기법 사용

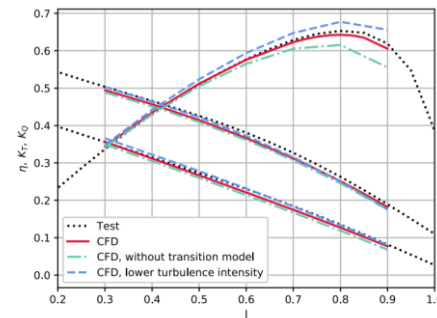
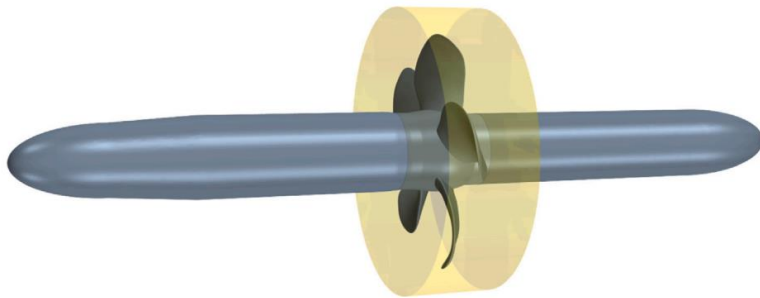


Fig. 7. Propeller A, model-scale test data vs. CFD predictions.

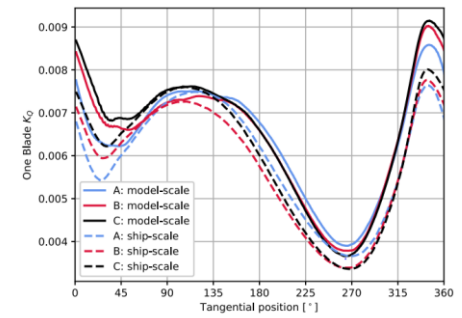


Fig. 11. CFD results for K_q variation in the wake.

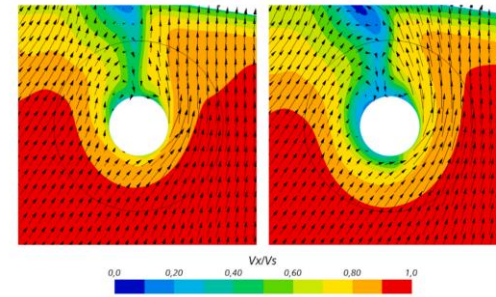
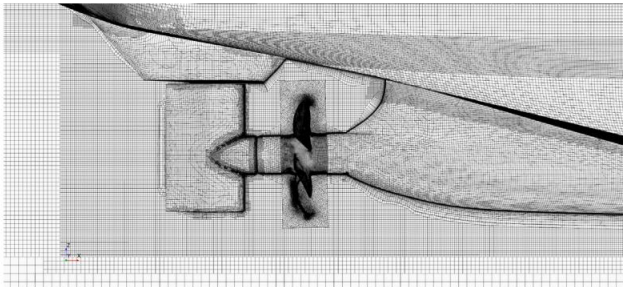


Fig. 10. Contour plots of nominal axial velocity (axial velocity normalized with V_0) with vectors for transverse velocities at propeller plane in ship-scale (left) and model-scale (right).

Andersson, J., Gustafsson, R., Johansson, R., & Bensow, R. E. (2022). Propeller-hull interaction beyond the propulsive factors—A case study on the performance of different propeller designs. *Ocean Engineering*

연구 목표 및 내용

■ 연구 목표

- MRF 기법을 사용하여 KP505 프로펠러의 회전 운동 모사 기법 제시
- 제시한 회전 운동 모사 기법으로 프로펠러 성능 추정

■ 연구 내용

- MRF 구역에서 cyclicAMI와 cyclic 경계조건의 차이 비교
 - 간단한 예시 케이스를 구성하여 해석 수행 후 비교
 - MRF 기법의 코드 분석
- 제시한 회전 운동 모사 기법을 사용하여 KP505의 성능 해석 수행
 - KRSIO에서 제시한 실험 데이터와 해석 결과의 비교

목차

I. 연구 개요

II. 이론적 배경

III. 해석 결과 및 분석

IV. 결론 및 향후 연구

지배 방정식

▪ Continuity & Navier-Stokes equation

- Continuity equation

- 비압축성 유동($\rho = \text{const.}$)에서 continuity equation

$$\frac{\partial u_i}{\partial x_i} = 0$$

- Navier-Stokes equation

- 비압축성 유동($\rho = \text{const.}$)에서 Navier-Stokes equation

$$\frac{\partial u_j}{\partial t} + \frac{\partial u_i u_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} - \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_i}$$

- Newtonian 유체 가정(유체 점성에 의한 전단 응력은 유체 변화률에 비례)

$$\tau_{ij} = -\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{2}{3} \mu \left(\frac{\partial u_i}{\partial x_j} \right) \delta_{ij} \quad \delta_{ij} : \text{Kronecker delta}$$

$= 0$ (\because continuity eqn.)

- 비압축성 Newtonian 유체의 운동량 보존 방정식

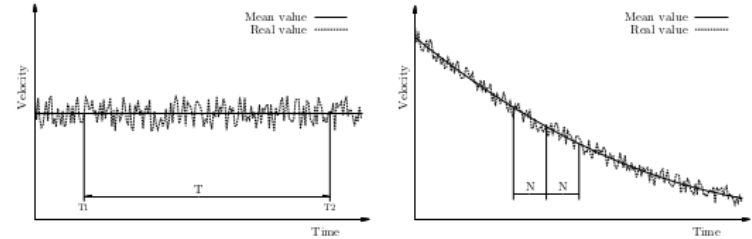
$$\frac{\partial u_j}{\partial t} + \frac{\partial u_i u_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} + \nu \frac{\partial}{\partial x_i} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

RANS 방정식

■ RANS (Reynolds averaged Navier-Stokes) equation

- 무질서한 난류 유동을 모사하기 위한 기법으로 유체 성분을 Reynolds 평균 성분과 요동 성분으로 분해

$$\bar{u}_i = \frac{1}{\delta} \int_{t_0}^{t_0+\delta t} u_i dt \quad \bar{p} = \frac{1}{\delta} \int_{t_0}^{t_0+\delta t} p dt \quad \Rightarrow \quad \begin{aligned} u_i &= \bar{u}_i + u'_i \\ p &= \bar{p} + p' \end{aligned}$$



- RANS equation

$$\frac{\partial \bar{u}_j}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} + \underbrace{\frac{\partial \overline{u'_i u'_j}}{\partial x_i}}_{= \text{Reynolds stress}} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_j} + \nu \frac{\partial}{\partial x_i} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$$

- RANS 방정식 상에 나타나는 Reynolds stress 성분을 추정하기 위해 Boussinesq approximation 적용

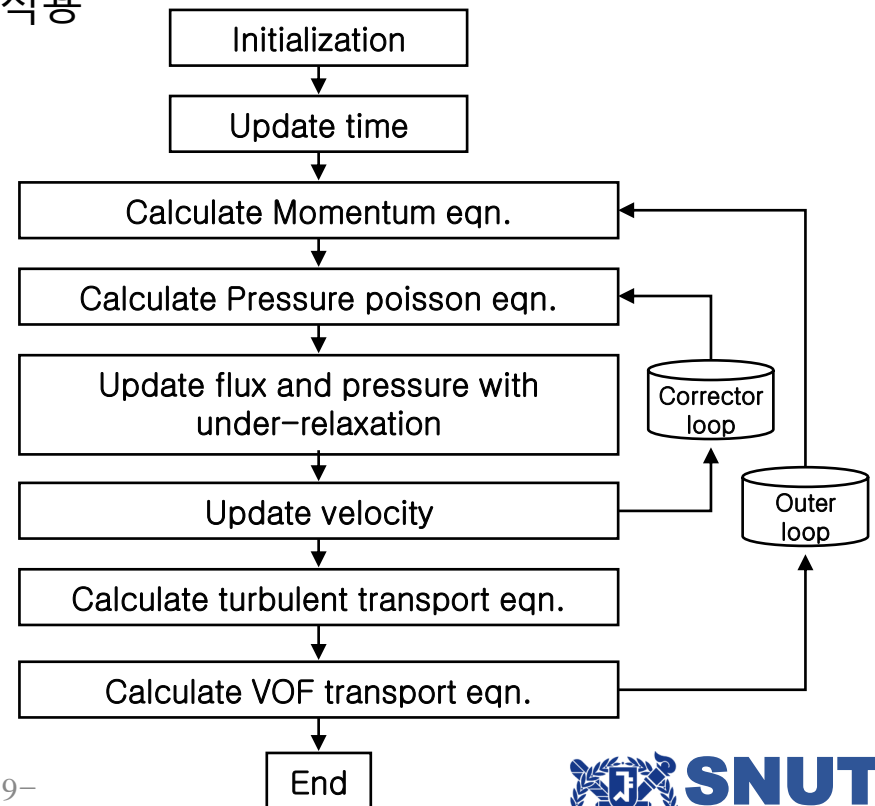
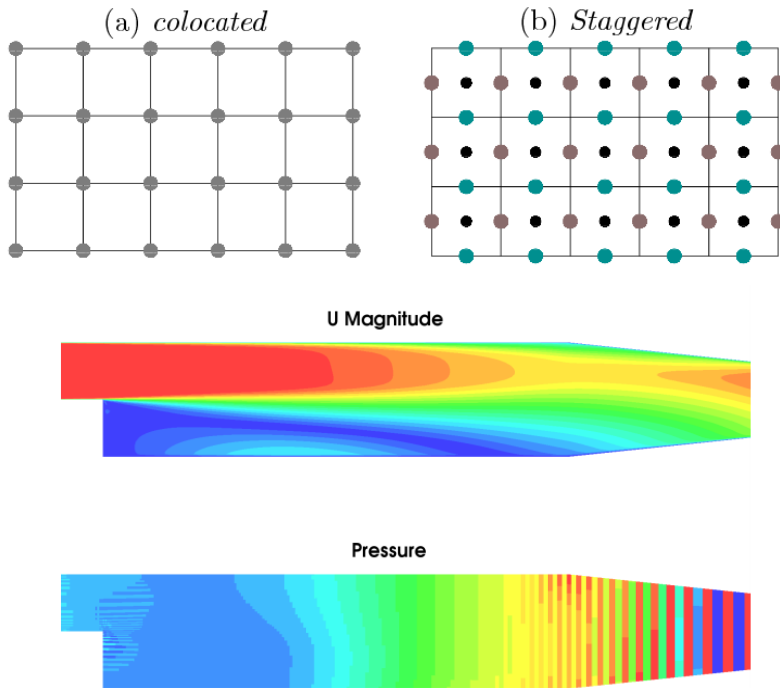
$$\frac{\partial \bar{u}_j}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_j} + (\nu + \nu_t) \frac{\partial}{\partial x_i} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial k}{\partial x_i} \delta_{ij}$$

- RANS 난류 모델로는 2 equations 모델 중 하나인 $k - \omega$ SST 모델 사용

속도-압력 연성 기법

■ 속도-압력 연성

- 비압축성 유동의 해를 구하기 위해 SIMPLE (semi-implicit method for pressure linked equations) 계열 알고리즘인 Merged PISO-SIMPLE 알고리즘 사용
 - 비압축성 유체의 비정상 유동 해석 기법
 - 비엇갈림 격자(collocated mesh)를 사용할 경우 발생하는 checker board 현상을 해결하기 위해 Rhie-Chow treatment 적용



MRF(Multiple Reference Frame) 기법

- 영역마다 서로 다른 관성좌표계를 기준으로 CFD 해석을 수행하는 기법
- 회전 좌표 기준 지배 방정식 (비압축성 유동)

- 연속 방정식

$$\nabla \cdot \vec{u}_I = \nabla \cdot (\vec{u}_R + \vec{\Omega} \times \vec{r}) = \nabla \cdot \vec{u}_R + \overbrace{\nabla \cdot (\vec{\Omega} \times \vec{r})}^{=0} = 0$$

$$\therefore \nabla \cdot \vec{u}_R = 0$$

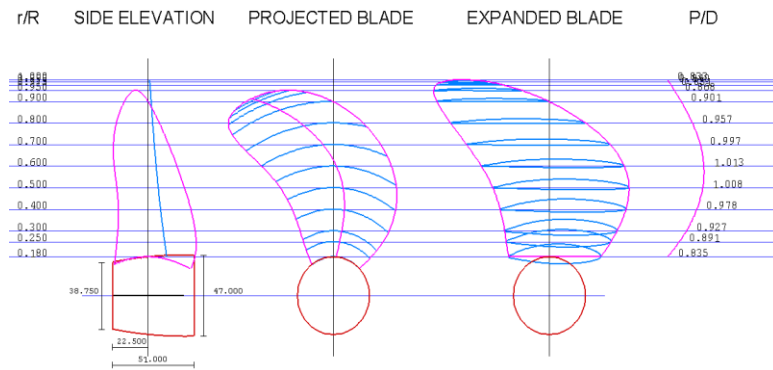
- Navier-Stokes 방정식

$$\begin{aligned} \nabla \cdot (\vec{u}_R \vec{u}_R) + 2\vec{\Omega} \times \vec{u}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} &= \nabla \cdot (\vec{u}_R \vec{u}_I) - \vec{\Omega} \times \vec{u}_R + 2\vec{\Omega} \times \vec{u}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} \\ &= \nabla \cdot (\vec{u}_R \vec{u}_I) + \vec{\Omega} \times \vec{u}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} \\ &= \nabla \cdot (\vec{u}_R \vec{u}_I) + \vec{\Omega} \times \overbrace{(\vec{u}_R + \vec{\Omega} \times \vec{r})}^{\vec{u}_I} \end{aligned}$$

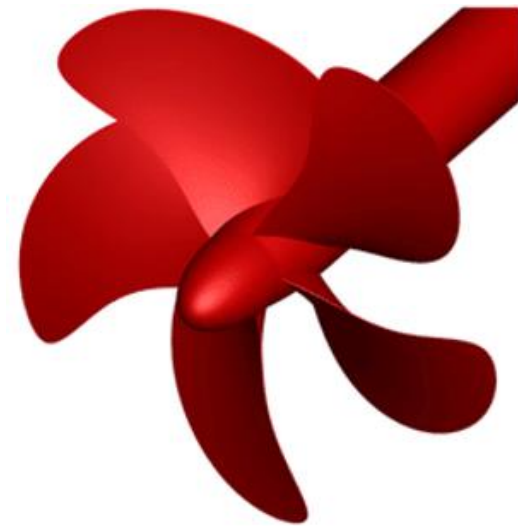
$$\therefore \nabla \cdot (\vec{u}_R \vec{u}_I) + \vec{\Omega} \times \vec{u}_I = -\frac{\nabla p}{\rho} + \nabla \cdot (v_{eff}(\nabla \vec{u}_I + (\nabla \vec{u}_I)^T))$$

해석 대상

- KP505 프로펠러
 - KRISO에서 NACA66 단면을 사용하여 설계한 프로펠러
 - 31.6 축척비의 모형으로 모형의 지름은 250mm
 - 날개 수: 5개
 - KCS Hull 추진용으로 KRISO에서 사용
- 주요 제원은 아래와 같음



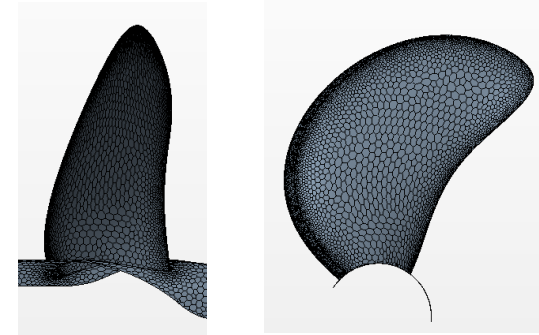
Propeller Principal Particulars					
Diameter(mm)	7900.0	Model Diam(mm)	250.000	Scale Ratio	31.6000
(P/D) mean	0.9500	(Rake/D) Tip	0.0000	Prop. Type	FFP
Ae/Ao	0.8002	Eff. skew(Deg)	32.00	Drawing Scale	1.80556
Hub Ratio	0.1800	(C/D) 0.7R	0.3590	Comment	
No. of Blade	5	(to/C) 0.7R	0.0140	Prop. Number	KP505
Section	NACA66	(b/D) 0.7R	0.0149		



해석 도메인

■ 계산 영역

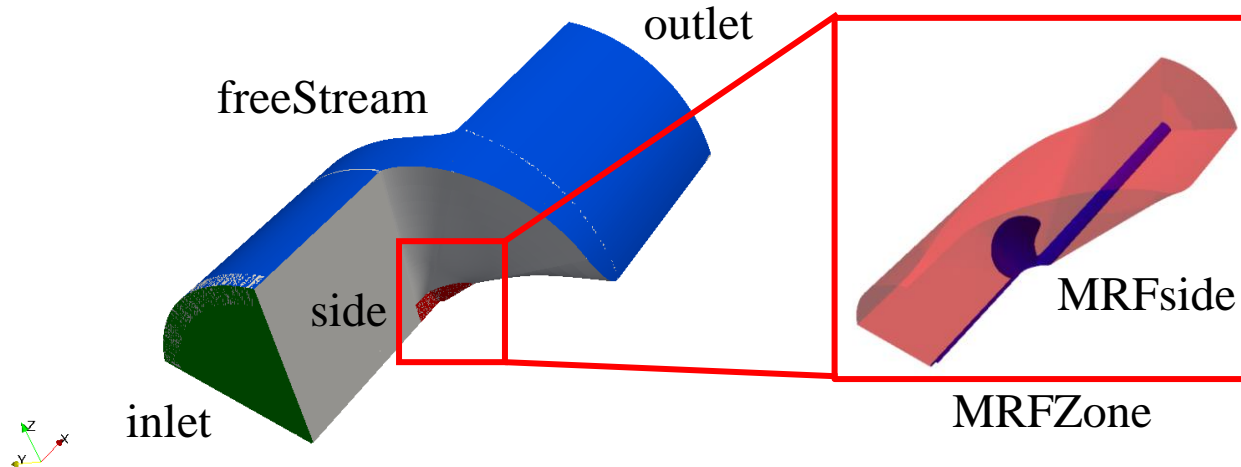
- 외부 정적 영역: $-6D < X < 8D, R = 5D$
- 내부 회전 영역: $-D < X < 2D, R = 0.7D$
- YZ 평면에서의 단면 각도를 72° 로 구성



< Surface mesh of KP505 >

■ 격자 생성

- 외부 정적 영역: 정렬 격자, 약 60만개
- 내부 회전 영역: 비정렬 격자, 약 40만개



<Computational domain for KP505 POW test>

해석 경계조건

- 수치해석 기법
 - Time discretization scheme
 - Euler
 - Spatial discretization scheme
 - Gradient scheme: Linear
 - Divergence scheme: Second-order upwind
 - Pressure-velocity coupling
 - Merged PISO and SIMPLE Algorithm
- 경계 조건은 아래 표에 정리하여 나타냄

Outer	Velocity	Pressure
Inlet	Dirichlet	Neumann
Outlet	Neumann	Neumann
freeStream	symmetry	symmetry
Shaft	No slip	Neumann
Side	Periodic	Periodic

<Boundary conditions for Outer domain of KP505 POW test>

Inner	Neumann	Neumann
MRFSide	Periodic	Periodic
Hub	No slip	Neumann
Blades	No slip	Neumann

<Boundary conditions for MRFZone of KP505 POW test>

목차

I. 연구 개요

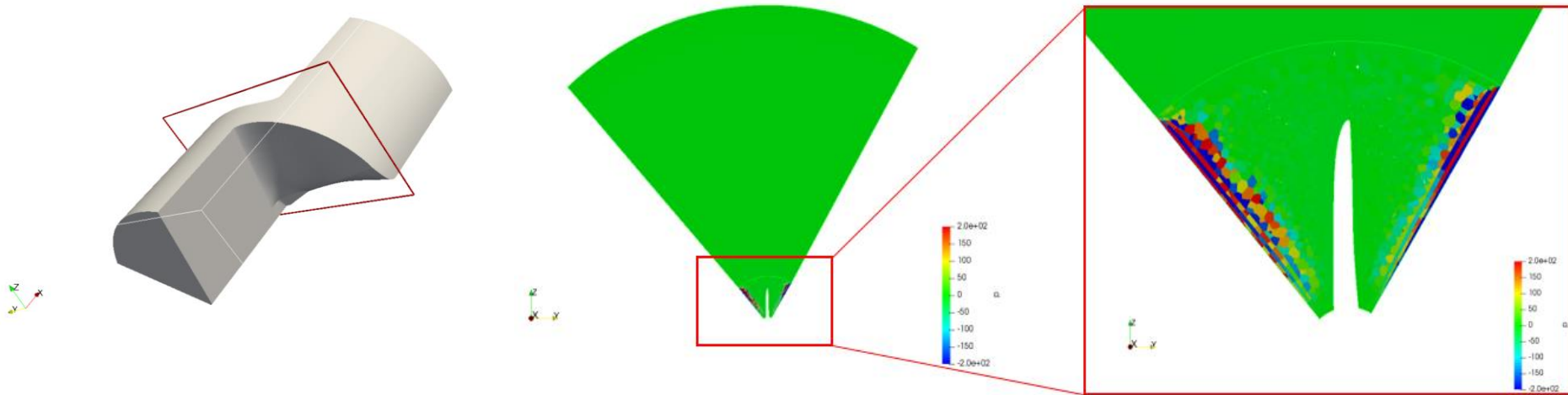
II. 이론적 배경

III. 해석 결과

IV. 요약 및 결론 및 향후 연구

도메인 단면의 압력 분포 추정

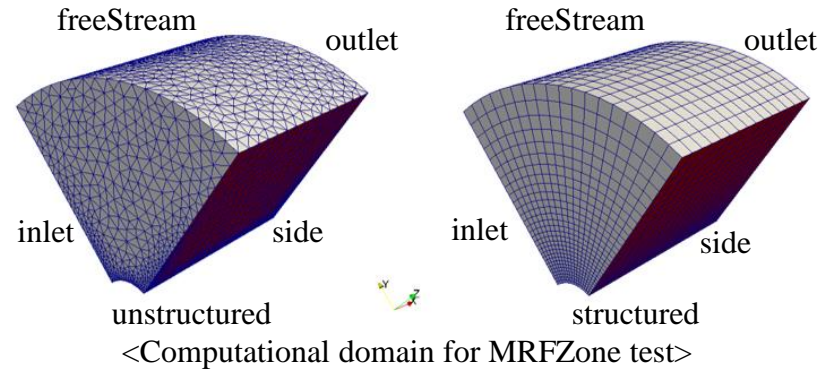
- 전진비 $J=0.5$ 에 대하여 해석 수행
- 도메인 단면의 압력 분포 추정 결과 checkerboard 현상 발견
 - 내부 회전 구역에서 checkerboard 현상이 나타남
 - 외부 정적 구역에서는 checkerboard 현상이 나타나지 않음
- 내부와 외부에서 격자의 종류 및 사용한 주기경계조건 차이가 존재
 - 내부 회전 구역: 비정렬 격자, cyclicAMI 경계조건
 - 외부 정적 구역: 정렬 격자, cyclic 경계조건



<도메인 단면에서의 압력 분포>

예시 케이스 구성

- 도메인 내부에 프로펠러가 없는 상황으로 간단한 예시 케이스 구성
 - 격자 수: 정렬격자 약 1만개, 비정렬격자 약 4만개
 - 도메인 전체를 MRF 구역으로 설정
- 경계조건 및 격자 종류에 따른 해석 수행
 - 경계조건: cyclicAMI, cyclic
 - 우측 도메인의 빨간 면에 경계조건 부여
 - 격자 종류: 정렬격자, 비정렬 격자
- 수치해석 기법
 - Time discretization scheme
 - Euler
 - Spatial discretization scheme
 - Gradient scheme: Linear
 - Divergence scheme: Second-order upwind
 - Pressure-velocity coupling
 - Merged PISO and SIMPLE Algorithm
- 경계조건은 오른쪽 표에 정리하여 나타냄

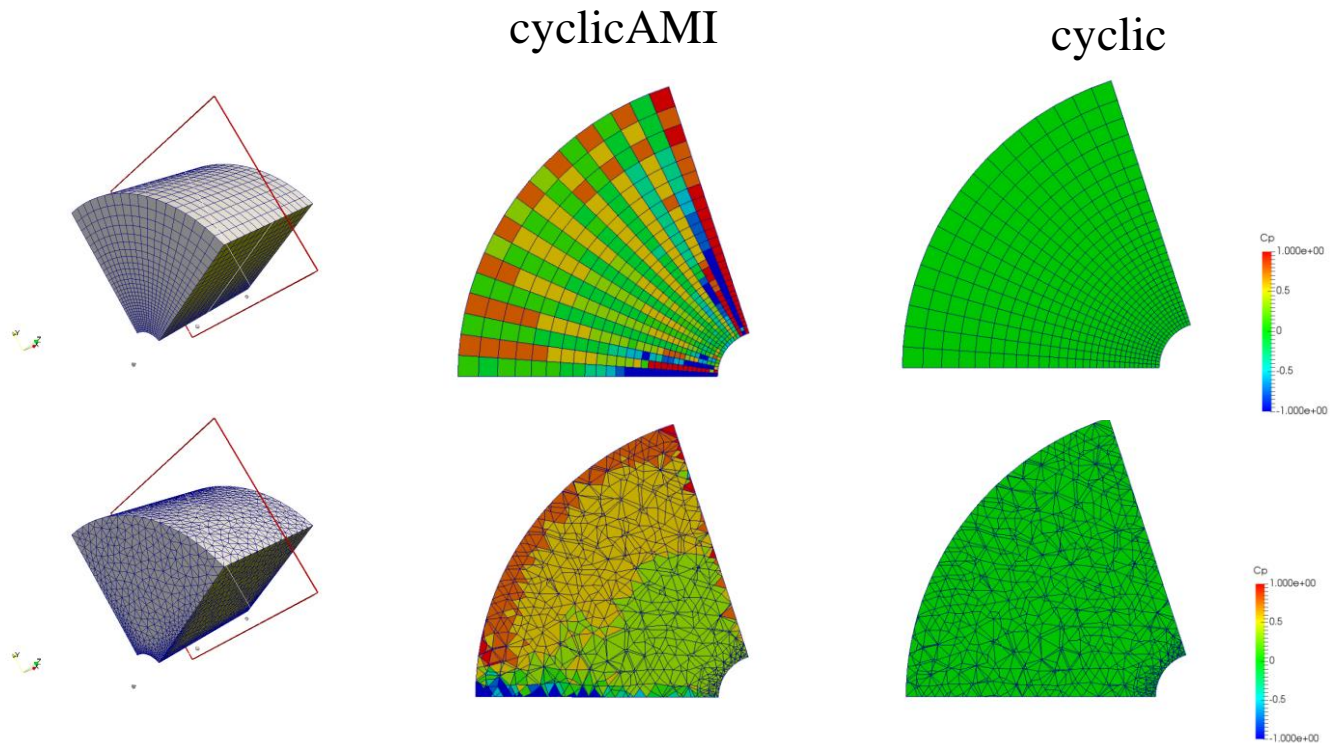


<Boundary conditions for MRFZone test>

	Velocity	Pressure
Inlet	Dirichlet	Neumann
Outlet	Dirichlet	Neumann
Side	periodic	periodic
Shaft	No slip	Neumann
freeStream	symmetry	symmetry

해석 결과

- 도메인 중앙 단면의 압력 분포를 관찰
- cyclicAMI 경계조건에서는 checkerboard 또는 발산이 나타남
- cyclic 경계조건에서는 checkerboard가 나타나지 않음



pimpleFoam에서의 MRF

- Momentum eqn을 푸는 과정에서 MRF.DDt(U)를 더해줌
 - MRF.DDt(U)는 addCoriolis를 수행
 - addCoriolis는 MRF 구역에 해당되는 cell에 $\vec{\Omega} \times \vec{u}_I$ 값을 더해줌

$$\nabla \cdot (\vec{u}_R \vec{u}_I) + \vec{\Omega} \times \vec{u}_I = -\frac{\nabla p}{\rho} + \nabla \cdot (v_{eff}(\nabla \vec{u}_I + (\nabla \vec{u}_I)^T))$$

```

1 // Solve the Momentum equation
2 MRF.correctBoundaryVelocity(U);
3
4 tmp<fvVectorMatrix> tUEqn
5 {
6     fvm::ddt(U) + fvm::div(phi, U)
7     + MRF.DDt(U)
8     + turbulence->divDevReff(U)
9     ==
10    fvOptions(U)
11 };
12 fvVectorMatrix& UEqn = tUEqn.ref();
13 UEqn.relax();
14
15 fvOptions.constrain(UEqn);
16
17 if (pimple.momentumPredictor())
18 {
19     solve(UEqn == -fv::grad(p));
20     fvOptions.correct(U);
21 }
22
23
24
    
```

UEqn.H

```

164 Foam::tmp<Foam::volVectorField> Foam::MRFZoneList::DDt
165 (
166     const volVectorField& U
167 ) const
168 {
169     tmp<volVectorField> tacceleration
170     (
171         volVectorField::New
172         (
173             "MRFZoneList:acceleration",
174             U.mesh(),
175             dimensionedVector(U.dimensions()/dimTime, Zero)
176         )
177     );
178     volVectorField& acceleration = tacceleration.ref();
179
180     forAll(*this, i)
181     {
182         operator[](i).addCoriolis(U, acceleration);
183     }
184
185     return tacceleration;
186 }
    
```

MRFZoneList.C

```

310 void Foam::MRFZone::addCoriolis
311 (
312     const volVectorField& U,
313     volVectorField& ddtU
314 ) const
315 {
316     if (cellZoneID_ == -1)
317     {
318         return;
319     }
320
321     const labelList& cells = mesh.cellZones()[cellZoneID_];
322     vectorField& ddtUc = ddtU.primitiveFieldRef();
323     const vectorField& Uc = U;
324
325     const vector Omega = this->Omega();
326
327     forAll(cells, i)
328     {
329         label celli = cells[i];
330         ddtUc[celli] += (Omega ^ Uc[celli]);
331     }
332 }
    
```

MRFZone.C

pimpleFoam에서의 MRF

- MRF 구역을 읽어올 때 nonRotatingPatches는 excludedPatch에 할당됨
- Pressure poisson eqn을 푸는 과정에서 excludedPatch의 patchFace에서 $\vec{\Omega} \times \vec{r}$ 를 빼줌
- cyclicAMI에서 \vec{u}_R 을 interpolate하는 과정 중 오류 발생

```
1 volScalarField rAU(1.0/UEqn.A());
2 volVectorField HbyA(constrainHbyA(rAU*UEqn.H(), U, p));
3 surfaceScalarField phiHbyA
4 {
5     "phiHbyA",
6     fvc::flux(HbyA)
7     + MRF.zeroFilter(fvc::interpolate(rAU)*fvc::ddtCorr(U, phi, Uf))
8 };
9
10 MRF.makeRelative(phiHbyA);
11
12 if (p.needReference())
13 {
14     fvc::makeRelative(phiHbyA, U);
15     adjustPhi(phiHbyA, U, p);
16     fvc::makeAbsolute(phiHbyA, U);
17 }
18
```

pEqn.H

```
236 Foam::MRFZone::MRFZone
237 (
238     const word& name,
239     const fvMesh& mesh,
240     const dictionary& dict,
241     const word& cellZoneName
242 ) :
243 {
244     mesh_(mesh),
245     name_(name),
246     coeffs_(dict),
247     active_(coeffs_.lookupOrDefault("active", true)),
248     cellZoneName_(cellZoneName),
249     cellZoneID_(),
250     excludedPatchNames_
251     (
252         wordReList(coeffs_.lookupOrDefault("nonRotatingPatches", wordReList()))
253     ),
254     origin_(coeffs_.lookup("origin")),
255     axis_(coeffs_.lookup("axis")),
256     omega_(Function1<scalar>::New("omega", coeffs_))
257 }
```

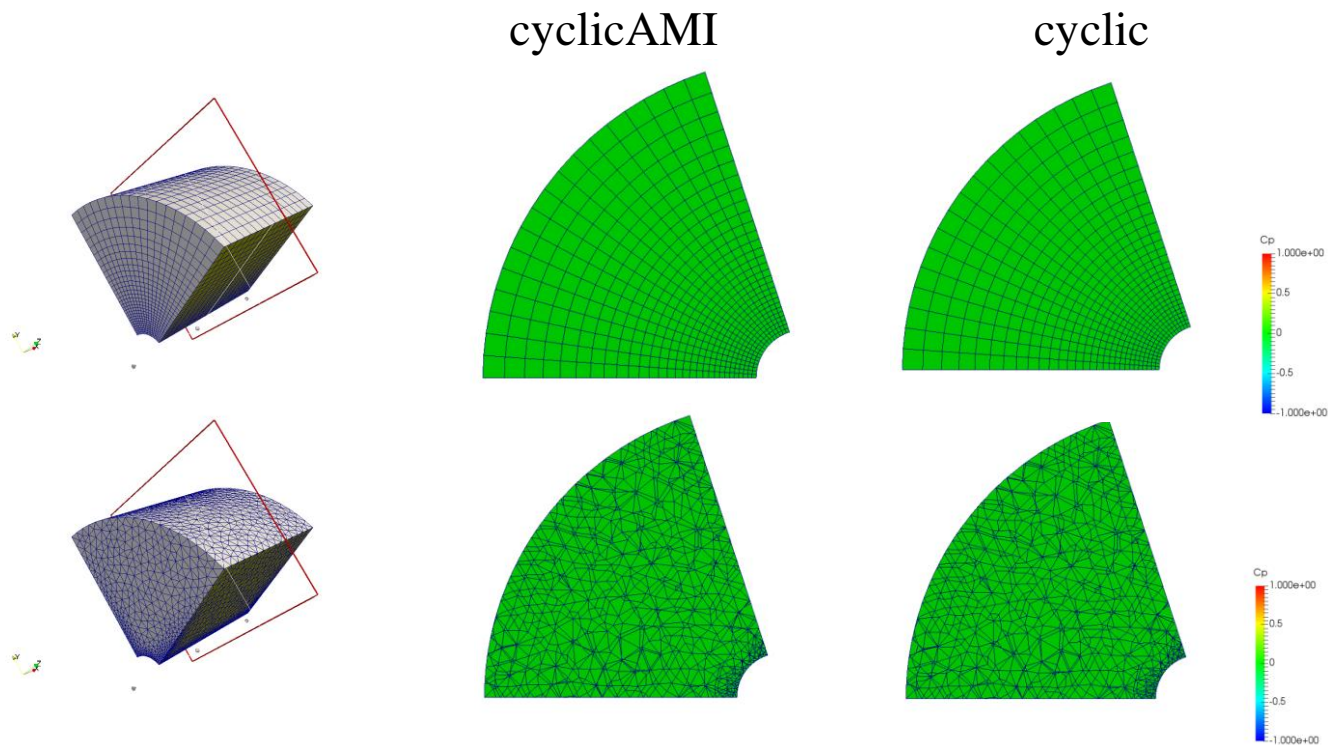
MRFZone.C

```
406 void Foam::MRFZone::makeRelative(volVectorField& U) const
407 {
408     if (cellZoneID_ == -1)
409     {
410         return;
411     }
412
413     const volVectorField& C = mesh_.C();
414
415     const vector Omega = this->omega();
416
417     const labelList& cells = mesh_.cellZones()[cellZoneID_];
418
419     forAll(cells, i)
420     {
421         label celli = cells[i];
422         U[celli] -= (Omega ^ (C[celli] - origin_));
423     }
424
425     // Included patches
426
427     volVectorField::Boundary& Ubf = U.boundaryFieldRef();
428
429     forAll(includedFaces_, patchi)
430     {
431         forAll(includedFaces_[patchi], i)
432         {
433             label patchFacei = includedFaces_[patchi][i];
434             Ubf[patchi][patchFacei] = Zero;
435         }
436     }
437
438     // Excluded patches
439     forAll(excludedFaces_, patchi)
440     {
441         forAll(excludedFaces_[patchi], i)
442         {
443             label patchFacei = excludedFaces_[patchi][i];
444             Ubf[patchi][patchFacei] -=
445                 (Omega
446                  ^ (C.boundaryField()[patchi][patchFacei] - origin_));
447         }
448     }
449 }
```

MRFZone.C

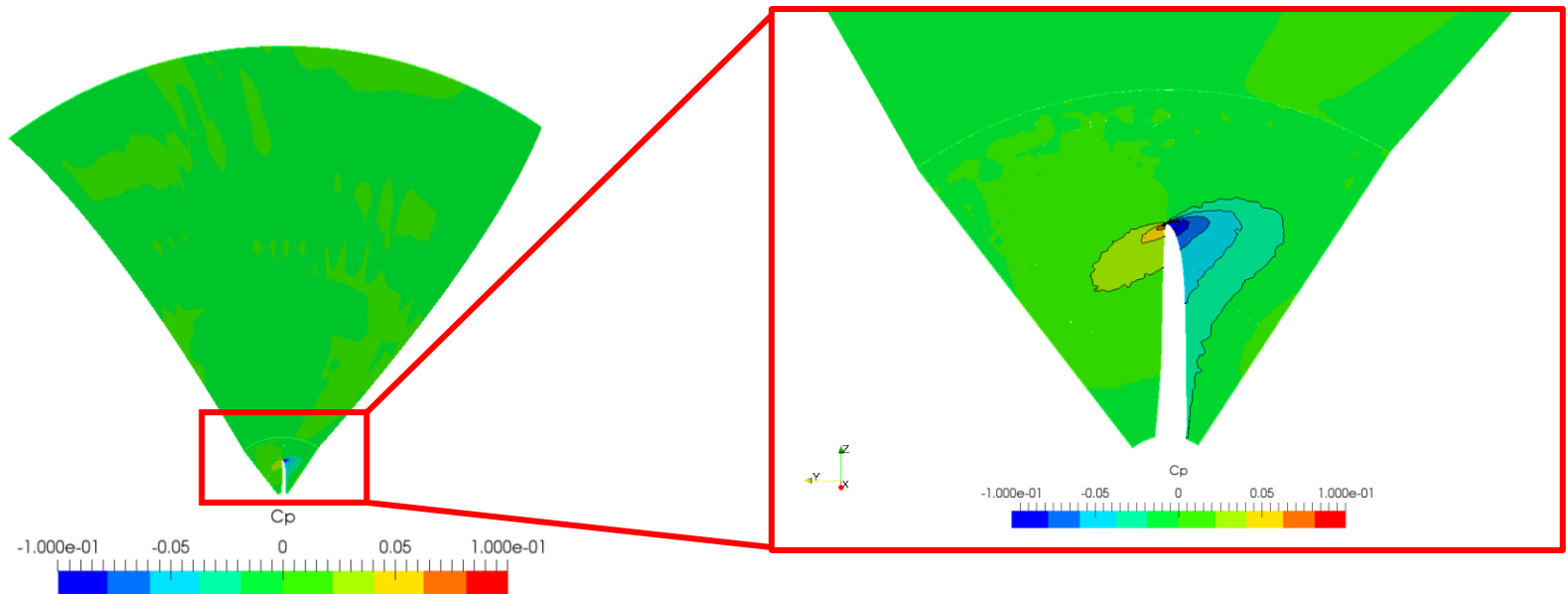
nonRotatingPatches 추가

- MRF 구역의 nonRotatingPatches에 주기 경계조건 추가 이후 해석 수행
 - 다른 조건들은 전부 동일하게 설정
- cyclicAMI, cyclic 경계조건 모두 checkerboard 및 발산하는 현상이 나타나지 않음을 확인



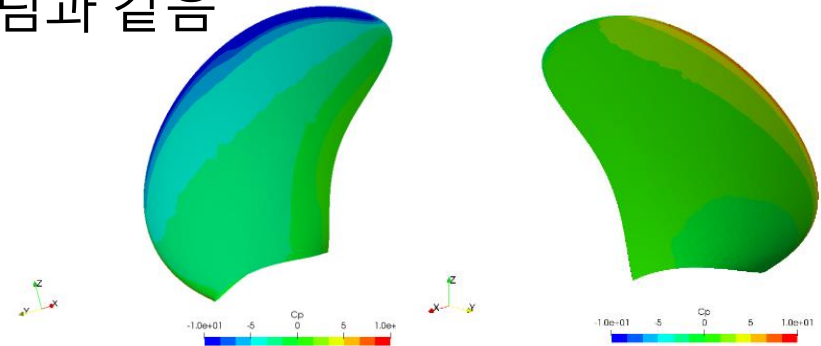
해석 결과

- 내부 회전 구역의 주기경계조건을 nonRotatingPatches에 포함하여 해석 수행
 - 해석 도메인 및 경계조건은 동일
 - 전진비 $J=0.5$ 인 케이스에 대하여 해석 수행
 - 주기경계조건에서 checkerboard가 나타나지 않음을 확인



KP505 성능 해석

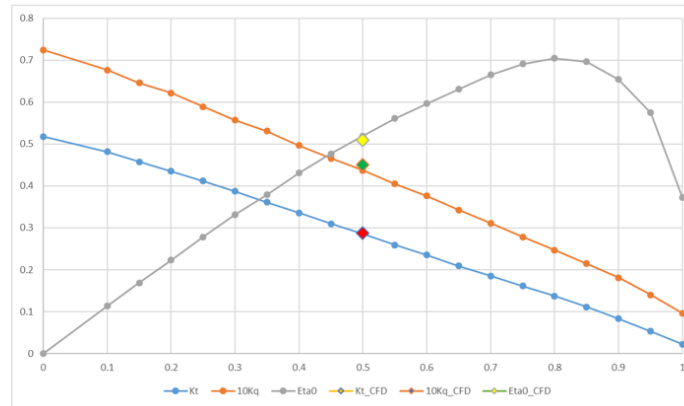
- Blade에서의 압력 분포는 오른쪽 그림과 같음



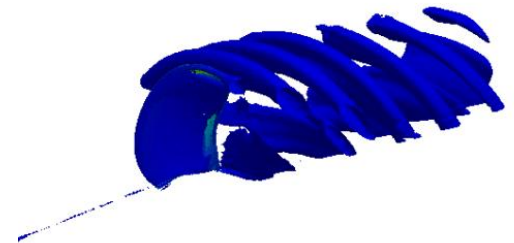
<Cp contour of KP505 POW test at J=0.5>

- KRISO의 모형 시험 결과와 비교한 결과 좌측 하단의 표와 같음
- Vorticity를 이용하여 q criteria를 도시화한 결과 우측 하단 그림과 같음

J=0.5	K_T	10K_Q	eta_0
EFD	0.285	0.437	0.519
CFD	0.2875	0.4495	0.509
DIFF(%)	0.88	2.86	-1.93



<KP505 POW model test data vs CFD>



<q criteria of KP505>

목차

I. 연구 개요

II. 이론적 배경

III. 해석 결과 및 분석

IV. 결론 및 향후 연구

결론 및 향후 연구

- MRF 기법을 사용하여 프로펠러의 회전 운동을 구현함
 - MRF 구역 내부에 cyclicAMI 경계조건이 있을 경우 이를 nonRotatingPatches로 설정하면 checkerboard를 방지할 수 있음
 - 해석자를 바탕으로 KP505의 성능 추정 수행
- 향후 연구
 - MRF에서 cyclicAMI의 interpolation 방식 수정
 - MPM (mixing plane method)를 이용한 rotor 회전운동 구현
 - 고정부인 stator의 도메인과 회전부인 rotor 사이 mixing plane 구성

감사합니다.