해양 CFD 해석을 위한 OpenFOAM 개발내용 소개

연성모

삼성중공업 조선해양연구소 선박해양연구센터

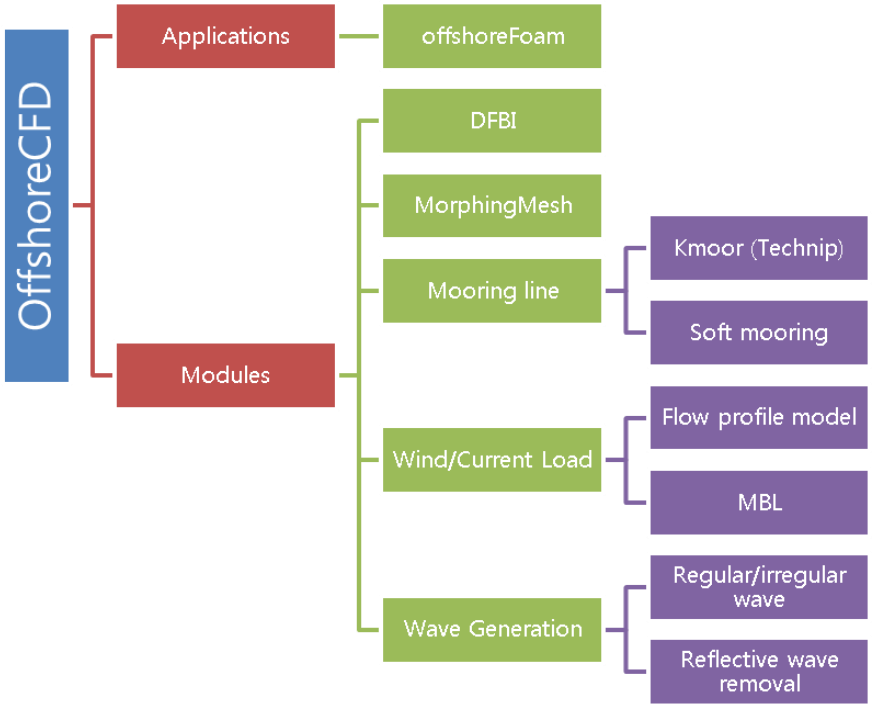
2019. 9. 26

Agenda

- OpenFOAM 버전별 병렬성능비교
- Wind load solution
- Global performance solution
- Conclusion and path forward

OffshoreCFD

- 해양 CFD의 특징
  - Unsteady
  - Fluid structure interaction by waves
    - Single body motion
    - Tow-body or multi-body relative motions
- 해양 CFD 대상
  - Wind/current load
  - VIM
    - Low-frequency motion
  - Global performance
    - 3hr in full-scale for irregular waves



OpenFOAM 벤치마크

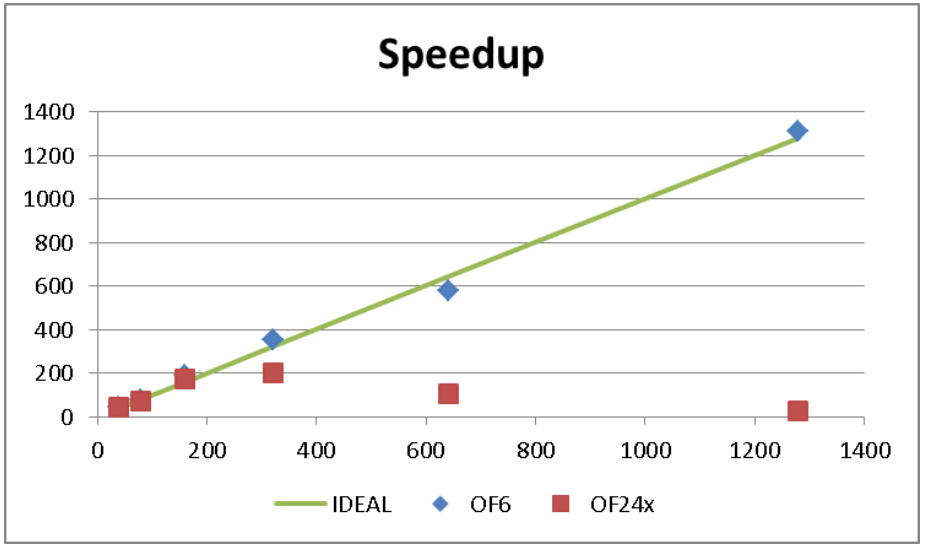
- Test Case
  - DTCHull tutorial case
    - morphingMesh
    - Two-phase flow
  - Mesh size
    - Abt. 28 Mil.
  - Unsteady
    - 초기 20 time step까지만 진행
- OpenFOAM 버전
  - OpenFOAM 2.4.x (2015년 release)
  - OpenFOAM 6 (2018년 release)

OpenFOAM 벤치마크

	Spec.
Nodes	32
Cores	1280 (40 cores/node)
Memory	192 GB/node
Storage	60 TB (user space) + 60 TB (scratch)
CPU	Intel Xeon Gold 6148 (SKL) 2.40 GHz
File System	Ext4 + gpfs
Interconnection	Infiniband EDR(100Gb/s)
OS	CentOS 7.3
Job Scheduler	Slurm 18.8

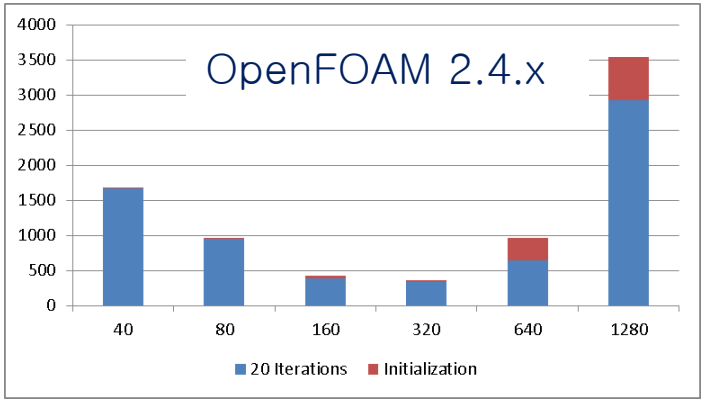
OpenFOAM 벤치마크

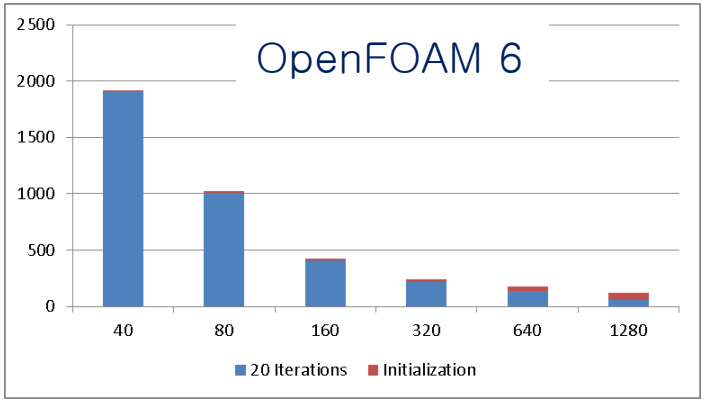
Cores	OpenFOAM 2.4.x	OpenFOAM 6
40	100%	100%
80	88%	95%
160	107%	117%
320	62%	111%
640	16%	90%
1280	2%	103%



OpenFOAM 벤치마크

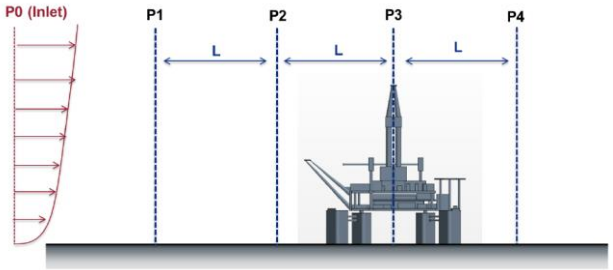
- OpenFOAM 2.4.x
  - 초기화 시간이 320 cores 이상에서 급증
  - Core간 통신 비용 또한 320 cores 이상에서 급증
- OpenFOAM 6
  - 초기화 시간 큰 폭 개선
  - 병렬 연산 성능 큰 폭 개선
- 관련 근거
  - 2016년 일본 HPC 京 OpenFOAM 지원보고서 (RIST, 井上 義昭)
  - OpenFOAM 4부터 RIST patch (RIST_A2A) 적용
  - Core간 정보를 matrix 형태에서 list 형태로 수정
    - O(N^2)에서 O(N) 복잡도로 개선한 형태

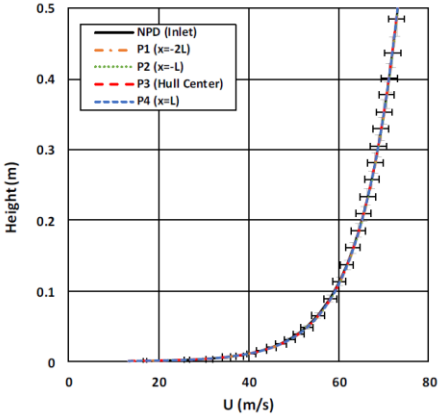




Wind Profile Implementation

- Numerical Issues
  - Sustainability of wind profile
    - Keep shape of profile in any location downstream
    - Preserve turbulence quantities (k, e, w)
  - Minimize computational domain
    - Minimize/remove fetch length (developing zone)





Wind Profile Implementation

- Previous studies on HHTBL implementation
  - Richard and Hoxey (1993)
    - The first successful implementation for a horizontally homogeneous turbulence boundary layer (HHTB)
    - Derivation of analytical boundary conditions (RH conditions) for sustainable wind profile
      - » Preservation of turbulence quantities as well as velocity profile
    - Bridge relation of CFD roughness model to roughness length
  - Richard and Norris (2011)
    - Studied numerical issue on implementing HHTBL with commercial S/W
  - SNAME OC-8 “Guidance on Wind Technologies”
    - Application of NPD profile to HHTBL
    - Analytic derivations of experimental variables (roughness length, roughness height) for NPD profile
    - Extension to typical RANS turbulence model (k-e, kw and Spallart-Almaras) of RH conditions

RH Conditions for NPD Profile

- NPD profile

$$U(z) = \frac{u^*}{\kappa} \cdot \ln\frac{z}{z_0} \equiv U(z) = U(z_{ref}) \cdot \left(1 + A \cdot \ln\frac{z}{z_{ref}}\right) \text{ with } A = 0.0573\sqrt{1 + 0.15 \cdot U(z_{ref})}$$

$$z_0 = z_{ref} \cdot Exp\left[-\frac{1}{A}\right] \text{ and } u^* = A \cdot \kappa \cdot U(z_{ref})$$

- Turbulence profiles

$$k(z) = \frac{u^{*2}}{\sqrt{\beta^*}}, \varepsilon(z) = \frac{u^{*3}}{\kappa(z+z_0)}, \omega(z) = \frac{u^*}{\kappa\sqrt{\beta^*}(z+z_0)}$$

- Roughness height (sand grain height)
  - Apply no-slip boundary condition on the bottom (mean sea level) with standard roughness wall-function
  - Roughness height (sand grain height) relation to roughness length

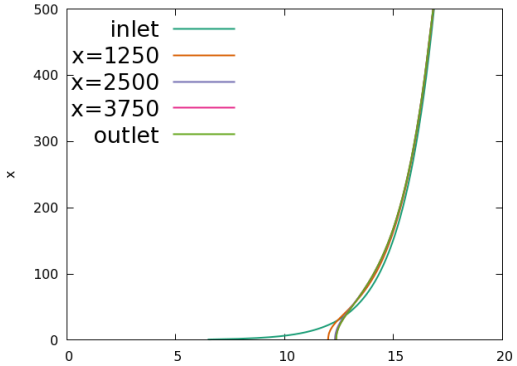
$$r = \frac{Ez_0}{C} - \frac{B\nu}{Cu^*} \approx 35.6z_0$$

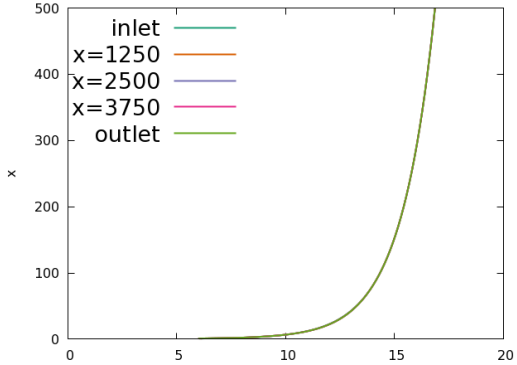
Wind Profile Comparison

Without RH conditions

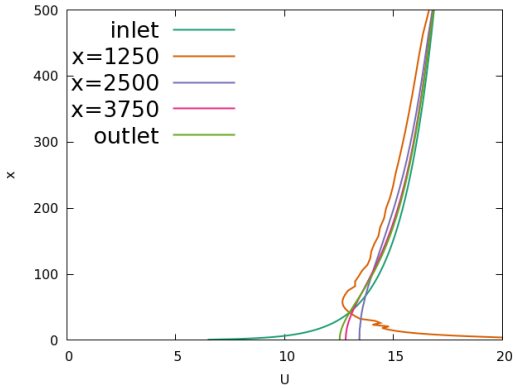
With RH conditions

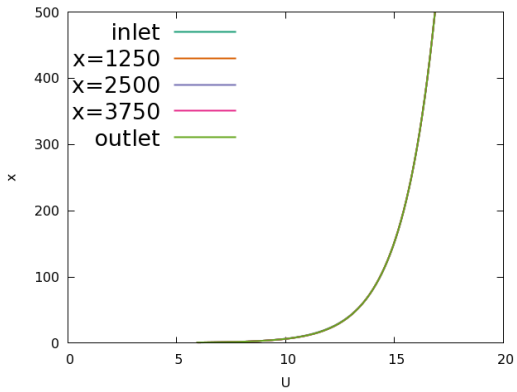
K-e model





K-w model





Fluid structure interaction

- OpenFOAM 제공 기능
  - sixDoFRigidBodyMotionSolver
    - sixDoFRigidbodyMotion 기반
  - rigidBodyMeshMotion
    - rigidBodyDynamics 기반

- Rigid body state monitoring 기능 부재
  - sixDoFRigidBodyState functionObject 개발 및 OpenFOAM.org에 제공
    - OpenFOAM 6부터 반영 및 OpenFOAM 8에 개선내용 반영 내용
  - rigidBodyState functionObject 개발 및 OpenFOAM.org에 제공
    - OpenFOAM 7 부터 반영

sixDoFRigidBodyMotion 클래스 검토

- sixDoFRigidBodyMotion 한계
  - Constraint 클래스
    - 모든 constraint 클래스는 고유의 degree of constraint (DoC)와 axis of action vector을 가지고 있음
      - » Mode: translation, rotation
      - » Degree of Freedom (DoF) = 3 – DoC
      - » pointConstraint 클래스 이용한 DoF 제어
        - + applyConstraint(const vector& cd)
        - + 주어진 constraint의 DoC를 더하는 것으로 constraint 합성이 일어남
    - 문제점
      - » 필요한 Rotation의 DoF = 2 인 경우
        - + Rotation의 DoC는 2 또는 3 이므로 가능한 DoF는 0, 1 또는 3 만 가능
        - + Seakeeping 문제에서 roll & pitch 연성 문제 해석 필요하나 불가능해짐

Class		Translation (DoC, axis of action)	Rotation (DoC, axis of action)
Rotation	axis	Free	(2, rotation axis)
	point	(3, zero)	Free
Translation	plane	(1, surface normal)	Free
	line	(2, translation direction)	Free
	orientation	Free	(3, zero)

sixDoFRigidBodyMotionSolver 클래스 수정

- 클래스 상속관계 변경
  - 기존: 단일 상속관계
    - displacementMotionSolver
  - 수정: 다중 상속관계
    - displacementMotionSolver
    - sixDoFRigidBodyMotion
  - 이유
    - sixDoFRigidBodyMotion 클래스만 사용해도 동일한 코드의 sixDoFRigidbodyState functionObject를 작성해야함
    - sixDoFRigidBodyState functionObject의 일반화를 위해 다중 상속관계로 수정
    - OpenFOAM 8에서 반영될 예정
  - But, sixDoFRigidBody system will be removed in the source tree in the future and rigidBody system will replace it. By Henry Weller (https://bugs.openfoam.org/view.php?id=3345)

```
bool write()
{
const dynamicMotionSolverFvMesh& mesh =
    refCast<const dynamicMotionSolverFvMesh>(obr_);

const sixDoFRigidBodyMotionSolver& motionSolver_ =
    refCast<const
sixDoFRigidBodyMotionSolver>(mesh.motion());

const sixDoFRigidBodyMotion& motion =
motionSolver_.motion();
}
```

```
const sixDoFRigidBodyMotion& motion() const
{
  const dynamicMotionSolverFvMesh& mesh =
    refCast<const dynamicMotionSolverFvMesh>(obr_);

  return (refCast<const sixDoFRigidBodyMotion>(mesh.motion()));
}

bool write()
{
  ...
  const sixDoFRigidBodyMotion& motion = this->motion();
}
```

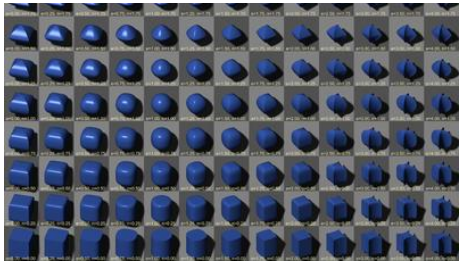
rigidBodyDynamics 클래스 검토

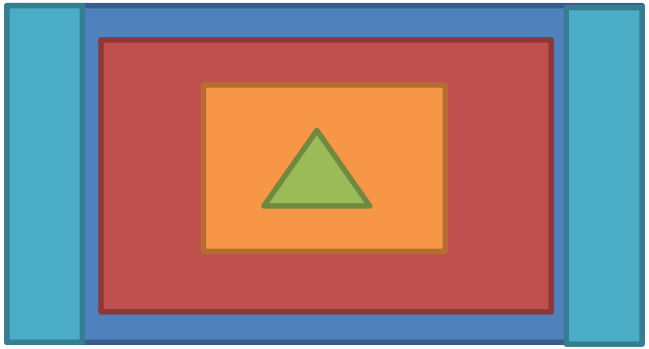
- rigidBodyDynamics 클래스
  - Constraint 제어
    - 각각의 constraint의 합성에 제한없음
    - 직관적
  - subBody 연결가능
    - 각종 부가물 (rudder, POD)등의 연성에 유용할 것으로 보임
  - 현재 해양 CFD 용으로 테스트중임

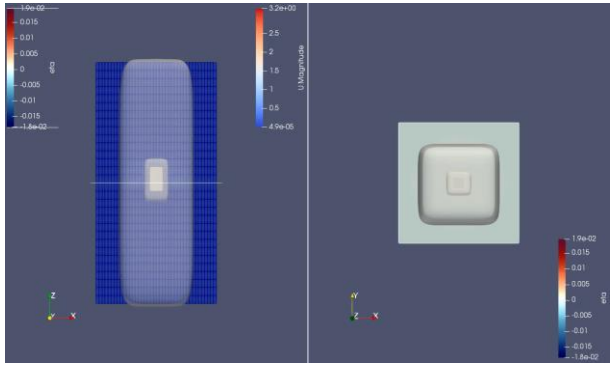
```
floatingObject
{
    type            rigidBody;
    parent          root;

    centreOfMass    (0 0 -0.25);
    transform       (1 0 0 0 1 0 0 0 1) (0 0 0);

    joint
    {
        type            composite;
        joints
        (
            {
                type Pz;
            }
            {
                type Rx;
            }
        );
    }

    patches         (floatingObject);
```

Algebraic Morphing Mesh

- Two levelsets : superellipsoid
  - Effective in/out determination
  - Various closed volume shape support
  - EOM zone protection







Wave elevation functionObject

- Wave elevation functionObject
  - surfaceElevationNew (waves2Foam 코드)
    - sampledSets functionObject 클래스의 단순 수정버전
      - + 상속관계가 아닌 copy & paste로 작성
      - + 불필요한 자료구조 존재
    - 개선 가능한 방향
      - » sampledSet, logFiles에서 상속
      - » Sampling line: gravity vector 이용한 ray tracing 이용
    - OpenFOAM 7 release version에서 사용시 오류발생
      - » sampledSets 객체 사용시 oldCellCentre 함수 호출 에러 발생
        - + OpenFOAM 7부터 새로이 도입된 변수/멤버함수임
        - + OpenFOAM 7 및 dev repository 9/5 이후 commit 버전에서 해결됨

Conclusions

- 해양 CFD
  - unsteady for a long time (3hr)
    - 병렬 성능이 중요
      - » OpenFOAM 코드 베이스 변경 : OpenFOAM 7 or higher 기반
- Wind load
  - Sustainable wind profile condition의 이점
- Global performance
  - 기존 sixDoFRigidBodyMotion 클래스의 한계
  - Rigid body state monitoring 도구 개발
  - morphingMesh 기법 개발
  - surfaceElevation functionObject 개선
- Path forward
  - Non-linear wave generation
  - Overset (based on foam-extend instead of OpenFOAM-plus)


Thank you very much!