

# 오픈소스 CFD 코드 OpenFOAM®에 대한 소개와 사용법 (上)



박 병 준 |

한국농어촌공사 농어촌연구원 수리시험연구팀  
연구원  
jiwhajai@ekr.or.kr

## 1. 개요

이수와 치수를 목적으로 하는 수리구조물들을 계획하고 시공함에 있어 해당 구조물의 건설 전·후에 대한 정확한 수리현상 검토는 필수적으로 선행되어야 할 과제이며, 공익을 위한 개발과 환경보전이 조화를 이룰 수 있도록 물의 거동을 연구하는 학자들과 실무자들이 양심을 걸고 최선을 다해야 할 사명 중 하나이다. 이러한 수리현상의 검토를 위해 해석적 방법, 수치모의, 수리모형실험, 현장관측 등이 통용되고 있는데, 수치모의의 경우 컴퓨터의 힘을 빌려 근사해를 얻어내는 방법으로 최근 하드웨어 사양의 발달과 더불어 자원이 허락하는 범위 내에서 정성적이고 정량적인 결과를 고해상도로 취득할 수 있을 뿐만 아니라, 모형의 변경과 수정이 용이하기 때문에 다수의 계획안에 대하여 수리검토를 할 수 있어 구조물의 계획 단계에 있어 경제적이고 안전한 재원 결정에 기여하고 있다.

수 년 전만 하여도 이러한 현상들을 수치기법으

로 모사하기 위해서는 해당 과업 종사자가 필요에 의해 직접 개발한 코드(in-house code)를 사용하는 것이 일반적이었으나, 직관적인 전·후처리 등 편리한 사용자 환경을 제공하며 꾸준한 개발이 진행된 상용 소프트웨어들의 약진으로, 최근 국내 대부분의 산·학·연에서는 전산유체역학(computational fluid dynamics, CFD)을 수행하기 위한 도구로 CFX™, FLUENT™, FLOW-3D® 등으로 잘 알려진 상용 소프트웨어를 사용하고 있는 상황이다.

상용 소프트웨어들은 지속적인 유지관리와 개발을 위한 체계가 잘 갖추어져 기하자료 및 격자를 생성하기 위한 모듈과 솔버, 결과물을 분석하고 도시하기 위한 도구들이 수치모의를 위한 흐름과정에 맞도록 하나의 패키지로 묶어 제공하고 있으며, 사용자가 이용하다 발생하는 문제점에 대한 지원이 신속하다는 장점이 있다. 그러나 고가의 라이선스 비용을 부담스러워 하는 사용자들도 적지 않은 실정이고, 더구나 최근 멀티코어 CPU의 보급과 하드웨어 가격 하락으로 병렬연산이 보편화 되고 있음에도, CPU 코어의 개수에 따라 사용료가 책정되는 라이선스 적용 방식은 효율적인 연구에 장애가 되고 있는 점도 있다.

반면, 공개 소프트웨어들은 대부분 솔버 코드만 제공되고, 전·후처리는 타 소프트웨어들을 사용해야 하는 경우가 대부분이라 과업을 수행하며 발생하는 문제점은 사용자가 해결해야 하기 때문에 사

용자가 기본 지식을 보유하고 있지 않거나 경험이 숙련되지 않으면 접근하기가 어려운 점도 있다. 그러나 이러한 점은 오히려 제한적 기능만을 제공하는 상용 소프트웨어와는 달리 솔버의 자유도가 높아 사용자가 자유롭게 솔버를 수정하거나 새로운 솔버를 생성할 수 있으며, 다수의 사용자가 즉시 코드를 개선하여 재배포 할 수 있다는 장점으로 부각되기도 한다. 지배방정식이 바뀌지 않는 한 어떤 CFD 코드를 사용하더라도 물리량의 계산 결과에는 큰 차이가 없고, 적용 수치기법에 따른 계산시간과 오차의 차이는 상대적으로 미묘하므로 사용료가 무료인 오픈 소스 CFD는 학계와 업계로부터 관심을 받을 자격이 충분하고, 경쟁력이 있다고 판단되어, 본 고에서는 오픈 소스 CFD 코드인 OpenFOAM®에 대한 간략한 소개와 사용법을 2회(上·下)에 걸쳐 다루고자 한다.

## 2. 코드소개

### 2.1 개요

OpenFOAM®(Open Field Operation and Manipulation)은 OpenCFD Ltd.<sup>1)</sup>에서 개발한 범용 3차원 전산유체역학 코드로서, GNU GPL(General Public License)를 사용하기 때문에 누구나 무료로 내려 받아 사용할 수 있다. 원래 OpenFOAM®은 1993년 영국의 Imperial College의 Gosman 교수 연구실에서 FOAM이라는 이름으로 Henry Weller와 Hrvoje Jasak에 의하여 CFD 코드의 재활용성을 위한 C++ 언어 기반의 CFD 솔버를 목적으로 개발이 시작되었고, 2004년 OpenFOAM®으로 개명하고 소스를 공개함으로써 전 세계의 많은 연구자들이 참여하여 지속적인 보완이 진행되어 2011년 6월 16일 버전 2.0<sup>2)</sup>이 발표되었다.

1) OpenCFD Ltd.는 2011년 8월 15일 Silicon Graphics Internation Corp.에 인수되었다.

2) 2011년 8월 4일 버전 2.0.1로 마이너 업데이트 되었다.

OpenFOAM®은 라이브러리 형태로 개발되며 실제 특정 문제를 해결하기 위한 솔버는 독자적으로 개발할 수 있는 환경을 제공한다. 이 라이브러리들은 우리가 해석하고자 하는 편미분 방정식과 최대한 유사하게 만들어져 있어 지배방정식을 잘 이해하고 있다면 C++ 언어에 대해 자세히 알지 못하더라도 쉽게 소스 코드를 만들 수 있는데, 표 1과 같이 넣어주면 된다.

표 1. OpenFOAM®에서의 편미분 방정식 코딩방법

|      |   |
|------|---|
| 수학적식 | $\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$    |
| 삽입코드 | <pre> solve (   fvm::ddt(rho,U) + fvm::div(phi,U)   - fvm::laplacian(mu,U) == -fvc::grad(p) ); </pre> |

OpenFOAM®은 또한 C++ 라이브러리들의 소스 파일 뿐 아니라 사전에 컴파일 된 많은 어플리케이션들을 함께 제공하는데, 솔버와 유틸리티로 구분된다. 기본적으로 제공되는 표준 솔버들은 특정 문제에만 맞도록 만들어져 있으며, 비압축성유동, 압축성유동, 화학반응, 다상유동, 구조해석, 전자기장해석, 분자동역학해석 등 60여 가지를 지원하고, 유틸리티들은 전·후처리, 데이터 파일 관리, 병렬 연산 등을 지원한다.

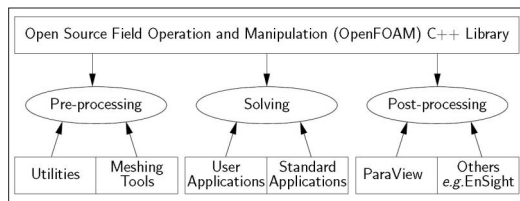


그림 1. OpenFOAM®의 개략구조

### 2.2 사용환경

OpenFOAM®의 작동환경은 LINUX를 기반으로 하고 있다. LINUX가 C/C++로 컴파일 된 오픈

소스 OS이기도 하고, 기본적으로 C/C++ 컴파일러를 내장하고 있기 때문에 C++ 기반으로 코딩된 OpenFOAM®의 개발환경과 이념이 잘 부합하였기 때문이라 추측된다. 어찌되었든, OpenFOAM®을 익숙하게 잘 사용하려면 LINUX를 사용하는 편이 훨씬 수월하다고 말하고 싶다. 물론, Cygwin을 이용하면 Microsoft® Windows 환경에서도 사용할 수 있겠지만 계산 속도가 현저하게 떨어지기 때문에 추천하지는 않고, 꼭 그럴 수밖에 없는 상황이라면 blueCFD<sup>3)</sup>의 사용을 고려하는 편이 나올 것 같다.

LINUX는 여러 종류의 무료판이 배포되어 있으나 현재 Red Hat 계열의 CentOS, Debian 계열의 Ubuntu가 널리 이용되고 있다. 과거와는 달리 최근엔 보편적인 하드웨어들을 다 지원하고, GUI 환경이 많이 개선되었다 하지만 LINUX는 여전히 터미널에서 직접 명령어를 입력해야 하는 경우들도 많이 부딪히게 된다. LINUX 사용 경험이 적고, GUI 환경에 익숙하다면 Ubuntu를, LINUX 사용 경험이 많고, 다수의 오픈소스 프로그램들을 컴파일해서 사용한 경험과 역량이 된다면 CentOS의 사용을 권장한다<sup>4)</sup>.

OpenFOAM®은 32bit와 64bit에서 모두 작동하고, MPI를 통한 병렬계산이 기본적으로 지원되기 때문에 CPU의 코어 개수가 많을수록 계산속도가 개선되므로 CPU의 클럭, 코어의 개수, 캐시메모리, 물리 메모리가 높은 PC를 준비할수록 유리하다. 사용할 PC의 메모리가 4GB 이하인 경우에는 32bit(x86) LINUX를 설치하고, 4GB를 초과하는 경우는 64bit(x64) LINUX를 설치하도록 한다. 32bit의 경우 OS가 인식할 수 있는 물리 메모리의 한계가 2<sup>32</sup>Byte이기 때문에 4GB를 초과하는 메모리가 설치되어 있더라도 OS가 32bit인 경우 그 이상은 인식하지 않는다는 것을 유념한다.

### 2.3 기본방정식

유체부에서의 지배방정식은 Navier-Stokes 식이다.

$$\nabla U = 0 \tag{1}$$

$$\frac{\partial \rho U}{\partial t} + \nabla(\rho U U) - \nabla(\mu S) = -\nabla p + \rho g + \sigma K \frac{\nabla \alpha}{|\nabla \alpha|} \tag{2}$$

여기서, U는 속도벡터장, p는 압력장, α는 두 가지의 유체로 정의되는 체적파면함수(volume fraction function)로 다음과 같이 정의된다.

$$\alpha = \begin{cases} 0 & : \text{volume occupied by air} \\ 1 & : \text{volume occupied by water} \end{cases} \tag{3}$$

S는 변형률텐서(strain rate tensor)로 S=1/2(∇U+∇U<sup>T</sup>)이고, σ는 표면장력, K는 표면곡률이다. 표면장력 문제는 그렇게 중요하지 않지만 식의 완성을 위하여 삽입되었다. 도메인에서의 밀도 ρ와 점성계수 μ는 다음과 같이 주어진다.

$$\rho = \alpha \rho_1 + (1-\alpha) \rho_2 \tag{4}$$

$$\mu = \alpha \mu_1 + (1-\alpha) \mu_2 \tag{5}$$

체적파면 α는 유체 속도장에 의해 이동되며, 체적파면 스칼라(scalar) α는 다음과 같다.

$$\frac{\partial \alpha}{\partial t} + \nabla(U\alpha) = 0 \tag{6}$$

수치확산(numerical diffusion)은 물과 공기 사

3) 포르투갈 blueCAPE의 상용 소프트웨어로 50유로에 판매되고 있다.

4) OpenFOAM®을 손쉽게 설치하려면 Ubuntu, 직접 컴파일해서 쓰려면 CentOS를 선택하면 된다.

이에 명확한 계면(sharp interface)으로 표현된다. 압축성 계면을 정의하는 기법인 CICSAM은 그 계면에 대해 재정립 해왔는데, 최근의 자유표면 모형과 CICSAM 기법의 상세내용은 Ubbink와 Issa(1999)에 의하여 찾을 수 있다.

유체부에서의 식은 전통적인  $k-\epsilon$  모형(Lauder와 Spalding, 1973)으로 구해진다.

$$\mu_t = C_\mu \rho \frac{k^2}{\epsilon} \quad (7)$$

$$\frac{\partial k}{\partial t} + \nabla \cdot (Uk) = \frac{1}{\rho} \nabla \cdot \left( \frac{\mu_t}{\sigma_k} \nabla k \right) + 2 \frac{\mu_t}{\rho} |\nabla U|^2 - \epsilon \quad (8)$$

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (U\epsilon) = \frac{1}{\rho} \nabla \cdot \left( \frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + 2 \frac{C_\mu \mu_t}{\rho} |\nabla U|^2 - \frac{\epsilon}{k} - C_{2k} \frac{\epsilon^2}{k} \quad (9)$$

여기서,  $\mu_t$ 는 난류와동점성계수(turbulence eddy viscosity),  $k$ 는 난류운동에너지(turbulence kinematic energy),  $\epsilon$ 는 난류에너지소산률(turbulence energy dissipation rate)이다. 식 (7), (8), (9)의 상수들은 Lauder와 Spalding(1973)이 제시한 값을 취한다.

### 3. 모형구축 : Pre-process

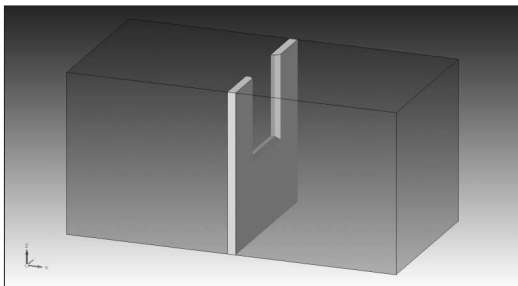


그림 2. 기하자료의 작성

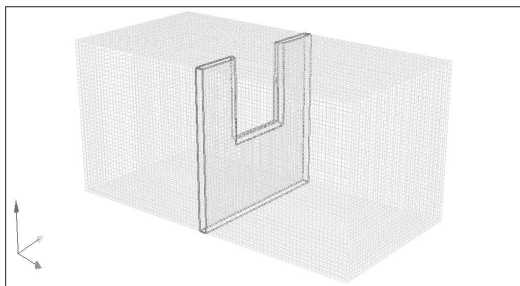


그림 3. 격자의 작성

### 3.1 기하자료 및 격자작성

OpenFOAM®은 사용자가 직접 격자를 생성할 수 있도록 blockMesh, snappyHexMesh와 같은 도구들을 제공한다. 유체장(fluid field)의 형상이 단순한 경우 blockMesh로 별도의 기하자료 작성을 위한 작업을 할 필요 없이 ASCII 텍스트 에디터를 통하여 격자만 정의하면 된다. 그러나 유체장의 형상이 복잡할 경우에는 사용자의 관심대상에 따라 비정렬·가변격자의 작성을 요구하게 될 수도 있다. 이 경우에는 기하자료를 작성하기 위한 작업이 추가로 필요하게 되고, STL(STereo Lithography) 파일과 snappyHexMesh를 활용하면 된다.

3차원 기하형상 작성을 위한 CAD 도구로는 오픈소스 프로그램인 Blender와 AutoCAD™, Solid Edge™, Solid Works™ 등의 상용 프로그램, Rhino™ 같은 기간제 프리웨어로 사용할 수 있는 것들도 있으므로 여건에 맞는 것을 이용하면 된다. 필자의 경우는 FLOW-3D®, FLUENT™, CFX™를 동시에 다루어 각 계산결과를 비교하는 과업을 진행 중이라 기하형상 작업은 Blender나 Solid Edge™를, 격자작성 작업은 Ansys Meshing™ 혹은 ICFM CFD™, Harpoon™<sup>5)</sup>을 애용하는 편이다. 그 이유는 OpenFOAM®에는 타 상용 프로그램으로 작성된 격자를 변환시켜주는 툴들이 내장되어 있어, FLUENT™나 CFX™용으로 작성된 격자도 OpenFOAM®용으로 변환<sup>6)</sup>이 가능하며, 격자를 직

5) 영국 Sharc Ltd.의 상용 소프트웨어로 (주)텍스트이엔에스(한국총판)에서 체험판을 받을 수 있다.

6) ASCII 형식으로 저장된 파일만 지원한다.

육면체 형태의 정렬격자로 구성하면 삼각망 사면체 형태의 격자에 비하여 계산시간을 단축할 수 있지만, 반대로 격자의 생성에는 상대적으로 시간이 더 소요되는데, Harpoon™의 경우 유체장 STL 파일만 있으면 기하형상에 맞는(body fitted) 육면체 격자를 가장 손쉽고 빠르게 작성할 수 있기 때문이다.

### 3.2 경계조건 및 초기조건 구성

OpenFOAM®의 모의안(case file) 작성은 [0], [constant], [system]이라는 디렉터리를 구성하고, 그 속에 각각의 값들을 작성한 파일들을 넣어줘야 한다.

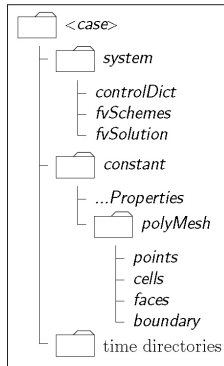


그림 4. 디렉터리 구조

#### 3.2.1 [0] (시간 디렉터리)

디렉터리 [0]에는 유체의 초기상태에 대한 물리량 파일들이 들어간다. 모의시간을 10초, 계산결과 저장간격을 0.1초로 하여 계산을 수행하면 0.1초마다 누적시간을 이름으로 한 디렉터리를 만들어 그 속에 유체의 물리량들 파일들을 저장한다.<sup>7)</sup> 처음 파일의 목록은 다음과 같다.

- alpha1, epsilon, k, p\_rgh, U

#### 3.2.2 [constant]

디렉터리 [constant]에는 중력가속도, 난류방정식, 수송방정식 등의 옵션들이 정의된 파일들과 서브 디렉터리인 [polyMesh]가 들어가는데, [polyMesh]에는 격자정보와 경계조건 정보가 들어간다. 처음 파일의 목록은 다음과 같다.

- g, RASProperties, transportProperties,

turbulenceProperties

- boundary, cellZones, faces, faceZones, neighbour, owner, points, pointZones

#### 3.2.3 [system]

디렉터리 [system]에는 모의시간, 계산시간의 간격, 계산결과와 저장 간격, 수치기법 옵션에 대한 정보가 들어간다. 처음 파일의 목록은 다음과 같다.

- controlDict, decomposeParDict, fvSchemes, fvSolution, setFieldsDict

### 4. 계산 : Solve

일반적으로 수공학에서 다루는 유체는 물이며, 대부분 자유수면을 가지는 개수로 형태를 취한다. 이는 유체역학적 측면에서 단상유체(single phase fluid)가 아닌 다상유체(multi phase fluid)에 해당하며, 특히 물과 공기의 경우 밀도차가 크고 혼합이 되지 않는 계면을 형성한다. 이런 현상의 해석을 위하여 VOF(Volume Of Fluid)와 LSM(Level Set Method) 등이 널리 이용되고 있는데, OpenFOAM®에서는 비압축성 비혼합 다상유체의 해석을 VOF로 풀어가도록 컴파일 된 interFoam이라는 기본 솔버가 제공된다. OpenFOAM®이 제공하는 솔버들 중 십중팔구 interFoam으로 대부분의 개수로 현상 모의가 가능하다고 생각하면 된다.

모의안 작성이 완료되었다면, 계산에 앞서 초기 상태에서의 유체장을 형성시켜야 하는데 모의안 디렉터리에서 setFields를 실행하면 되고, 완료되면 interFoam를 실행하면 된다. 그림 5는 계산이 진행 중인 모습으로, pyFoam이라는 썬드파티 패키지를 이용하여 우측편에 연속성과 잔차를 실시간으로 도시하고 있다.

7) [0.1], [0.2], [0.3], [0.4], ... , [10.0]의 형식인 시간별 디렉터리가 생성된다.

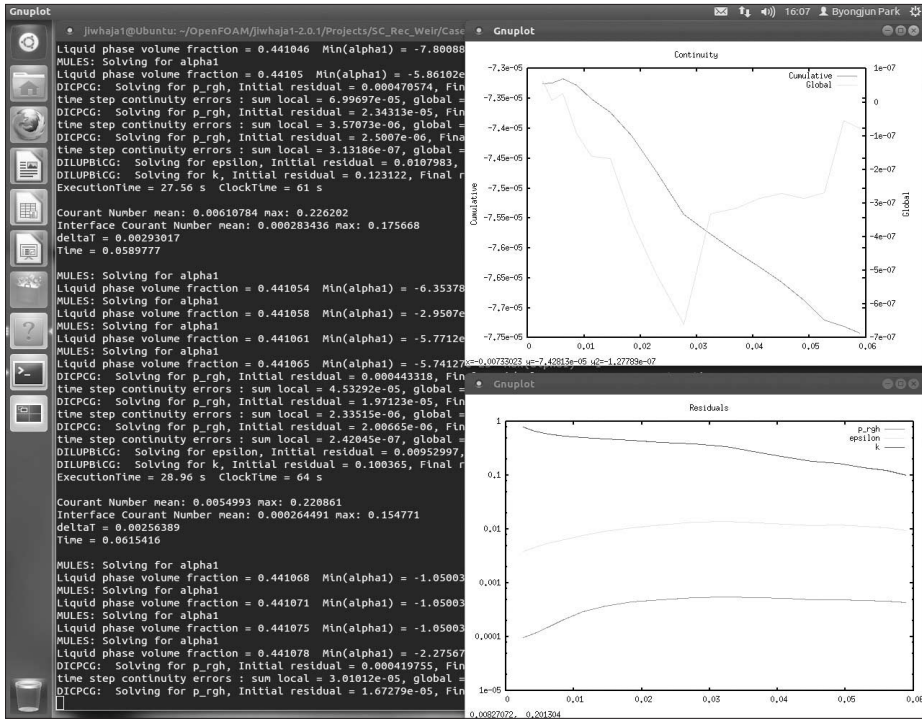


그림 5. 계산의 실행장면

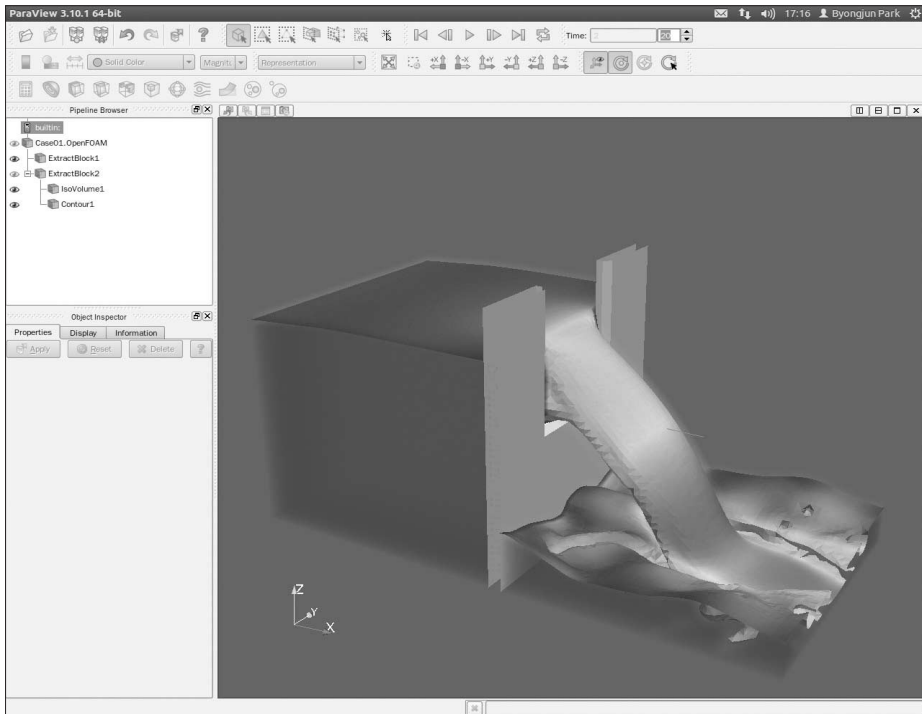


그림 6. 계산결과의 도식



## 5. 결과도시 : Post-Process

기본적으로 OpenFOAM®에서는 결과 도시용 패키지가 제공되지 않으나 계산된 결과물을 오픈소스 프로그램인 ParaView에서 읽을 수 있도록 paraFoam이라는 패키지를 제공한다. 따라서, 계산이 종료되었다면 모의안 디렉터리에서 paraFoam을 실행하면 곧바로 ParaView에 계산 결과를 도시할 수 있다. EnSightCFD™의 경우 기본적으로는 상용 소프트웨어지만 절점 수 200만개 이내에서는 기능제한 없이 무료버전으로 이용이 가능하고, 상용 소프트웨어인 FieldView™, Tecplot 360™에서도 OpenFOAM®으로 계산한 결과의 도시를 지원한다.

## 6. 정리

지금까지 오픈 소스 CFD 코드인 OpenFOAM®에 대한 개략적인 소개를 다루었다. 지면이 한정되고, 저자의 공부가 부족하여 상세한 설명을 수록할 여력이 되지 않았음도 시인한다. 다만, 이 부족한 내용이 무료 소프트웨어만을 이용하여도 얼마든지 양질의 CFD 과업을 수행할 수 있음을 알릴 수 있다면 저자로서는 더 없는 보람이 될 것이며, 여러 선배님들의 조언들을 통한 시너지도 기대해 본다. 다음 호에서는 OpenFOAM® 및 부수적인 Software들의 설치방법과 간단한 예제를 선정하여 따라하기 식 실습에 대한 내용을 다룰 예정이다. ☕

## 참고문헌

1. 박준권, 강관형 (2001). Source 공개 코드 OpenFOAM에 대한 리뷰, *한국전산유체공학회지*, 제15권, 제3호, pp.46-53.
2. Franz Jacobsen (2009). Application of OpenFOAM for designing hydraulic water structures. *Open Source CFD International Conference 2009*.
3. OpenCFD Ltd. (2011), OpenFOAM User's Guide(Ver 2.0.0).
4. OpenCFD Ltd. (2011), OpenFOAM Programmer's Guide(Ver 2.0.0).
5. Xiaofeng Liu, Marcelo H. Garcia (2006). Numerical Simulation of Local Scour with Free Surface and Automatic Mesh Deformation. *World Environmental and Water Resource Congress 2006, ASCE*.