

# 오픈폼(OpenFOAM)에 LU-SGS 알고리즘을 이용한 밀도 기반의 내재적 Navier-Stokes Solver 코드 개발 및 검증

## A Development and Verification of Density Based Implicit Navier-Stokes Solver Using LU-SGS Algorithm in OpenFOAM

김정현<sup>1\*</sup>, 김규홍<sup>1</sup>, 김지홍<sup>2</sup>, 신훈범<sup>2</sup>, 김병윤<sup>3</sup>, 길재홍<sup>3</sup>  
서울대학교<sup>1</sup>, 한국항공우주산업<sup>2</sup>, 넥스트폼<sup>3</sup>

### 초 록

본 연구에서는 기존의 오픈 소스 기반의 전산유체역학 프로그램인 오픈폼(OpenFOAM)에 LU-SGS 를 이용한 밀도 기반의 내재적 해석자에 대한 코드개발 및 검증을 수행하였다. 현재 오픈폼에서 압축성 해석자는 압력 기반의 내재적 해석자 혹은 밀도 기반의 외재적 해석자가 상용화 되어 있다. 본 연구에서는 오픈폼에서 압축성유동 해석의 정확성을 높이기 위해 LU-SGS 를 이용한 새로운 밀도 기반의 내재적 해석자에 대한 코드개발을 수행하였다. 기존의 오픈폼에서 제공하는 있는 해석자에 비해 빠른 수렴성과 정확도를 확인할 수 있었다. 게다가 현재 오픈폼에서 상용화 되어 있지 않은 리만 경계조건에 대해 기존의 오픈폼에서 제공하는 경계조건과 비교함에 따라 활용가능성을 확인할 수 있었다. 연구 결과를 바탕으로 코드검증 시 널리 사용되고 있는 검증 모델을 선정하여 검증을 수행하였다. 그 결과 개발된 코드의 활용가능성을 확인하였고 이를 바탕으로 향후 수행할 연구 방향을 제시하였다.

### ABSTRACT

A development of density based implicit solver using LU-SGS(Lower Upper Symmetric Gauss Seidel) algorithm in OpenFOAM was performed. Within OpenFOAM as standard solvers, compressible flow equations are addressed using different approaches such as pressure implicit based or density explicit based solvers. In this study, a new density based Navier-Stokes solver was developed. A faster convergence and more accuracy are expected from this new type of solver. In addition, there are no characteristic boundaries in OpenFOAM so that the boundary condition, namely Riemann invariant, was implemented as well. To validate the solver developed for using in OpenFOAM, some validation models which are widely used to validate developed codes were analyzed. Preliminary results indicated that our setup in OpenFOAM was correct and additional results showing the comparison between an experimental and computational data of aircraft and fighter will be shown in the nearest future.

**Key Words:** OpenFOAM(오픈폼), Density Based Implicit Code(밀도 기반 내재적 코드), LU-SGS(Lower Upper Symmetric Gauss Seidel), Open Source Code(오픈 소스코드), Riemann Boundary Condition(리만경계조건)

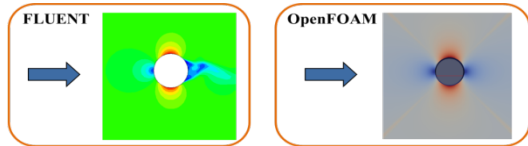
## 1. 서 론

유체역학의 현상에 대한 연구는 다음과 같은 접근방법에 의해 접근할 수 있다: 1)이론적인 접근, 2)실험적인 방법, 3)전산유체역학. 그러나 이론적인 접근은 지배방정식을 풀기 위한 한계가 여전히 존재하고 있고 실험적인 방법은 시험 설비 설치 및 유지, 시험 모델 제작 등에 사용되는 비용의 부담이 큰 문제로 작용하고 있다. 이에 따라 전산유체역학을 통한 유동 해석 연구가 다양한 분야에서 활용되고 있는 것이 사실이

다.<sup>(1)</sup>

사실 전산유체역학을 통한 유동 해석 연구를 수행하기 위해서는 In-house 코드를 개발하여 사용하거나 FLUENT 와 같은 상용 전산유체역학 해석 프로그램을 사용해야 한다. 상용 해석 프로그램을 사용하는 경우에는 해석 코드를 개발해야 하는 시간과 노력을 절약할 수 있으며 사용방법 학습이 용이하여 사람들에게 쉽게 다가갈 수 있다. 그러나 이 경우에는 초기 구매 비용과 유지보수 비용이 비싸고 별도의 라이선스 비용을 지불해야 하는 문제점이 존재할 뿐만 아니라 코드의 수정이 불가능하며 특정 문제를 해

결하는데 사용자가 제한을 받을 수 있다.<sup>(1)</sup> 예를 들어 FLUENT의 경우 오픈 소스 기반의 전산유체역학 프로그램인 오픈폼(OpenFOAM)에서 이미 해석자로 가지고 있는 포텐셜 유동에 대한 해석을 수행할 수 없다.



**Fig.1. FLUENT vs. OpenFOAM (Potential Flow Solution)**

그에 반해 In-house 코드 개발을 통한 전산유체역학 연구는 자신의 연구 분야에 적용하기에 적절한 해석기법을 사용하기 때문에 높은 정확도를 가지며 개발자에 의한 수정이 용이하다는 장점을 지니고 있다. 그러나 In-house 해석 코드를 개발하기 위해서 필요한 시간과 노력이 상당하며 개발자 이외의 연구자가 사용하기에는 상당한 노하우가 개발자에 의해 전수되어야 하는 단점 또한 지니고 있다.

위에서 언급한 사항들을 보완하기 위해 오픈 소스 전산유체역학 프로그램인 오픈폼이 다음과 같은 이유 때문에 사용되고 있다: 1)공개 프로그램을 지향함에 따라 쉽게 코드에 접근할 수 있다. 2)비교적 메모리를 적게 요구한다. 3)비용이 들지 않는 GPL(General Public License)를 사용한다. 4)특정한 연구분야에 적용하기에 용이하다.<sup>(1)</sup>

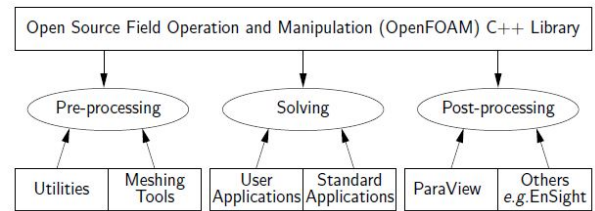
이에 따라 본 연구에서는 기존의 오픈 소스 기반의 전산유체역학 프로그램인 오픈폼에 LUSGS를 이용한 밀도 기반의 내재적 해석자에 대한 코드개발 및 검증은 수행하였다. 현재 오픈폼에서 압축성유동의 해석자는 압력 기반의 내재적 해석자 혹은 밀도 기반의 외재적 해석자가 상용화 되어 있다. 본 연구에서는 오픈폼에서 압축성유동의 해석의 정확성을 높이기 위해 LUSGS를 이용한 새로운 밀도 기반의 내재적 해석자에 대한 코드개발을 수행하였다. 기존의 오픈폼에서 제공하는 있는 해석자에 비해 빠른 수렴성과 정확도를 확인할 수 있었다. 게다가 현재 오픈폼에서 상용화 되어 있지 않은 리만 경계조건에 대한 활용가능성을 기존의 오픈폼에서 제공하는 경계조건과 비교함에 따라 확인할 수 있었다.

본 논문은 오픈폼의 개요, 코드 검증 및 확인, 검증모델의 전산해석 결과, 결론 및 향후 연구 계획으로 구성되어 있다.

## 2. 본 론

### 2.1 오픈폼(OpenFOAM)의 개요

오픈폼은 LINUX OS 하에서 C++ 언어 기반으로 개발되었으며 객체지향방식으로 현재까지 사용자들이 개발한 압축성, 비압축성, 다상유동, 연소와 열전달 등과 관련된 다양한 해석자들이 포함되어 있다.<sup>(4)</sup>



**Fig.2. Overview of OpenFOAM Structure**

객체 지향방식으로 개발된 오픈폼은 기본적으로 유한체적기법으로 유동현상을 해석한다. 예를 들어 식(1)과 같은 해석하고자 하는 식이 있을 경우에는 그림 3 과 같이 코드를 작성하여 실행한다.<sup>(4)</sup>

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p \quad (1)$$

```

solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);

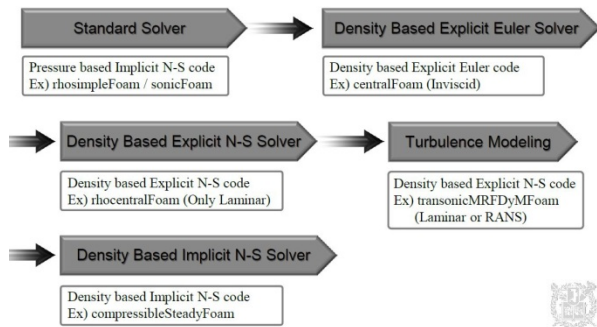
```

**Fig.3. Equation Representation**

### 2.2 코드 검증

#### 2.2.1 코드 검증 개요

오픈폼에 압축성유동의 해석자를 위한 이전 연구 및 본 연구에서의 코드 개발의 개요는 그림 4 과 같다.



**Fig.4. Outlines for the developed density based implicit code in OpenFOAM**

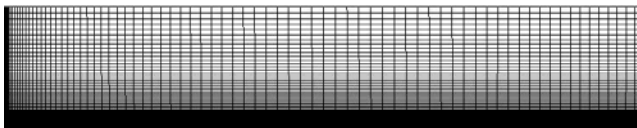
본 연구에서는 Oliver Borm<sup>(6)</sup>에 의해 개발된 외재적 밀도 기반의 코드를 기반으로 정상상태에 대한 내재적 밀도 기반의 코드개발을 수행하였다.

### 2.2.2 외재적 밀도 기반의 코드 검증

코드의 검증을 위해 아래와 같은 격자와 경계 조건 및 수치기법을 사용하였다. 해석격자는 상용격자 생성 프로그램인 Gambit 2.4(ANSYS, Canonsburg, PA, USA)를 사용하였고 오픈폼에서 사용할 수 있도록 격자를 전환하였다.

**Table 1. Flow conditions (Altitude 15km)**

$\mu_{\infty}$	1.4216E-05 Ns/m <sup>2</sup>
$T_{\infty}$	216.65 K
$P_{\infty}$	12111.4 Pa
$a_{\infty}$	295.07 m/s
$M_{\infty}$	1.5



**Fig.5. Grid system for verification of the code**

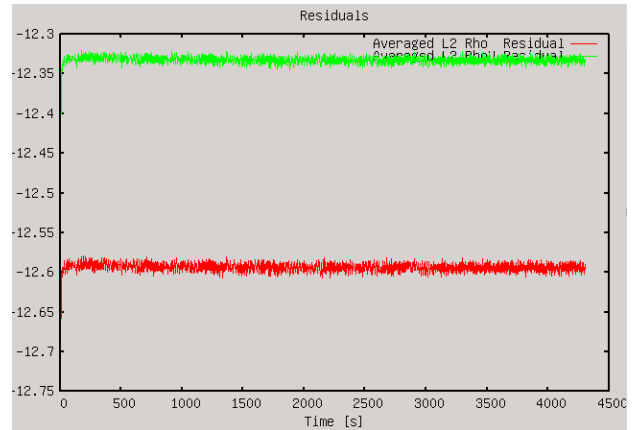
**Table 2. Boundary Conditions**

Left Side	Fixed Value (P,T, and U)
Right Side	1st order Extrapolation
Top/Bottom	1st order Extrapolation

**Table 3. Numerical Schemes (Explicit code)**

Time discretization	Explicit 4 <sup>th</sup> Runge-Kutta
Space discretization	Roe scheme
Convergence criterion	Double precision

정상상태 해석을 수행하였으며 수치적인 오류를 줄이기 위해 Courant Number 는 1 보다 작게 설정하였다. 아래의 그림에서 볼 수 있듯이, 주어진 조건에서 Residual 이 Machine accuracy 영역에서 기록되는 것을 확인할 수 있었다.



**Fig.6. Residuals of Explicit density based code**

### 2.2.3 내재적 밀도 기반의 코드 검증

Oliver Borm 에 의해 개발된 외재적 밀도 기반의 코드를 기반으로 검증을 완료한 후 LU-SGS 를 이용한 밀도 기반의 내재적 해석자에 대한 코드개발을 수행하였다. 코드의 검증을 위해 앞서 사용된 격자와 경계조건을 동일하게 적용하였고 수치기법은 아래와 같은 방법을 통하여 해석을 수행하였다.

**Table 4. Numerical Schemes (Implicit code)**

Time discretization	LU-SGS
Space discretization	Roe scheme
Convergence criterion	Double precision

정상상태 해석을 수행하였으며 그림 7 에서 볼 수 있듯이, 주어진 조건에서 Residual 이 Machine accuracy 영역에서 기록되는 것을 확인할 수 있었다. 또한 내재적 기반의 코드가 갖는 빠른 수렴성의 특성을 확인하기 위해 널리 사용되고 있으며 간단한 형상인 Bump 문제에 대해 2 차원 비점성유동의 해석을 수행하였다. 그 결과 그림 8 에서 볼 수 있듯이 기존의 외재적 밀도 기반의 코드에 비해 내재적 기반의 코드가 가지는 빠른 수렴성의 특성을 확인할 수 있었다. (3)

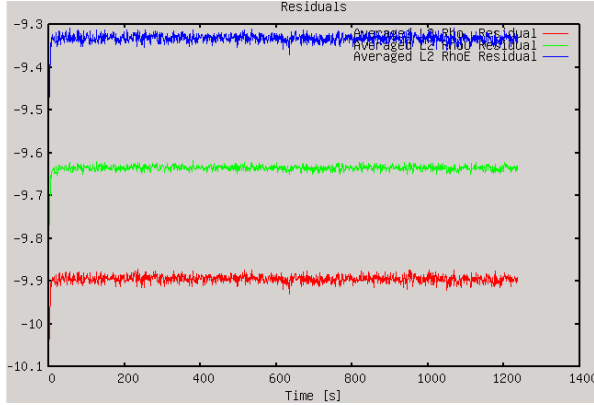


Fig.7. Residuals of Implicit density based code

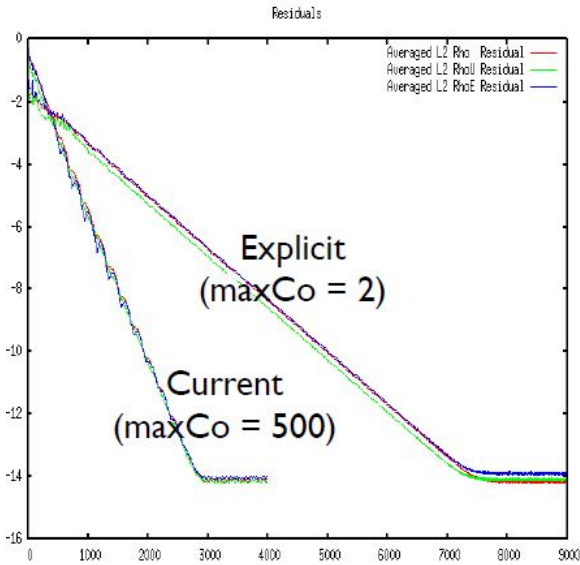


Fig.8. Explicit vs. Implicit (Convergence history)

#### 2.2.4 리만 경계조건의 코드 검증

일반적으로 수치해석의 해를 구하기 위해서는 다양한 경계조건 및 초기조건들이 적용된다. 오픈폼에서는 고정값 조건인 Dirichet 방식과 고정 기울기 조건인 Neumann 방식을 기본으로 다양한 경계 조건들을 제공한다. (4) 본 연구에서는 오픈폼에서 제공하지 않는 대표적인 특성 경계 조건인 리만 경계조건을 개발하고 이를 검증하였다. 보존량을 wave의 전파 방향에 따라 내삽 혹은 외삽하는 형태로 구성하였으며 이론적인 식을 바탕으로 다음과 같이 식을 구성하였다.

$$R^+ = U_{nB} + \frac{2C_B}{r-1} \quad (2)$$

$$R^- = U_{nB} - \frac{2C_B}{r-1} \quad (3)$$

이때  $R^+$ 는 전산해석영역으로 들어오는 wave를 의미하며 이것은 외부조건을 이용하여 구하도록 코드를 구성하였다. 반대로  $R^-$ 는 전산해석영역에서 나가는 wave로 내부조건을 이용하여 구하도록 코드를 구성하였다. 기본적으로 리만 경계조건은 유입류와 유출류가 혼재되는 경우 그 구분을 위해 사용되는데 이때 아음속에서 사용되는 유입류와 유출류의 정의는 아래의 식과 같다.

$$R^+(\text{Inflow}) = U_{nB} + \frac{2C_B}{r-1} = U_{n\infty} + \frac{2C_\infty}{r-1} \quad (4)$$

$$R^-(\text{Inflow}) = U_{nB} - \frac{2C_B}{r-1} = U_{ni} - \frac{2C_i}{r-1} \quad (5)$$

$$R^+(\text{Outflow}) = U_{nB} + \frac{2C_B}{r-1} = -U_{n\infty} + \frac{2C_\infty}{r-1} \quad (6)$$

$$R^-(\text{Outflow}) = U_{nB} - \frac{2C_B}{r-1} = -U_{ni} - \frac{2C_i}{r-1} \quad (7)$$

코드의 검증을 위해 아래와 같은 격자와 유동 조건을 적용하였다. 해석격자는 상용격자 생성 프로그램인 Gambit 2.4(ANSYS, Canonsburg, PA, USA)를 사용하였고 오픈폼에서 사용할 수 있도록 격자를 전환하였다. 경계조건으로는 개발된 리만 경계조건을 사용하였고 기존에 오픈폼에서 제공하는 자유류 경계조건을 해석한 후 비교분석 하였다.

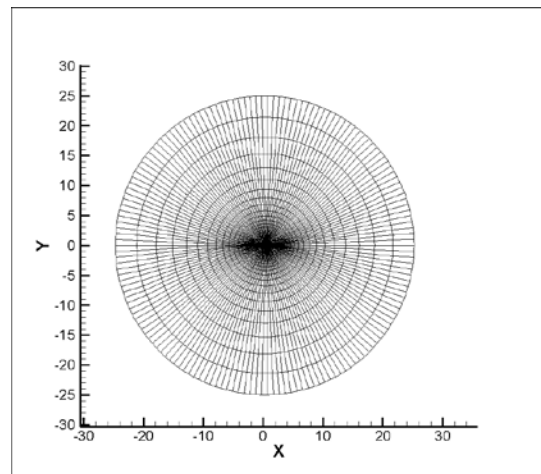


Fig.9. Grid system of RAE2822 airfoil

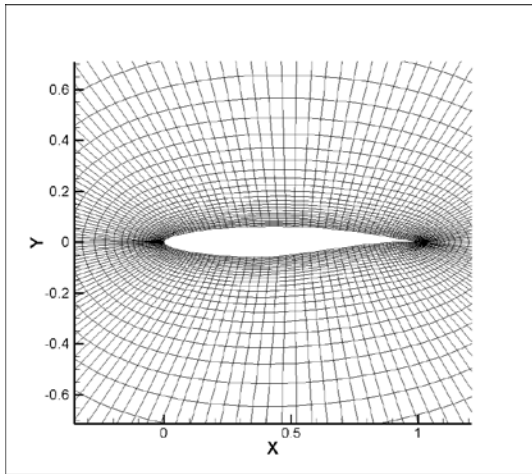


Fig.10. Grid system of RAE2822 airfoil (Close-Up)

Table 5. Flow Conditions

$T_{\infty}$	255.5 K
$P_{\infty}$	108987.393 Pa
$M_{\infty}$	0.729
Angle of Attack	2.31 degree
Characteristic length	1

아래의 그림에서 볼 수 있듯이, 전산해석영역이 코드길이의 25 배에서 10 배로 감소함에 따라 기준에 오픈폼에서 제공하는 경계조건(Free-stream condition)은 해석에 있어 전산해석영역의 크기에 따라 오류를 발생하는 반면 개발된 리만 경계조건은 그 차이가 미미한 것을 확인하였다. 또한 그림 13 과 같이 리만 경계조건을 사용하였을 때 코드길이의 15 배와 10 배에서 압력계수값의 차이가 없는 것을 확인하였다.

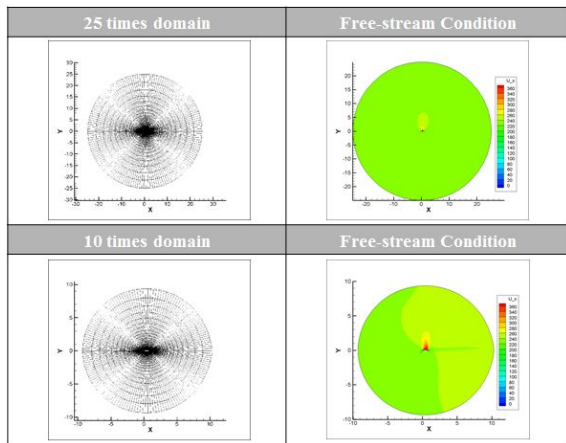


Fig.11. OpenFOAM Boundary Conditions

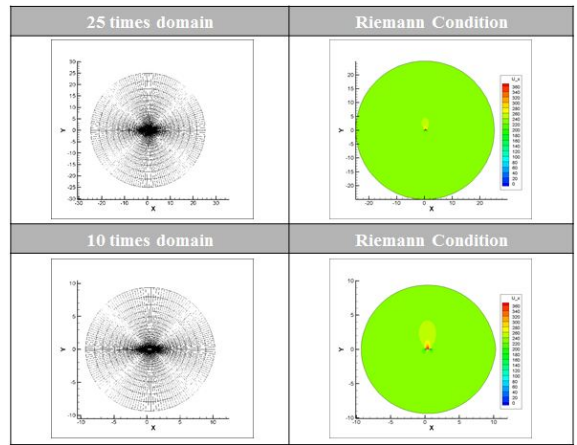


Fig.12. Riemann Boundary Conditions

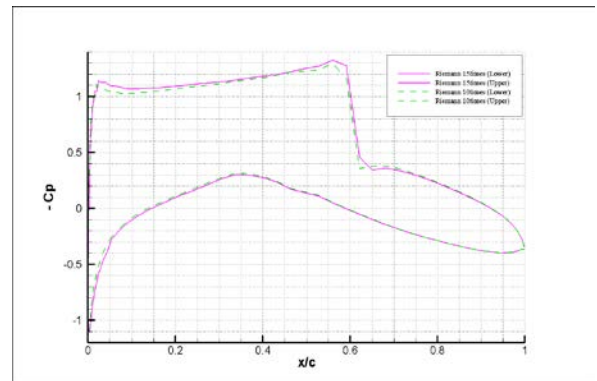


Fig.13. Riemann BCs (10 vs. 15 times Domain)

## 2.3 검증모델의 전산해석 결과

### 2.3.1 2 차원 썰기 경사충격파 해석

개발된 코드의 확인 및 검증을 위해 2 차원 썰기의 경사충격파에 대한 전산해석을 수행하였다.

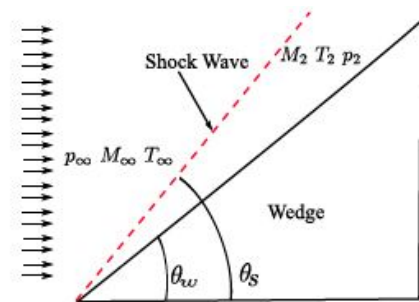
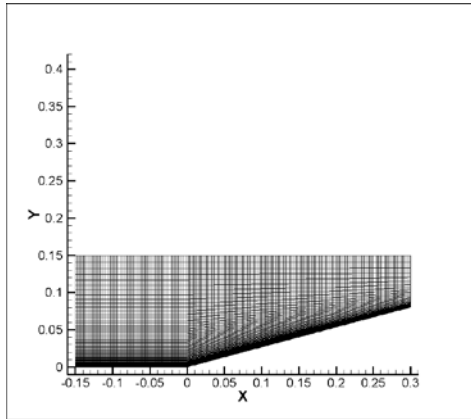


Fig.14. Schematic with flow geometry for wedge

해석을 위해 다음과 같은 격자와 조건을 적용하였다.

**Table 6. Flow Conditions**

$\mu_\infty$	1.78E-05 Ns/m <sup>2</sup>
$T_\infty$	288.88 K
$P_\infty$	101352.58 Pa
$M_\infty$	2.5
Wedge half angle	15 degree

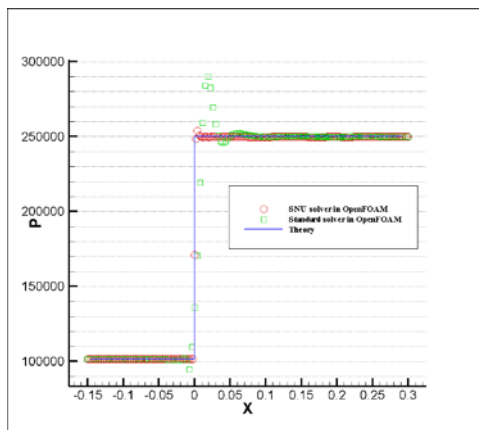


**Fig.15. Grid system for wedge**

초음속 영역의 해석 및 이론과의 비교를 위해 비점성, 압축성 유동의 조건으로 해석을 수행하였으며 이론식과의 비교분석을 통해 (5), (9) 코드의 검증 및 확인을 수행하였다. 아래의 표에서 확인할 수 있듯이 이론적인 값과 개발된 코드의 오차는 약 0.1% 되는 것으로 확인되었다.

**Table 7. Comparison (Theory vs. Computation)**

<i>Inviscid, Steady, and Compressible flow</i>			
	Theory	OpenFOAM	Error (%)
Pressure ratio	2.4675	2.4655	0.08
Temperature ratio	1.8665	1.8641	0.12
Density ratio	1.3219	1.3236	0.13



**Fig.16. Pressure distribution along the wall**

동일한 격자와 조건하에 오픈폼의 표준 해석자인 SonicFoam 과의 비교분석을 수행하였다. SonicFoam 의 경우는 압축성 유동에 맞도록 개발된 PISO 알고리즘을 바탕으로 개발되었으나 개발 당시 밀도기반이 아닌 압력기반으로 개발이 이루어졌다. 그림 16 과 같이 기준에 오픈폼에서 사용되고 있는 SonicFoam 의 경우 충격과 이후에 수치적인 오류가 크게 발생하고 진동이 심한 것을 확인할 수 있었다. 그러나 개발된 코드의 경우 충격과 이후에도 이론 값을 쉽게 예측하는 것을 확인할 수 있었다.

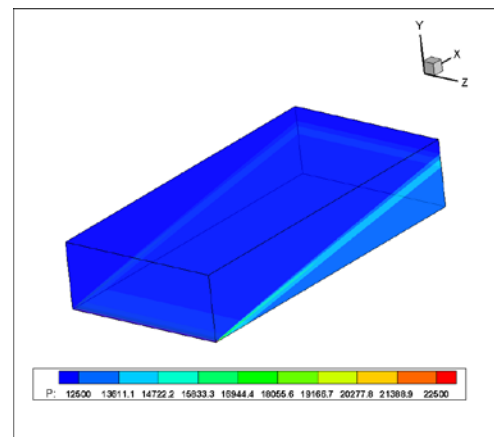
**2.3.2 3차원 평판 해석**

개발된 코드의 정확성 및 효율성에 대한 검증을 위해 2차원 평판 문제를 3차원 평판으로 확장하여 전산해석을 수행하였다. 본 검증에서 사용된 유동은 자유류 속도 마하 8 인 유동이 특성길이가 1 이고 표면 온도가 1000K 인 평판 위를 지날 때 고도 15km 조건에서 수치적으로 해석하여 Van Driest 결과(10)와 비교하였다. 해석을 위해 다음과 같은 조건을 적용하였다.

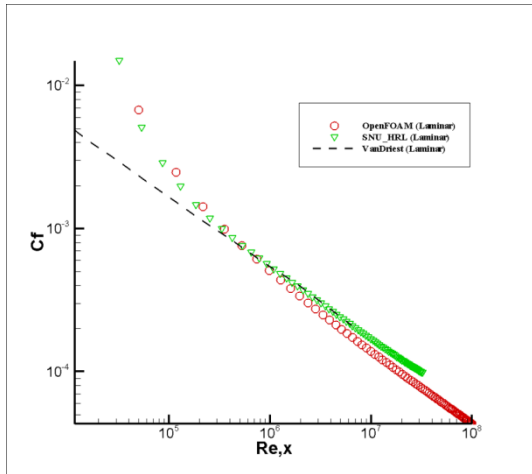
**Table 8. Flow Conditions (Altitude 15km)**

$\mu_\infty$	1.4216E-05 Ns/m <sup>2</sup>
$T_\infty$	216.65 K
$P_\infty$	12111.4 Pa
$a_\infty$	295.07 m/s
$M_\infty$	8

계산에 사용된 격자계의 크기는 80x160x3 이며 아래 그림은 초음속 평판 유동의 해석 결과이다.



**Fig.177. Pressure contour of flat plate**



**Fig.188. Comparison Skin friction coefficient of laminar flow**

그림 18 에서 확인할 수 있듯이 기존 In-house 코드(SNU\_HRL)와 비교하여 오픈폼의 결과는 다소 차이가 있는 것을 확인할 수 있었으며 이는 컴퓨터에 의해 계산된 전단응력을 오픈폼 후처리 기능을 이용하여 산출하는데 문제가 있을 가능성이 있으며 현재 세밀하게 확인 중에 있다.

### 3. 결론 및 향후 연구계획

본 연구에서는 In-house 코드와 상용 전산유체역학 프로그램의 단점을 보완하기 위하여 오픈소스 기반의 전산유체역학 프로그램인 오픈폼(OpenFOAM)을 사용하였다. 이러한 오픈폼에 압축성유동의 해석의 정확성을 높이기 위해 LU-SGS(Lower Upper Symmetric Gauss Seidel) 를 이용한 새로운 밀도 기반의 내재적 해석자에 대한 코드개발 및 검증은 수행하였다. 게다가 현재 오픈폼에서 상용화 되어 있지 않은 대표적인 특성 경계조건인 리만 경계조건에 대해 기존의 오픈폼에서 제공하는 경계조건과 비교함에 따라 그 활용가능성을 확인하였다. 코드의 검증을 위해 널리 사용되고 있는 검증모델을 선정하여 검증 및 확인을 수행하였다. 그 결과 기존의 오픈폼에서 제공하는 있는 해석자에 비해 빠른 수렴성과 정확도를 확인할 수 있었으며 개발된 코드의 활용가능성을 확인하였다. 3 차원 평판의 경우 In-house 코드의 결과 및 이론적인 결과와 비교하여 신뢰할만한 결과를 얻을 수 있었으나 오픈폼 자체의 후처리 기능을 이용한 전단응력 산출에 문제가 있을 가능성이 있으며 향후 전단응

력 산출 부분의 오류를 수정하고 난류모델의 검증과 실제 항공기 및 전투기의 전산해석 결과와 실험값을 비교함에 따라 개발된 코드의 정확성 문제를 보다 정량적으로 해석할 예정이다.

### 참고문헌

- 1) J.H.Kim, C.F. Lange, and C.R. Koch: A Computational Study of a Circular Cylinder at Low Reynolds number for Open Loop control of Von Karman vortex shedding, 8th OpenFOAM conference, South Korea, 2013.
- 2) Oliver Borm, Aleksandar Jemcov, and Hans-Peter Kau: Density Based Navier Stokes Solver for Transonic Flows, 6th OpenFOAM conference, USA, 2011.
- 3) R.F.Chen and Z.J. Wang: Fast, Block Lower-Upper Symmetric Gauss-Seidel Scheme for Arbitrary Grids, AIAA Journal, 2000.
- 4) OpenFOAM, the open source CFD Toolbox, User guide.
- 5) Luis f. Gutierrez Marcantoni, Jose P. Tamagno, and Sergio A. Elaskar: High speed flow simulation using OpenFOAM, Argentina journal, 2012.
- 6) Oliver Borm: Transonic Density based Flow Solver, 5th OpenFOAM conference, Sweden, 2010.
- 7) P.L. Roe: Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of computational physics, 43:357-372, 1981.
- 8) Catherine M. Maksymiuk and Thomas H. Pulliam: Viscous transonic airfoil workshop results using ARC2D, AIAA Journal, 1987.
- 9) John. D. Anderson: Fundamentals of Aerodynamics.
- 10) Christopher J. Roy and Frederick G. Blottner: Methodology for Turbulence Model Validation - Application to Hypersonic flows, Journal of Spacecraft and Rockets, June 2003.
- 11) T.W.Kim, S.J.Oh, and K.J.Yee: Verification of the Open Source Code, OpenFOAM to the External Flows, KSAS Journal, July 2011.