

오픈소스 CFD 코드 오픈폼(OpenFOAM)을 이용한 가스터빈엔진 노즐-터빈 유동 해석

김 병 윤^{1*}, 김 병 휘², 길 재 흥³

CFD Simulation of nozzle-turbine in gas turbine engine using open source CFD code OpenFOAMN

B.Y. Kim, B.H. Kim and J.H. Gill

OpenFOAM is widely used open source CFD code. In this study the performance of OpenFOAM to simulate nozzle-turbine flow is evaluated. Single stage nozzle-turbine of radial gas turbine engine is simulated. To simulate nozzle-turbine flow, the capability of compressible flow, MRF, cyclic boundary condition and non-matching grid interface are needed. snappyHexMesh of OpenFOAM is used to generate mesh and steady DensityBasedTurbo solver of OpenFOAM-1.6-ext is used for flow compressibility. cyclicGgi boundary condition is used and overlapGgi method for non-matching grid interface between nozzle and turbine. We compared results of OpenFOAM with commercial CFD code. OpenFOAM shows enough capability for industrial turbine simulation and shows some advantages and disadvantages comparing to commercial CFD code.

Keywords: 전산유체역학(CFD), 오픈폼(OpenFOAM), 가스터빈(Gas turbine)

1. 서 론

가스터빈 엔진의 노즐-터빈은 고온 고압의 연소가스를 이용하여 터빈을 회전시켜 에너지를 얻기 위한 장치로 노즐은 정지 상태에서 터빈으로 유입되는 유동을 제어하기 위한 장치이며 터빈은 연소가스에 의해 회전하는 장치이다. 본 연구의 대상은 원심형 터빈이다. 노즐-터빈의 CFD 해석을 위해서는 우선 작동유체의 높은 온도와 압력에 의한 압축성 유동 특성과 터빈의 회전을 고려하여야 한다. 노즐 및 터빈 전체를 해석할 수도 있으나 계산의 효율성을 위해 각각의 블레이드 하나에 대한 유로만을 계산할 때 경계면에서의 주기조건이 필요하다. 또한 노즐과 터빈의 블레이드 개수가 다르기 때문에 각각의 경계면의 형상이 일치하지 않아 데이터의 보정이 필요하다. 이런 해석 기법은 오래 전부터 정립되어 왔으며 본

연구에서는 오픈폼에서 격자생성, 경계면처리, 압축성유동해석 등이 얼마나 잘 구현될 수 있는지를 검토하고 상용프로그램의 대안으로 적용가능성을 평가하였다. 그리고 사용상의 편의성 향상을 위해 그래픽 사용자 환경을 개발하였다.

2. 본론

2.1 격자 생성

오픈폼이 제공하는 snappyHexMesh 격자생성 유틸리티가 노즐-터빈 문제에 적절한 격자를 생성할 수 있는지를 평가하였다. 노즐-터빈 문제는 블레이드를 포함한 벽면 주위의 경계층 격자를 잘 구현하는 것이 매우 중요하다. snappyHexMesh는 stl 파일을 이용하여 block mesh, castellate mesh, snap mesh, add layer의 순서로 작업이 이루어지며 각 단계에서 필요한 계수들을 입력값으로 받아들인다. OpenFOAM-2.1.x의 stl 파일로부터 feature line을 만들어 내는 surfaceFeatureExtract 유틸리티를 사용하였다. 원심형 터빈과 같이 곡률이 많이 변하는 경우 형상을 정확히 구현하기 위해서는 snap mesh 과정에서 입

1 비회원, (주)넥스트폼

2 비회원, STX 종합기술원

3 비회원, (주)넥스트이엔에스

* TEL : (070) 8796-3011

* Corresponding author E-mail: bykim@nextfoam.co.kr

력값들에 대한 세심한 주의가 필요하였다. 형상을 정확히 따라가기 위해 표면에 어느 정도 이상의 격자가 필요하였고 이 때문에 경계면들에 필요 이상의 많은 격자가 요구되는 부분이 존재하였다.

원하는 높이의 육면체 경계층격자를 자동으로 생성하는 것은 큰 문제가 없었다. 입력값에 따라 격자 생성에 소요되는 시간은 크게 차이가 나지만 격자 생성과정이 병렬연산을 지원하기 때문에 다른 격자생성 프로그램에 비해 격자생성에 소요되는 시간은 문제가 되지 않을 것으로 판단된다. 주기 조건을 적용하는 면에서 cyclic GGI(General Grid Interface) 조건을 사용하기 때문에 두 면의 표면격자를 같게 만들 필요가 없어 격자 생성 시 따로 고려할 사항은 없다. 노즐과 터빈 부분의 격자를 각각 생성하여 mergeMeshes 유틸리티를 사용하여 두 격자를 하나로 합쳐 계산에 사용하였다. 노즐에 305,778개, 터빈에 954,665개 격자가 사용되어 전체 120만개 정도의 격자가 사용되었다. 대부분이 육면체 격자이며 일부 프리즘, 사면체, 다면체격자가 사용되었다. Fig.1에 전체 형상과 격자를 나타내었다.

오픈폼의 격자 생성 기능을 사용하지 않고 상용 격자 생성 프로그램인 turbo grid을 사용하여 만들어진 격자는 fluent3DMeshToFoam이라는 유틸리티를 사용하면 문제없이 오픈폼 격자로 변환할 수 있다.

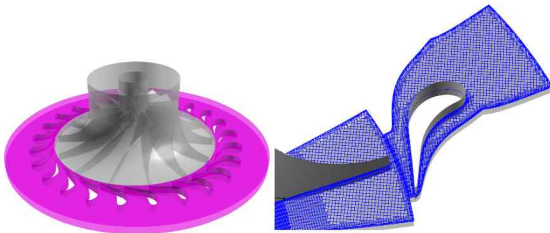


Fig. 1 Geometry and Mesh of Radial turbine

2.2 경계조건 및 난류모델

노즐 및 터빈의 주기면들은 cyclic GGI 경계조건을 사용하여 구현하였다. 노즐과 터빈의 경계면은 partial overlap GGI 경계조건을 사용하였다.[1] 이 조건은 형상이 일치하지 않는 경계면에서 경계면에 포함되지 않는 면들은 회전 주기조건을 사용하여 데이터를 복사하여 사용한다. Fig.2에 두 면의 반경 방향 속도 분포를 나타내었다. 두 면에서 속도와 스칼라 값들의 분포는 매우 정확히 일치하는 결과를 보여주지만 그림에서 나타났듯이 두 면이 겹치지 않는 부분에서 로터 입구면에서 약간 높은 속도를 나타내고 있다. 이 때문에 유량이 약 1~1.5% 정도의 오차를 보여주는데 이 부분은 추가적인 보완이 필요한 부분으로 생각된다.

터빈의 회전을 고려하기 위해서는 MRF 조건을 사용하였다.

노즐의 입구는 일정한 전압력 조건, 등엔트로피 전온도 조건, zero gradient 속도 조건을 사용하였으며, 터빈의 출구는 일정한 정압 조건, zero gradient 속도 및 압력 조건을 사용하였다. 터빈 출구의 압력이 일정하다는 조건보다는 출구 압력의 평균값이 일정하다는 조건이 더 타당하지만 현재 오픈폼의 버전에서 fixedMeanValue 경계조건은 병렬연산시 문제점이 있어 보완이 필요하다. 터빈 출구에서의 압력파의 반사를 없앨 수 있는 non-reflecting 경계조건은 아직 제공되지 않으며 추후 개발이 필요할 것으로 생각된다.

출구의 압력을 모르고 대신 입구의 전압력과 유량을 알고 있을 때 계산할 수 있는 target mass flow rate 경계조건도 개발이 필요할 것이다.

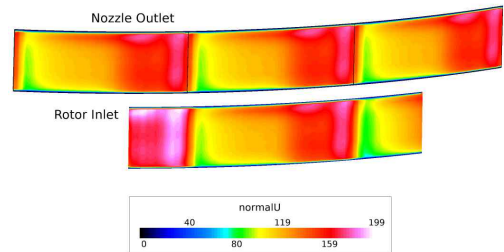


Fig. 2 Normal velocity contour at overlapGgi

난류모델은 SST k-omega 모델을 사용하였다. 입구에 k는 난류강도(intensity) 경계조건을 사용하였고 omega는 mixing length 경계조건을 사용하였다.

2.3 압축성 유동해석 솔버 및 해석 결과

압축성 유동해석 솔버로는 OpenFOAM Extend Project의 OpenFOAM-1.6-ext 버전에서 사용할 수 있는 Oliver Borm의 density based turbo 솔버를 사용하였다.[2] 이 솔버는 Jameson, Schmidt, Turket의 Runge-Kutta time-stepping 기법을 사용하는 압축성 유동 솔버이다.[3] Oliver Borm은 여러가지의 솔버를 제공하는데 그 중 transonicMRFDyMFoam이라는 솔버를 정상상태 솔버로 수정하여 사용하였다. 4차 Runge-Kutta 기법과 vanAlbadaSlope limiter를 사용하였고 gradient 및 divergence 계산은 Gauss linear 기법을 사용하였다.

압축성 유동 솔버 검증을 위해 실험결과가 공개되어 있는 RAE2822 transonic airfoil을 대상으로 계산을 우선 수행하였다. 사각형 경계층격자와 삼각형 원방격자로 구성된 36,000개 정도의 2차원 격자를 사용하였고 SST k-omega 난류 모델을 사용하였다. 속도는 마하수 0.729이다. Fig. 3과 Fig. 4에 압력분포와 NASA의 실험결과와 오픈폼의 계산 결과를 비교하여 나

타내었다.[4] 실험결과와 매우 일치하는 것을 확인할 수 있다.

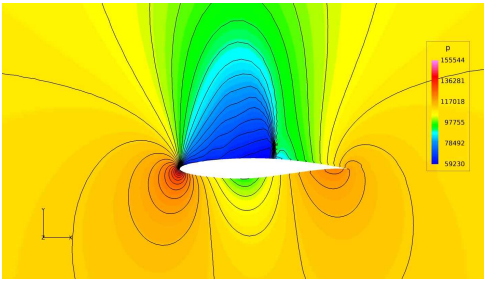


Fig. 3 Pressure contour of RAE airfoil

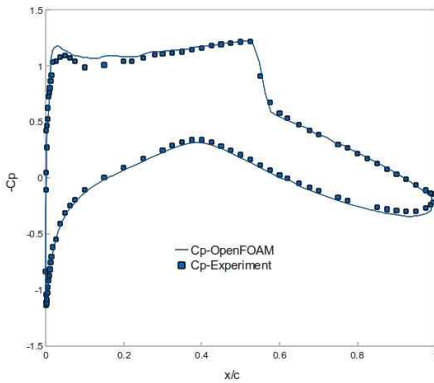


Fig. 4 Pressure coefficient of airfoil

노즐-터빈 해석결과는 검증할 수 있는 실험결과가 없어 상용 프로그램인 Fluent의 기존 결과와 비교하였으며 Fig. 5에 압력분포를 비교하여 나타내었다. Fluent의 계산은 turbo grid에서 만든 육면체 격자를 사용하였으며 density based implicit 솔버를 사용하였고 노즐-터빈의 경계면은 mixing plane 기법을 사용하여 계산한 결과이다.

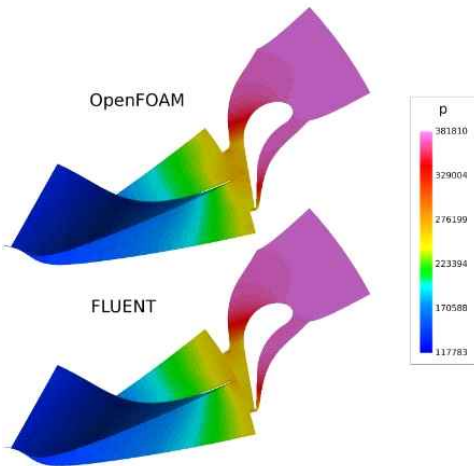


Fig. 5 Pressure contour of OpenFOAM and Fluent

압력분포는 전체적으로 거의 동일하게 나타나고 있으나 온도분포는 노즐-터빈 인터페이스면과 노즐의 trailing edge 부근에서 조금 다른 경향을 나타낸다. Fluent의 결과는 mixing plane면에서 유량이 정확히 일치하는데 유량을 보정하면서 부근의 온도분포가 일부 비연속적인 분포를 나타내는 부분이 존재하는 것으로 추정된다. 노즐의 trailing edge 부근은 비정상상태 특성이 강하게 나타나는 부분인데 정상상태 해석을 수행하면서 국부적으로 심한 압력이나 온도의 변화가 나타나는 것으로 보인다.

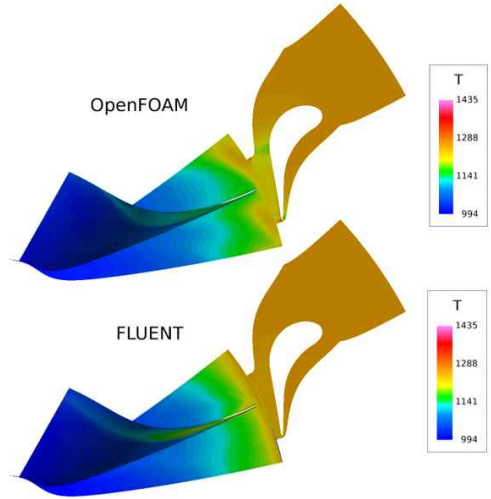


Fig. 6 Temperature contour of OpenFOAM and Fluent

터빈의 토크는 두 솔버에서 0.1% 이내의 오차 범위로 일치하는 값을 얻을 수 있었으며, 터빈의 효율은 약 1.1%의 차이를 두 솔버가 보여주었다.

2.4 병렬연산 기능

오픈폼의 병렬연산 기능은 최근 많이 사용되고 있는 상용 CFD 프로그램들의 기능과 거의 동일한 수준으로 발표되고 있다. 그러나 본 연구에서는 노즐과 터빈에 cyclicGGI가 한 쌍씩 있고 노즐과 터빈 경계면에 overlapGGI면이 있다. 이러한 경우 병렬연산 성능은 일반적인 유동해석보다 성능이 떨어질 수밖에 없을 것이다. 오픈폼에서 GGI 면들에 대해 병렬연산을 위해 GGI 면들을 선연해주는 것 외에 특별한 설정을 할 필요는 없는 것으로 알려져 있다. 그러나 본 연구의 결과 ggi와 cyclicGGI에서는 문제가 없으나 overlapGGI 면이 여러 개의 계산영역으로 분할될 때 문제점이 발생함을 알 수 있었다. 이런 문제를 해결하기 위해 두 개의 overlapGGI면을 하나의 계산영역에 포함되도록 설정해야 하며 patchConstrained 영역 분할 기법을 사용하면 간단히 해결할 수 있다.

격자수가 많지 않기 때문에 CPU 코어 수의 증가에 따른

성능 향상이 크게 나타날 수는 없지만 전체적인 병렬 성능은 인터페이스가 없는 문제에 비해 매우 떨어지는 것을 확인하였다. 이런 병렬 연산 성능의 저하는 다른 상용 코드들에서도 분명히 발생할 것인데 그 정도에 대한 평가는 추후에 진행할 예정이며, 병렬연산 성능 향상을 위한 방안의 마련이 필요할 것으로 판단된다.

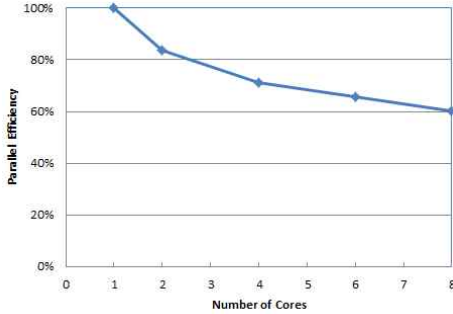


Fig. 7 Parallel performance, $\text{efficiency} = [\text{time using } N \text{ cores}] / [\text{time using } 1 \text{ core}]$

2.5 GUI 환경의 개발

오픈폼은 별도의 GUI 환경이 없기 때문에 사용에 불편한 점이 있다. 본 연구의 노즐-터빈 문제의 경우 격자생성은 OpenFOAM-2.1.x를 사용하고, 솔버는 OpenFOAM-1.6-ext를 사용해야 하는 번거로움이 추가되고, 두 격자를 합치고 인터페이스를 만들고, 후처리를 위한 각종 설정들이 추가되어 불편함이 더욱 커진다. 이러한 번거로운 작업들을 프로그래밍을 통해 자동화하고 그래픽 환경에서 변수들을 조정할 수 있도록 GUI 환경을 개발하였다. GUI 환경은 python을 사용하여 개발했으며 pyFoam, swak4Foam을 사용하여 사용의 편의성 및 후처리 기능을 보완하였다. 개발된 프로그램은 노즐-터빈, 압축기-디퓨저, 터빈 혹은 압축기 단독 해석시 사용할 수 있도록 개발되었다. snappyHexMesh를 사용한 격자의 생성을 지원하며 starCCM+, Fluent 형식의 격자 파일의 사용을 지원한다. 후처리는 paraview 혹은 Fieldview를 사용할 수 있다.

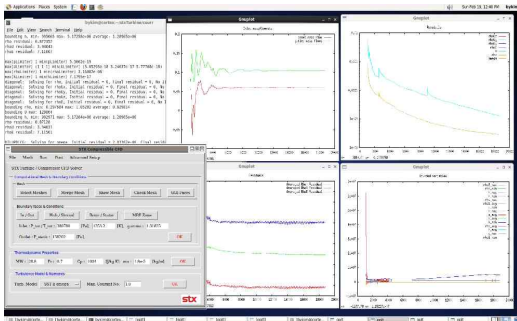


Fig. 8 GUI 환경

3. 결론

실제 가스터빈 엔진의 노즐-터빈 유동해석을 통해 오픈소스 CFD 코드인 오픈폼의 산업체 적용가능성을 평가하였다. 본 연구의 대상은 정확한 실험 결과가 없어 아직 정량적으로 해석 결과의 정확성을 분석할 수는 없다. 단지 상용 코드인 Fluent의 해석결과와 비교하여 상용코드와 유사한 수준의 결과를 얻을 수 있음을 확인할 수 있었다. 격자생성을 위한 snappyHexMesh는 인터페이스면 부근에서 세심한 주의가 필요하지만 원하는 수준의 격자를 얻을 수 있었다. 솔버와 터보기계해석을 위한 각종 기능들은 실제 산업체의 연구개발에 적용하기에 문제가 없음을 확인할 수 있었다.

Non-reflecting 경계조건, 입구에 유량과 전압력 조건을 동시에 사용할 수 있는 target mass flow rate 기법, 병렬성능의 개선 등이 이루어진다면 좀 더 좋은 결과를 빠른 시간에 얻을 수 있을 것으로 생각한다. Partial overlapggi 면에서의 유량 보정 및 mixing plane 조건의 검증도 추후 필요한 것으로 생각한다.

참고문헌

- [1] 2011, Hrvoje Jasak and Martin Beaudoin, "OpenFOAM turbo tools; From general purpose CFD to turbomachinery simulations", Proc. ASME-JSME-KSME Joint Fluid Engineering Conf. 2011, AJK2011-05015
- [2] 2011, Oliver Borm, Aleksandar Jemcov and Hans-Peter Kau, "Density based Navier Stokes solver for Transonic flows", 6th OpenFOAM workshop
- [3] 1981, Jameson, Schmidt and Turkel, "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA 1981-1259
- [4] www.grc.nasa.gov/WWW/wind/valid/raetaf/raetaf01/raetaf01.html