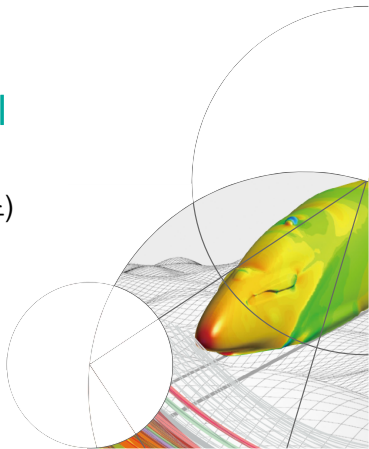


오픈 소스 코드를 이용한 터널내 화재 확산 예측 프로그램 개발

2015 한국전산유체공학회 춘계학술대회

김병윤, 길재흥(넥스트폼), 조현보(가온코스)



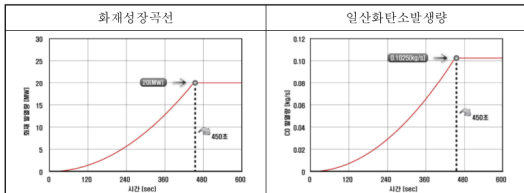
연구 배경

- 오픈 소스 코드를 이용한 사용자 맞춤형 코드 개발의 일환
 - 항공기 공력해석, 교반기, 건조기, 도장공장 환기해석, 선박 저항해석, 냉장고 ...
- 터널 내 차량 화재시 안전한 대피를 위한 연소 가스 확산 예측
- 터널 설계 및 엔지니어링 업체에서 많이 사용하고 있는 문제
- 대공간, 긴 시간에 대한 비정상상태 해석이 필요한 문제
- 보다 정밀하고 효율적인 예측을 위한 기반 기술 개발

Open  FOAM

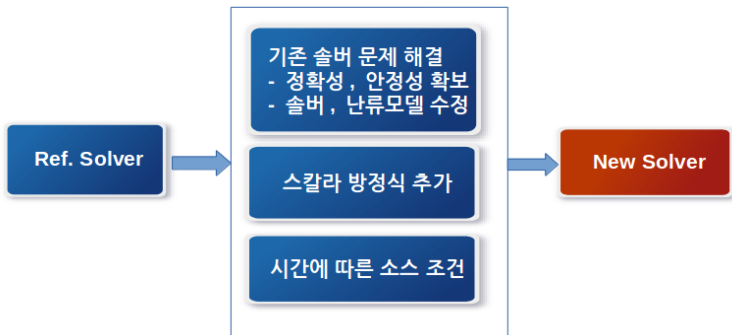
문제의 개요

- 비정상상태 난류 유동
- 부력이 포함된 열전달 및 연소가스에 대한 스칼라 계산
- 정상상태 계산을 통한 초기화
- 시간에 따른 에너지, 스칼라 소스항
- 솔버 개발 + 사용자 환경 개발



솔버 개발

- 개발 방법



- reference solver : buoyantPimpleFoam

Reference solver 수정

- 지배방정식 수정
 - Reference density(density의 volume average) 도입

$$\begin{aligned} \nabla \cdot (\rho \tilde{u} \tilde{u}) - \nabla \cdot \tau &= -\nabla p_{rgh} - (\tilde{g} \cdot \tilde{r}) \nabla \rho \\ p_{rgh} &= p - \rho(\tilde{g} \cdot \tilde{r}) \end{aligned}$$



$$\begin{aligned} \nabla \cdot (\rho \tilde{u} \tilde{u}) - \nabla \cdot \tau &= -\nabla p_{rgh} + (\rho - \rho_0) \tilde{g} \\ p_{rgh} &= p - \rho_0(\tilde{g} \cdot \tilde{r}) \end{aligned}$$

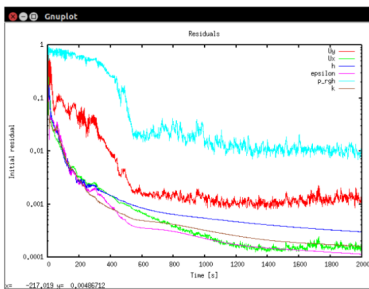
- 알고리즘 수정
 - 기존 : flux correction → pressure relax → velocity correction
 - 수정 : flux correction → velocity correction → pressure relax
- Operating Pressure에 따른 상대압력 사용
 - round-off error 최소화

kEpsilon 난류 모델 수정

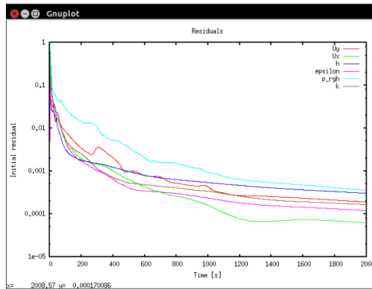
- 문제점
 - 안정성의 문제 - k, epsilon의 초기값에 매우 민감
 - 계산 도중 turbulent viscosity ratio 값이 매우 커지면서 발산
- k, epsilon의 최소값을 다르게 설정
 - 기존 : $k_{\text{Min}} = \epsilon_{\text{Min}} = 1e-15$
 - 수정 : $k_{\text{Min}} = 1e-14$, $\epsilon_{\text{Min}} = 1e-20$
- turbulent viscosity의 최소값, 최대값 설정
 - $\text{min} = 1e-4 * \text{laminar viscosity}$
 - $\text{max} = 1e5 * \text{laminar viscosity}$

솔버, 난류모델 수정 결과

- residual



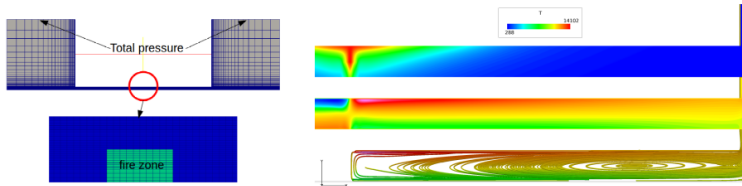
(a) 기존 코드



(b) 수정 코드

2D 단순 모델 해석 결과

- 길이 500m 2차원 덕트 중앙에서 차량 화재 조건
- 비현실적인 온도 분포
- 유동의 성층화가 나타나지 않음
- 난류 모델의 문제

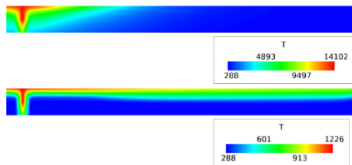
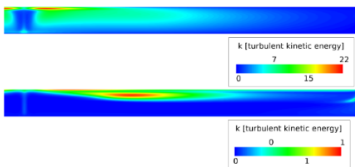


난류 모델의 수정

- 부력에 의한 난류에너지 생성항 추가

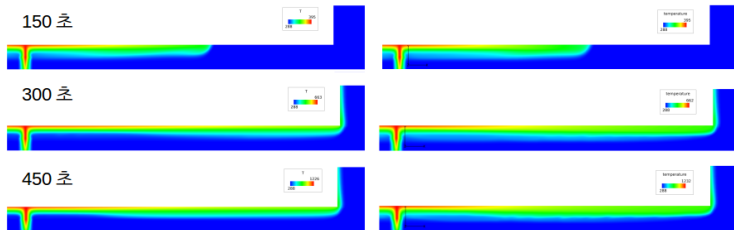
$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho k u_i)}{\partial x_i} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_t} \right) \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \epsilon$$

$$G_b = -g_i \frac{\mu_t}{Pr_t} \frac{\partial \rho}{\partial x_i}$$



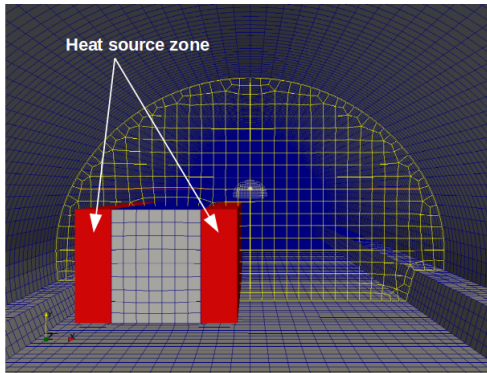
난류 모델의 수정

- 개발코드(좌)와 상용 코드(우) 온도 분포 비교



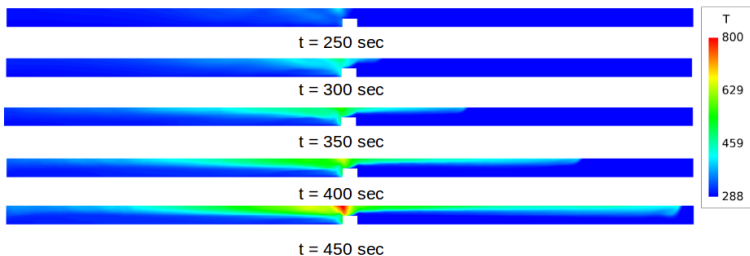
3차원 실제 모델 계산

- 형상 및 격자
 - STL 파일을 이용하여 cfMesh로 격자 생성

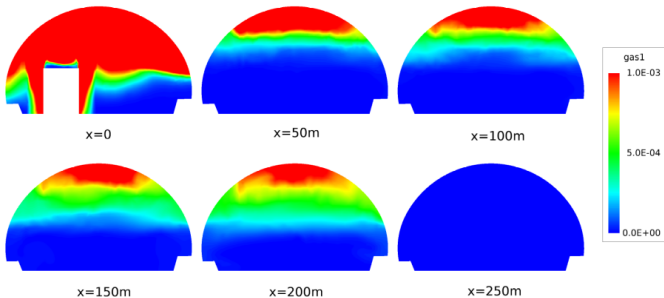
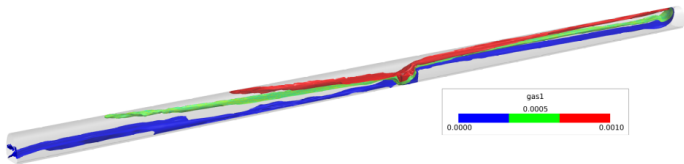


3차원 실제 모델 계산

- 계산 결과
 - 20MW 버스 화재 조건, 2% 경사
 - 초기 유입속도 = 2.5 m/s



3차원 실제 모델 계산



사용자 환경

NEXtfoam
open source CFD consulting

Mesh Setup

- create mesh
- import mesh
- read mesh
- check mesh
- boundary type

General Conditions

Tur. Model: standardKEpsilon

중력 방향: -y

터널 가울기 (%): 0

화재 강도 (MW): 20

화염 체적 (m3): 67.2

CO 발생량 (kg/s): 0.1025

유입 경계조건 (m/s): 0

터널 내 공기 온도 (K): 288

Flash over time (sec): 450

Inlet patch: end

Outlet patch: start

Fire zone: fire

Numerical Conditions

fvScheme setup

time: Euler

gradient: Gauss linear

laplacian: Gauss linear correc

snGradient: corrected

div. U: upwind

div. tur.: upwind

div. h: upwind

fvSolution setup

pSolver: GAMG

USolver: smoothSolver

hSolver: smoothSolver

momentumPredictor

nOuterCorrectors: 3

nCorrectors: 3

non-Ortho.Correctors: 0

relaxation factor: p|U|tur,|h

0.3 0.7 0.7 1.0

Run Conditions

start from: startTime

start Time: 0

end time: 450

time step size: 1

adjustTimeStep

maxCo: 1.0

maxDeltaT: 1

write control: adjustabl

write interval: 50

purge write: 0

write format: ascii

write precision: 6

Data Compression

Plot Residual

number of cores: 1

Parallel type: SMP

Steady Initialize

Start Transient Run

Stop

/home/bykim/Doing/TunnelFoam/GUI/tests/test5

결론

- 오픈 소스 기반으로 터널 내 화재 확산 예측 프로그램을 개발
- 효율적인 CFD 활용 및 정밀한 예측을 위한 기술 개발의 기반 마련
- 향후 개발 계획
 - 복사열전달 모델 추가
 - 화학종 전달 방정식 추가
 - 후처리 자동화 기능 개발

감사합니다

