Two-Layer Realizable k-ε 난류모델 개발

길재흥
넥스트폼

realizable k-ε 난류 모델

- 지배방정식
  - k-equation

$$\frac{\partial(\rho k)}{\partial t}\nabla\cdot\left(\rho\vec{U}k\right)-\nabla\cdot\left\{\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla k\right\}=G_k-\rho\varepsilon \qquad (1)$$

  - ε-equation
    - Realizable k-ε 모델은 standard k-ε 모델과 ε-equation의 생성항이 다름

$$\frac{\partial(\rho\varepsilon)}{\partial t}\nabla\cdot\left(\rho\vec{U}\varepsilon\right)-\nabla\cdot\left\{\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right\}=\rho C_1 S\varepsilon-\rho C_2\frac{\varepsilon^2}{k+\sqrt{\nu\varepsilon}} \qquad (2)$$

    - 생성항 Coefficients

$$\sigma_k=1.0,\ \sigma_\varepsilon=1.2$$

$$C_1=max\left[0.43,\frac{\eta}{\eta+5}\right],\quad \eta=S\frac{k}{\varepsilon},\quad S=\sqrt{2S_{ij}S_{ij}}$$

$$C_2=1.9$$

realizable k-ε 난류 모델

- 난류점성계수
  - Model equation

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{3}$$

  - C_μ
    - Standard k-ε 모델과 난류점성계수의 정의는 같지만 C_μ의 정의가 다음과 같이 다름

$$C_\mu = \frac{1}{A_0 + A_S \frac{kU^*}{\varepsilon}} \tag{4}$$

    - C_μ Coefficients

$$A_0 = 4.04$$

$$A_S = \sqrt{6}\cos\varphi, \quad \varphi = \frac{1}{3}\cos^{-1}\left(\sqrt{6}W\right), \quad W = \frac{S_{ij}S_{jk}S_{ki}}{\tilde{S}^3}, \quad \tilde{S} = \sqrt{S_{ij}S_{ij}}$$

$$U^* \equiv \sqrt{S_{ij}S_{ij} + \tilde{\Omega}_{ij}\tilde{\Omega}_{ij}}, \quad \tilde{\Omega}_{ij} = \Omega_{ij} - 2\epsilon_{ijk}\omega_k, \quad \Omega_{ij} = \overline{\Omega}_{ij} - \epsilon_{ijk}\omega_k$$

- $\overline{\Omega}_{ij}$: mean rotation rate viewed in rotating reference frame
- ω_k : angular velocity of rotating reference frame
- ϵ_{ijk} : Levi-Civita symbol
- $2\epsilon_{ijk}\omega_k$: Omitted in OpenFOAM(not compatible with MRF or Sliding Mesh)

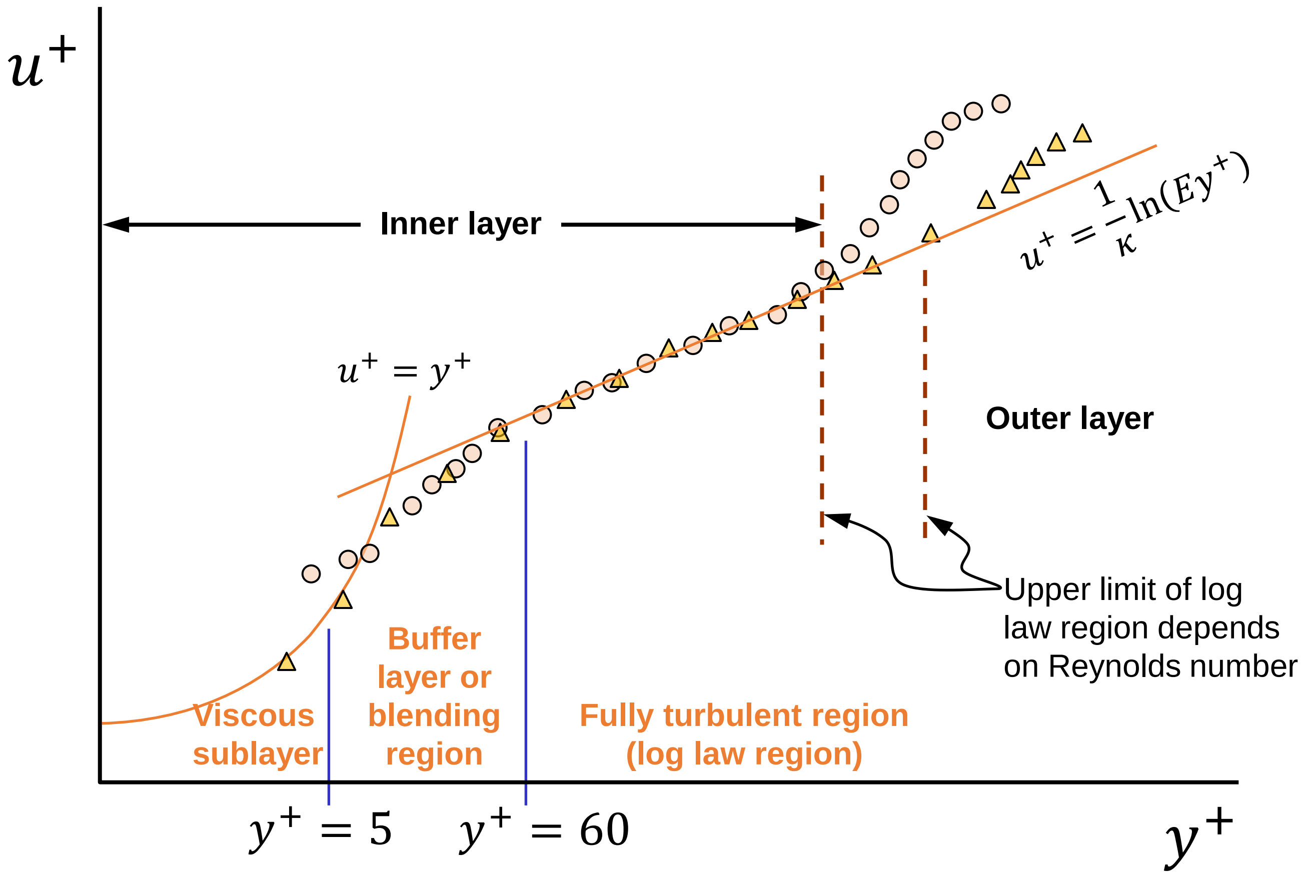
Realizable k-ε 난류 모델

- 특징
  - Standard k-ε 모델에 비해 jet 유동의 spreading rate를 잘 예측
  - 유동박리(separation) 또는 역압력구배(adverse pressure gradient)가 존재하는 경계층 유동을 더 잘 예측
  - 회전(Rotation), 재순환(recirculation) 유동을 더 잘 예측

- 한계
  - Standard k-ε 모델과 마찬가지로 High-Re 난류 유동에 적용되는 모델이므로 벽면 근처의 유동을 예측하기에 적합하지 않음

Realizable k-ε 난류 모델의 한계

- The Universal Law of The Wall
  - 압력구배가 없는 경계층내의 속도분포는 다음과 같이 일반화 된다.



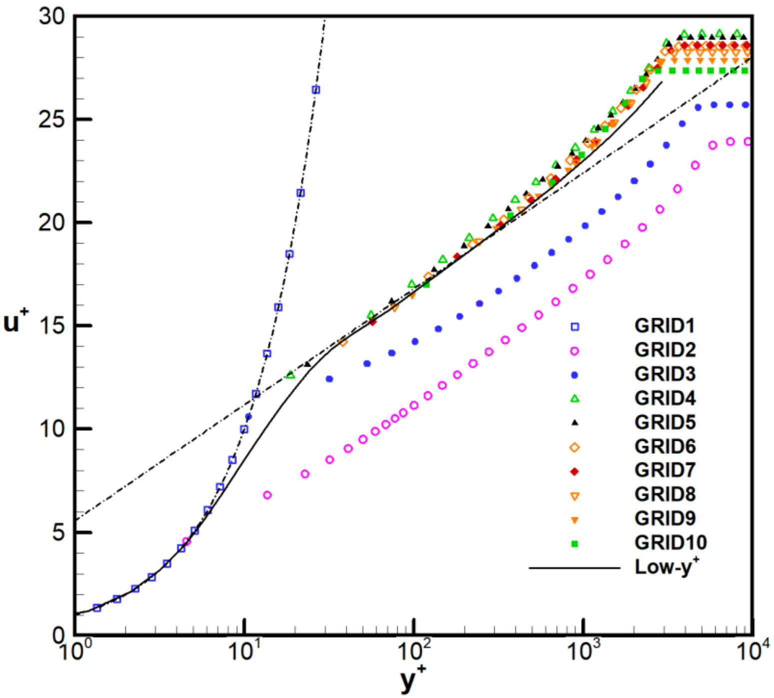
$$u^+ = \frac{u}{u_\tau}$$

$$y^+ = \frac{u_\tau y}{\nu}$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

Realizable k-ε 난류 모델의 한계

- 벽근처 첫번째 격자의 높이 (y^+) 에 따른 속도분포 계산
  - 첫번째 격자의 y^+가 30 미만인 격자계에서는 속도분포를 올바르게 예측할 수 없음



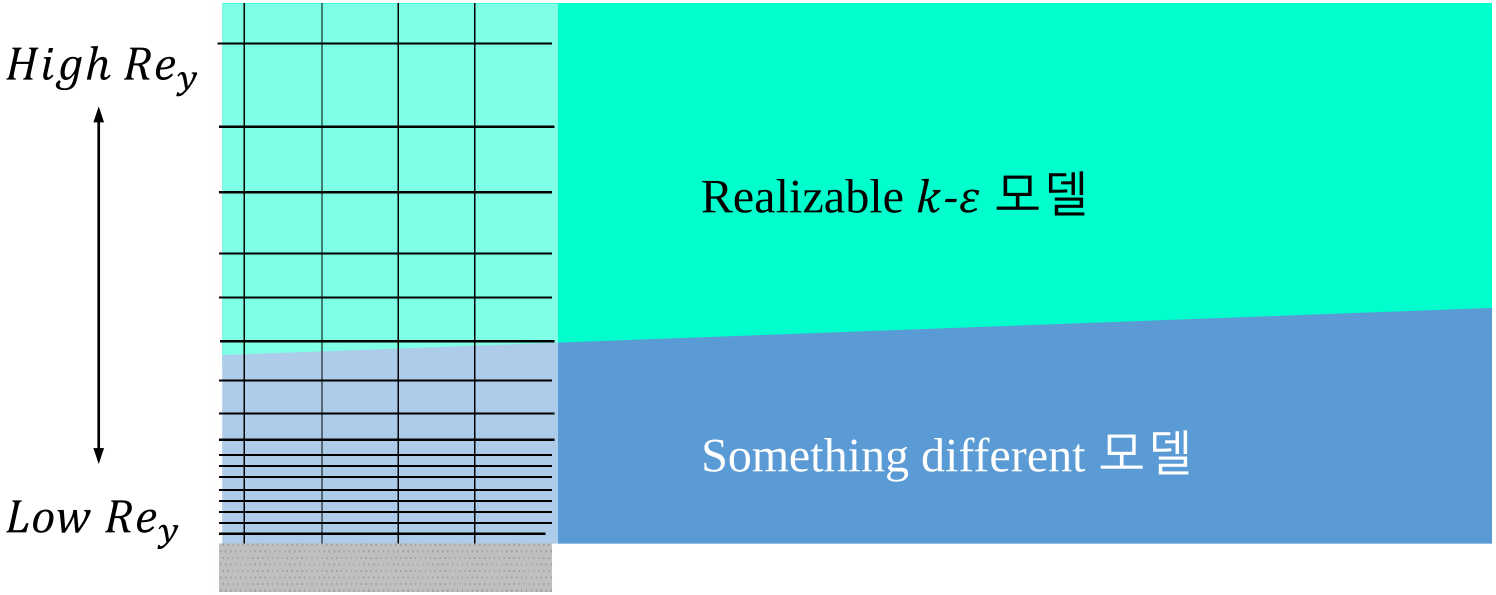
평판 위를 흐르는 난류 유동에 대한 OpenFOAM 해석 결과

Two-Layer Realizable k-ε 난류 모델

- y^{+}가 30 미만인 격자계에서 경계층 내의 속도분포를 올바르게 예측할수 없는 Realizable k-ε 모델에 대한 개선 모델
- 경계층을 높이방향으로 두 영역으로 나누어 각 영역에 적절한 난류모델을 적용는 방식
  - 대표적인 상용코드(STAR-CCM+, FLUENT 등)에 구현되어 있음
  - Standard OpenFOAM에는 구현되어 있지 않음

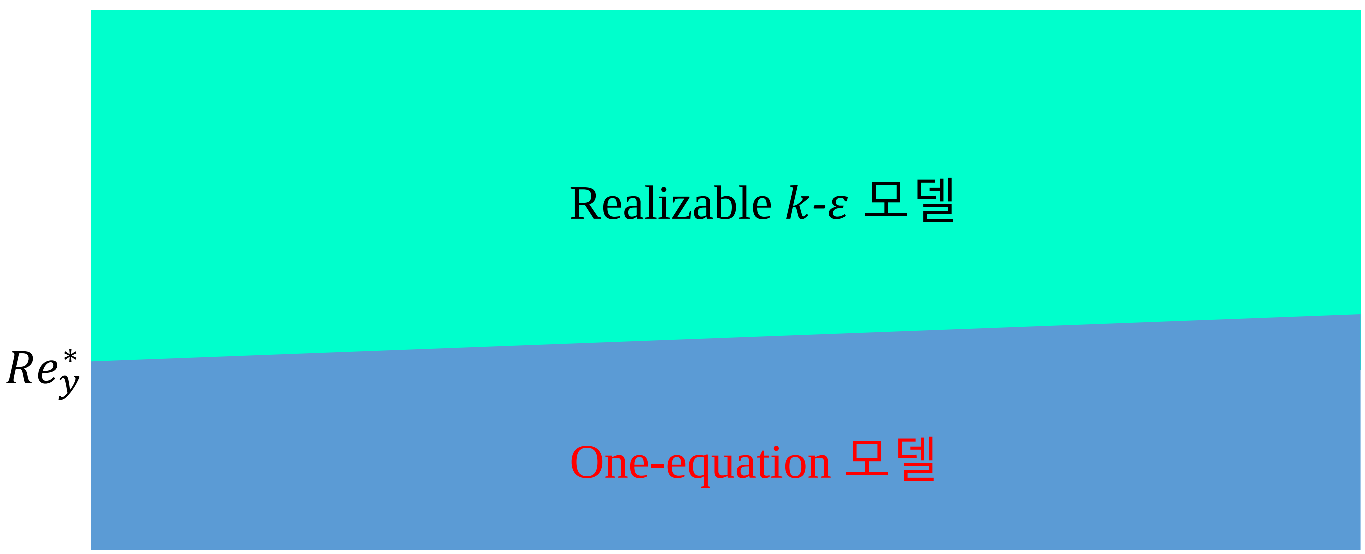
Two-Layer Realizable k-ε 난류 모델

- Two-Layer formulation
  - 경계층을 $Re_y \left(= \frac{y\sqrt{k}}{\nu}\right)$를 기준으로 두 층으로 나누어 서로 다른 난류모델 적용



Two-Layer Realizable k-ε 난류 모델

- Wolfstein Two-Layer Model
  - 부력이 지배적이지 않은 유동에 적합
  - 대표적인 상용코드에서 공통적으로 사용
  - Re_y^*($Re_y = 60\sim200$) 를 기준



Wolfstein **Two-Layer Realizable** k-ε **난류모델**

- 지배방정식
  - $Re_y > Re_y^*$ 일 경우
    - Realizable k-ε 모델의 지배방정식 사용 (식 (1) 및 식 (2))
  - $Re_y < Re_y^*$ 일 경우
    - k-equation
      - Realizable k-ε 모델의 k-equation 과 동일 (식 (1))
    - **ε-equation**
      - **Algebraic relation with k using length scale function l_ε**

$$\boldsymbol{\varepsilon = \frac{k^{3/2}}{l_\varepsilon}} \qquad (5)$$

$$l_\varepsilon = yC_l^*\left[1 - e^{-Re_y/A_\varepsilon}\right]$$

$$C_l^* = \kappa C_\mu^{-3/4}, \quad A_\varepsilon = 2C_l^*$$

Wolfstein **Two-Layer Realizable** k-ε **난류모델**

- 난류점성계수
  - $Re_y > Re_y^*$ 일 경우
    - Realizable k-ε 모델의 난류점성계수 모델식을 사용 (식 (3))
  - $Re_y < Re_y^*$ 일 경우
    - **Algebraic relation with k using length scale function l_μ**

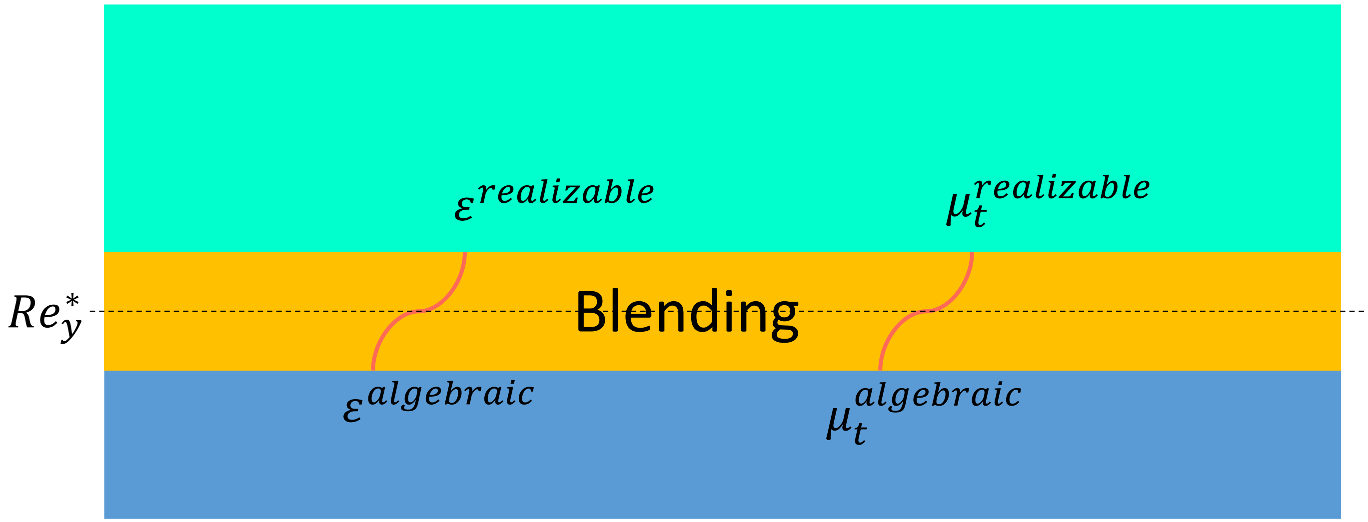
$$\mu_t = \rho C_\mu l_\mu \sqrt{k} \qquad (6)$$

$$l_\mu = y C_l^* \left[1 - e^{-Re_y/A_\mu}\right]$$

$$C_l^* = \kappa C_\mu^{-3/4}, \quad A_\mu = 70$$

Wolfstein **Two-Layer Realizable** k-ε **난류모델**

- ε 및 μ_t 의 blending
  - Re_y^* 의 경계에서 갑작스런 해의 변화를 방지하기 위해 blending 필요



  - Blending factor, λ

$$\boldsymbol{\lambda} = \frac{\mathbf{1}}{\mathbf{2}}\left[\mathbf{1} + \tanh\left(\frac{Re_y - Re_y^*}{A}\right)\right]$$

$$A = \frac{|\Delta Re_y|}{\tanh^{-1}(0.98)}, \quad |\Delta Re_y| = 3 \sim 40$$

NEXT*foam* open source CFD consulting

Wolfstein **Two-Layer Realizable** k-ε **난류모델**

- Blending factor를 적용하여 표현한 μ_t
  - 난류점성계수 (μ_t) 는 k 와 ε 에 대한 선형대수 방정식을 풀어서 얻은 해를 이용하여 계산 되기 때문에 해석 영역의 모든 격자에서 아래와 같은 식을 적용하여 계산하면 된다.

$$\mu_t = \lambda \mu_t^{realizable} + (1 - \lambda)\mu_t^{algebraic}$$

- Blending factor를 적용하여 표현한 ε
  - 이산화한 방정식 레벨의 blending
    - ε 의 경우는 선형대수 방정식을 풀어서 구해지는 값이므로 선행대수 방정식으로 구성되는 행렬의 모든 요소에 blending factor를 적용하여 표현해야 한다.

Discretization of Blended ε-equation

- $Re_y > Re_y^*$ 일 경우

$$\frac{\boldsymbol{a_P}}{\boldsymbol{\alpha}}\boldsymbol{\varepsilon_P^*} + \sum_{\boldsymbol{N}} \boldsymbol{a_N}\,\boldsymbol{\varepsilon_N^*} = \boldsymbol{b} + \frac{\mathbf{1}-\boldsymbol{\alpha}}{\boldsymbol{\alpha}}\boldsymbol{a_P}\boldsymbol{\varepsilon_P^{n-1}} \quad (7)$$

a_P : diagonal coefficient of discretize equation

a_N : off-diagonal coefficient of discretize equation

b : source vector component of discretize equation

α : under-relaxation factor

$\varepsilon_P^*, \varepsilon_N^*$: solution on current iteration for cell $\boldsymbol{P}$ and it's neighbour $\boldsymbol{N}$

ε_P^{n-1} : solution on previous iteration for cell $\boldsymbol{P}$

- $Re_y < Re_y^*$ 일 경우

$$\boldsymbol{\varepsilon_P^*} = \boldsymbol{\alpha}\left(\boldsymbol{\varepsilon_P^s} - \boldsymbol{\varepsilon_P^{n-1}}\right) + \boldsymbol{\varepsilon_P^{n-1}} \quad (8)$$

$\varepsilon_P^s = \dfrac{k^{3/2}}{l_\varepsilon}$: algebraic relation with current solution of k

Discretization of Blended ε-equation

- 식(7)과 식(8)로부터

$$\boldsymbol{\varepsilon_P^{realizable}} = \frac{\boldsymbol{\alpha}}{\boldsymbol{a_P}}\left(\boldsymbol{b} + \frac{\boldsymbol{1-\alpha}}{\boldsymbol{\alpha}}\boldsymbol{a_P\varepsilon_P^{n-1}} - \sum_{\boldsymbol{N}}\boldsymbol{a_N\,\varepsilon_N^*}\right) \qquad (9)$$

$$\boldsymbol{\varepsilon_P^{algebraic}} = \boldsymbol{\alpha\left(\varepsilon_P^s - \varepsilon_P^{n-1}\right) + \varepsilon_P^{n-1}} \qquad (10)$$

  – Blending factor, λ 를 이용해 식(9)와 식(10)을 결합하면 다음과 같고,

$$\varepsilon_P^* = \lambda\varepsilon_P^{realizable} + (1-\lambda)\varepsilon_P^{algebraic}$$

$$= \lambda\left[\frac{\alpha}{a_P}\left(b + \frac{1-\alpha}{\alpha}a_P\varepsilon_P^{n-1} - \sum_N a_N\,\varepsilon_N^*\right)\right] + (1-\lambda)[\alpha(\varepsilon_P^s - \varepsilon_P^{n-1}) + \varepsilon_P^{n-1}]$$

  – 위의 식을 정리하면 다음과 같이 blended ε-equation 의 이산화된 식을 얻을 수 있다.

$$\boldsymbol{\frac{a_P}{\alpha}\varepsilon_P^* + \lambda\sum_N a_N\,\varepsilon_N^* = \lambda\left(b + \frac{1-\alpha}{\alpha}a_P\varepsilon_P^{n-1}\right) + (1-\lambda)a_P\left(\varepsilon_P^s + \frac{1-\alpha}{\alpha}\varepsilon_P^{n-1}\right)} \qquad (11)$$

Implementation into OpenFOAM

- Realizable k-ε 난류모델의 소스코드
  - 식 (7)과 같은 형태로 ε-equation 을 이산화 하는 코드

```
    // Dissipation equation
    tmp<fvScalarMatrix> epsEqn
    (
        fvm::ddt(alpha, rho, epsilon_)
      + fvm::div(alphaRhoPhi, epsilon_)
      - fvm::laplacian(alpha*rho*DepsilonEff(), epsilon_)
     ==
        C1*alpha*rho*magS*epsilon_
      - fvm::Sp
        (
            C2_*alpha*rho*epsilon_/(k_ + sqrt(this->nu()*epsilon_)),
            epsilon_
        )
      + epsilonSource()
      + fvOptions(alpha, rho, epsilon_)
    );

    epsEqn.ref().relax();
```

Implementation into OpenFOAM

- Realizable k-ε 난류모델의 소스코드를 활용
  - 식 (11)과 같은 형태로 ε-equation 을 이산화 하기 위해 이산화 함수들(`fvm::ddt`, `fvm::div`, `fvm::laplacian` 등)을 재정의 하기 보다는 식 (7)의 형태로 이산화해서 얻은 행렬시스템의 요소들을 식 (11)의 형태로 수정하는 방식으로 구현

```
// Dissipation equation
tmp<fvScalarMatrix> epsEqn // 식(7)의 형태로 이산화된 행렬시스템 객체
(
    // 생략
);

// Blend with one-equation turbulence model near the wall
{
    dimensionedScalar Aeps = 2.0*kappa_/Cmu75_;
    volScalarField::Internal leps
    (
        y_()*kappa_/Cmu75_*(1 - exp(-Rey_/Aeps))
    );
    volScalarField::Internal es(pow(k_(), 1.5)/leps);

    blendAndRelax(epsEqn.ref(), lambda_, es);
}
```

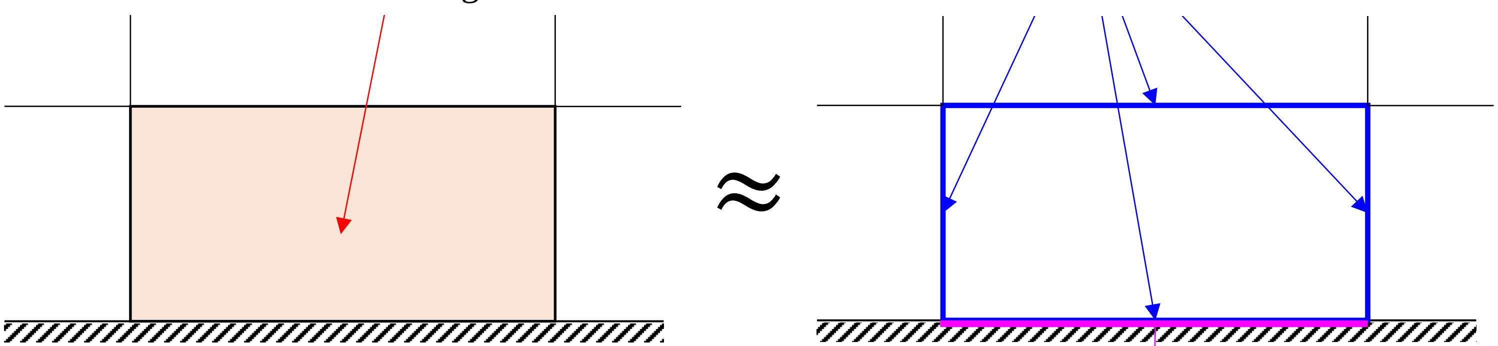
Implementation into OpenFOAM

- Two-Layer Realizable k-ε 모델이 올바르게 구현되기 위해서는 다음과 같은 벽면 경계조건도 적절하게 구현되어야 한다.
  - 운동량보존 방정식에 대한 벽면 경계조건
  - k-equation 에 대한 벽면 경계조건
  - ε-equation 에 대한 벽면 경계조건

Implementation into OpenFOAM

- 운동량보존 방정식에 대한 벽면 경계조건
  - 확산항 이산화에 의해 벽면 격자에 momentum source로 작용
    - 층류유동이거나 벽면 격자가 점성저층(viscous sublayer) 안에 있을 경우에는 no-slip 조건 이외에 별도의 경계조건 처리가 필요하지 않음

$$\underbrace{\int \nabla\cdot\bar{\bar{\tau}}\,dV}_{\text{volume integral}} = \oint \bar{\bar{\tau}}_f\cdot d\vec{S} \approx \underbrace{\sum_f \bar{\bar{\tau}}_f\cdot\vec{S}_f = \sum_f \left[(\mu+\mu_t)\nabla\vec{U}\right]_f\cdot\vec{S}_f}_{\text{summation over the control volume surface}}$$



at wall surface

$\mu_t = 0$ for laminar or in viscous sublayer

$$\bar{\bar{\tau}}_f\cdot\vec{S}_f = \left(\mu\nabla\vec{U}\right)_f\cdot\vec{S}_f \Leftrightarrow \underbrace{\tau_w\cdot\vec{S}_w = (\mu+\mu_t)_w\left(\frac{du}{dy}\right)_w S_w}_{\text{momentum source}}$$

❖ 하첨자 f 는 control volume surface에서의 값을 나타냄
❖ 하첨자 w 는 벽면에서의 값을 나타냄

Implementation into OpenFOAM

- 운동량보존 방정식에 대한 벽면 경계조건
  - 난류 유동에서 벽면 격자가 점성저층보다 위에 있을 경우
    - $\tau_w = (\mu + \mu_t)_w \left(\frac{du}{dy}\right)_w$ 의 식이 성립하지 않음
    - 이 경우 확산항 이산화에 의해서 벽면 격자에 올바른 momentum source가 적용되기 위해서는 벽면 격자에 한해서만 이산화 방법을 변경하거나 이미 이산화된 행렬시스템에서 벽면 격자에 적용된 생성항을 수정해야 함
  - OpenFOAM에서는 운동량 보존방정식의 이산화 과정에서 벽면 격자에서만 다른 방법으로 처리를 하는 번거로움을 피하기 위해 벽면의 난류점성계수 $(\mu_t)_w$ 를 정의하여 사용
    - $(\mu_t)_w$ 는 운동량보존방정식에 대한 경계조건을 부여하는 수치적 방법일 뿐 물리적인 의미는 없음

Implementation into OpenFOAM

- 운동량보존 방정식에 대한 벽면 경계조건
  - Spalding's formula
    - Viscous-sublayer, buffer-layer 및 log-layer의 속도분포를 하나의 식으로 통합

$$\boldsymbol{y^+ = u^+ + \frac{1}{E}\left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{(\kappa u^+)^2}{2} - \frac{(\kappa u^+)^3}{6}\right]}$$

$$\Rightarrow \quad y = \frac{\nu u}{u_\tau^2} + \frac{\nu}{u_\tau E}\left[e^{\kappa u/u_\tau} - 1 - \frac{\kappa u}{u_\tau} - \frac{1}{2}\left(\frac{\kappa u}{u_\tau}\right)^2 - \frac{1}{6}\left(\frac{\kappa u}{u_\tau}\right)^3\right] \qquad (12)$$

    - 위의 식에서 Newton-Raphson method를 이용해 u_τ 를 구하면 확산항 이산화에 의해 표현된 식으로부터 다음과 같이 $(\mu_t)_w$를 표현할 수 있다.

$$\tau_w = \rho u_\tau^2 = (\mu + \mu_t)_w \left(\frac{du}{dy}\right)_w$$

$$\therefore \quad (\mu_t)_w = \frac{\rho u_\tau^2}{\left(\frac{du}{dy}\right)_w} - (\mu)_w \quad \boldsymbol{or} \quad \boldsymbol{(\nu_t)_w = \frac{u_\tau^2}{\left(\frac{du}{dy}\right)_w} - (\nu)_w}$$

Implementation into OpenFOAM

- k-equation 에 대한 벽면 경계조건
  - 난류 유동에서 벽면 격자가 점성저층보다 위에 있을 경우
    - 벽면 격자에서 k-equation 의 생성항, $G_k = \mu_t S^2$, 의 식이 성립하지 않음
    - 벽면에 대한 경계조건을 zero-gradient 로 적용하여 k-equation 을 이산화 한 후 벽면 격자에 해당하는 생성항 요소를 다음과 같이 수정

$$\boldsymbol{G_k = g\mu_t S^2 + (1-g)\rho u_\tau^2 \frac{\partial u}{\partial y}}$$

$$where \;\; g = exp\left(-\frac{Re_y}{11}\right)$$

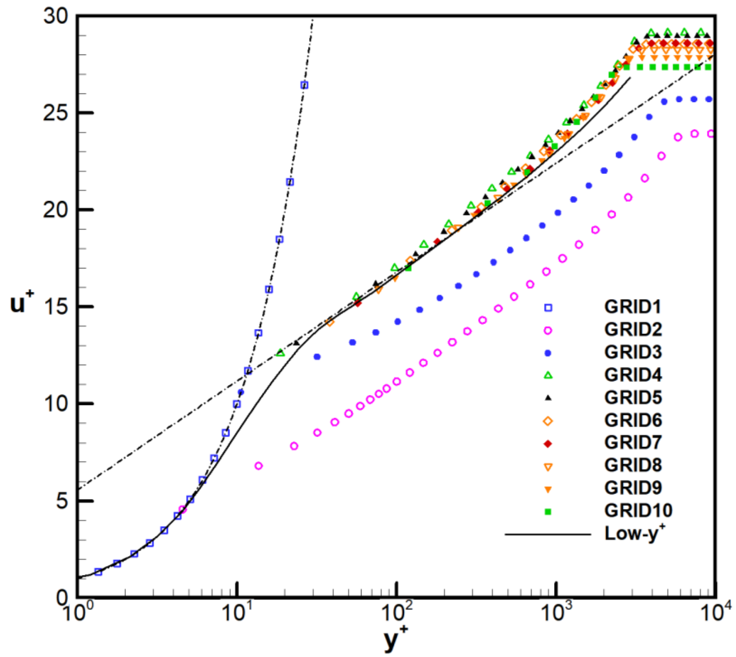
- ε-equation 에 대한 벽면 경계조건
  - 난류 유동에서 벽면 격자가 점성저층보다 위에 있을 경우
    - 벽면에 대한 경계조건을 zero-gradient 로 적용하여 ε-equation 을 이산화 한 후 벽면 격자에 해당하는 행렬요소의 값을 다음 식과 같이 수정
      - 행렬시스템을 inversion하여 해를 구하여도 벽면 격자에 해당하는 해는 아래 식으로 주어진 값을 유지하도록 함

$$\boldsymbol{\varepsilon_P = g\frac{2\nu k}{y^2} + \frac{k^{3/2}}{l_\varepsilon}}$$

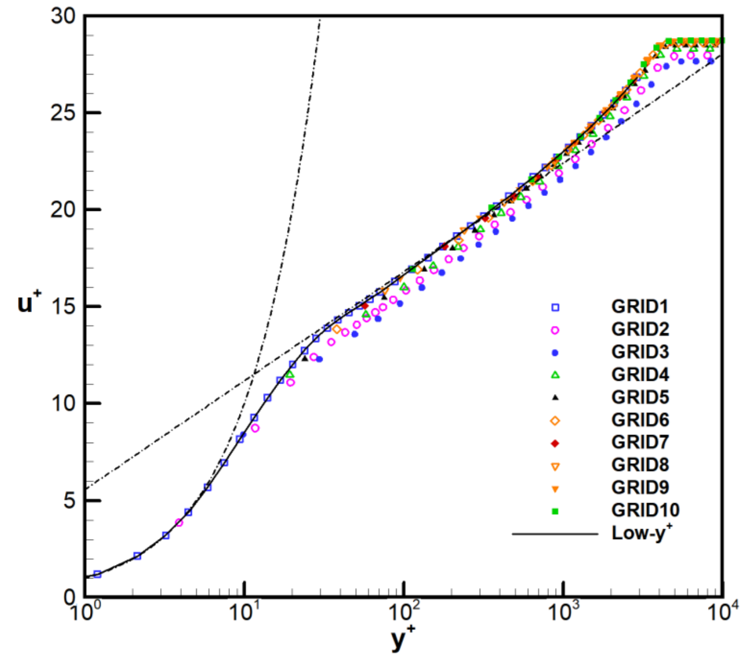
Results

- 개발된 Two-Layer Realizable k-ε 난류모델을 이용한 벽근처 첫번째 격자의 높이 (y^+) 에 따른 속도분포 계산 검증
  - y^+ 30 미만의 격자에서도 속도분포를 적절히 예측함을 확인

Realizable k-ε



Two-Layer Realizable k-ε



평판 위를 흐르는 난류 유동에 대한 OpenFOAM 해석 결과