Tsinghua University

Steady RANS Simulation of the TUDa-GLR-OpenStage Using the Open-source Code SU2

Xuedong Zheng[1], Chuanxiang Yan[1], Xiao He[2], Baotong Wang[1],

1. Tsinghua University, Beijing, China
2. Imperial College, London, UK

2024.09.04

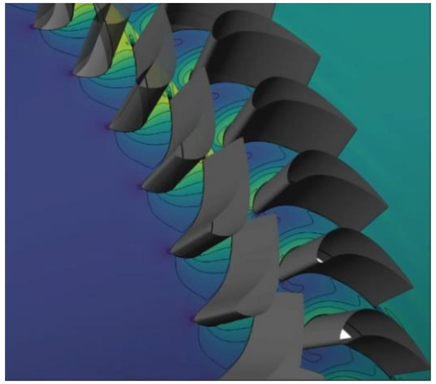
Content

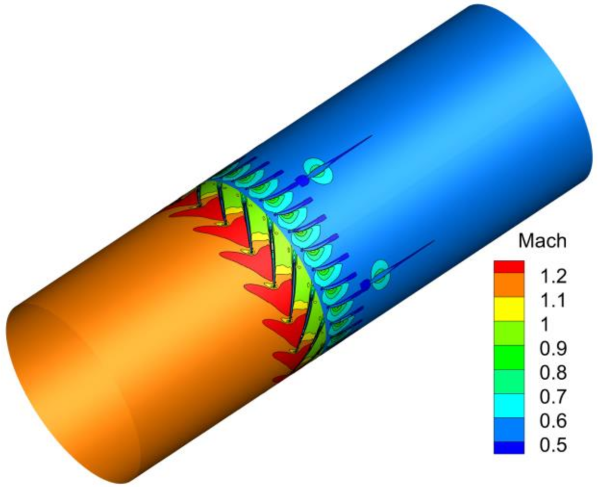
- The open-source code SU2
- Computation Setup
- Overall Performance
- Radial Profiles

The Open-Source Code SU2

- **Features of SU2**
  - **Adjoint Optimization**
  - **Compressible Flow from start**
  - **Unstructured grid**
  - **Dynamic Mesh**

- **Turbo Features of open SU2 up to now**
  - **Multi-stage RANS**
  - **Axial/Radial turbomachinery**
  - **Harmonic Balance**
  - **Body Force**


Aachen Turbine*



TUDa-GLR-OpenStage

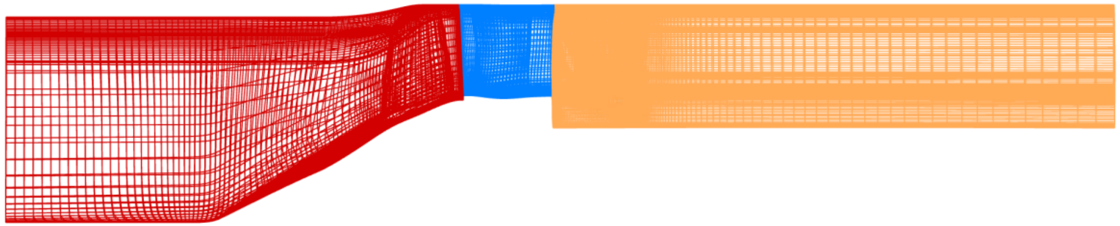
Validated with Aachen Turbine case and NASA Rotor 67/Stage 35 and some cascade configurations on aerodynamics and aeroelasiticity**

*Vitale, S., M. Pini, and Piero Colonna. "Multistage turbomachinery design using the discrete adjoint method within the open-source software su2." *Journal of Propulsion and Power* 36.3 (2020): 465-478.
**Yan, C., Wang, B., He, X., Zhao, F., Zheng, X., Vahdati, M., and Zheng, X. "Extension and Validation of the Turbomachinery Capabilities of SU2 Open Source Computational Fluid Dynamic Code." . Journal of Turbomachinery.(2024); 146(6).

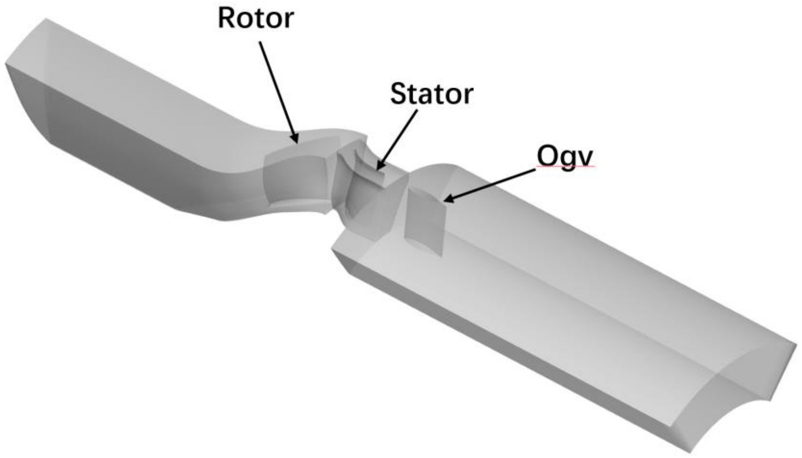
Computation Setup

☐ Grids

- **Geometry with rotor pinch included**

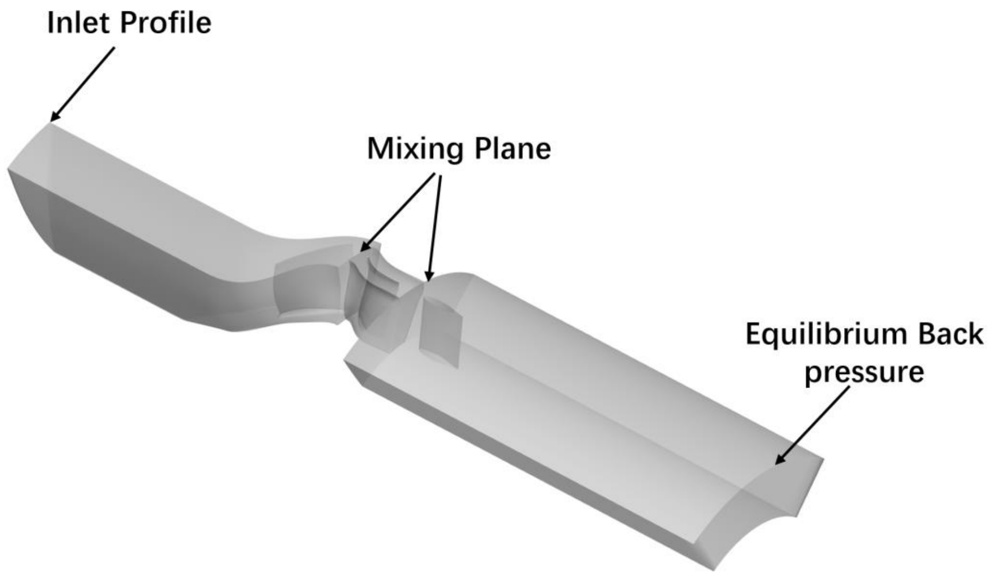


	Ultracoarse	Coarse	Medium	Fine	Ultrafine
Rotor	0.11	0.30	1.05	3.29	11.60
Stator	0.04	0.15	0.51	1.76	5.71
OGV			8.24		



Computation Setup

- **RANS Solver**
  - JST convective scheme
  - FGMRES linear solver
  - Turbulence model：SST-2003/1994, SA, SA-PGOmega*
  - Without wall function



- **Boundary Conditions**
  - Inlet: official inlet profile
  - Outlet: radial equilibrium static pressure
  - Mixing Plane Interface

*He, X., Zhao, F., and Vahdati, M. (September 19, 2022). "**A Turbo-Oriented Data-Driven Modification to the Spalart–Allmaras Turbulence Model**."

Computation Setup

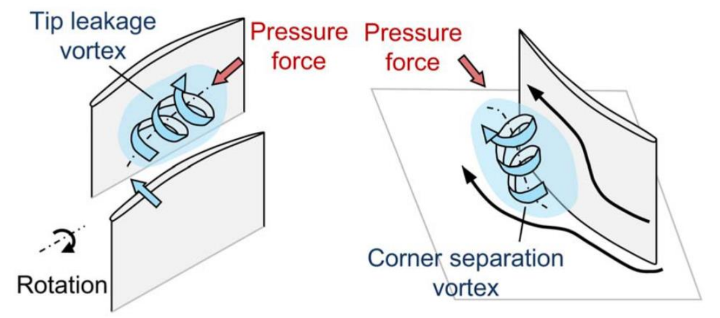


Fig. 1 Illustration of compressor blockage cells

◻ PGOmega

He, X., Zhao, F., and Vahdati, M. (September 19, 2022). "**A Turbo-Oriented Data-Driven Modification to the Spalart–Allmaras Turbulence Model**."

2.3 Proposal of SA-PG$_\omega$ Model. To start with, the flow topologies of compressor blockage cells are illustrated in Fig. 1. A compressor blockage cell is typically formed by the tip leakage vortex or the corner separation vortex, both of which are featured by 3D swirling motion, flow stagnation, and adverse pressure gradient. In the blockage cells, the relative vorticity vector is almost aligned with the pressure gradient vector, which generally points from the upstream suction surface (SS) to the downstream pressure surface (PS). Based on these observations, an empirical compressor blockage identifier is proposed in Eq. (8)

$$\hat{p}_\omega = \left| \frac{\partial p}{\partial x_i} \frac{\omega_i}{\Omega} \right| \Big/ \frac{\rho^2 W^3}{\mu \mathrm{Re}_{ref}} \tag{8}$$

Finally, the compressor blockage identifier $\hat{p}_\omega$ is applied to boost eddy viscosity of the original SA model by multiplying the modified strain rate $\hat{S}$ with the $\beta_{p\omega}$ function

$$\beta_{p\omega}(\hat{p}_\omega) = \begin{cases} 1 + c_{p\omega 1} \tanh\left(c_{p\omega 2} \hat{p}_\omega^{c_{p\omega 3}}\right) & \text{if } \dfrac{\partial p}{\partial x_i} w_i \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

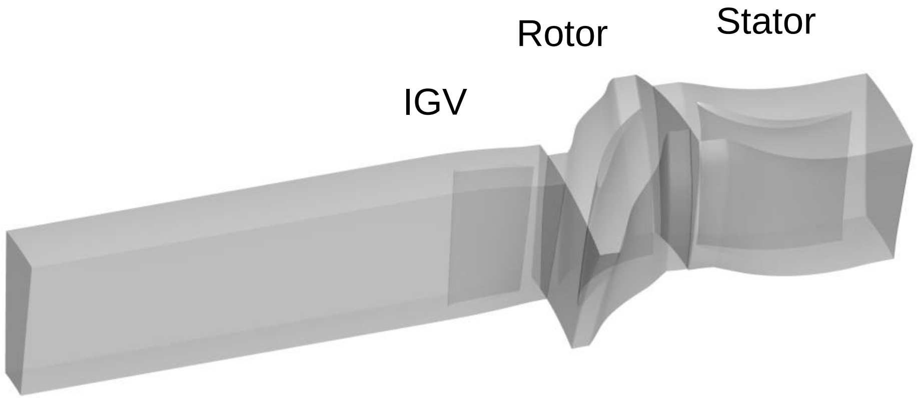
Computation Setup

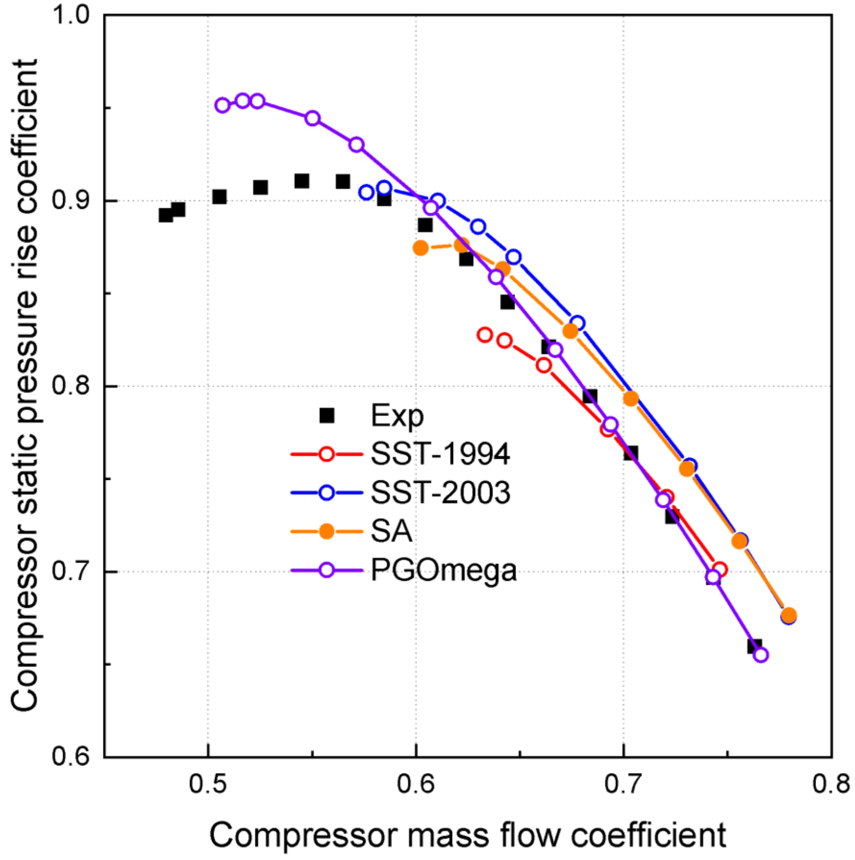
☐ PGOmega

```cpp
/*--- Evaluate Omega with a rotational correction term. ---*/
// pgomega by Xiao He 2022
  if (options.pgomga) {

    /*!\brief  PG-OMEGA*/
    su2double dpds = 0, dpomg = 0, omg = 0, velocityMag = 0;
    su2double Velocity_Rel[3] = {0.0, 0.0, 0.0};
    su2double Vorticity_Rel[3] = {0.0, 0.0, 0.0};
    su2double RotationalVelocity[3] = {0.0, 0.0, 0.0};
    su2double RotationVelocityCrossR[3] = {0.0, 0.0, 0.0};
    for (int iDim = 0; iDim < nDim; iDim++) {
      RotationalVelocity[iDim] = config->GetRotation_Rate(iDim) / config->GetOmega_Ref();
    }

    GeometryToolbox::CrossProduct( RotationalVelocity,Coord_i, RotationVelocityCrossR);

    for (int iDim = 0; iDim < nDim; iDim++) {
      Velocity_Rel[iDim] = V_i[idx.Velocity() + iDim] - RotationVelocityCrossR[iDim];
      Vorticity_Rel[iDim] = Vorticity_i[iDim] - RotationalVelocity[iDim];

    }

    velocityMag = max(GeometryToolbox::Norm(3, Velocity_Rel), 0.001);
    omg = max(GeometryToolbox::Norm(3, Vorticity_Rel), 0.001);

    for (int iDim = 0; iDim < nDim; iDim++) dpds += PrimVar_Grad_i[idx.Pressure()][iDim] * Velocity_Rel[iDim];      You, 3周前 • Uncommitted changes

    if (dpds >= 0.0) {
      su2double RefReynold = 1.0e6;
      dpomg = GeometryToolbox::DotProduct(3, PrimVar_Grad_i[idx.Pressure()], Vorticity_Rel) * laminar_viscosity *
              RefReynold / (omg * pow(velocityMag, 3) * pow(density, 2));  // reynold should be modified
      //cout<<"dpomg="<<dpomg<<endl;
      dpomg=abs(dpomg);
      var.beta_PGO = var.cpw1 * tanh(var.cpw2 * pow(dpomg, var.cpw3));
    }
    SetPgomegaBridge(dpomg);
    //var.Omega = omg * (1.0 + var.beta_PGO);
  }
Omega::get(Vorticity_i, nDim, PrimVar_Grad_i + idx.Velocity(), var);
```

- Common/include/option_structure.hpp
- SU2_CFD/include/numerics/CNumerics.hpp
- SU2_CFD/include/numerics/turbulent/turb_sources.hpp
- SU2_CFD/include/variables/CTurbSAVariable.hpp
- SU2_CFD/include/variables/CVariable.hpp
- SU2_CFD/src/output/CFlowOutput.cpp
- SU2_CFD/src/solvers/CTurbSASolver.cpp
- SU2_CFD/src/variables/CTurbSAVariable.cpp

PGOmega-BUAA

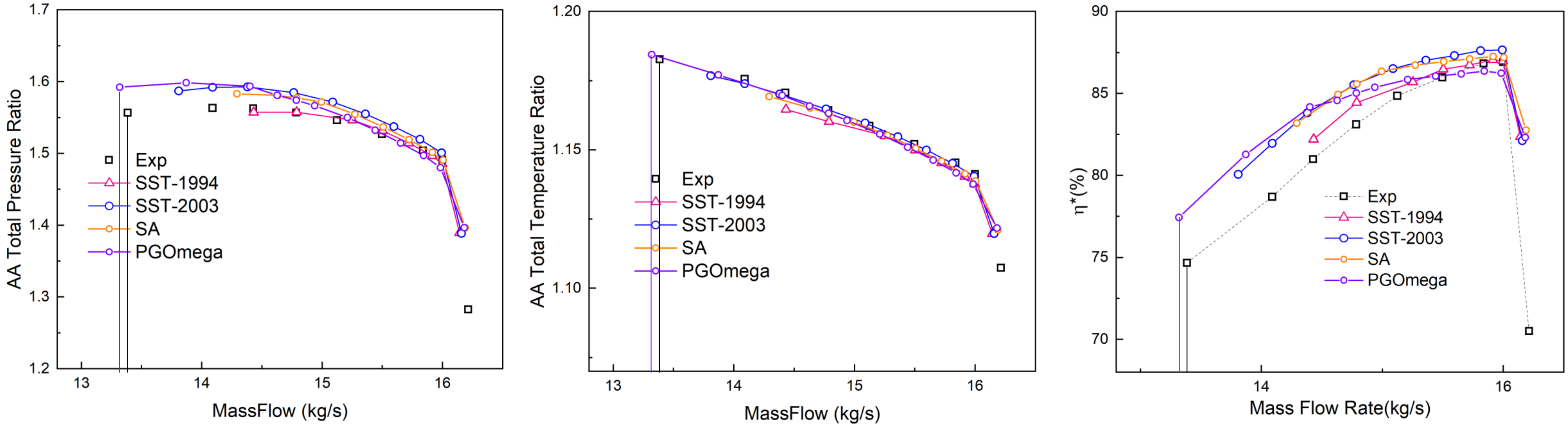




BUAA

Overall Performance

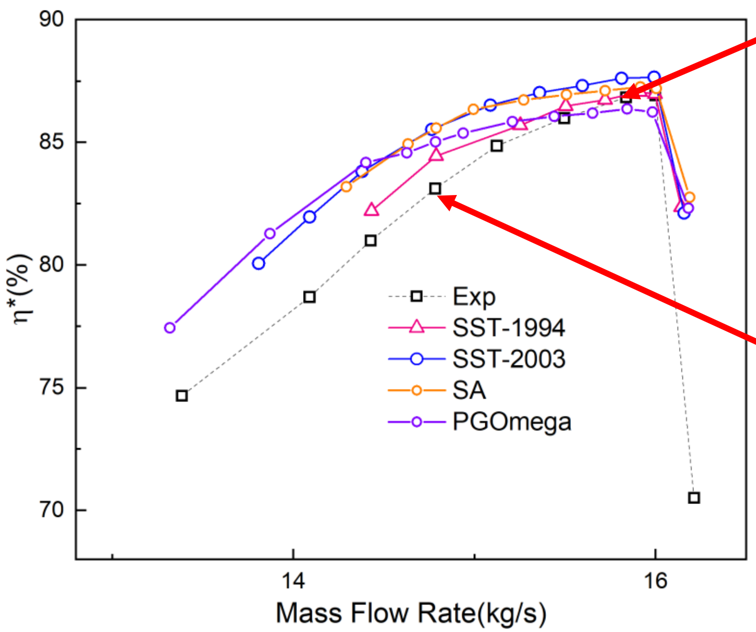
☐ Comparison between Different Turbulence Models



Mass Flow difference:0.56%

Radial Profiles

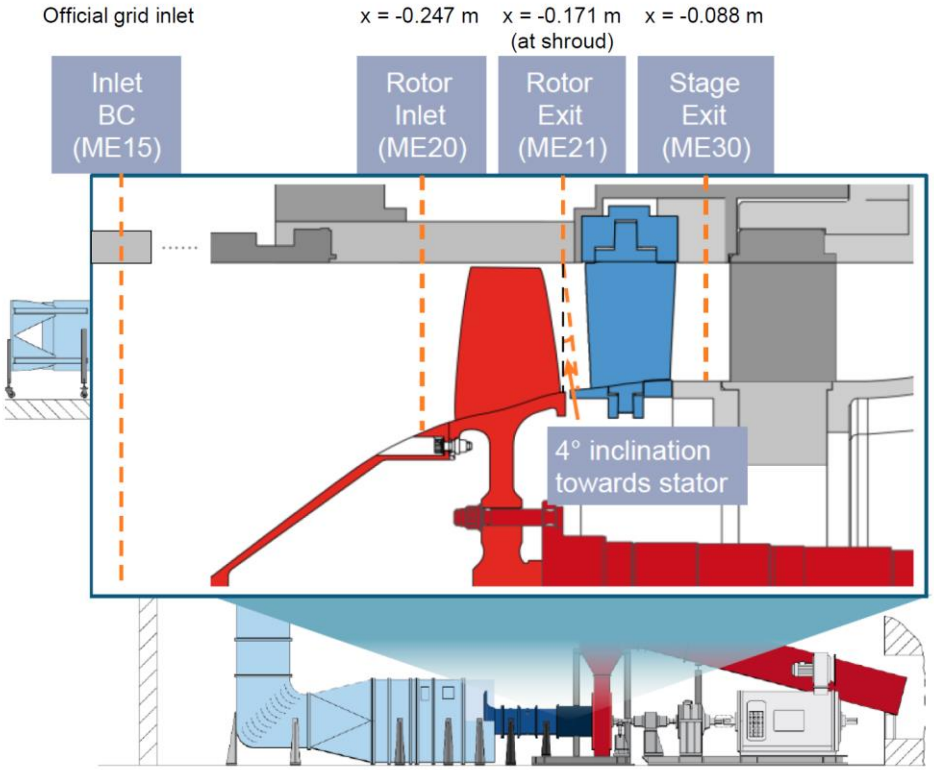
☐ Comparison between Different Turbulence Models



PE ME21

ME30

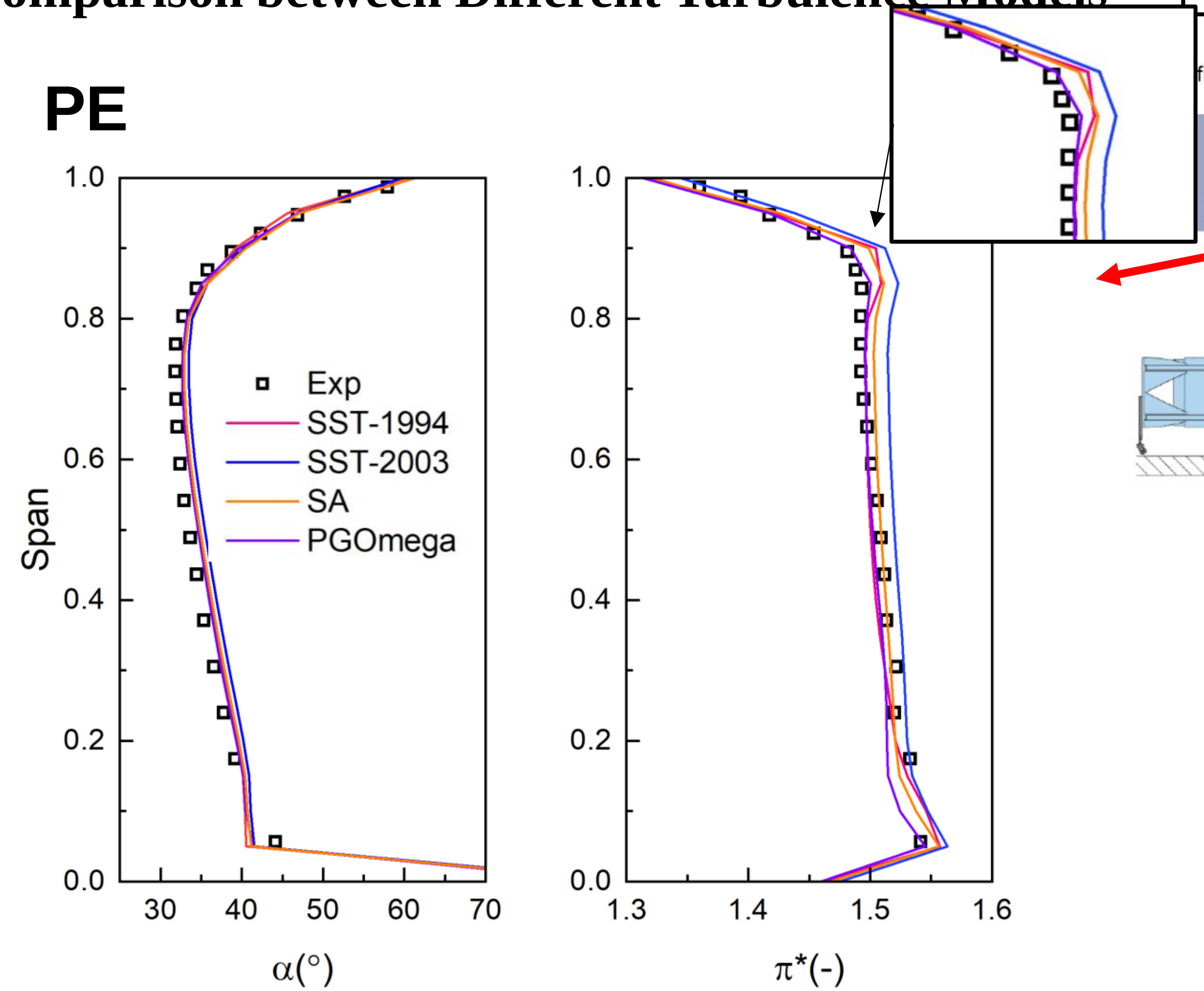
NS ME30

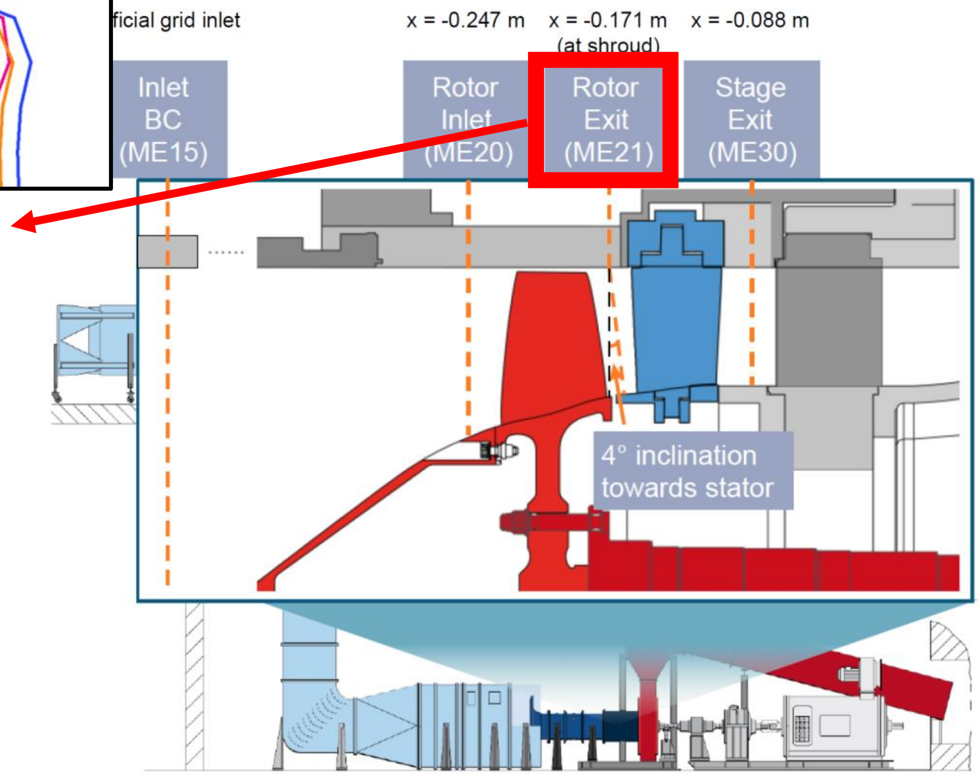


Comparison between Different Turbulence Models

alpha: the angle of circumferential velocity

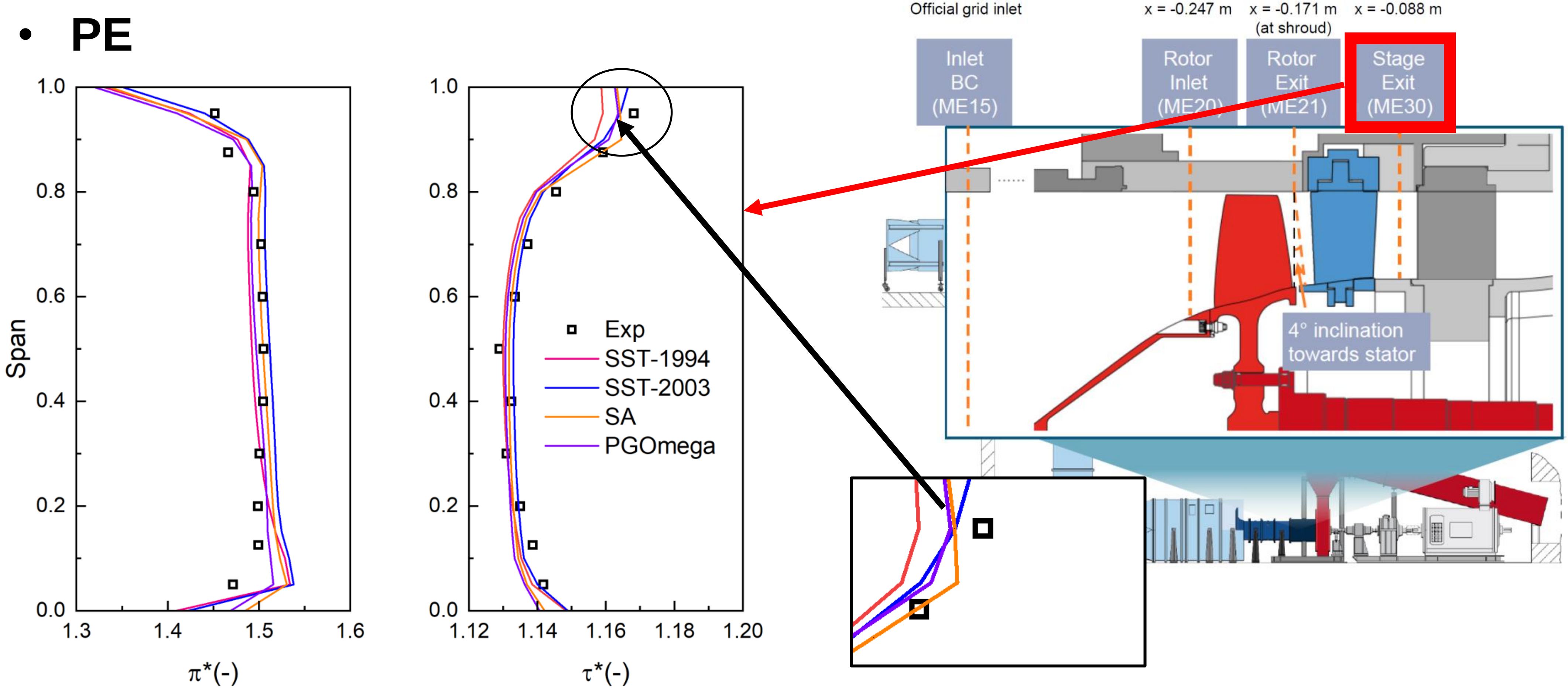
- **PE**





Comparison between Different Turbulence Models

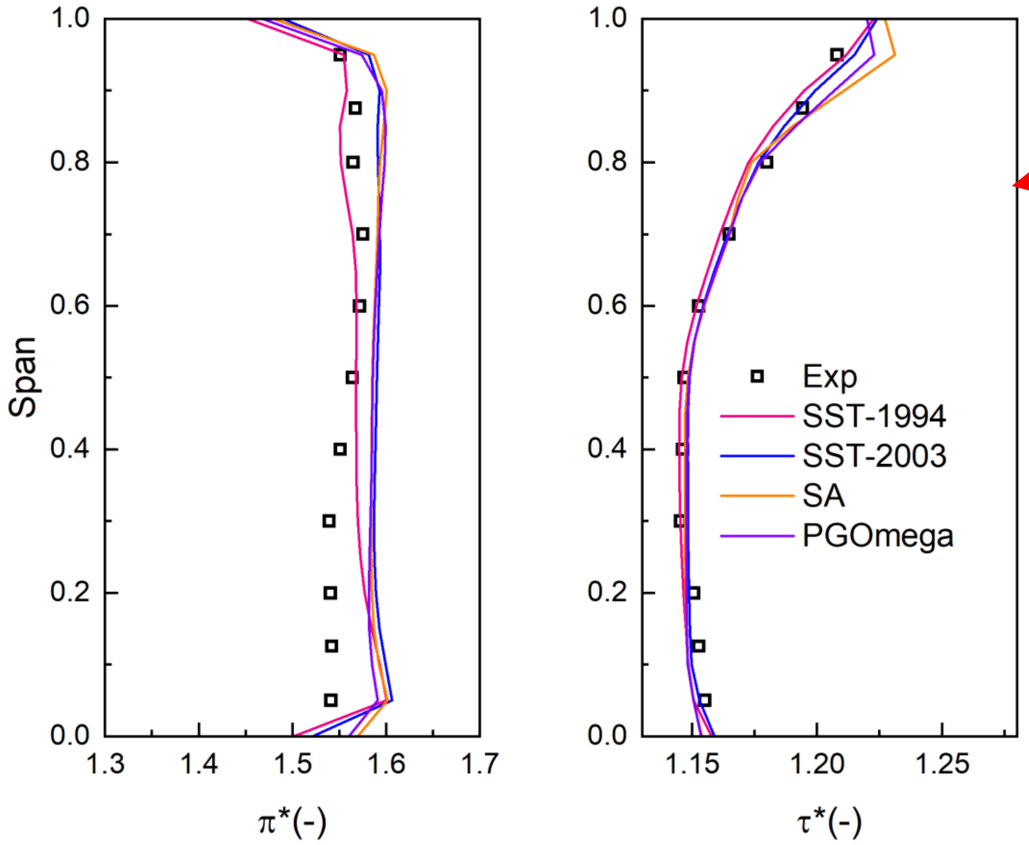
- **PE**

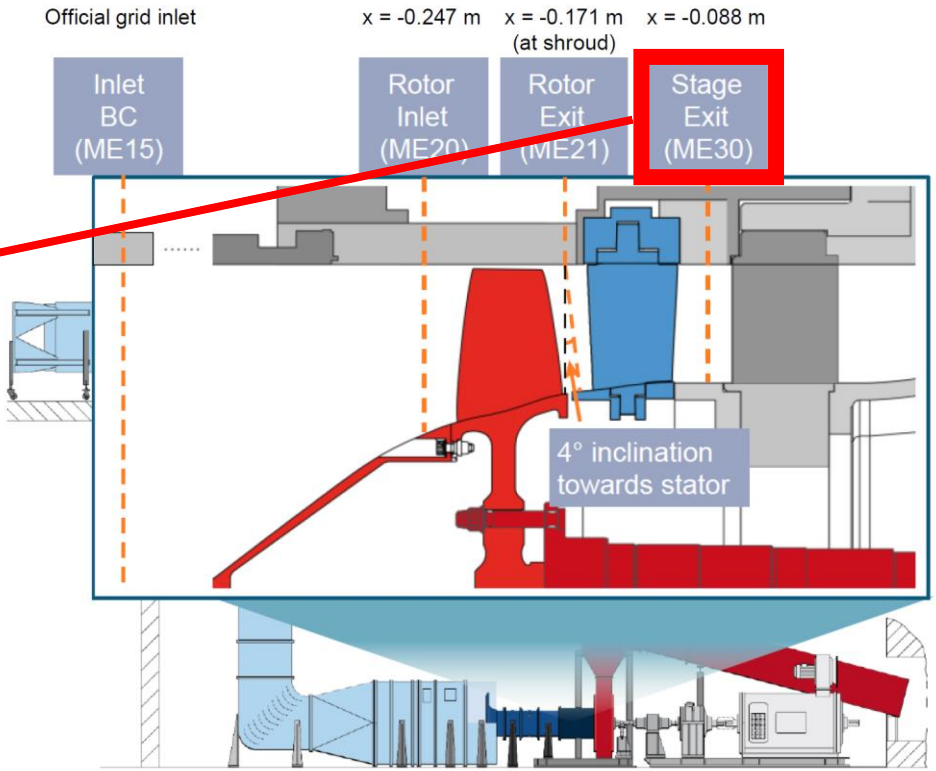


Radial Profiles

◻ Comparison between Different Turbulence Models

- **NS**



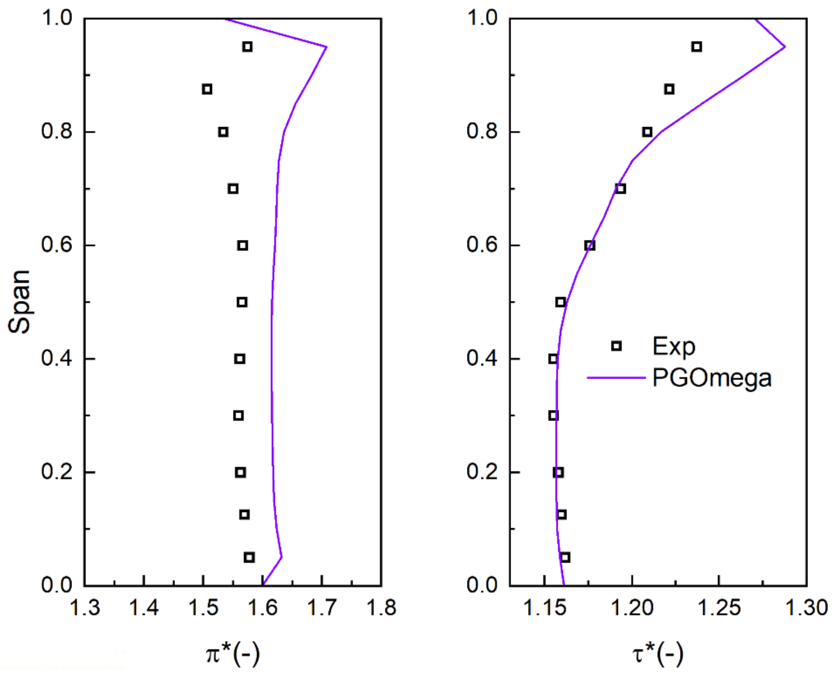


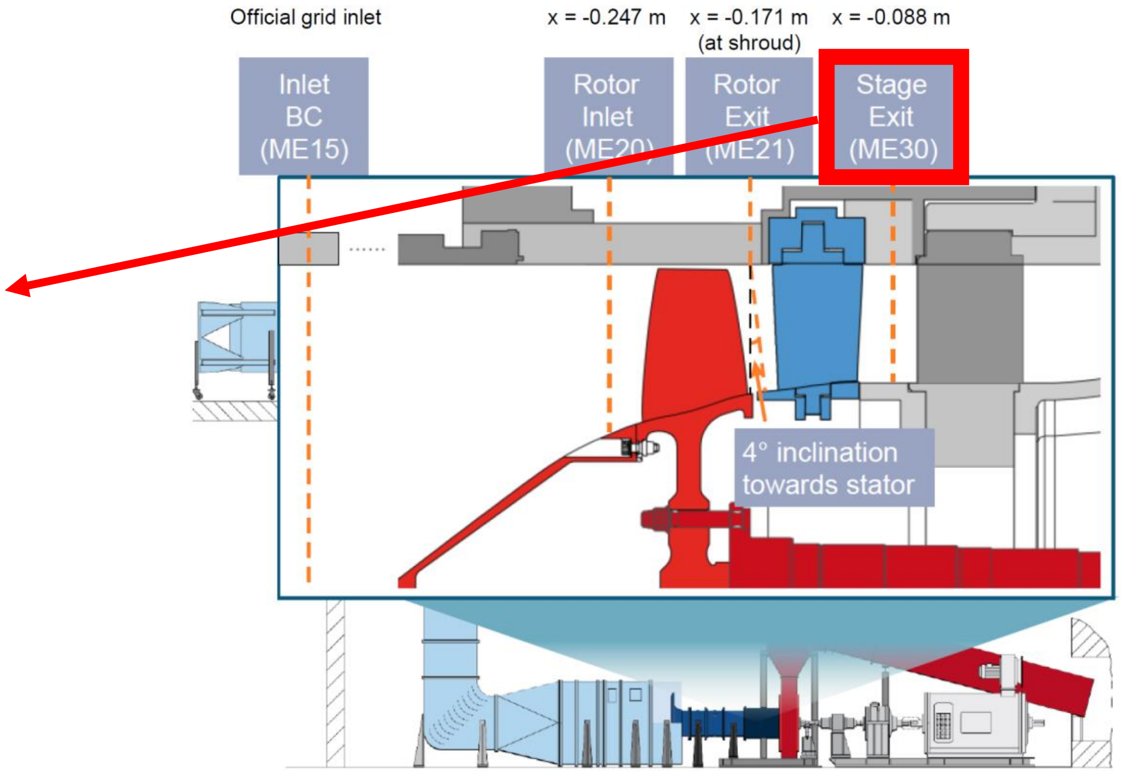
Thanks

Radial Profiles

Comparison between Different Turbulence Models

- **Stall**





Radial Profiles

☐ Comparison between Different Turbulence Models

- **Stall**