Axial Compressor Simulation of a Single Rotor at Different Tip Gap Sizes

Motivation

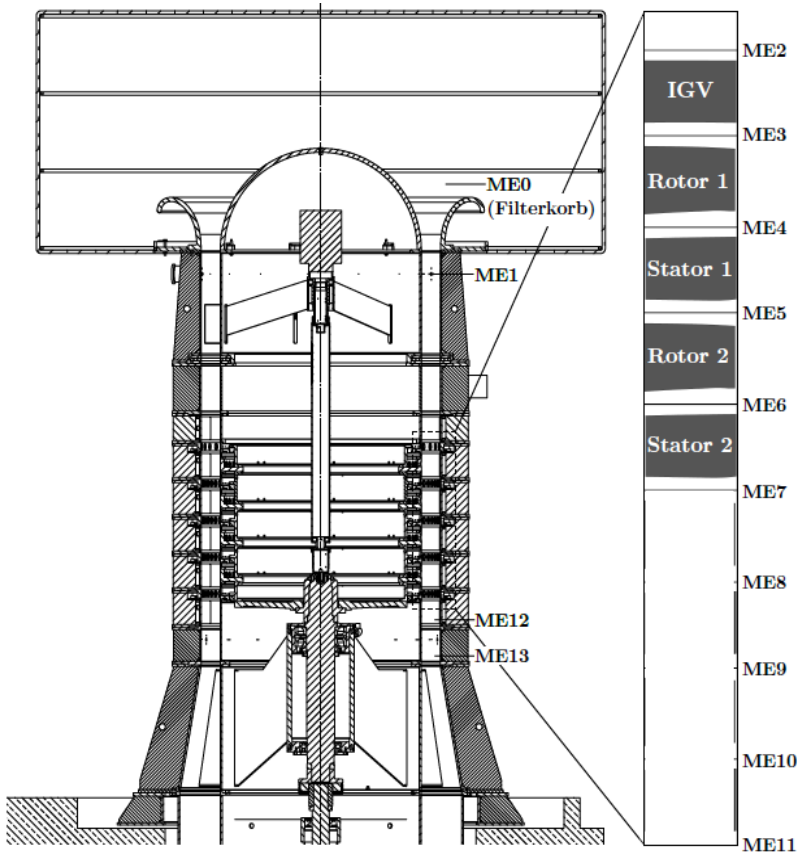
Short biography (OF related):

- 2020: Research internship at TUD
  Simple pipe simulation
- 2022: Internship at VKI in Belgium
  Two-phase simulation of an Inertial Particle Separator
- 2023: Master thesis at MTU Aero Engines
  CHT simulation of a rotating cavity (in a jet engine)
- 2023: Aerospace degree (Dipl. Ing)
- 2023 – Present: Research associate at TU DRESDEN
  Chair of Turbomachinery and Flight propulsion

Current work:

- Experimental and numerical work at Chair of Turbomachinery and Flight propulsion LSRC
- Research to investigate influence of circumferential grooves in axial compressors
- Variation of tip gaps
- CFD simulations carried out in CFX

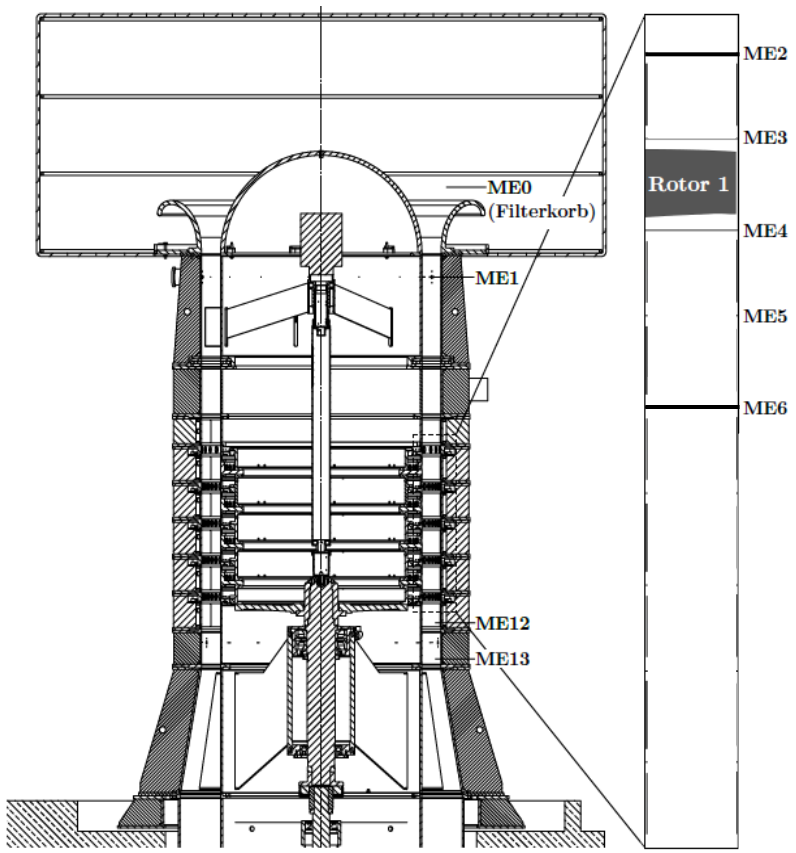
Motivation

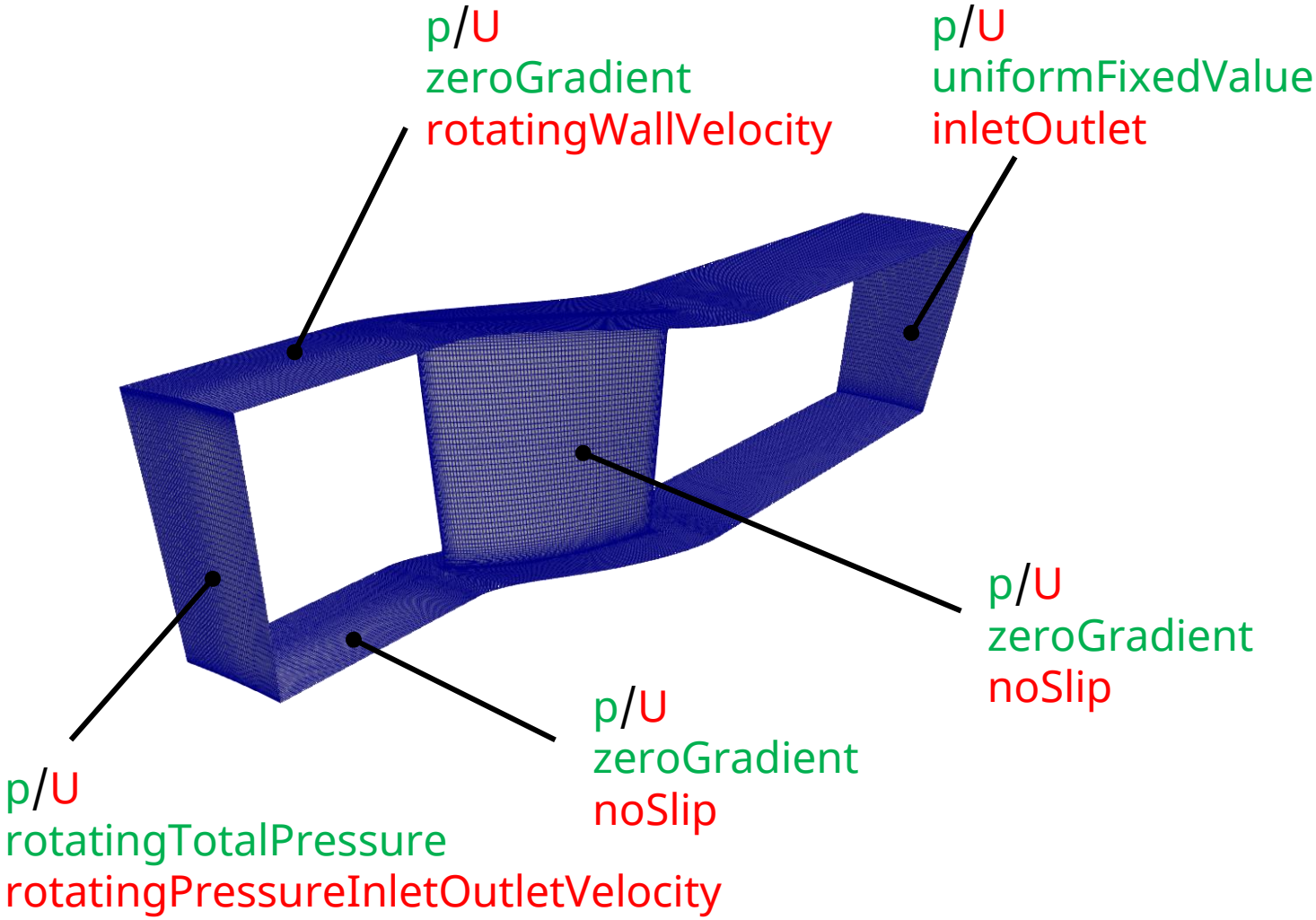


Current Work:

- Experimental and numerical work at the chairs LSRC
- Research to investigate influence of circumferential grooves in axial compressors
- Variation of tip gaps
- CFD simulations carried out in CFX
- $\Longrightarrow$ Could we use OpenFOAM?

Case setup – General configuration

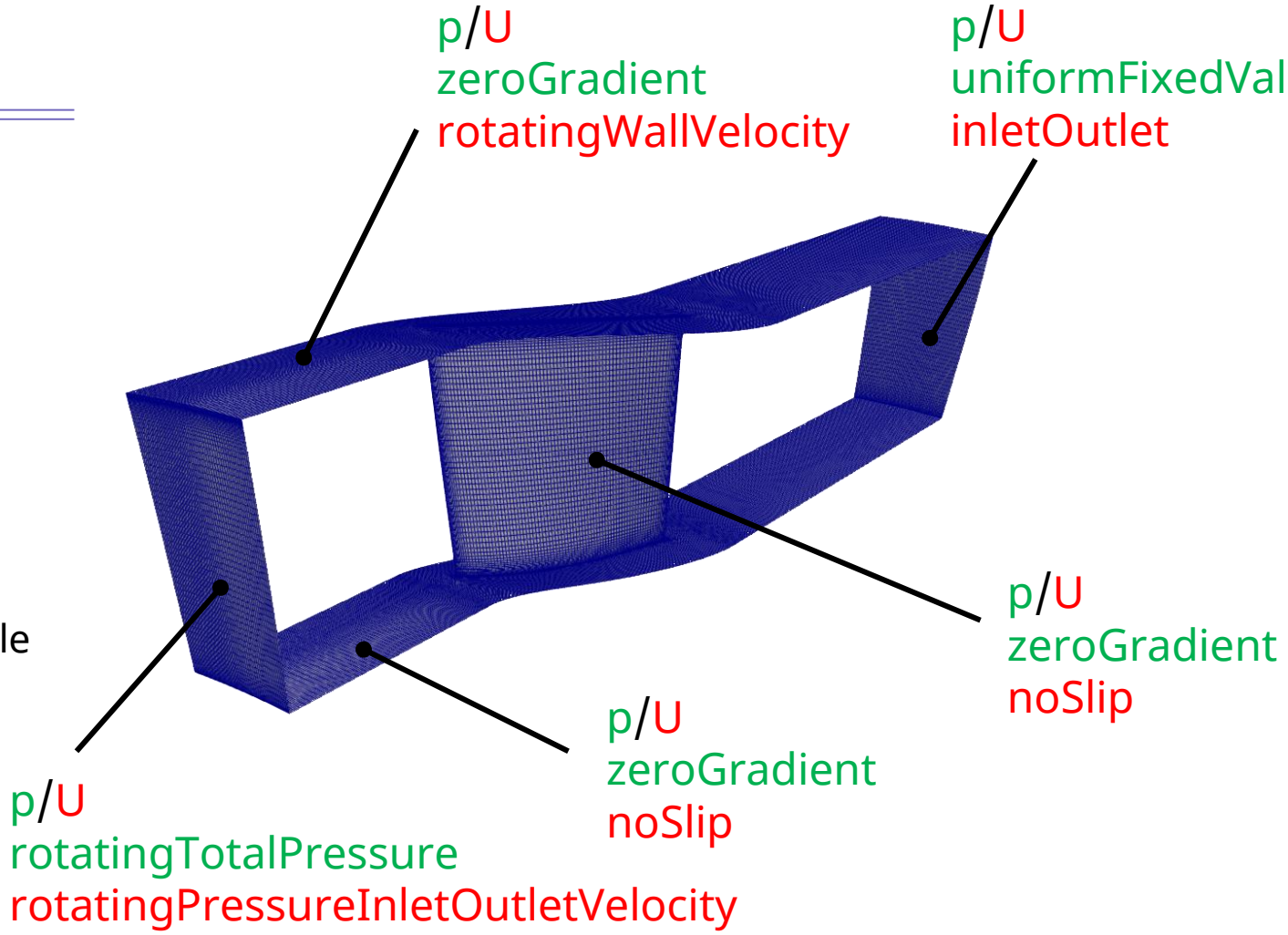




Case setup – General configuration

General information:

OF-Version:	10
Gitter-Größe:	2.4 Mio Zellen
Y+:	~ 1
Type:	Steady state
Solver:	rhoSRFSimpleFoam
BC:	**Inlet:** Total pressure profile **Outlet:** Ramped pressure
Rotational Vel:	1000 rpm

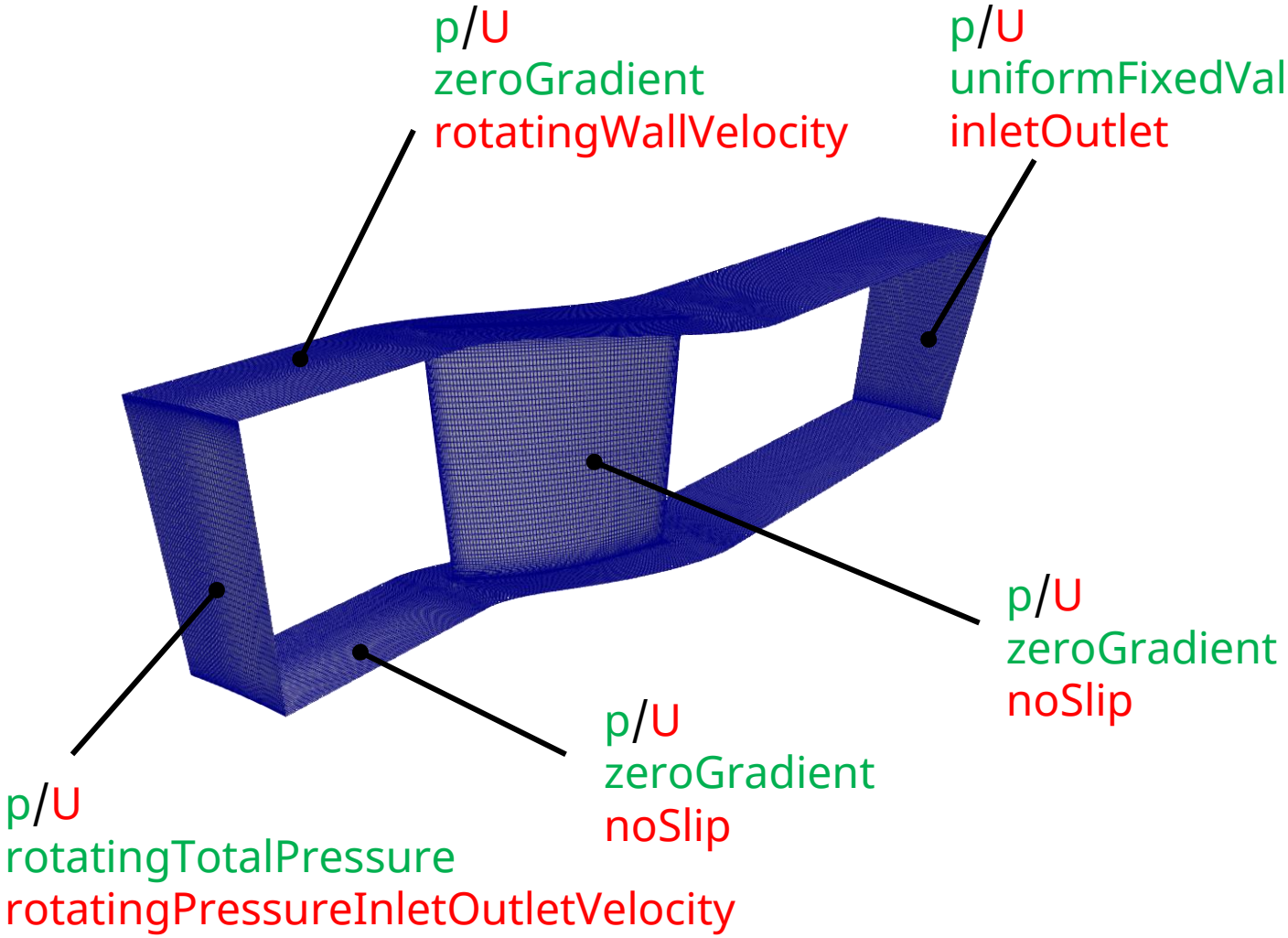


Case setup – Boundary conditions for SRF

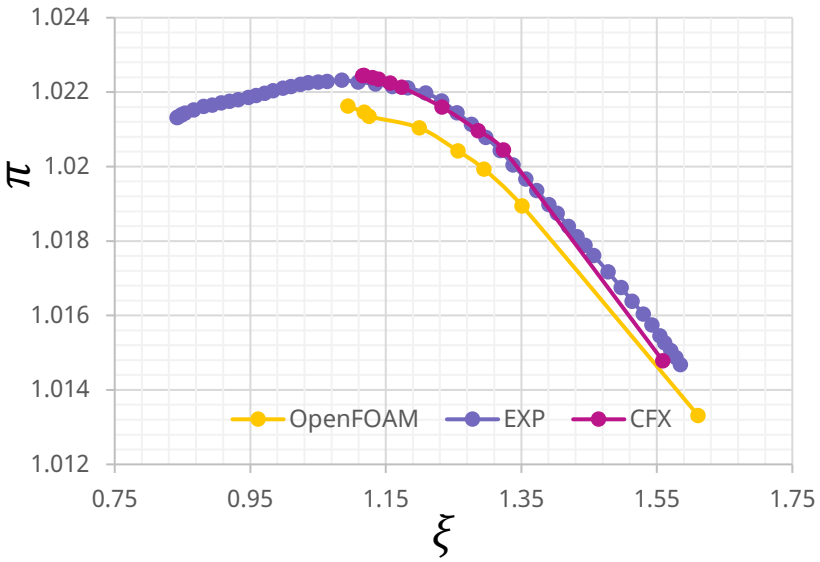
Mesh generation in AutoGrid5 2025.V2

Dummy case for mesh preparation

Simulation (SRF - MRF)

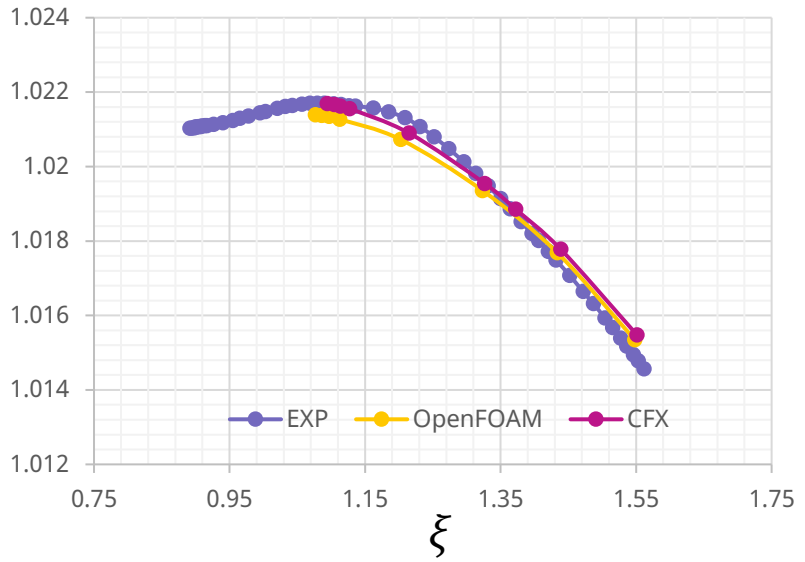


Current state of research – Compressor maps



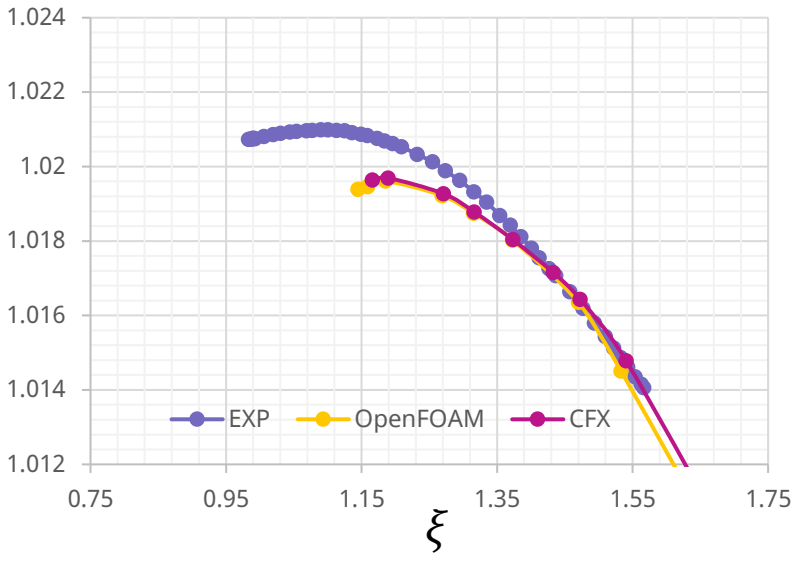
Small tip gap size

$x_{rel} = 1.25\ \%$



Medium tip gap size

$x_{rel} = 2.5\ \%$



Big tip gap size

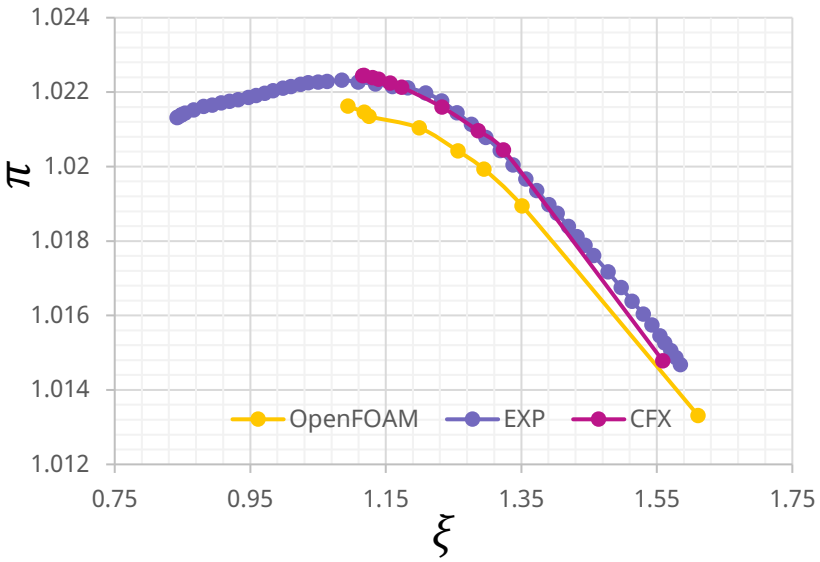
$x_{rel} = 4.2\ \%$

- OF and CFX simulations are carried out with the same mesh
- With increasing tip gap size difference between solvers disappear
- Difference between CFD and EXP grow towards bigger tip gap size

$$x_{rel} = \frac{x_{TG}}{h} \cdot 100\ \%$$

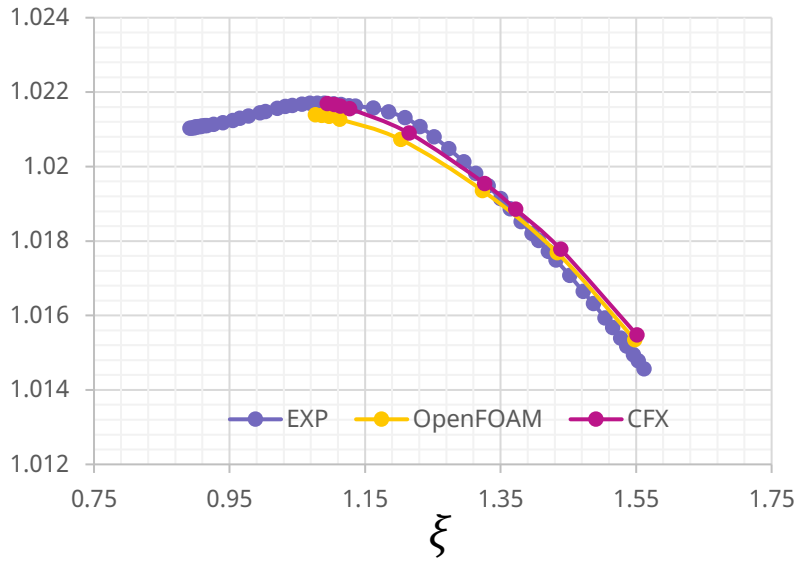
$$\xi = \frac{\dot{m}}{\dot{m}_{AP}} \qquad \pi = \frac{p_{out}}{p_{in}}$$

Current state of research – Compressor maps



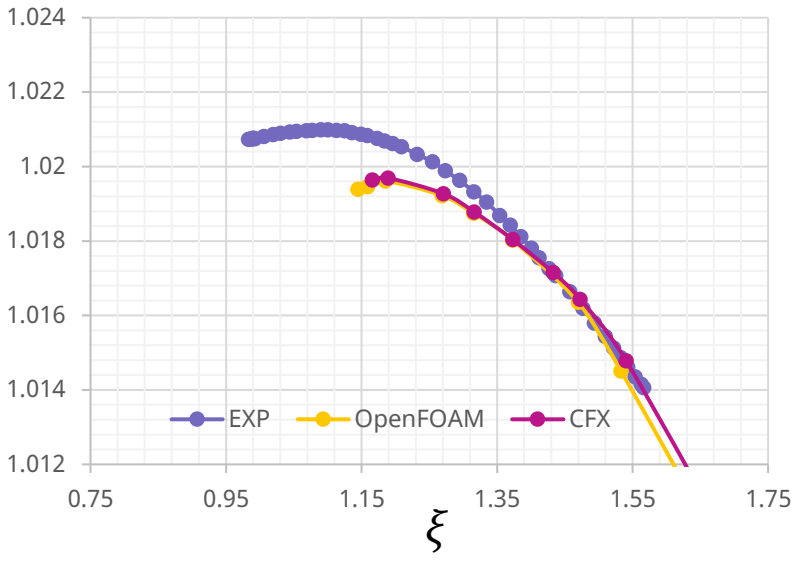
Small tip gap size

$x_{rel} = 1.25\ \%$



Medium tip gap size

$x_{rel} = 2.5\ \%$



Big tip gap size

$x_{rel} = 4.2\ \%$

- **First Issue:** What causes the difference between CFX and OF

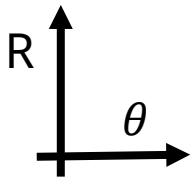
$$x_{rel} = \frac{x_{TG}}{h} \cdot 100\ \%$$

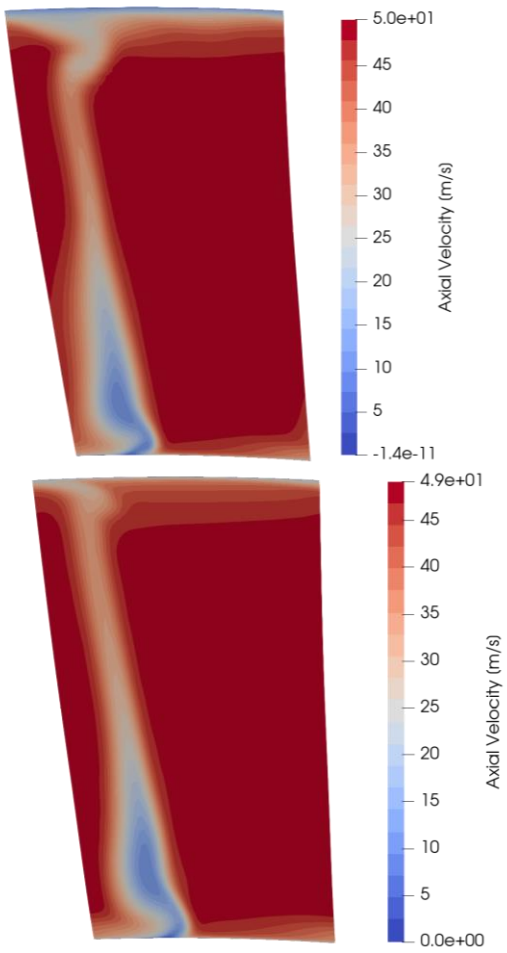
$$\xi = \frac{\dot{m}}{\dot{m}_{AP}} \qquad \pi = \frac{p_{out}}{p_{in}}$$

Current state of research – Compressor rotor outflow flow

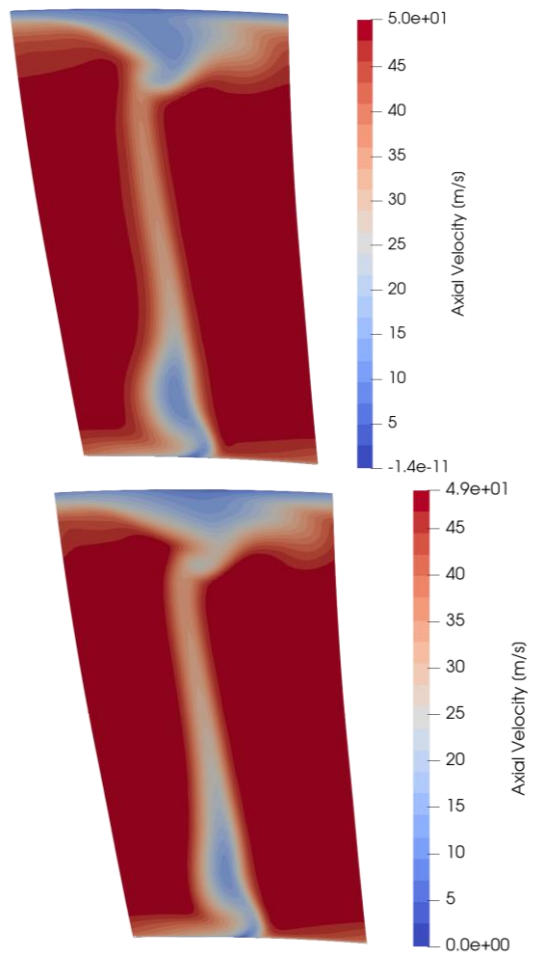
OF

CFX

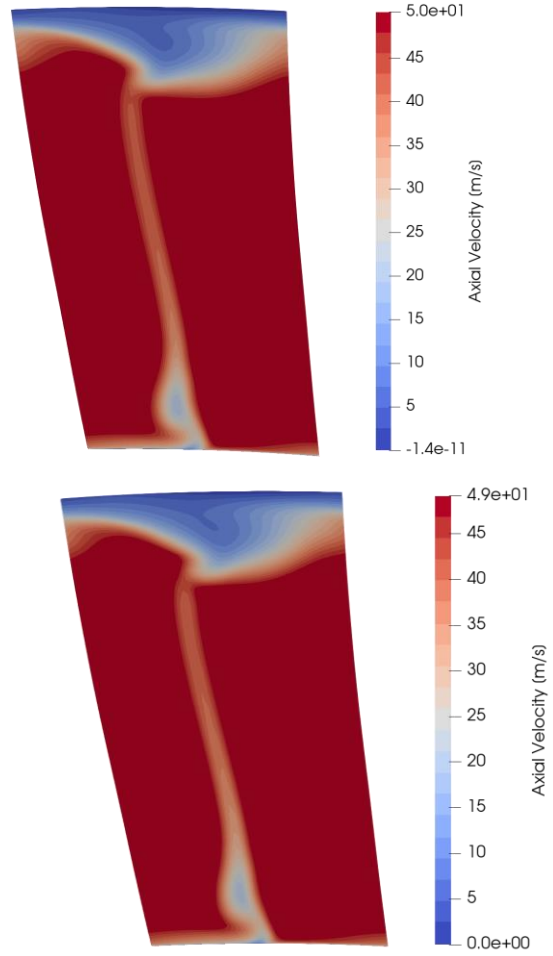




Small tip gap size
$x_{rel} = 1.25\ \%$



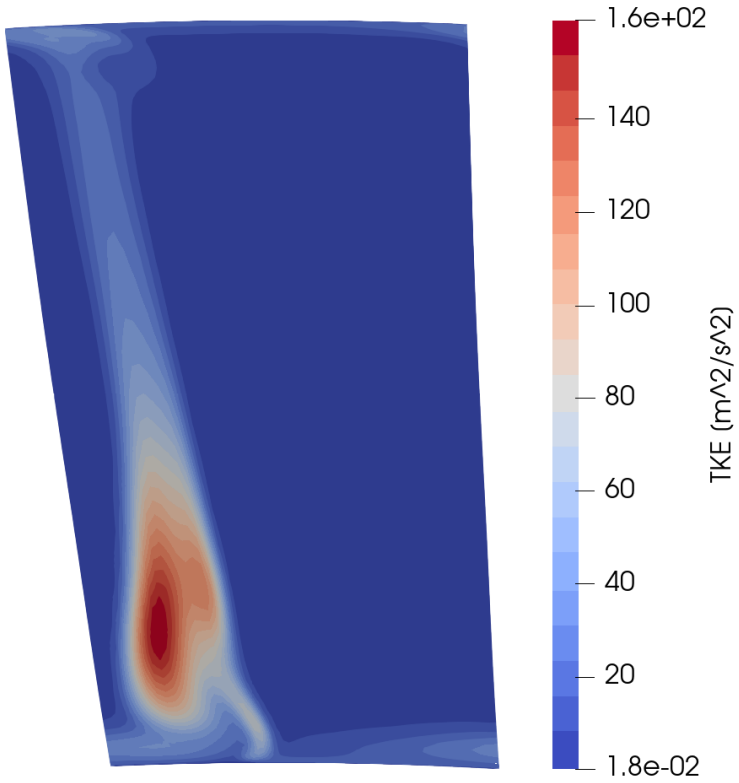
Medium tip gap size
$x_{rel} = 2.5\ \%$



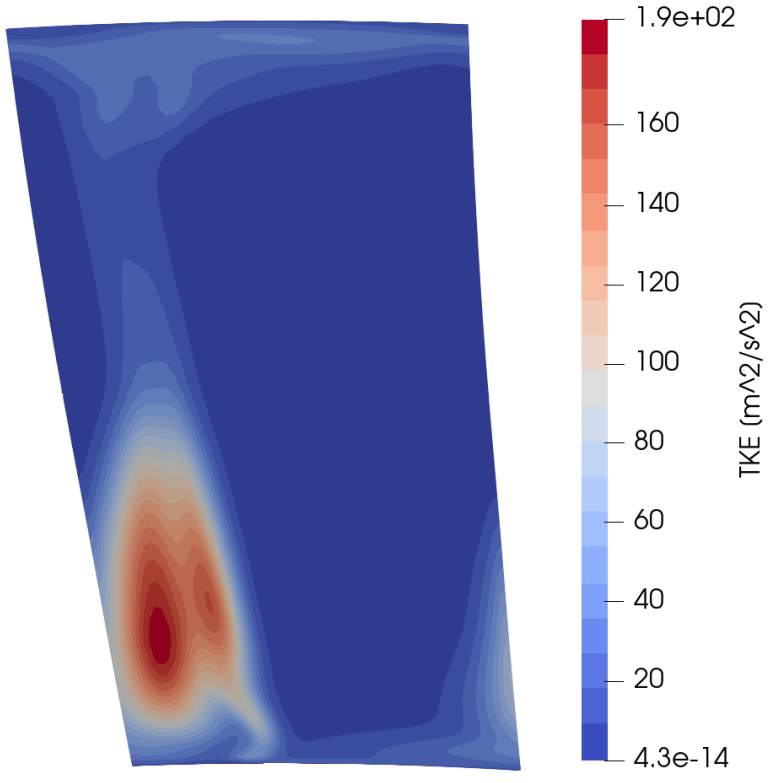
Big tip gap size
$x_{rel} = 4.2\ \%$

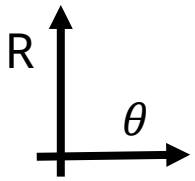
Current state of research – Compressor rotor flow

OF



CFX





- **First issue:** What could cause the difference in TKE production?

Additional issues:

Second issue: Computation times are two long compared to CFX (~ 3h vs 12h)

Third issue: General questions regarding the chosen fvSolution and fvSchemes:
What can be optimized?

Fourth issue: Best mesh quality requires NCC at cyclic boundaries, but all simulation crash between (10k – 20k iterations)

Error message

```
==== backtrace (tid:3931134) ====
 0 0x00000000000243e5 uct_ib_mlx5_completion_with_err()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/uct/ib/mlx5/ib_mlx5_log.c:162
 1 0x0000000000037db2 uct_ib_mlx5_poll_cq()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/uct/ib/mlx5/ib_mlx5.inl:91
 2 0x0000000000037db2 uct_rc_mlx5_iface_progress()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/uct/ib/rc/accel/rc_mlx5_iface.c:173
 3 0x0000000000037db2 uct_rc_mlx5_iface_progress_cyclic()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/uct/ib/rc/accel/rc_mlx5_iface.c:178
 4 0x000000000003764a ucs_callbackq_dispatch()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/ucs/datastruct/callbackq.h:211
 5 0x000000000003764a uct_worker_progress()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/uct/api/uct.h:2589
 6 0x000000000003764a ucp_worker_progress()  /dev/shm/build/UCX/1.12.1/GCCcore-11.3.0/ucx-1.12.1/src/ucp/core/ucp_worker.c:2636
 7 0x000000000002e20b opal_progress()  ???:0
 8 0x0000000000052745 ompi_request_default_wait_all()  ???:0
 9 0x000000000008f04f MPI_Waitall()  ???:0
10 0x00000000000084d3 Foam::UPstream::waitRequests()  ???:0
11 0x00000000000fa9d1 Foam::GeometricBoundaryField<Foam::Tensor<double>, Foam::fvPatchField, Foam::volMesh>::evaluate()  ???:0
12 0x00000000011a5df5 Foam::fv::faceLimitedGrad<Foam::Vector<double> >::calcGrad()  ???:0
13 0x00000000001a6b65 Foam::fv::gradScheme<Foam::Vector<double> >::grad()  ???:0
14 0x00000000001a7230 Foam::fvc::grad<Foam::Vector<double> >()  ???:0
15 0x00000000001a76ae Foam::fvc::grad<Foam::Vector<double> >()  ???:0
16 0x00000000001c0e52 Foam::kOmegaSST<Foam::eddyViscosity<Foam::RASModel<Foam::compressibleMomentumTransportModel> >, Foam::compressibleMomentumTransportModel>::correct()  ???:0
17 0x000000000042d59e main()  ???:0
18 0x00000000000295d0 __libc_start_call_main()  ???:0
19 0x0000000000029680 __libc_start_main_alias_2()  :0
20 0x000000000042edce _start()  ???:0
```

rhoSRFSimpleFOAM

- Based on SRFSimpleFOAM
- Includes additional density to account for compressibility

```
// Solve the Momentum equation

tmp<fvVectorMatrix> tUrelEqn
(
    fvm::div(phi, Urel)
  + turbulence->divDevTau(Urel)
  + rho*SRF->Su()
 ==
    fvModels.source(rho, Urel)
);
fvVectorMatrix& UrelEqn = tUrelEqn.ref();

UrelEqn.relax();

fvConstraints.constrain(UrelEqn);

solve(UrelEqn == -fvc::grad(p));

fvConstraints.constrain(Urel);
```