Conservative approach for rotor-stator coupling in turbomachinery computations

B. de Laage de Meux – T. Pasutto – B. Audebert

Rotor-stator interactions modelling

Frozen rotor
- frozen geometry for rotor and stator
- flow resolved in the relative frame of reference attached to the blades (Coriolis and centrifugal pseudo-forces in the rotor)

Unsteady rotor-stator
- rotor mesh actually rotating
- flow resolved in a galilean frame of reference (ALE formulation in the rotor)

A new treatment in *Code_Saturne*

Previous approach: code-code coupling
- one calculation for the rotor and one calculation for the stator
- boundary conditions at rotor-stator interface: coupling scheme based on the closest cells on both side of the interface

New approach: interface joining
- Single *Code_Saturne* calculation on joint mesh

Frozen Rotor:
- mesh joining at the beginning of the computation
- incompressible momentum equation in the rotor inverted in the form:
  \[
  \frac{\partial u_A}{\partial t} + \nabla \cdot (u_A \otimes u_A) + \Omega \times u_A = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nabla u_A)
  \]
- with:
  - \(u_A\) : absolute velocity (primary variable)
  - \(u_R = u_A - \Omega \times x\) : relative velocity

Unsteady rotor-stator:
- update geometry and join meshes at each time step
- partition mesh at the beginning of the computation (parallelism)

Validation case

- Genova’s pump: centrifugal pump with vaned diffuser, quite simple geometry
- Gourdain’s pump: centrifugal impeller + casing

Industrial case

- CPU time of joining operations / total CPU with mesh joining algorithm: 15 %
- total CPU with mesh joining algorithm / total CPU with code-code coupling algorithm: 80 %

Performances of unsteady rotor-stator computations

- Similar results compared with previous algorithm
- Better convergence in frozen rotor
- Computation savings in unsteady rotor-stator

Visualisation of the mesh (1.2 M cells)

Specification of the pump: total head

Visualisation of the flow in subrate (left) and nominal (right) conditions