Ulrich Heck, Martin Becker

Application-oriented tools based on Open-source solutions: New potentialities for CFD integration into the Design Process

Introduction

Benchmark and solver requirement

OpenFOAM® adaptations

CAD model based workflow

Conclusion

DHCAE Tools GmbH, Germany

CFD-Solutions based on OpenFOAM® -Technology

Engineering:

CFD-Services with OpenFOAM

Software
Standard/ Customized:

GUIs, Extensions

User Support Training:

OpenFOAM/our Extensions

This offering is not approved or endorsed by ESI Group, the producer of the OpenFOAM® software and owner of the OPENFOAM® and OpenCFD® trade marks.

Tool for extrusion dies

Why OpenFOAM® based:

- Reliable, stable and established CFD-methods
- Scales excellent in parallel
- Wide variety of physical models for complex CFD applications
- Cost aspects (open-source CFD)
- Most important in this context: Easy to extend by object-orientated C++-structure
- Text-file based: Easy to modify

Potential difficulties in particular for target group: Design engineer

- Available only on Linux
- No GUIs: Editing Text Files
- Many features not seen, keyword needs to be known
- Meshing: STL, script based
- High demands on the CFD knowledge

Extensions on top:

Support for Linux and Windows ports

GUI solutions with selectable graphical keywords and job control, solution templates

CAD-based solutions

Specific solver adaptation, Training, Support

Study: extrusion die

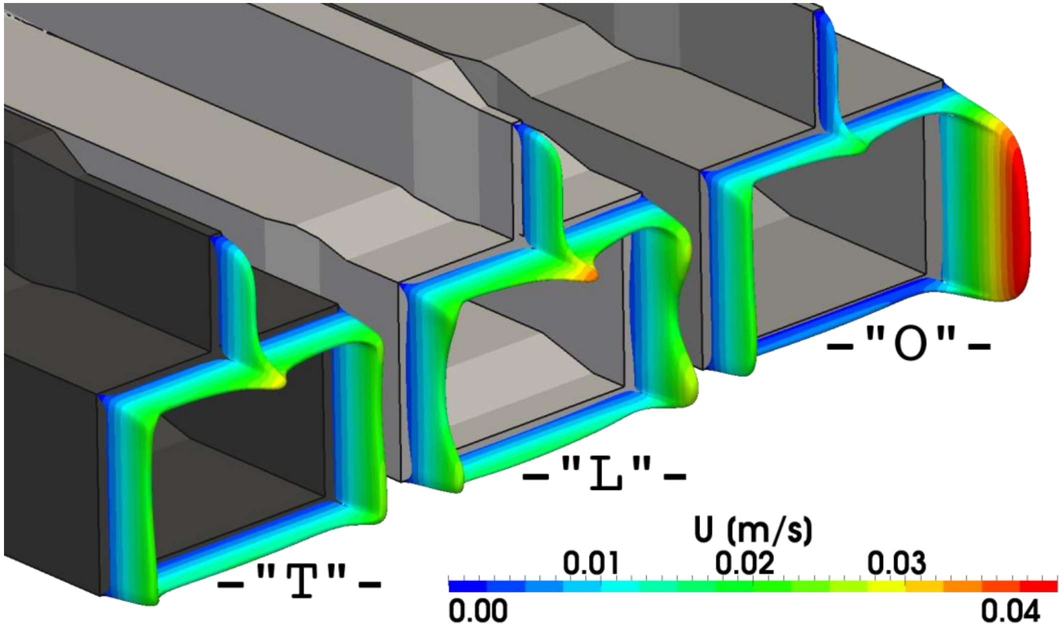
Example for development of modelling environment for extrusion die

Benchmark: Study by Nóbrega et al:

Nóbrega, J.M. and Carneiro, O.S. and Pinho, F.T. and Oliveria, P.J. (2004), Flow Balancing in Extrusion Dies for Thermoplastic Profiles: Part III: Experimental Assessment. *Intern. Polymer Processing XIX (2004)*

Target:
- Judging different designs for optimized flow

Design criteria:
- Uniform velocity profile
- Pressure loss



Flow configuration

- Steady state solver for high viscous flow (Re << 1)
- Non-Newtonian flow model
  - $BirdCarreau$: $\mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1+(k \cdot \dot{\gamma})^2)^{\frac{1-n}{2}}}$
- Temperature effects:
  - Heat transfer
  - viscous dissipation
  - Arrhenius-model for viscosity

Road map for customized solver

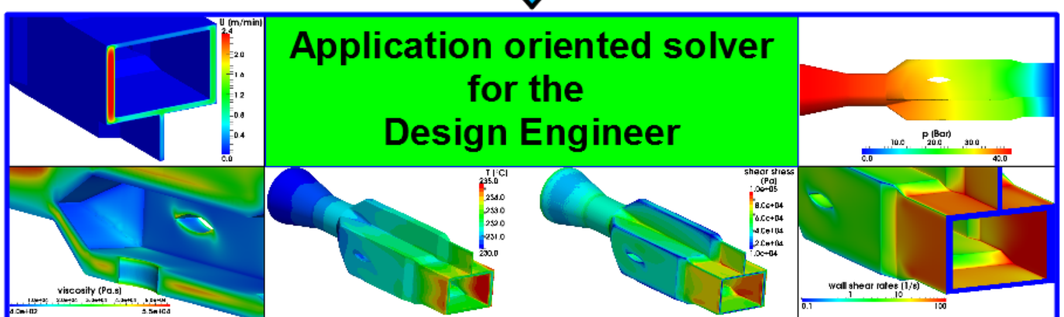
Polymer specific boundary conditions and material models
- Wall slip model library
- Non-Newtonian viscosity model library

Temperature
- Viscous dissipation
- Heat transport
- Non-Newtonian viscosity model library

Postprocessing
- Wall shear rate
- Wall shear stress
- Residence times

Work flow
- Field initializations
- Simulation stabilizer
- Simulation acceleration

Basis solver
simpleFoam
+
extensions



```
dimensionedScalar t_celsius("t_celsius", dimensionSet(0,0,0,1,0,0,0), 273.15);
const volScalarField& tm = U_.mesh().lookupObject<volScalarField>("T");
volScalarField shift =
    (
        exp
        (
                (alpha_ - t_celsius)
                *
                (
                    scalar(1.0) / (tm - t_celsius)
                  - scalar(1.0) / (T0_ - t_celsius)
                )
        )
     );
return
    nuInf_
  + (shift * (nu0_ - nuInf_))
   *pow(scalar(1)
    + sqr(k_*strainRate()*shift)
, (n_ - scalar(1.0))/scalar(2.0))
  ;
```

Temperatur dependency (Arrhenius shift)

$$BirdCarreau\colon\ \mu(\dot{\gamma}) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{\left(1 + (k \cdot \dot{\gamma})^2\right)^{\frac{1-n}{2}}}$$

OpenFOAM: solver adaptation

- Creeping flow:
- Additional expressions can be added or removed easily:
- Here convection term in momentum equation

```
    // Momentum predictor

    tmp<fvVectorMatrix> UEqn
    (
        // fvm::div(phi, U) // <-- creeping flow assumed
      + turbulence->divDevReff(U)
      ==
        sources(U)
    );
    UEqn().relax();
    sources.constrain(UEqn());
    solve(UEqn() == -fvc::grad(p));

```

Additional solver/material adaptions

• Further solver adaptions:	
• Start from reasonable initial fields	speed up + stability
• Retarded Non-Newtonian Flow	stability
• Implementation: Underrelaxation viscosity	stability
• Pressure extrapolation:	speed–up
• Residuum smoothing:	speed–up
• Multi-grid methods :	speed–up

Temperature effects in solver:

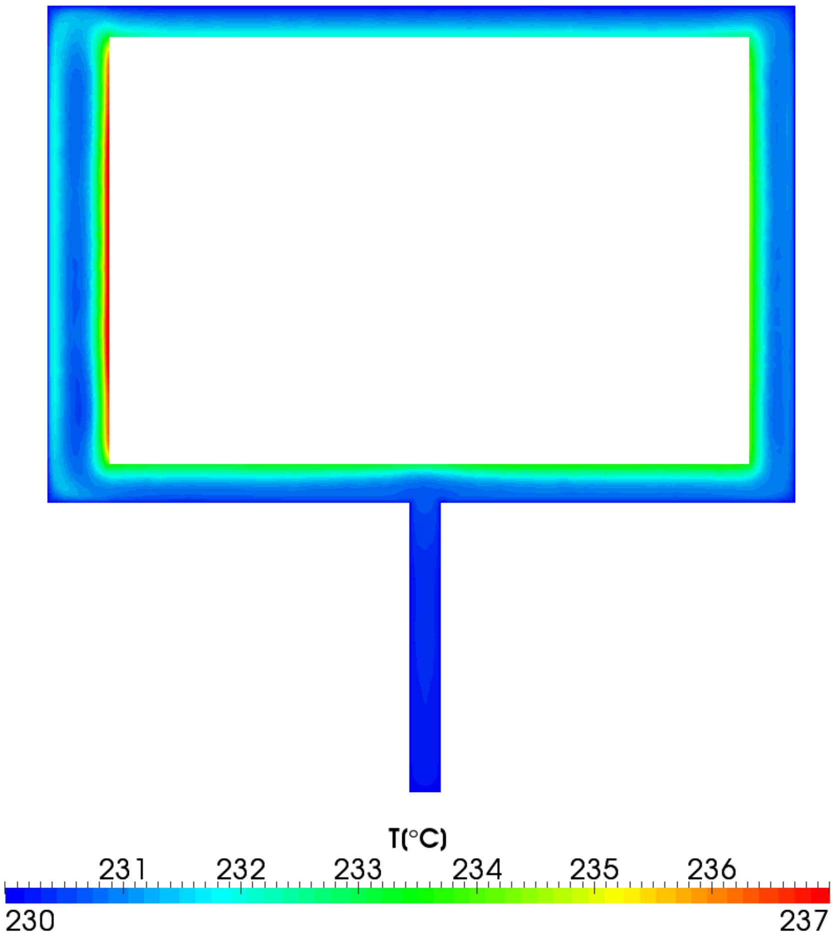
- Viscous dissipation

Additional material properties

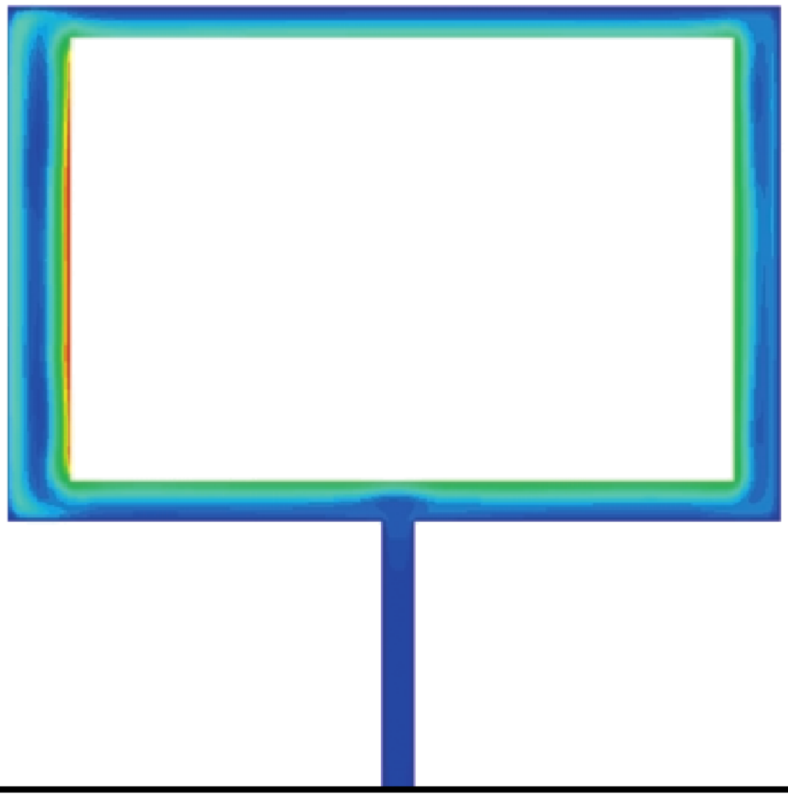
- E.g. pressure dependent viscosity
- WLF (Williams-Landel-Ferry)

Results comparison

Results from customized OpenFOAM solver



DieINI

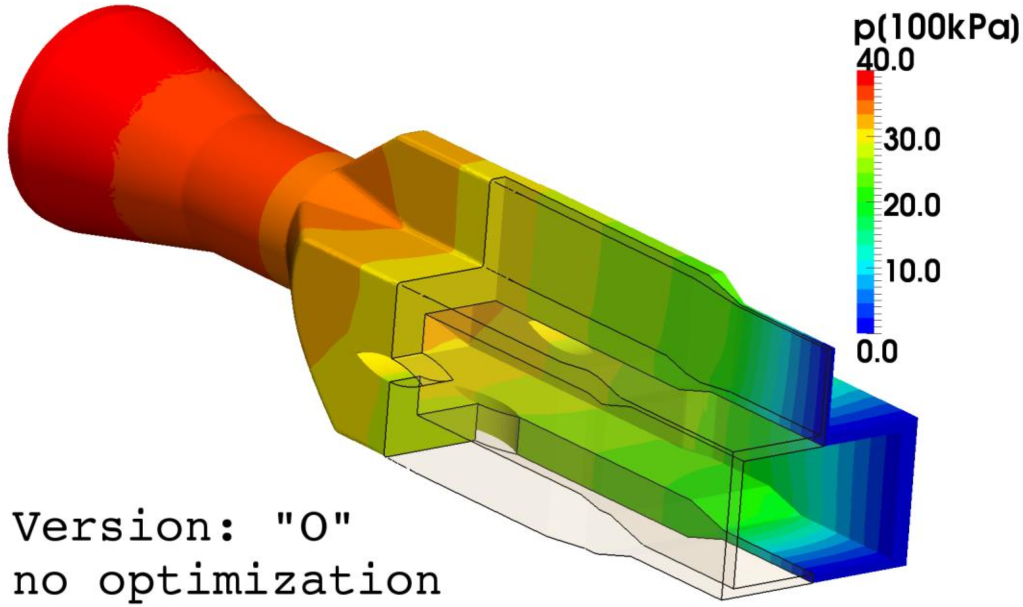


Outlet temperature from Fig.14 from Paper "Flow Balancing in Extrusion Dies for Thermoplastice Profiles, Part III"

Final solver comparison:

Good agreement is found for

- temperature distribution compared to other numerical results (can‘t be measured easily)
- measurable design parameters:
  E.g. pressure loss.



Result for “DieINI”	Paper Nóbrega (2004)	Customized OpenFOAM solver result
Average T @ outlet	231.4°C	231.4°C
Maximum T @ outlet	237°C	237°C
Pressure drop	4.00 MPa measured value	4.02 MPa

Solver-Modification: Postprocessing

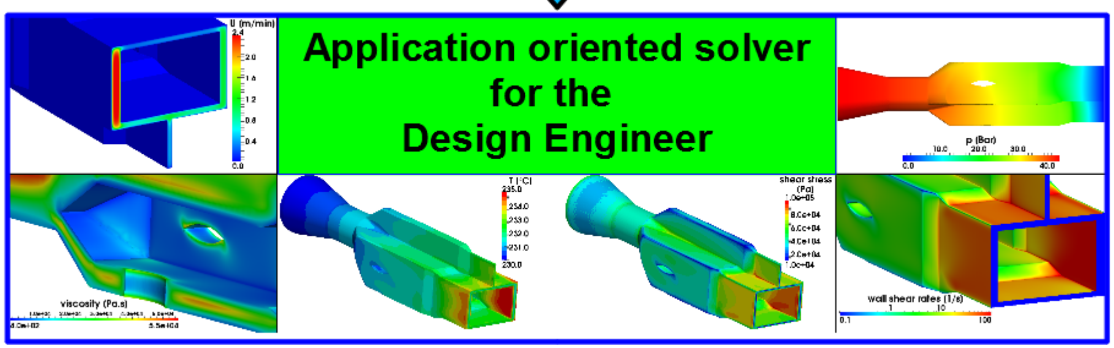
Polymer specific boundary conditions and material models
- Wall slip model library
- Non-Newtonian viscosity model library

Temperature
- Viscous dissipation
- Heat transport

Postprocessing
- Wall shear rate
- Wall shear stress
- Residence times

Basis solver
simpleFoam
+
extensions

Work flow
- Field initializations
- Simulation stabilizer
- Simulation acceleration



Post-processing

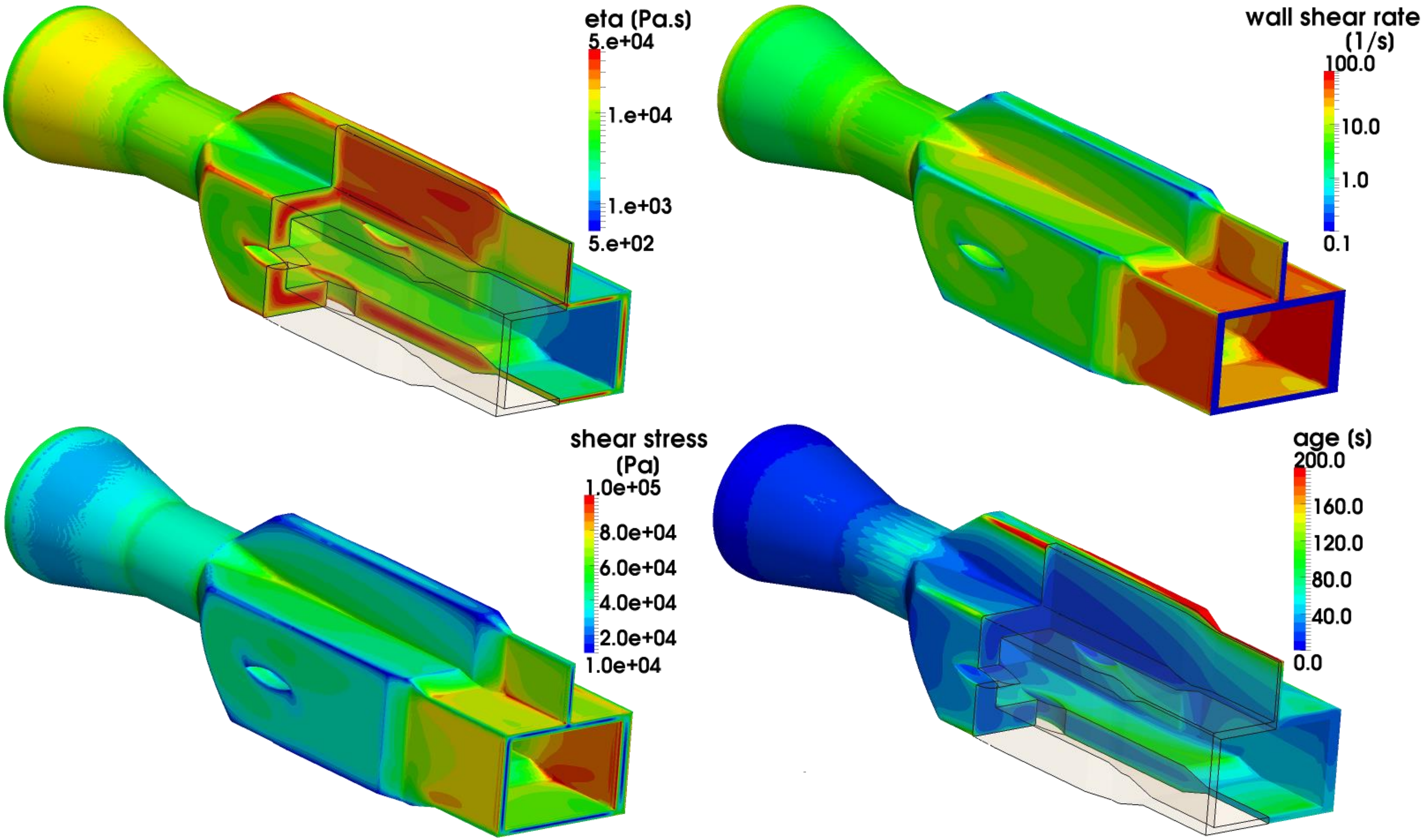
Calculation of relevant post-processing data by the solver or post processing tools:

- Viscosity
- Wall shear rate and wall shear stresses
- Residence time
- Conversion to industry standard units (m / min, bar, ° C, Pa s)

Local parameters if required:

- Local residuals (convergence control)
- Cell-Re, Pr-cell, cell-Pe, Pe thermal flow type, etc.

Post-processing



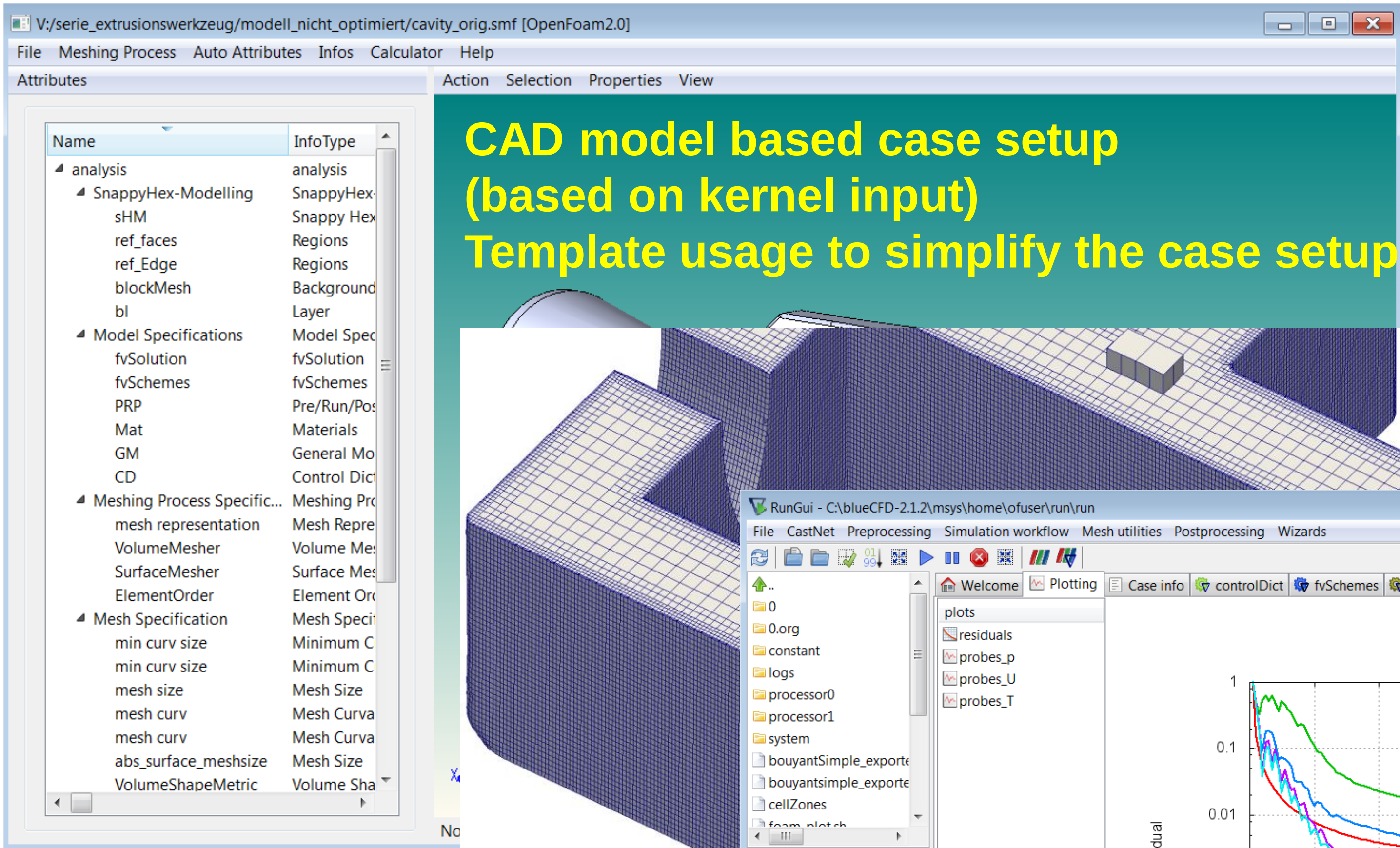
Process workflow integration:

- Independent from the CFD-solution workflow
- Depends on:
  - Preferred case setup procedure: Typically GUI based for design engineers
  - CAD environment
  - CFD-knowledge: More or less options/ Template usage
  - IT infrastructure: Linux or Windows

Typically:

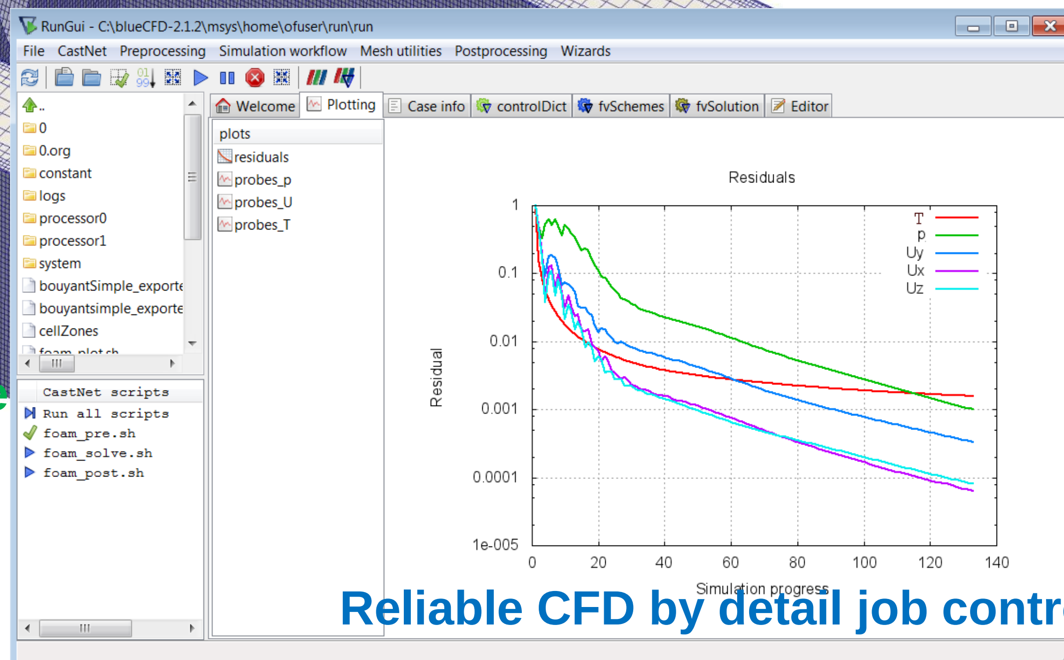
- Text file based work process workflow based on STL geometry input can be established with OpenFOAM Tools alone (complete open-source)
- GUI based working needs additional proprietary tools

Process workflow integration



CAD model based case setup
(based on kernel input)
Template usage to simplify the case setup

Automate



Reliable CFD by detail job control

Note:

- GUIs are not part of the OpenFOAM package and not open-source
- Provided by DHCAE Tools based on proprietary CAD-import technology

Conclusion

OpenFOAM® : very good solutions and calculation capabilities for a wide range of challenging CFD analyses

Base for application-specific calculation tools:

- Defined solving skills of a "solver“
- Availability of source code
- Object-oriented structure
- Adaptability
- Communication based on file

According needs: Integration into process workflow:

- Complete Open Source Solutions: Text file based, Stl-Input
- CAD based with GUIs: proprietary add-on tools available