Implementation of an Alternate Chemistry Library into OpenFOAMTM

Markus Rehm
Framatome GmbH

Implementation of an Alternate Chemistry Library into OpenFOAM™

Bernhard F.W. Gschaider[1] Markus Rehm[2] Peter Seifert[2]
Bernd Meyer[2]

[1]ICE Strömungsforschung
Leoben, Austria

[2]Institute of Energy Process Enineering and Chemical Engineering,
TU Bergakademie Freiberg, Germany

Open Source CFD International Conference
4th/5th December

Outline

1. **Introduction/Motivation**
   - Current status
   - Cantera
   - Desired behaviour
2. **Implementation**
   - Alternate Chemistry Library
   - Transient Solver
   - Steady Solver
   - Other topics
3. **Results Plug Flow Reactor**
   - Transient PFR
   - Steady-state PFR
4. **Results Sydney Bluff-Body Flame (HM1)**
   - Results HM1
   - Discussion HM1
5. **Conclusion**
   - Summary
   - Availibility
   - Outlook/Future Work
   - Acknowledgements

IEC

Strömungsforschung GmbH
ICE
Innovative Computational Engineering

The `chemistryModel`-class

- Part of the OpenFOAM^{TM}-distribution
    - Solves a set of chemical reactions and returns a **reaction-rate**
    - Used in the solvers `dieselFoam` and `reactingFoam`
- Chemical reactions can be specified in two formats:
    - A generic OpenFOAM^{TM}-format (allows the inclusion of non-standard reaction rates)
    - The Chemkin^{TM}-format which is very “Fortran”
- Both formats are converted to the same internal representation
- … and are solved using a number of different solvers

Cantera

Cantera [1] is a open-source chemical kinetics package

- Very **flexible** and easy to handle
- Important functions for chemical species, (heterogeneous) kinetics, equilibrium, transport, diffusion, heat conduction are available and efficiently implemented
- Kernel written in C++; accessible via Python, Matlab, Fortran and of course C/C++
- Good Chemkin$^{\text{TM}}$-Lexer
- A set of ideal reactors is available
- Uses the CVODE-Solver of the Sundials Suite [2] for integration of the ODEs.

Goals of this work

As mentioned in the Milan presentation [3], CANTERA would be benefitial for the development of solvers for reacting flows:

- Flexible exchange of **chemistry engines** without the necessity to change the solver
- Easy access to **thermochemical functions**
- Better **stability** (JANAF-error)
- Stationary chemistry solver

The Alternate Chemistry Library

- A Library that provides a **framework** for adding **further chemistry solvers**
- Should require only **minimal changes** to existing solvers
  - Solvers should be implementation-agnostic (by using the RunTime-selection mechanism)
- This is achieved by replacing the `chemistryModel` object (usually called `chemistry`) in the solvers with a `autoPtr<alternateChemistryModel>`
  - Direct subclassing of `chemistryModel` is not possible because necessary properties are `private`
  - Usually every call `chemistry.xxxx()` is replaced by a `chemistry().xxxx()`

Interface

- Only a **small part** of the original `chemistryModel`-interface is needed by the solvers:

  solve Solve the reaction system

  RR Access the reaction rates

  tc Access the chemical time-scale

- Only these have to be implemented by a new chemical solver class
- In addition to these the methods

  calcDQ for the calculation of `dQ`

  tf Access the flow time-scale (described later)

  were added to the interface

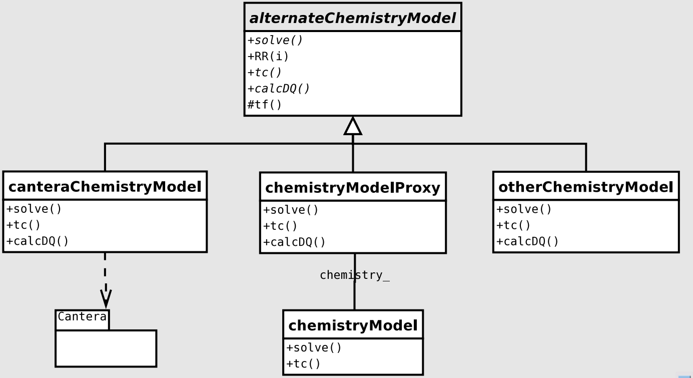
chemistryModelProxy

- Provides access to the functionality of the original `chemistryModel`-class
- Has a private `chemistryModel`-object
- Calls are passed through to the `chemistryModel`-object
    - `solve()` and `tc()`
    - `calcDQ()` implemented after example from original solvers
- With this chemistry model implementation the solver is equivalent to the original `reactingFoam`

The CANTERA-model

- Implements interface to the CANTERA-package
- Requires the `thermoType` to be a `hMixtureThermo<CanteraMixture>`
    - Instead of the usual `hMixtureThermo<reactingMixture>`
    - Currently required to have consistent calculations of the mixture and the chemistry
    - Calculation of thermophysical properties is done in `CanteraThermo` a wrapper to the CANTERA-class `IdealGasMix`
- Reactions are read from a `cti`-file (CANTERA Input)
    - Also the thermophysical data of the species. Thus the requirement on the `thermoType`
- Calculations in `solve()` are done via the `Reactor`/`ReactorNet`-classes of CANTERA
    - Solution of the ODEs is done via CVODE

Class overwiew



alternateReactingFoam

Solver almost unchanged. Basis: reactingFoam. Remember the Chalmers **PaSR-Model** for turbulent combustion:

$$\kappa = \frac{dT + tc}{dT + tc + tk} \tag{1}$$

Where *tc* is the **chemical time scale** and *tk* is the **turbulent time scale**.

- Implementation of tc()-method for the CANTERA calculation necessary.
- For the CANTERA solver no ODE solver from OF necessary (choice of the ODE solver in chemistryProperties doesn't have any effect).

alternateSteadyReactingFoam

Steady solvers are a must when simulating **large-scale chemical reactors** or **burners**.

Necessary changes from the transient case:

- Usage of the SIMPLE-algorithm for p-U-coupling
- Coupling of chemistry to the flow-time
- **Stabilization** of the solution by reduction of κ if $|Y_i^{old} - Y_i^{new}|$ is too large (only at start-up!)

Flow time scale

In **unsteady** calculations Co$<$ 1 and integration time equals dT. For **steady-state** solvers Co$>>$1. So we limit the integration time by the residence time in the cell i:

$$tf_i = \frac{D_i}{U_i} \tag{2}$$

and hence we substitute dT in (1) by df:

$$\kappa = \frac{tf + tc}{tf + tc + tk} \tag{3}$$

We define the **flow time scale** tf.

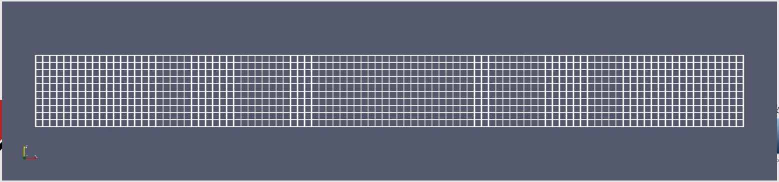
CVODE

An OF library that makes use of the CVODE-Solver of the Sundials Suite [2] was written.

Plug flow reactor (PFR) test case

Simple test case, which is easily comparable to ideal PFR models as available in CHEMKIN^{TM} or CANTERA:

- Ignitable premixed composition at inlet: $Y_{H2} = 0.009$, $Y_{O2} = 0.026$, $Y_{Ar} = 0.965$, $T = 1600$ K, $p = 1$ bar
- Tube idealized as 2D-domain $x_{max} = 1$ m, $y_{max} = 0.1$ m, 10x100 cells, inlet left patch, outlet right patch, upper and lower patch symmetry
- $H_2 - O_2$ mechanism with 9 species and 27 reactions



PFR alternateReactingFoam

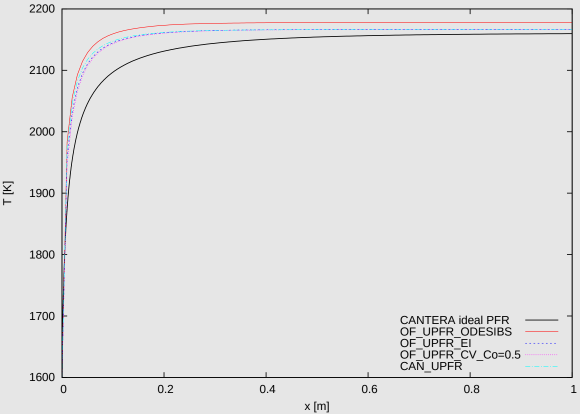
For the **transient simulations** we were interested in

- Behaviour of the CANTERA chemistry library
- Choice of ODE solvers and their performance

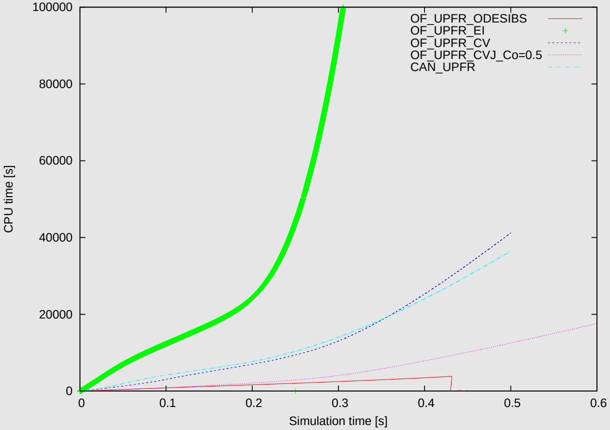
All plots were done along the **x direction**. The ideal Solution was obtained by calculating the time into a position with U and ρ.

PFR alternateReactingFoam ODE-solvers (T)

ODESIBS: ODE SIBS; EI: Euler Implicit; CV: CVODE



PFR alternateReactingFoam ODE CPUTime



PFR alternateReactingFoam Summary

Transient PFR results:

- With implementation of CANTERA chemistry and tc() results for both chemistry engines give **identical results** for PFR.
- ODE SIBS results deviate sligtly from the others (EI, CV) (due to semi-impliciteness?)
- CVODE more stable than SIBS, EI but sometimes slower
- CVODE benefits from **larger integration steps** (internal time-stepping). $\rightarrow$ So use higher Co, if results aren't affected.
- Compared to ideal reactor solvers are 'faster' and there is a temperature difference of 5 K. Possibly induced by discretisation errors or idealisation.

PFR alternateSteadyReactingFoam

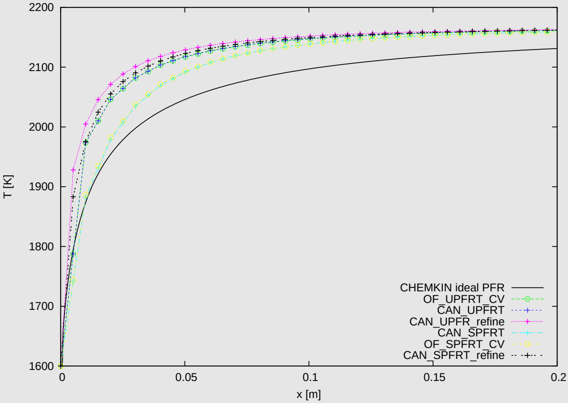
The following aspects for the runs of the **steady-state simulations** are of particular interest:

- **A** Performance and stability of the steady-state implementation in relation to transient case and the ideal case.
- **B** Influence of the implementation of κ
- **C** Influence of Cmix

Plots are similar to the transient ones but scaling was changed.

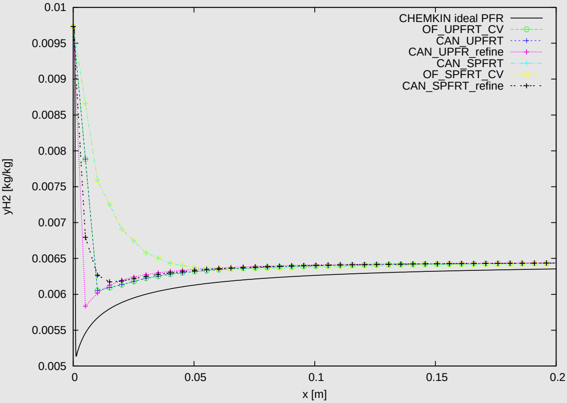
PFR alternateSteadyReactingFoam (T)

A Transient vs. steady



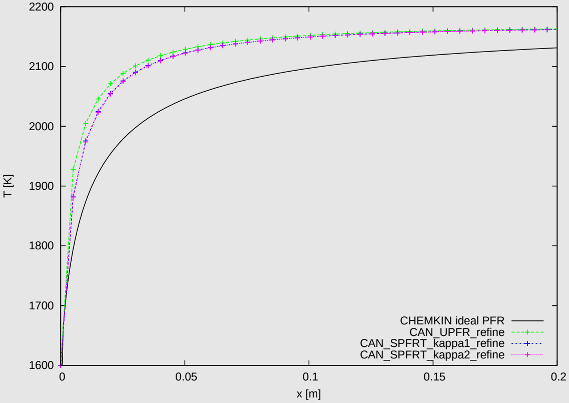
PFR alternateSteadyReactingFoam (H2)

A Transient vs. steady



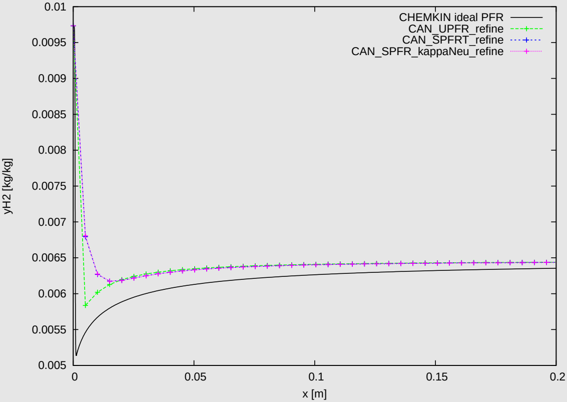
PFR alternateSteadyReactingFoam (T)

B Influence of κ



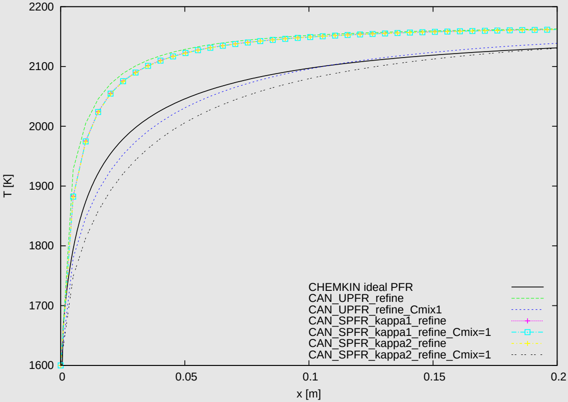
PFR alternateSteadyReactingFoam (H2)

B Influence of κ



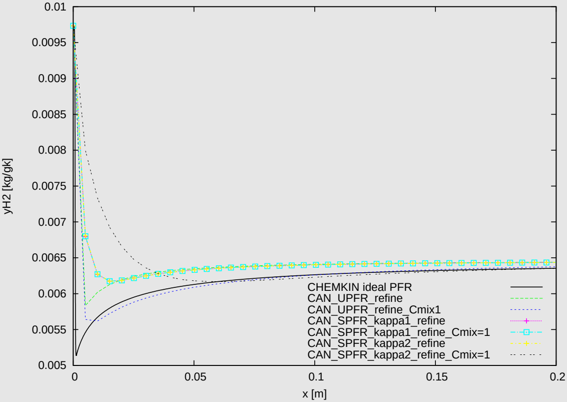
PFR alternateSteadyReactingFoam (T)

C Influence of Cmix



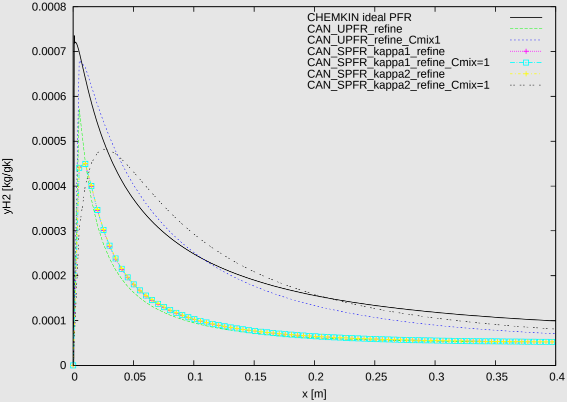
PFR alternateSteadyReactingFoam (H2)

C Influence of Cmix



PFR alternateSteadyReactingFoam (OH)

C Influence of Cmix



PFR Steady Summary

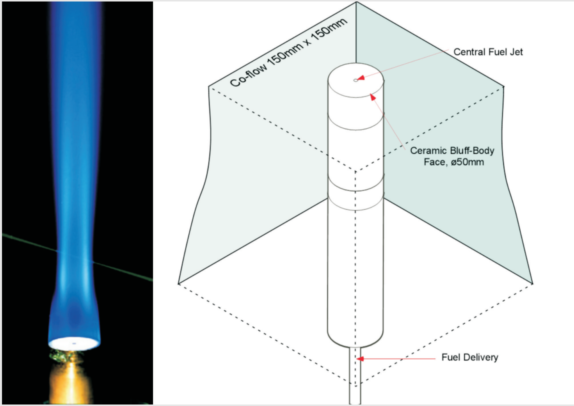
A:

- Transient/Steady solutions do **not agree totally** but **refinement reduces deviation**

B+C:

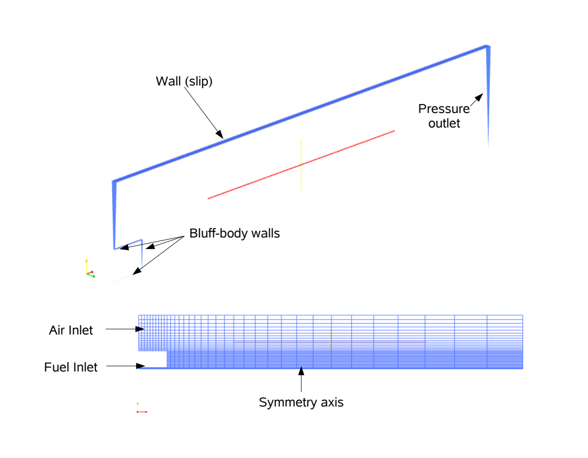
- Definition of new κ according to (3) brings no changes for small values of Cmix. But for Cmix=1 the solution seems closer to ideal solution.
- However, no clear conclusion could be drawn.

Sydney Bluff-body Flame (HM1)



Data from Sandia bluff-body flame HM1 conducted at Sydney university [4].

HM1 case set-up



HM1 set-up

Boundary Conditions:

- Diameters: $D_{jet} = 3.6\text{mm}$; $D_{channel} = 150\text{mm}$ (300mm), $D_{bluffbody} = 50\text{mm}$
- Turbulence:8.5 % (2.5%) turbulence intensity , 0.135 mm (5.625 mm) mixing length for jet (co-flow)
- Mesh: from blockMesh with 1095

Inlet conditions:

- HM1: $U_{jet} = 118\text{m/s}$ $U_{coflow} = 40\text{m/s}$

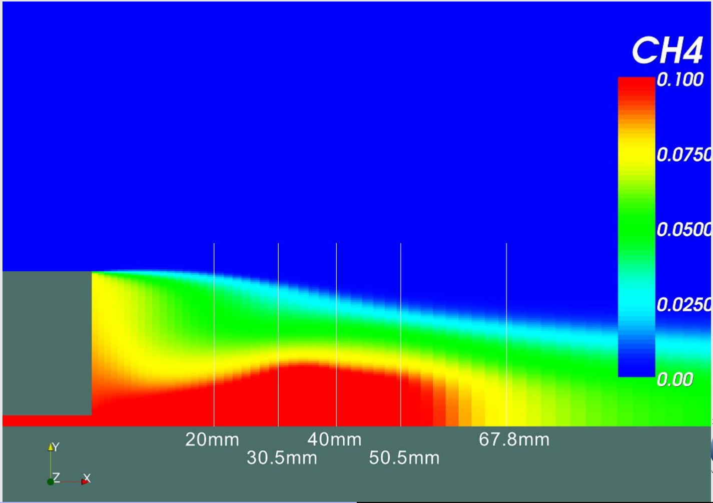
Other models:

- $k - \epsilon$ turbulence model, ATRMech (reduced GRI 3.0)
- OF used KRR4 ODE solver for chemistry

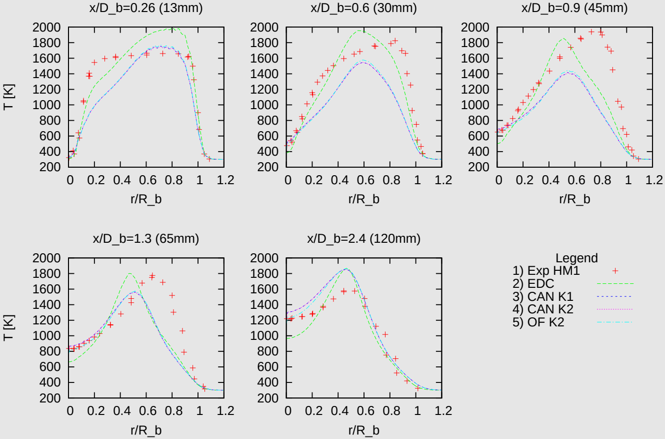
HM1 Plots & Points of Measurement

- Plots of **radius** against measurement **values** (**Exp HM1**) at different **x-positions** are shown together with simulation results.
- **EDC** referes to the EDCSimpleFoam solver [3].
- **CAN** and **OF** mark CANTERA- and OPENFOAM^{TM}-solution
- **K1** and **K2** stands for κ_1 and κ_2

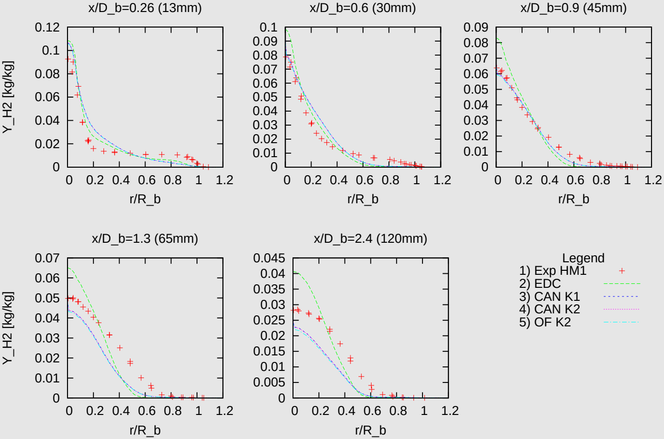
HM1 Plots & Points of Measurement



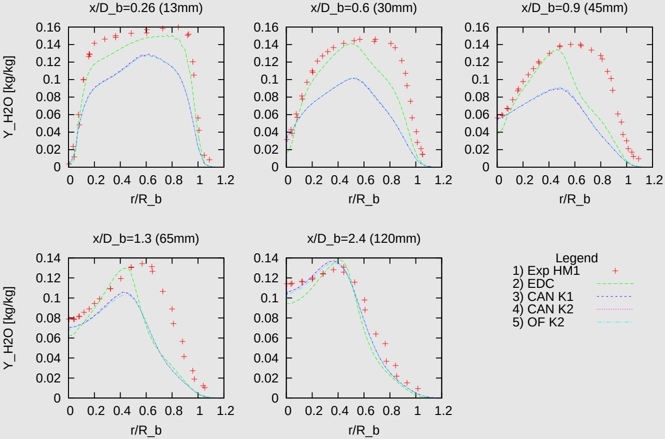
HM1 results T



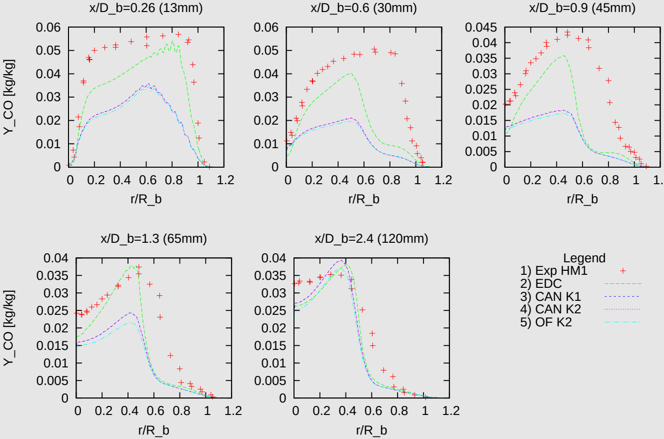
HM1 results H2



HM1 results H2O



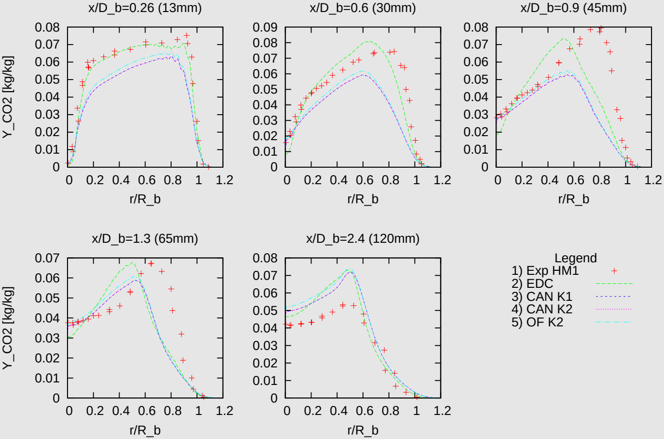
HM1 results CO



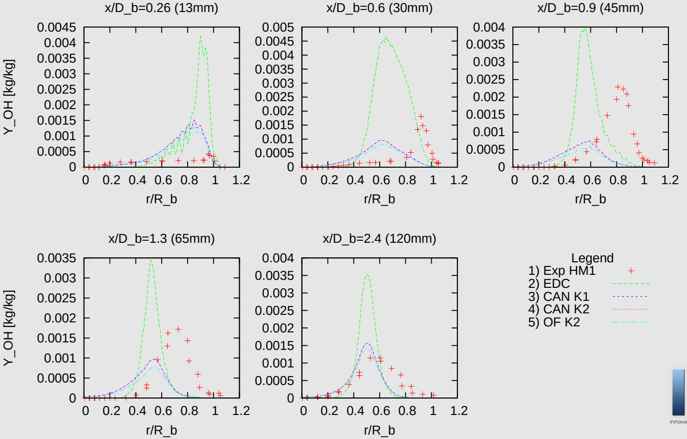
IEC

Strömungsforschung GmbH
ICE
Innovative Computational Engineering

HM1 results CO2



HM1 results OH

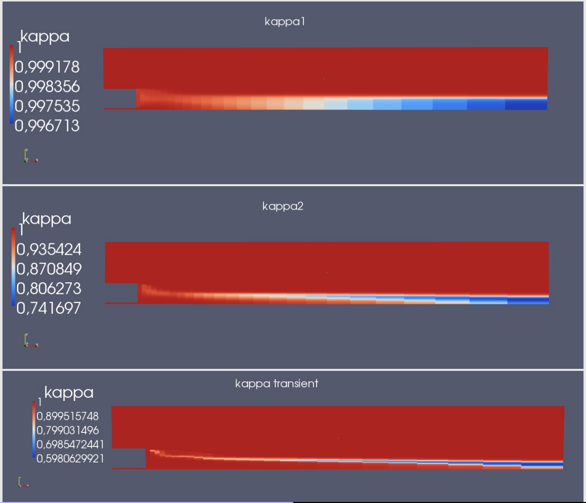


Discussion HM1

Why are all the resuls **so similar**? (No influence of κ-implementation or Cmix (not shown))

- In case Cmix=0.005 no difference in κ notable - too fast mixing as in PFR case.
- Furthermore, in the domain **close to the inlets** $\kappa \approx 1$ in every case and influence of Cmix can only be seen **further downstream** where no measurements are available.
- Comparison with **transient simulation** of HM1 indicates that κ_2 seems more appropriate (see next slide)
- Still, further testing is needed

HM1 comparison κ_1, κ_2 for Cmix=1



Conclusion HM1

- Flame was rendered with **acceptable quality** by the new solvers
- Steady solvers are **very robust** if the right ODE solvers are used (CVODE for CANTERA, KRR4 for OF)
- Checking of dY_i provides further stability and and prevents composition from being driven to unphysical states.
- Solutions for a finer grid (20000 cells) are still not available - flame **blows off** very easily.

Summary

The implementation of the CANTERA chemistry into OPENFOAM^{TM}can be considered successful and provides:

- A **flexible interface** to access a wide range of thermophysical data,and chemistry functions
- Robust **solvers** for transient and steady-state calculations
- Valuable **test-cases** for validation and further investigation
- **Collaborators** are welcome to test the code and contribute own developments.

How to get the library

- The libraries and solvers presented here will be made available on the `openfoam-extend`-SVN on `sourceforge.net`

  Libraries Two independent libraries:

  alternateChemistryModel The general framework an the OpenFOAM^{TM}-interface

  canteraChemistryModel The interface to Cantera

  Solvers The steady and the transient solver with the test-cases PFR and HM1

- `URL` will be given on the Wiki
- Interfaces to other **chemistry libraries**, more **solvers**, critique, and improvements are encouraged

Future Work

There is always room for improvement:

- Both chemistry solvers read the same input files
  - Converter between CANTERA and OPENFOAM™-format
- Make the CANTERACHEMISTRYMODEL indepentent of `canteraMixture`
- Implementation of reduction/tabulation techniques for more time-efficient solution.
- More combustion models.

Acknowledgements

A part of the development and implementation work was done by the IEC in the context of the research project "Grosstechnisch bertragbare Untersuchungen zur Weiterentwicklung der IGCC-Kohlekraftwerkstechnik mit CO2-Abtrennung - **COORAMENT**". This project is supported by the **German Federal Ministry of Economics and Technology** (Project ID 0327113B).
Furthermore we thank N. Nordin for a valuable posting regarding the tc()- implementation, all developers who contributed to OpenFOAM$^{\text{TM}}$, and people who inspired that work.

Bibliography

[1] DG Goodwin. Cantera: Object-oriented software for reacting flows. Technical report, California Institute of Technology, 2002.

[2] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[3] M. Rehm, P. Seifert, and B. Meyer. Towards a CFD Model for Industrial Scale Gasification Processes. In *OpenFOAM Workshop 2008, Milan*, 2008.

[4] BB Dally, AR Masri, RS Barlow, and GJ Fiechtner. Instantaneous and Mean Compositional Structure of Bluff-Body Stabilized Nonpremixed Flames. *Combustion and Flame*, 114(1-2):119–148, 1998.