Fire Simulation with adaptive FEM

April 3, 2018 | Marc Fehling | Civil Safety Research (CSR)

Introduction

- Marc Fehling
- M. Sc. Physics, Ruhr-University Bochum, Germany, 2015

- since November 2015:
  PhD candidate at
  Jülich Research Centre (FZJ),
  institute Civil Safety Research (CSR),
  associated with the University of Wuppertal (BUW)

Introduction: Location

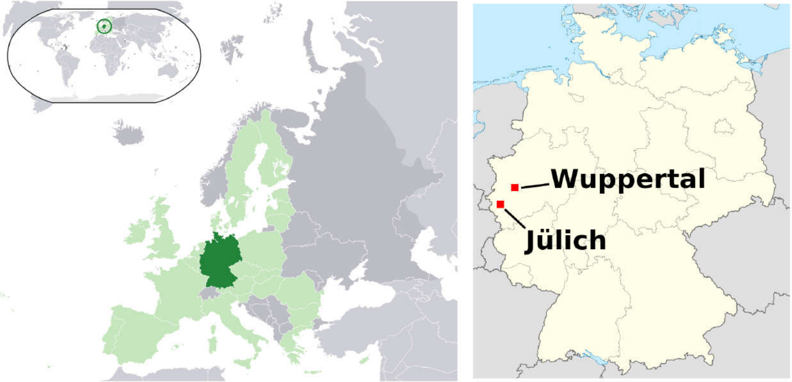


Figure: Locations of the Jülich Research Centre and the University of Wuppertal. (source: wikipedia)

JÜLICH FORSCHUNGSZENTRUM

Introduction: Civil Safety Research

- Concerns: Pedestrian behavior and safety
  - i.e. in emergency situations (e.g. caused by fires)
- Scientists from various fields
  - mathematicians, physicists, engineers, programmers, etc.

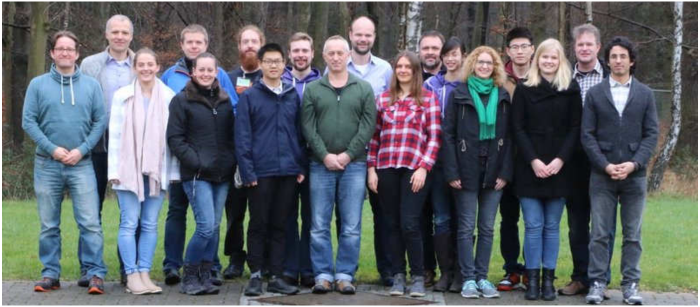


Figure: Members of the institute Civil Safety Research.

Introduction: Supercomputers

- Jülich Supercomputing Centre (JSC) houses supercomputers
  - General purpose cluster JURECA
  - High scaling cluster JUQUEEN
    - Shutdown in March 2018. Replaced by successor JUWELS.

Figure: Supercomputers at Research Centre Jülich. (source: JSC)

Table of contents

Motivation

Modeling fires

Numerical methods

Adaptive Mesh Refinement (AMR)

Unstructured grids

High–performance computing

Summary & Outlook

Motivation: Fire Safety

- Safety measures increase chance of survival of inhabitants in case of emergency.
- Fire safety measures:
  - Fire doors, building materials, smoke extraction, sprinklers, ...
- German legislation prescribes 'building codes'.
  - For 'normal' purposes:
    - residential buildings
  - For 'special' purposes:
    - event halls
    - malls
    - garages
    - factories
    - skyscrapers
    - airports / train stations
    - ...

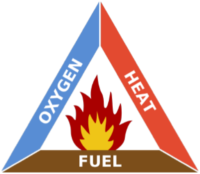


Figure: Fire triangle. (source: wiki)

Motivation: Architecture

- Modern architecture and large–scale projects may not fit in 'building code'
- Individual considerations necessary:
  - Model experiments, CFD investigations, ...

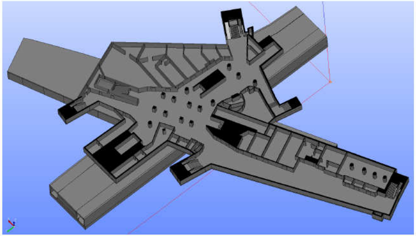
Figure: Interior of BMW World, Munich. (source: mazdaspeed.pl)

Motivation: Project Orpheus

ORPHEUS

Optimierung der **R**auchableitung und **P**ersonenführung in U-Bahn**h**öfen: **E**xperimente **u**nd **S**imulationen

- Smoke and fire spread in complex metro stations
  - High dimensional parameter space for fire scenarios, including underground climate



Motivation: Project Orpheus II

Figure: Video of experiments in the metro station.

Modeling: Fires

Processes to consider:

- Fluid dynamics
  - Navier–Stokes equations
  - Turbulence
- Combustion
- Radiation
- Conduction
- Pyrolysis

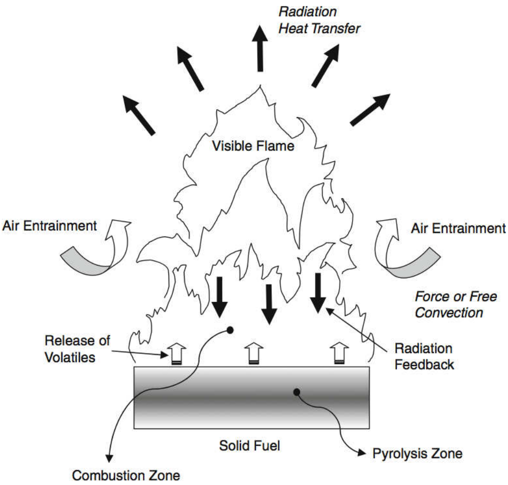


Figure: Burning solid fuel in air with physical processes involved. [1]

Modeling: Buoyancy driven flows

- Consider low Mach number limit ($\mathrm{Ma} = u/c_{\mathrm{sound}} \to 0$)
- Propagation of fluids with incompressible Navier–Stokes (INS) equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \rho_0 \left[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}\right] + \nabla p - \nabla(2\,\mu\,\epsilon_{ij}(\mathbf{u})) &= \mathbf{f}(T) \\ \rho_0 c_p \left[\partial_t T + (\mathbf{u} \cdot \nabla) T\right] - \nabla \cdot (\kappa \nabla T) &= g(\mathbf{u}) \end{aligned}$$

  with strain rate tensor $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right]$.
- Prandtl number connects viscous to thermal diffusion:

$$\mathrm{Pr} = \frac{\nu}{\alpha} = \frac{c_p \mu}{\kappa}$$

  with $c_p = \frac{5}{2}\frac{R}{M}$ specific heat for monoatomic ideal gases

Modeling: Turbulence

- Avoid resolving the dissipative scales by modeling turbulence
  - Kolmogorov length of air ~25 µm
- Apply filtering to model effect of small eddies
  - time filtering: Reynolds-Averaged-Navier-Stokes (RANS)
    - separation of observables in mean and fluctuations
    - averages over fluctuations
    - justification: transport processes happen on large time scale
  - space filtering: Large Eddy Simulation (LES)
    - averaging over filter width Δ on subgrid scales (SGS)
    - result: effect of small dissipating eddies by adjusting viscosity

Modeling: LES

- Filter width typically related to mesh: $\Delta = (h_x\, h_y\, h_z)^{1/3}$
- Superposition of viscosity: $\mu = \mu_{mol} + \mu_{turb}$
- Smagorinsky-Lilly model:

$$\mu_{turb} = \rho (C_s \Delta)^2 \left|\left|\epsilon_{ij}(\mathbf{u})\right|\right|_2 \ , \qquad C_s = 0.2$$

- Deardorff model:

$$\mu_{turb} = \rho\, C_V\, \Delta \frac{1}{\sqrt{2}} ||\bar{\mathbf{u}} - \hat{\bar{\mathbf{u}}}||_2 \ , \qquad C_V = 0.1$$

with $\bar{\mathbf{u}}$ cell-averaged and $\hat{\bar{\mathbf{u}}}$ a weighted average, i.e.

$$\hat{\bar{u}}_{x,ijk} = \frac{\bar{u}_{x,ijk}}{2} + \frac{\bar{u}_{x,i-1,jk} + \bar{u}_{x,i+1,jk}}{4}$$

Modeling: Source terms

- Momentum sources: Gravity and buoyancy
- Buoyancy effects modeled with Boussinesq approximation

$$\mathbf{f}(T) = \underbrace{\rho_0\,\mathbf{g}}_{\text{gravity}} - \underbrace{\rho_0\,\mathbf{g}\,\beta(T - T_0)}_{\text{buoyancy}}$$

with $\beta = T_0^{-1}$ thermal expansion coefficient for ideal gases

- Temperature sources: Convective heat and friction

$$g(\mathbf{u}) = \underbrace{\dot{q}'''_{conv}}_{\text{convective heat}} + \underbrace{\nabla\left(2\,\mu\,\mathbf{u}\cdot\epsilon_{ij}(\mathbf{u})\right)}_{\text{friction}}$$

Modeling: Heat transfer

- Heat released by combustion transfers into three different types

$$\dot{q}'''_{combustion} = \dot{q}'''_{conduction} + \dot{q}'''_{convection} + \dot{q}'''_{radiation}$$

- There is no rule on how generated heat is partitioned
- Needs to be determined empirically / estimated

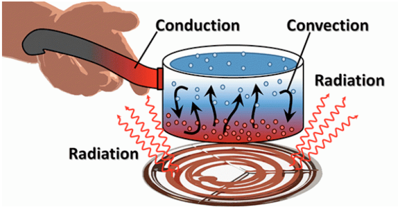


Figure: Types of heat transfer.
(source: machinedesign.com)

Modeling: Combustion: Infinitely fast chemistry

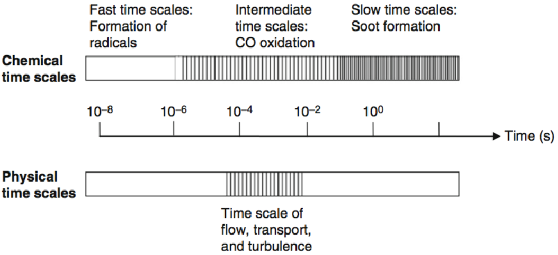


Figure: Chemical and physical time scales. [1]

- Simplification: Infinitely fast reaction of lumped species

$$\dot{q}'''_{\text{comb}} = -\sum_{\alpha} \Delta h_{f,\alpha}\, \dot{m}'''_{\alpha}$$

with heat of formation Δh_f and mass generation $\dot{m}'''$

Modeling: Combustion: Finite rate formulation

- Keep track of each lumped species via mass fractions

$$Y_i = \frac{m_i}{m}\,, \quad \partial_t(\rho\, Y_i) + \nabla \cdot (\rho\, \mathbf{u}\, Y_i) - \nabla \cdot (\rho\, D_i\, \nabla\, Y_i) = R_i$$

with reaction rate of fuel depending on concentrations

$$R_{fuel} = k(T)\, C_{fuel}\, C_{oxy} = k(T)\, \rho^2\, Y_{fuel}\, Y_{oxy}$$

- Specific reaction rate as Arrhenius expression

$$k(T) = A_0\, T^n \exp\left(-\frac{E_a}{k_B\, T}\right)$$

where A_0, n, E_a are empirically determined

Modeling: Combustion: Conserved scalars

- Mixture of oxygen and fuel gases determine reaction

$$f\,Y_f + (1-f)\,Y_A = Y_M \qquad \Rightarrow \qquad f = \frac{Y_M - Y_A}{Y_F - Y_A}$$

- With equal diffusivities, transport eqs. can be reduced to

$$\partial_t\,(\rho\,f) + \nabla\cdot(\rho\,\mathbf{u}\,f) - \nabla\cdot\left(\left(\frac{\mu}{\mathrm{Sc}} + \frac{\mu_T}{\mathrm{Sc}_T}\right)\nabla f\right) = 0$$

with Schmidt number $\rho D = \mu\mathrm{Sc}$.
Subscript T denotes turbulent properties.

- Source terms cancel, thus f is a conserved scalar

Modeling: Combustion: Conserved scalars II

- Determine mass fractions Y via state relationships

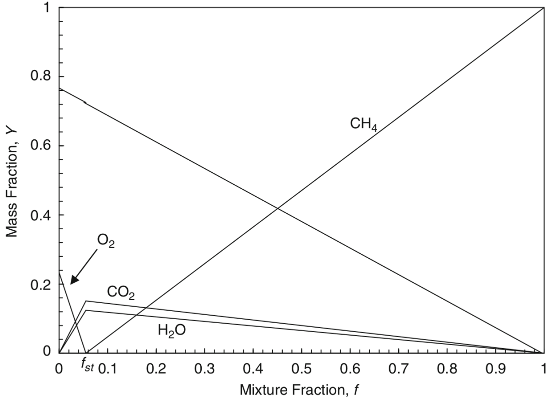


Figure: Single step combustion state relationships for methane. [1]

Modeling: Even more to come...

Model can be arbitrarily complex. What we haven't dealt with

- Pyrolysis – thermochemical decomposition of solid fuel
- Radiation
- Thermal conduction through solid materials
- Flame propagation
- Turbulent combustion
- ...

Literature recommodations:

- NIST FDS documentation. https://pages.nist.gov/fds-smv
- Yeoh & Yuen. CFD in Fire Engineering. doi:10.1016/B978-0-7506-8589-4.X0001-4

Member of the Helmholtz-Association

JÜLICH FORSCHUNGSZENTRUM

Numerical methods

We consult numerical methods to solve the model equations

Spatial discretization methods for computational fluid dynamics and software packages using it:

- Finite Difference Method (FDM)
  - Open source: NIST Fire Dynamics Simulator (FDS), ...
- Finite Volume Method (FVM)
  - Open source: FireFOAM (OpenFOAM), ...
- Finite Element Method (FEM)
- Lattice–Boltzmann Method (LBM)

FEM: Features

Why use FEM?

FEM: Features

Why use FEM?

- Unstructured grids
  - Allow for better domain representation without aliasing

FEM: Features

Why use FEM?

- Unstructured grids
  - Allow for better domain representation without aliasing
- hp–adaptivity
  - Dynamic resolution of numerical grid
  - Adaptive polynomial degree of basis functions

  $\rightarrow$ Increase accuracy where the action is happening!

FEM: Features

Why use FEM?

- Unstructured grids
  - Allow for better domain representation without aliasing
- hp–adaptivity
  - Dynamic resolution of numerical grid
  - Adaptive polynomial degree of basis functions

  $\rightarrow$ Increase accuracy where the action is happening!
- Discontinuous Galerkin (DG) methods
  - Allow discontinuities across cell borders
  - Continuous Galerkin (CG) methods unstable for advection like problems, thus require stabilization

JÜLICH
FORSCHUNGSZENTRUM

Our implementation

- Motivation of PhD project:
  - Investigate applicability of adaptive methods for fire simulation
  - Use FEM with its unique feature of hp–adaptivity
- Use open–source library deal.II [2]
  - 'Toolbox' for the creation of FEM codes

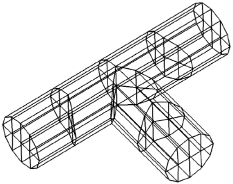
deal.II

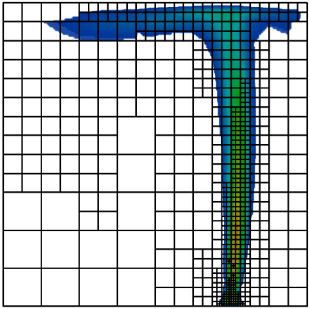
Differential **E**quations **A**nalysis **L**ibrary

Our implementation: Features

Implemented:

- Unstructured grids
- Continuous Galerkin Methods
- Adaptive mesh refinement (AMR)
- MPI parallelization
- Utilization of CAD models as manifolds for mesh refinement





Current field of activity:

- Verification of flow solver
- Discontinuous Galerkin methods
- p–adaptivity

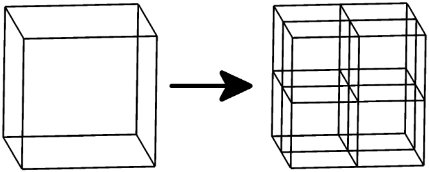
Adaptive mesh refinement (AMR)

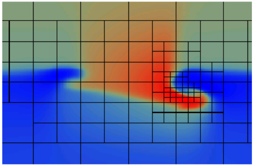
- Solution of differential equations with *FEM* requires:
  - Discretization of space in cells of length h
  - Shape functions with polynomial degree p
- But: The more accurate the solution by choosing h and p, the longer the computation

Adaptive mesh refinement (AMR)

- Solution of differential equations with *FEM* requires:
  - Discretization of space in cells of length *h*
  - Shape functions with polynomial degree *p*
- But: The more accurate the solution by choosing *h* and *p*, the longer the computation

→ **Adaptive refinement** as a 'compromise'
  - Adjustment of parameters locally where necessary
  - Variable mesh precision at runtime





J.Dreher, RUB

AMR: Algorithm

How to determine refinement criteria?

Set up decision criterion for refinement/coarsening
Our choice:

- Calculate $||\nabla u||$ on each cell
- Normalize values with respect to all cells

AMR: Algorithm

How to determine refinement criteria?

Set up decision criterion for refinement/coarsening
Our choice:

- Calculate $||\nabla u||$ on each cell
- Normalize values with respect to all cells
- Flag top/lower 30% for refinement/coarsening at each step
- Define max/min refinement levels as upper/lower bounds

AMR: Algorithm

How to determine refinement criteria?

Set up decision criterion for refinement/coarsening
Our choice:

- Calculate $||\nabla u||$ on each cell
- Normalize values with respect to all cells
- Flag top/lower 30% for refinement/coarsening at each step
- Define max/min refinement levels as upper/lower bounds
- Neighboring cells are not allowed to differ by more than one level of refinement

AMR: Verification Example

- Demonstration of adaptive mesh refinement via moving vortex test case as a shape–preserving potential stream

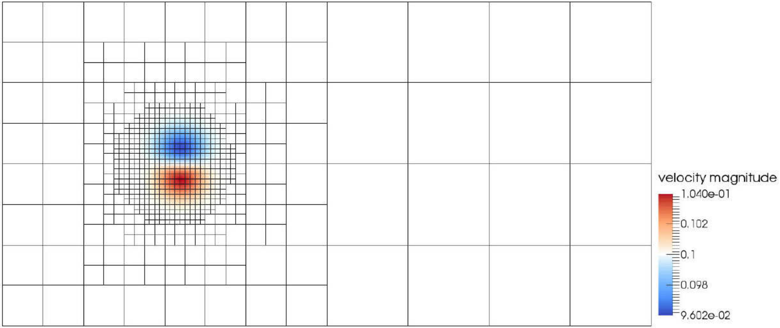
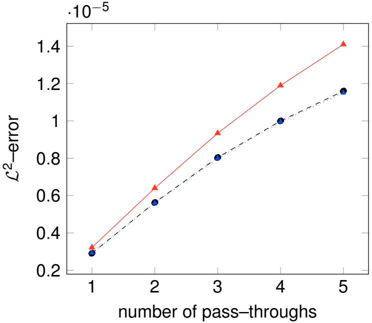


Figure: Video of velocity magnitude of moving vortex, overlayed with corresponding mesh.

AMR: Benefits

- Comparison of runtime and accuracy between uniform and adaptively refined meshes with the moving vortex example



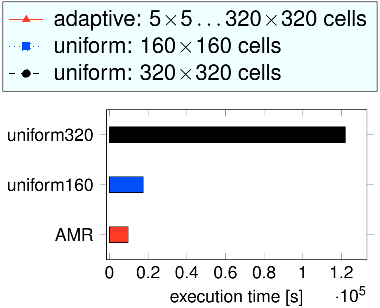


Figure: Global $\mathcal{L}^2$–errors at each periodic pass–through of the vortex.

Figure: Execution time of the vortex simulation, run in serial on a common desktop computer.

AMR: Errors

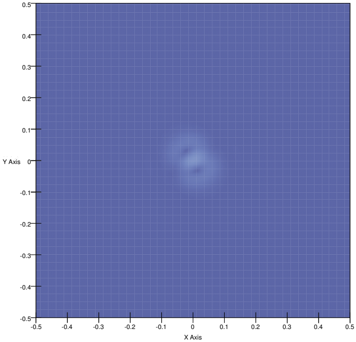


Figure: Static mesh.

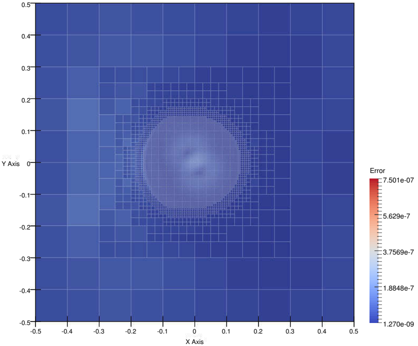


Figure: Adaptive mesh.

1 pass–through.

AMR: Errors

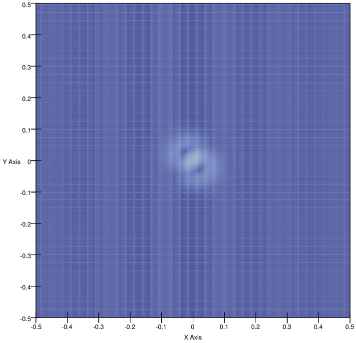


Figure: Static mesh.

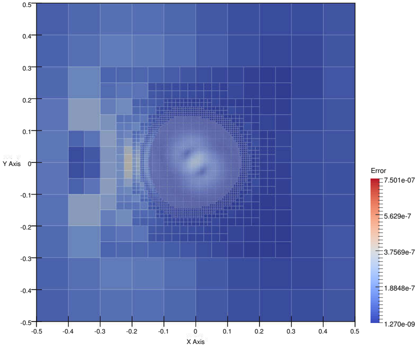


Figure: Adaptive mesh.

2 pass–throughs.

AMR: Errors

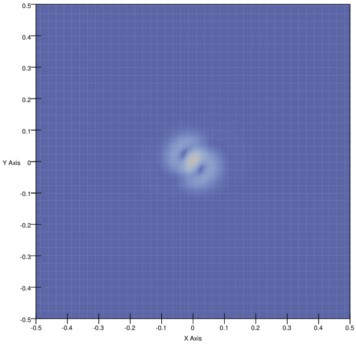


Figure: Static mesh.

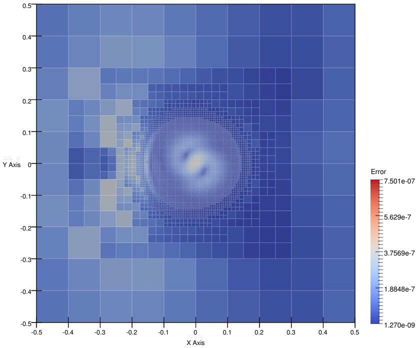


Figure: Adaptive mesh.

3 pass–throughs.

AMR: Errors

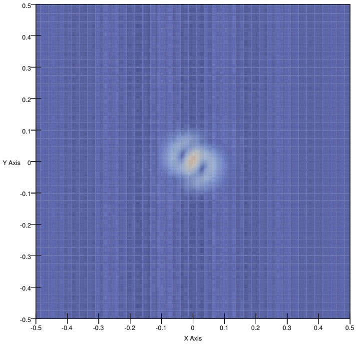


Figure: Static mesh.

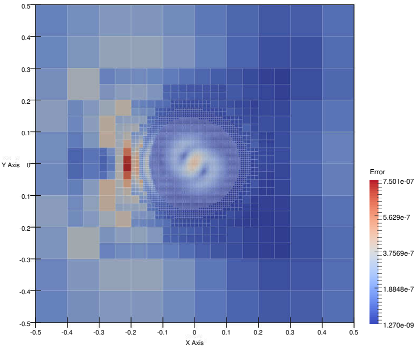


Figure: Adaptive mesh.

4 pass–throughs.

AMR: Errors

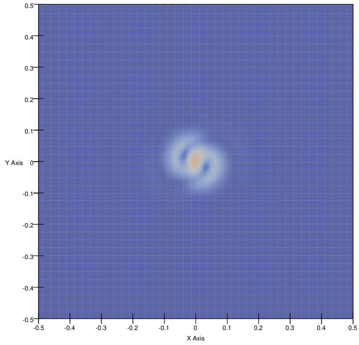


Figure: Static mesh.

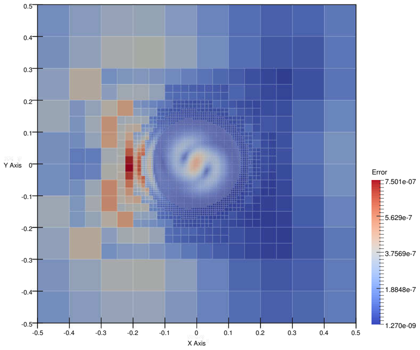


Figure: Adaptive mesh.

5 pass–throughs.

AMR: Buoyancy example

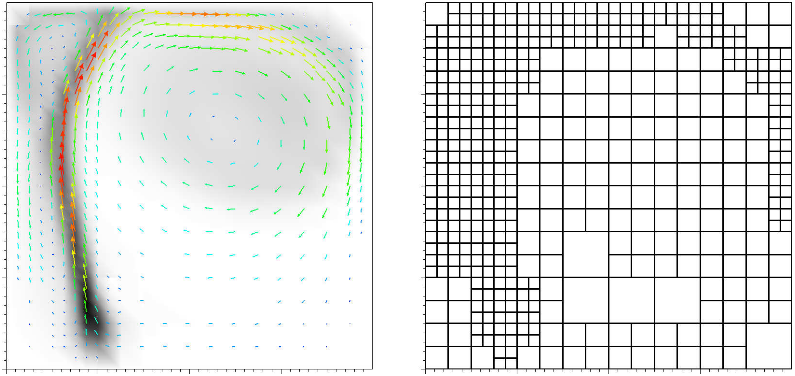
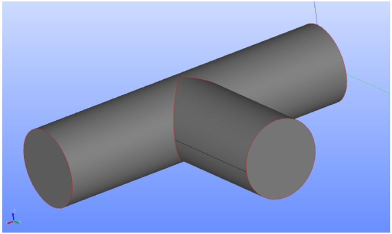


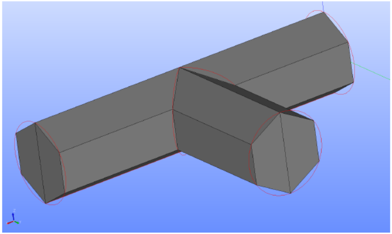
Figure: Examplary simulation of a buoyancy driven flow with a Gaussian heat source.

Unstructured Grids: Example: T–pipe

- Why T–pipe?
  - Compound bodies
  - Crooked areas (→ cylinder casing area)
  - Discontinuous edges (→ 'welding seam')
- Develop procedure for the creation of appropriate initial meshes

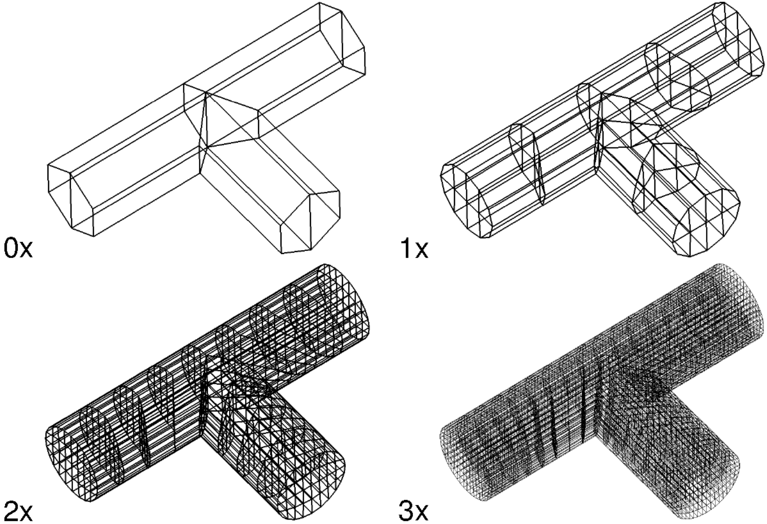


CAD model.



Initial mesh.

Iterations of global refinement:



HPC on JURECA: Parallelization

- `MPI` parallelization with `Trilinos` and `p4est` through `deal.II` backends

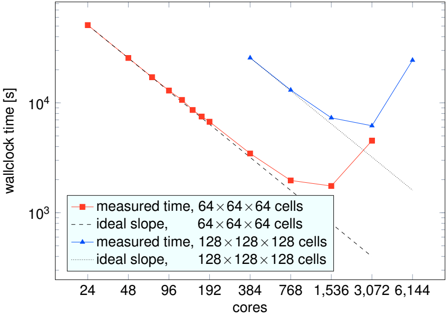


Figure: Strong scaling on JURECA for fixed 3D problems with 262,144 and 2,097,152 cells, respectively.

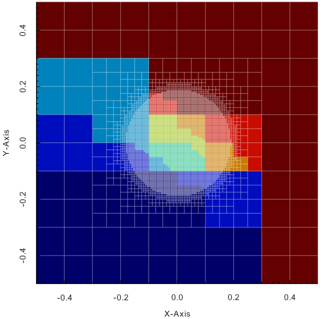


Figure: Examplary domain decomposition with `p4est` in an AMR case.

Summary & Outlook

Summary:

- Adaptive Mesh Refinement works satisfactorily (for now)
- Flow solver accomplishes first verification tests

Future work:

- Implementation of models
  - Radiation, combustion, pyrolysis, ...
- Extension of numerical methods
  - DG methods, p-adaptivity, ...
- Comparison with other fire solvers
- Validation using experiments

Thank you for your kind attention!

References I

- Guan Heng Yeoh and Kwok Kit Yuen.
  Computational Fluid Dynamics in Fire Engineering – Theory, Modelling and Practice.
  Butterworth–Heinemann, 1st edition, 2009.

- D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells.
  The `deal.II` library, version 8.5.
  Journal of Numerical Mathematics, 25(3):137–146, 2017.

References II

Cedric Taylor and P. Hood.
A numerical solution of the Navier–Stokes equations using the finite element technique.
Computers & Fluids, 1:73–100, 1973.

Bernardino Roig.
One–step Taylor Galerkin methods for convection–diffusion problems.
Journal of Computational and Applied Mathematics, 204:95–101, 2007.

References III

N.D. Heavner and L.G Rebholz.
Locally chosen grad–div stabilization parameters for finite elemnt discretizations of incompressible flow problems.
SIAM Undergraduate Research Online (SIURO), 7.

Randy McDermott.
A non–trivial analytical solution to the 2–D incompressible Navier–Stokes equations, August 2003.
[cited 04/02/2018].

Fire Simulation with adaptive FEM: Addendum

April 3, 2018 | Marc Fehling | Civil Safety Research (CSR)

JÜLICH FORSCHUNGSZENTRUM

FEM: Formulation

- Solve variational equation from 'weak' formulation of the differential equation with bilinear form $a(u, v)$:

$$\exists u \in V : \forall v \in V : a(u, v) = f(v)$$

- Choose subspace V_h with basis w_i, out of which the approximate solution $u_\mathrm{h} = \sum u_i w_i \in V_\mathrm{h}$ will be constructed:

$$a(u_\mathrm{h}, w_\mathrm{j}) = \sum a(w_i, w_\mathrm{j})\; u_i = f(w_\mathrm{j}) \quad \rightarrow \quad AU = F$$

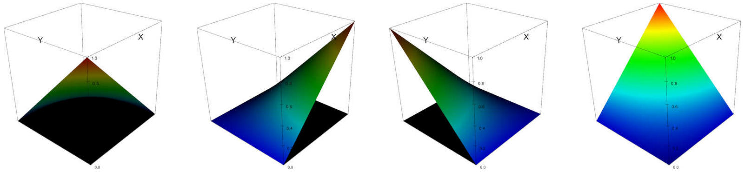


Figure: Q_1 elements in 2D (source: deal.II)

Our Implementation: Algorithm

- Time evolution with backward differentiation formula (BDF)
- Taylor–Hood elements [3]: $(\mathbf{u}, p) \in Q_{k+1}^{\mathrm{dim}} \times Q_k$
- Decoupling of $(\mathbf{u}, p)$ by projection scheme
  - Leads to the pressure–Poisson equation
- Stabilization of momentum equation
  - Taylor–Galerkin stabilization [4] for Taylor–Hood elements
    - Additional diffusion in flow direction
  - Grad–div stabilization [5] to enforce $\nabla \cdot \mathbf{u} = 0$
- Neumann series for fast matrix assembly
- Boussinesq approximation for buoyancy force density

Verification: Richardson Extrapolation

- How to get convergence rates of time **and** space dependent problems?
- Cancel out redundant error with Richardson extrapolation. Constant ratio r for successive refinement (here: space).

$$\left.\begin{aligned} f_1 - f_0 &= c_t\,\Delta t^{p_t} + c_h\,\Delta h^{p_h} \\ f_2 - f_0 &= c_t\,\Delta t^{p_t} + c_h\,(r\Delta h)^{p_h} \\ f_3 - f_0 &= c_t\,\Delta t^{p_t} + c_h\,(r^2\Delta h)^{p_h} \end{aligned}\right\} \quad \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) = p_h\ \ln(r)$$

	Space	Time
FDS	1.9244	2.0721
JuFire	3.0894	0.9715

Table: Examplary convergence rates for a McDermott testcase [6].

Buoyancy driven flows – Work in progress

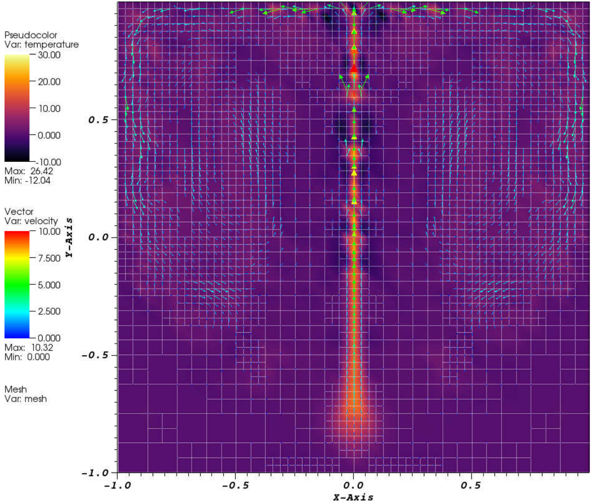


Figure: Video of velocity & temperature with constant heat source.

Buoyancy driven flows – Possible error source

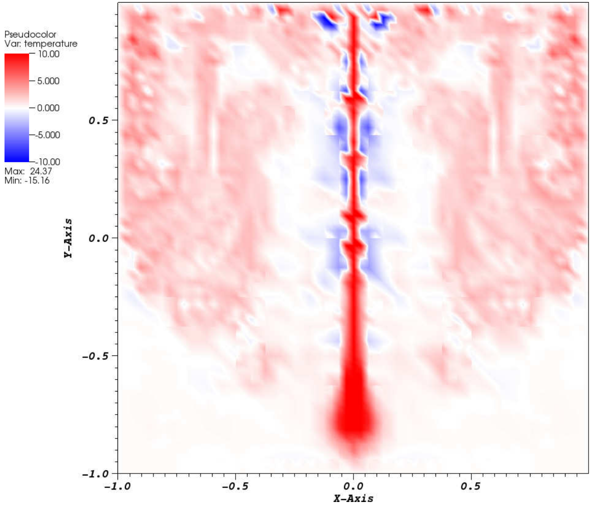


Figure: Temperature with constant heat source.

Heat equation with FEM – Possible error source

$$\partial_t u - \nu \nabla^2 u = 0$$

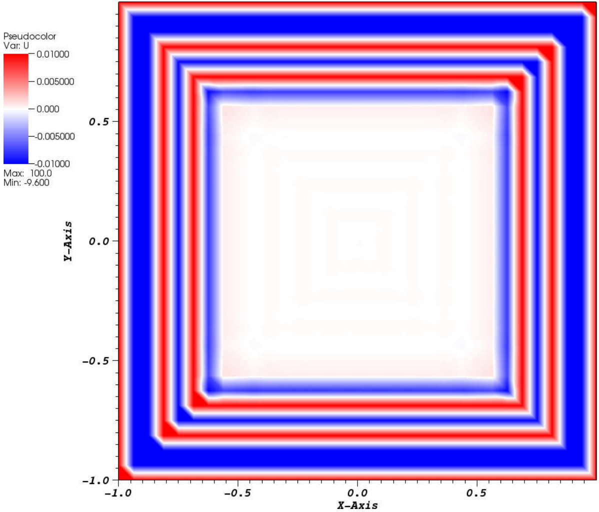


Figure: Solution of heat equation at advanced time.