



Introduction to the 4C Multiphysics Framework @UKACM & GACM Autumn School 2025

^aSebastian Brandstaeter, ^bGeorg Hammerl, ^{a,c}Matthias Mayr,
^dGil Robalo Rei, ^bIngo Scheider, ^dChristoph P. Schmidt

^aInstitute for Mathematics and Computer-Based Simulation, University of the Bundeswehr Munich

^bInstitute of Material Systems Modeling, Helmholtz-Zentrum Hereon, Geesthacht

^cData Science & Computing Lab, University of the Bundeswehr Munich

^dInstitute for Computational Mechanics, Technical University of Munich

Reminder

- Please **download** the **virtual box**, if not yet already done.
- Detailed instructions are available in the course material at https://github.com/mayrmt/UKACM_GACM_Tutorial_4C_QUEENS/blob/main/PREPARATION.md .

Introduction to the 4C Multiphysics Framework

- **What is 4C Multiphysics?**
- Capabilities of 4C Multiphysics
- How to build 4C Multiphysics
- The input file
- Tips and tricks to create a 4C input file

What is 4C Multiphysics?

- Modular, parallel, and open-source simulation environment
- Solves single- and multiphysics real-world problems
 - Described by differential equations (meso and macro scale)
 - Focus on finite element and particle methods
 - Applied in engineering, science, and biomedicine
 - Ready to use applications/solvers
- Extensible framework for methods development and new applications (C++)
 - Single fields
 - Coupling algorithms (surface-to-surface, volume-to-volume, (non-)matching grids)
 - Linear Algebra/Solvers/Preconditioners



What is 4C Multiphysics?

- Not yet heard of 4C Multiphysics?
 - Open source since 10/2024 under LGPL-v3.0-or-later
<https://github.com/4C-multiphysics/4C>
 - Roots date back >20 years, > 50 PhD theses, > 400 papers
- Ca. active 35 developers
- Quarterly release, latest v2025.3.0

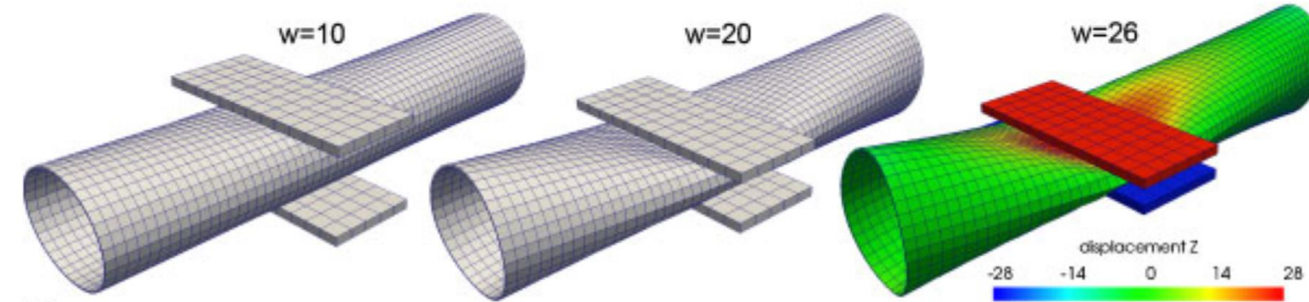


Introduction to the 4C Multiphysics Framework

- What is 4C Multiphysics?
- **Capabilities of 4C Multiphysics**
- How to build 4C Multiphysics
- The input file
- Tips and tricks to create a 4C input file

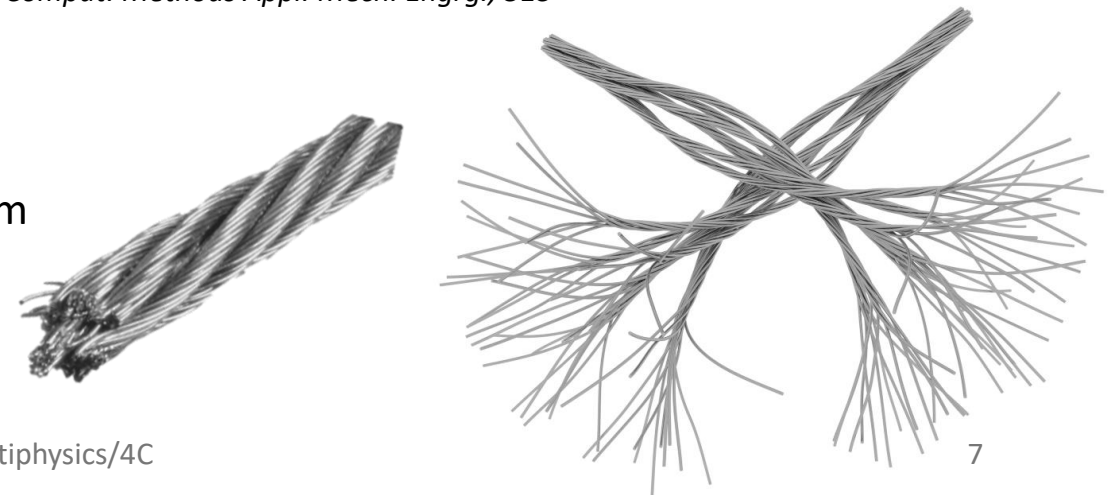
Solid mechanics

- 3D solids (EAS, Fbar), 2D (plane strain, plane stress), shells, beams
- Small / finite deformations
- Explicit / implicit time integration
- Various material models
 - (Hyper-)elasticity
 - Viscosity
 - Plasticity
 - Fiber reinforced materials
- Contact formulations:
 - Penalty, Lagrange Multipliers, Nitsche
 - node-to-segment, segment-to-segment (mortar)
- Example 1: Large-scale dynamic contact analysis of thin-walled pipe (continuum shell)
- Example 2: Dynamic failure of steel ropes modelled on the scale of individual wires based on Cosserat continua employing a beam formulation



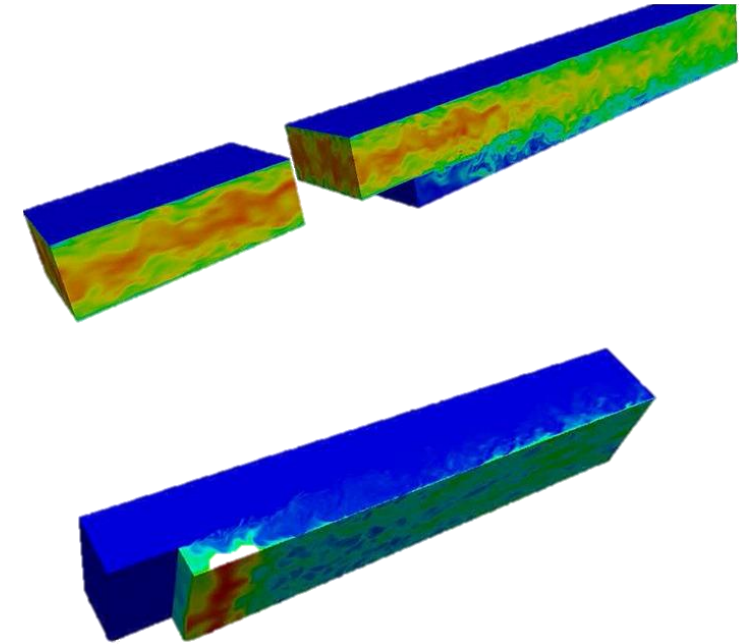
A. Popp, M. Gitterle, M. W. Gee, W. A. Wall (2010). A dual mortar approach for 3D finite deformation contact with consistent linearization. *Int. J. Num. Meth. Engng.* 83(11)

C. Meier, W. A. Wall, A. Popp (2017). A unified approach for beam-to-beam contact. *Comput. Methods Appl. Mech. Engrg.*, 315



Fluid mechanics / Turbulence / Heat

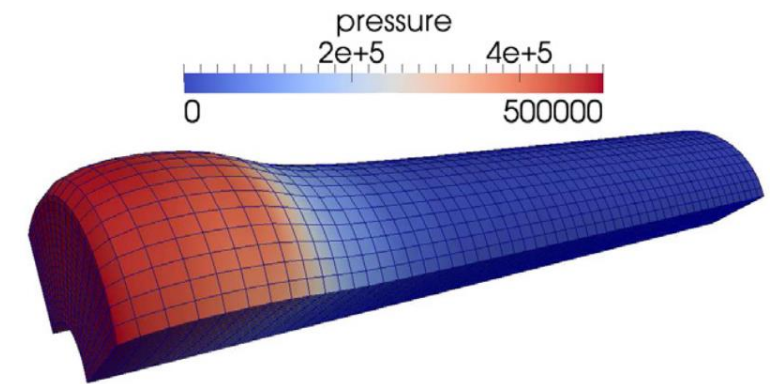
- 2D / 3D fluid flow solver based on finite elements
 - Incompressible Navier-Stokes equations
 - Weakly compressible variable-density low-Mach number flows
 - with Eulerian or Arbitrary Lagrangian-Eulerian (ALE) observer (mesh motion)
 - Monolithic solution approach
- Turbulence modeling via Large-eddy simulation (LES) with different turbulence models
 - Dynamic Smagorinsky
 - Dynamic Vreman
 - Multifractal Subgrid Scales
- Example: LES fluid flow through channel heated from the bottom



U. Rasthofer, G. C. Burton, W. A. Wall, V. Gravemeier (2010). An algebraic variational multiscale-multigrid-multifractal method (AVM4) for large-eddy simulation of turbulent variable-density flow at low Mach number. *Num. Methods in Fluids*, 76

Porous media

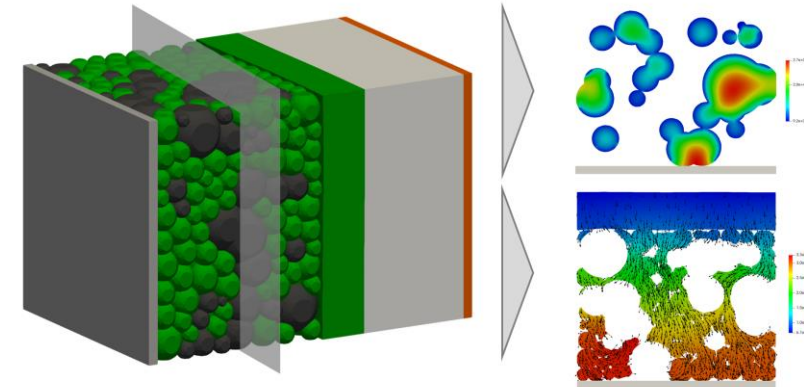
- Darcy & Darcy-Brinkmann flows
- Finite deformation formulation
- Monolithic solution approach
- Multiphase variant with several fluid phases
- Example:
 - Darcy–Brinkman flow through a quarter of a hollow cylinder
 - Pressure wave travels through the porous media



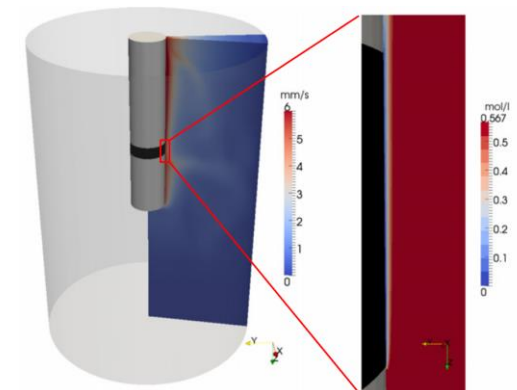
A.-T. Vuong, C. Ager, W.A. Wall (2016). Two finite element approaches for Darcy and Darcy–Brinkman flow through deformable porous media—Mixed method vs. NURBS based (isogeometric) continuity. *Comput. Methods Appl. Mech. Engrg.* 305

Electrochemistry

- Analysis of solid state batteries
- Interaction of solid mechanics, electrochemistry, and optionally temperature effects
- Inelastic expansion due to charge/temperature changes
- Contact mechanics at electrode-electrolyte interface (delamination phenomena)
- Concentration fields governed by convection-diffusion equation
- Example 1:
 - solid-state battery, structured from left to right with a current collector, a composite cathode, a solid electrolyte separator, a lithium metal anode, and another current collector.
 - Visualization of the lithium concentration distribution in the active material and the electric potential, including local flux visualization, within the composite cathode of a solid-state battery.
- Example 2:
 - Rotating cylinder electrode: interplay of natural convection with forced convection in the electrolyte due to the rotating cylindrical Copper cathode. (left: velocity field, right: Copper concentration boundary layer close to the cathode)



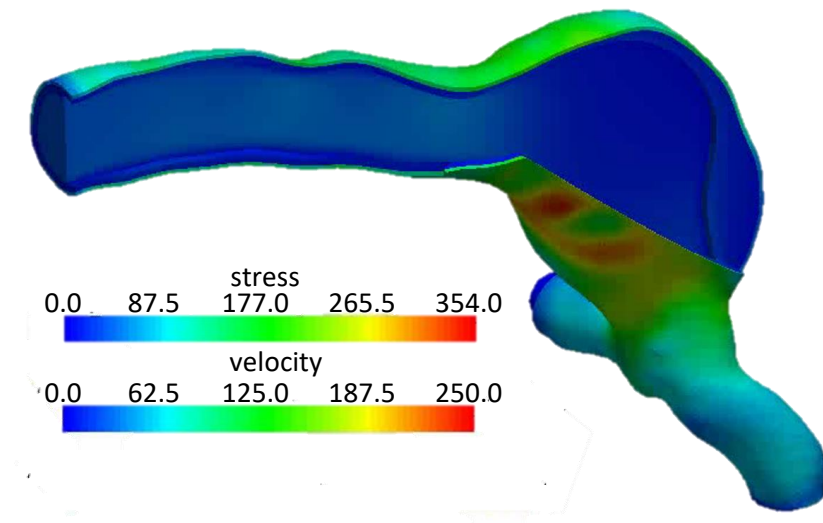
C.P. Schmidt, S. Sinzig, V. Gravemeier, W.A. Wall (2023). A three-dimensional finite element formulation coupling electrochemistry and solid mechanics on resolved microstructures of all-solid-state lithium-ion batteries. *Comput. Methods Appl. Mech. Engrg.*, 417



A. Ehrl, G. Bauer, V. Gravemeier, W.A. Wall (2013). A computational approach for the simulation of natural convection in electrochemical cells. *J. Comput. Phys.*, 235

Fluid-structure interaction (ALE)

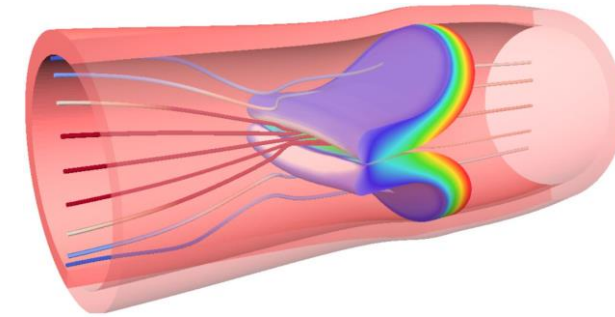
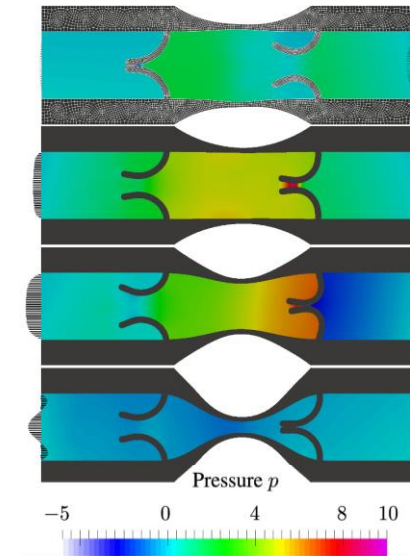
- Monolithic/partitioned (iterative) FSI (2D / 3D)
- **Moving mesh approach** for the fluid
(Arbitrary Lagrangian Eulerian formulation)
- Fluid described by incompressible Navier-Stokes equations
- Lagrangean description for solid (various material models)
- Example: Blood flow through abdominal aorta including aneurysm



A. Maier, et al (2010). A comparison of diameter, wall stress, and rupture potential index for abdominal aortic aneurysm rupture risk prediction. *Annals of biomedical engineering*, 38(10)

Fluid-structure interaction (CutFEM)

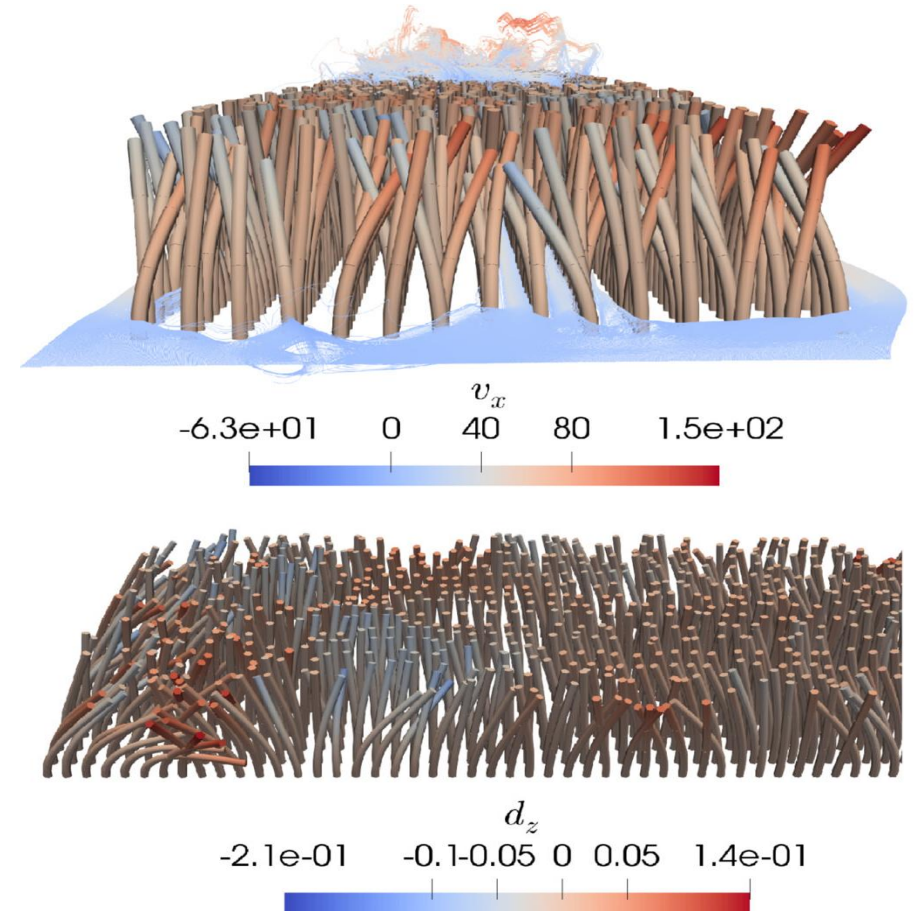
- Monolithic FSI (2D / 3D)
- **CutFEM approach** for the fluid (**fixed grid**)
- Fluid described by incompressible Navier-Stokes equations
- Lagrangean description for solid (various material models)
- Solids may collide
- Example: Fluid flow through double-leafed valve with poroelastic leaflets in an elastic tube
 - Color scale: velocity of fluid streamlines, pressure on leaflets' surface
 - Opening and closing of the valves includes contact



C. Ager, et al. (2021): A consistent and versatile computational approach for general fluid-structure-contact interaction problems. *Int. J. Num. Meth. Engng.*, 122(19)

Fluid-structure interaction (mixed-dimensional)

- Two-way partitioned mixed-dimensional coupling
- Fibers: Slender one-dimensional (1D) fibers based on geometrically exact nonlinear beam theory
- Flow: 3D incompressible Navier-Stokes equations
- Regularized mortar-type finite element discretization
- Example: submersed vegetation canopy

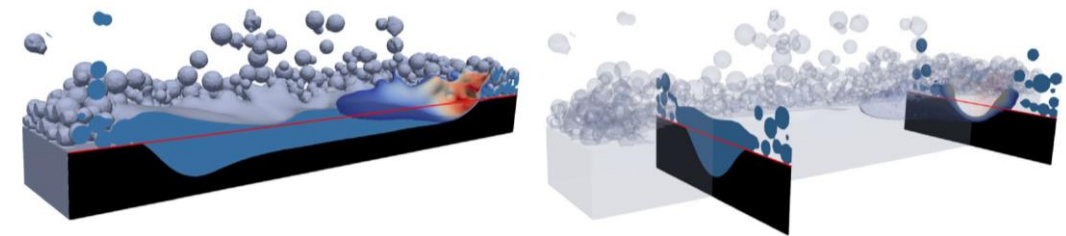
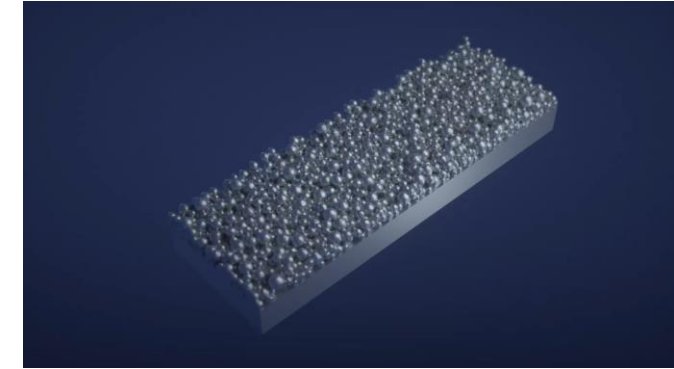


N. Hagmeyer, M. Mayr, A. Popp (2024).

A fully coupled regularized mortar-type finite element approach for embedding one-dimensional fibers into three-dimensional fluid flow. *Int. J. Num. Meth. Eng.*, 125

Fluid-structure interaction (SPH)

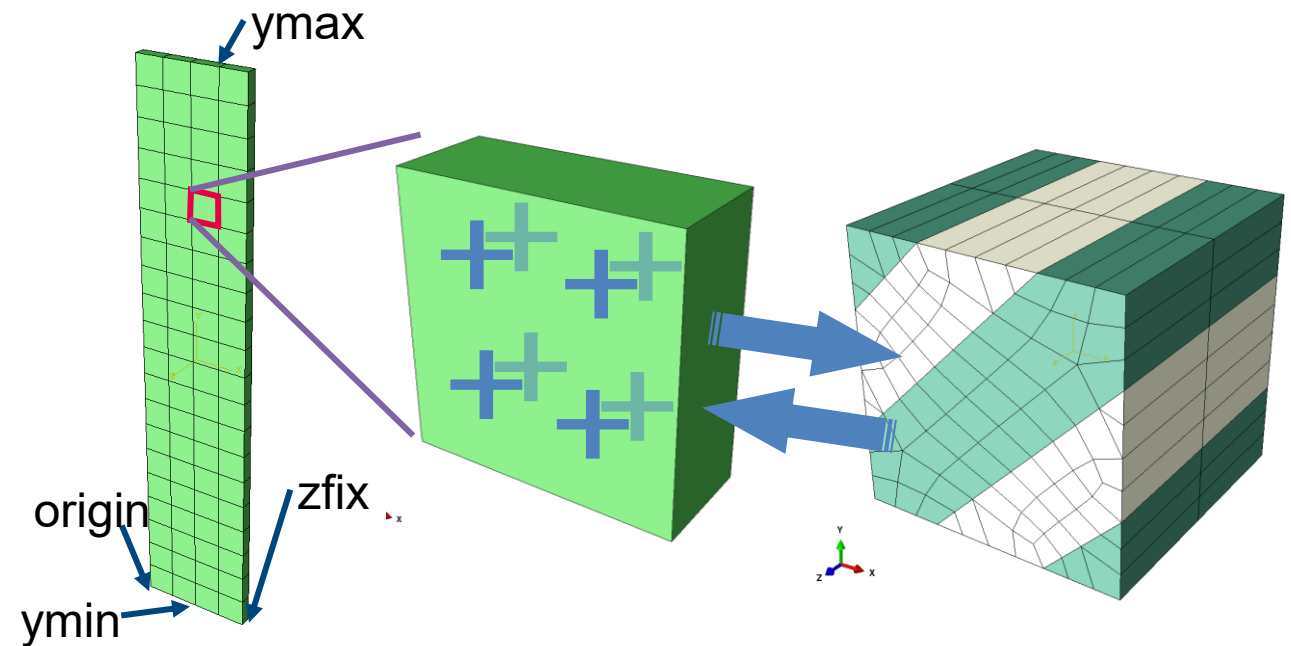
- Available interaction types
 - DEM (also in combination with elastic solids)
 - SPH
 - Peridynamics (WIP)
- Fluid-structure interaction
 - Solids
 - mobile but undeformable of arbitrary shape
 - Can collide
 - Fluid with different phases
 - Optionally thermal conduction
 - Optionally reversible phase changes
- Example: Highly dynamic particle simulation of powder bed fusion in additive manufacturing. Metal powder particles are heated using a laser beam moving along. Eventually, the particles melt and combine with the underlying substrate. (17e⁶ SPH particles, 5e⁵ time steps)



S.L. Fuchs, et al. (2022): A versatile SPH modeling framework for coupled microfluid-powder dynamics in additive manufacturing: binder jetting, material jetting, directed energy deposition and powder bed fusion. *Engineering with Computers* (<https://doi.org/10.1007/s00366-022-01724-4>)

Multiscale modeling

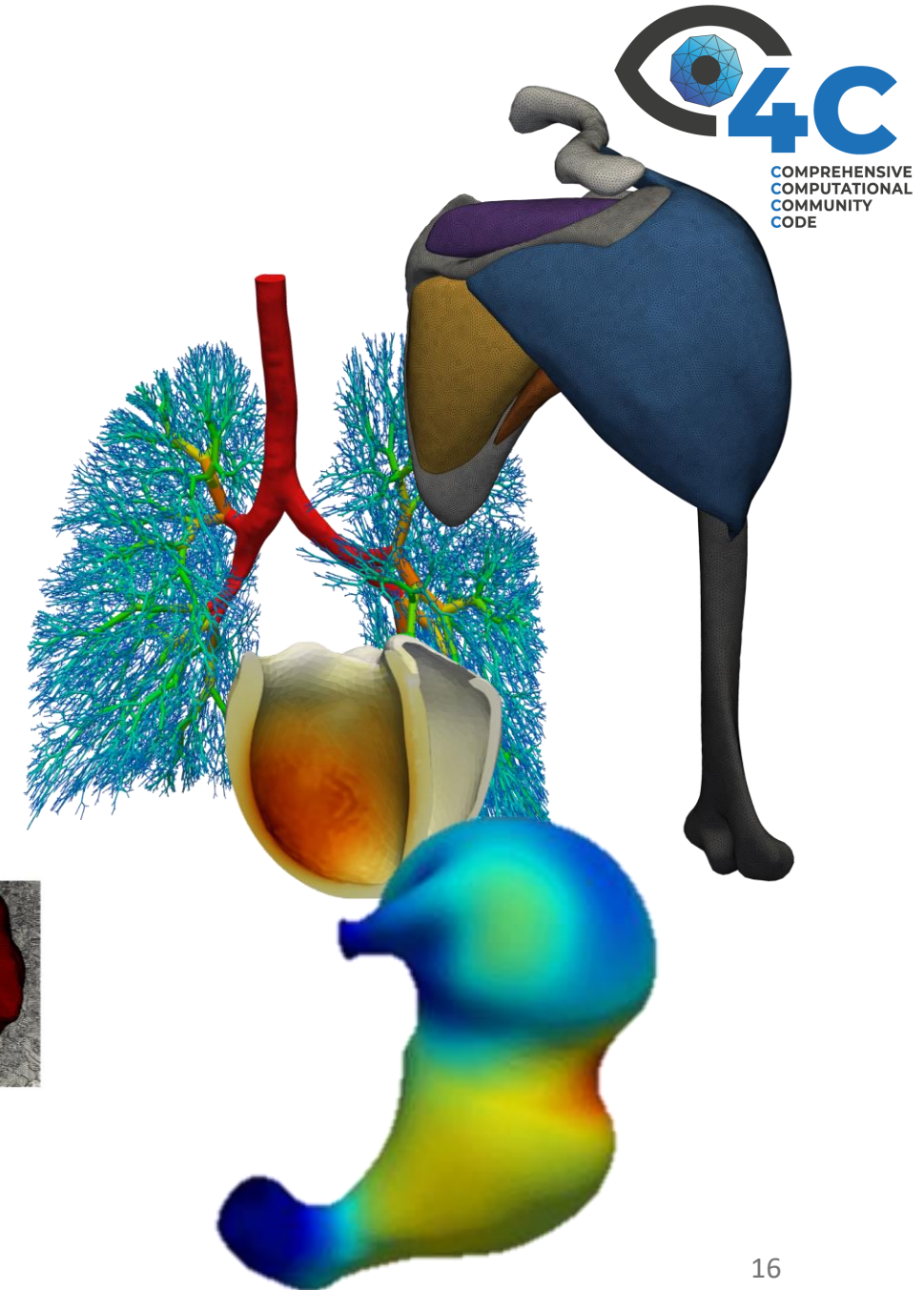
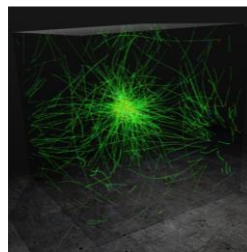
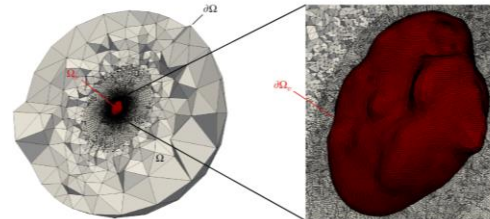
- FE² approach
- Microstructure
 - is calculated at each macro Gauss point
 - can be different in each macro element
 - can be run in parallel, with additional processes for the microstructure
- Simulation time increases exponentially
- Example
 - Tensile bar with laminar microstructure
 - Instead of 80 elements, the model comprises $80 \times 8 \times 176 = 112.640$ elements



Biomechanics applications

(<https://4c-multiphysics.org/publications/>)

- Lung modeling: Mixed dimensional modeling (0D, 1D, 3D)
 - Roth, C.J. Ismail, M., Yoshihara, L., Wall, W.A. (2016) Numer. Meth. Biomed. Eng. 33(1)
- Musculoskeletal systems
 - Engelhardt, L., Sachse, R., Burgkart, R., & Wall, W. A. (2025) Int. J. Num. Meth. Biomed Eng. 41(4)
- Heart
 - Gebauer, A., Pfaller, M., Bräu, F., Cyron, C., Wall, W. (2023) Biomech Model Mechanobiol 1-20
- Stomach
 - Henke, M. S., Brandstaeter, S., Fuchs, S. L., Aydin, R. C., Gizzi, A., Cyron, C. J. (2025): submitted for publication, preprint: arXiv:2509.02486
- Tumor / Cancer modeling
 - Kremheller, J., Brandstaeter, S., Schrefler, B.A., Wall, W.A. (2021) Int J Numer Method Biomed Eng
- Cellbiology
 - Müller, K.W., Cyron, C.J., Wall, Proc. Roy. Soc. A 471 (2182)



Introduction to the 4C Multiphysics Framework

- What is 4C Multiphysics?
- Capabilities of 4C Multiphysics
- **How to build 4C Multiphysics**
- The input file
- Tips and tricks to create a 4C input file

Build & run 4C (Docker Image)

- Docker image with pre-built 4C executable
https://github.com/4C-multiphysics/4C/tree/main/docker/prebuilt_4C
- Start the docker image
docker run --interactive --tty ghcr.io/4c-multiphysics/4c:latest
- Run simulation
 - Serial: *path/to/4C path/to/<input-file.4C.yaml> output-prefix*
 - Parallel: Prepend *mpirun -np <numProcs>*

Build & run 4C (manual build)

- Clone the repo: <https://github.com/4C-multiphysics/4C>
- Here is the installation:
<https://4c-multiphysics.github.io/4C/documentation/installation/installation.html>
- Install dependencies
- Create build folder, therein
 - *cmake --preset=<name-of-preset> <4C_source_dir>*
 - *ninja -j <numProcs>*

Build & run 4C (manual build)

- Dependencies
 - <https://github.com/4C-multiphysics/4C/blob/main/docker/dependencies/Dockerfile> (Ubuntu 24.04)
 - C++ compiler with C++20 compatibility
 - MPI / CMake / Ninja
 - TPLs (most important)
 - Trilinos
 - Suitesparse, SuperLUDist
 - BLAS, LAPACK
 - Metis, ParMETIS
 - HDF5
 - Several install scripts: <https://github.com/4C-multiphysics/4C/tree/main/dependencies/>
- For code development, integration with CLion and Visual Studio Code is documented, see <https://4c-multiphysics.github.io/4C/documentation/installation/installation.html#set-up-your-ide>

```
apt-get update && apt-get install -y \  
doxygen \  
graphviz \  
texinfo \  
lcov \  
libblas-dev \  
libboost-all-dev \  
libc++-dev \  
libhdf5-dev \  
libhdf5-openmpi-dev \  
libnetcdf-dev \  

```

Introduction to the 4C Multiphysics Framework

- What is 4C Multiphysics?
- Capabilities of 4C Multiphysics
- How to build 4C Multiphysics
- **The input file**
- Tips and tricks to create a 4C input file

4C input file

- ASCII file in yaml format **.4C.yaml*
- Further files can be referenced
 - Linear/nonlinear solver
 - Mesh
 - Include other yaml files
- Two parts are necessary in a 4C input file
 - Parameter/header block
 - Mesh block

4C input file (header block)

- PROBLEMTYPE → 40 different “applications”
- DYNAMIC section(s) → for each field and many interaction types
- IO → mostly VTK runtime output
- SOLVER
 - References to xml files
 - Templates
 - Iterative linear solvers
 - Preconditioners
 - https://github.com/4C-multiphysics/4C/blob/main/tests/input_files/xml
- MATERIALS
 - <https://4c-multiphysics.github.io/4C/documentation/materialreference.html>
 - https://4c-multiphysics.github.io/4C/documentation/analysis_guide/materials.html
(to be extended)

```
SOLVER 2:  
SOLVER: "Belos"  
AZPREC: "Teko"  
AZREUSE: 10  
SOLVER_XML_FILE: "xml/linear_solver/iterative_gmres_template.xml"  
TEKO_XML_FILE: "xml/block_preconditioner/thermo_solid.xml"
```

4C input file (header block)

- FUNCT
 - Functions of space and time (interpolation and symbolic math)
 - <https://4c-multiphysics.github.io/4C/documentation/furtherreference.html#functions-reference>
- Boundary conditions
 - Dirichlet bc → prescribe from scalar to FUNCT
 - Neumann bc → in reference and spatial configuration (“orthopressure”)
 - Periodic bc
 - Many more (e.g., coupling conditions, initial conditions, constraints, ...)
 - <https://4c-multiphysics.github.io/4C/documentation/conditionreference.html>

4C input file (mesh block)

- Variant 1
 - Reference mesh file containing node sets, node coordinates, and element connectivity
 - Supported file formats
 - *.e/*.exo (EXODUS II from Cubit)
 - *.vtu
 - *.msh (GMSH – to come soon)

```
STRUCTURE GEOMETRY:
  FILE: "/tmp/pw_m3.exo"
  SHOW_INFO: "summary"
  ELEMENT_BLOCKS:
    - ID: 1
      ELEMENT_NAME: SOLID
      ELEMENT_DATA: "MAT 2 KINEM nonlinear TECH eas_full"

FLUID GEOMETRY:
  FILE: "/tmp/pw_m3.exo"
  SHOW_INFO: "detailed"
  ELEMENT_BLOCKS:
    - ID: 2
      ELEMENT_NAME: FLUID
      ELEMENT_DATA: "MAT 1 NA ALE"
```

4C input file (mesh block)

- Variant 2
 - Node sets (DNODE/DLINE/DSURF/DVOL–NODE TOPOLOGY)
 - NODE COORDS
 - Unique numbering
 - ELEMENTS
 - Different element types within one file possible
 - Individual material per element
 - Several fields within one file possible for multiphysics, e.g.,
STRUCTURE ELEMENTS,
FLUID ELEMENTS,
TRANSPORT ELEMENTS,
...

```

- "NODE 5 DVOL 1"
- "NODE 7 DVOL 1"
- "NODE 4 DVOL 1"
NODE COORDS:
- "NODE 1 COORD 0.00236432494009"
- "NODE 2 COORD 0.09009273926518"
- "NODE 3 COORD -0.0711680774560"

```

2 styles possible

```

"DVOL–NODE TOPOLOGY": [
  "NODE 6 DVOL 1",
  "NODE 1 DVOL 1",
  "NODE 4 DVOL 1",
  "NODE 2 DVOL 1",
  "NODE 7 DVOL 1",
  "NODE 5 DVOL 1",
  "NODE 3 DVOL 1",
  "NODE 8 DVOL 1"
],
"NODE COORDS": [
  "NODE 1 COORD 0.0023643 0.0023643 0.0023643",
  "NODE 2 COORD 0.0900927 1.0900927 0.0900927",
  "NODE 3 COORD -0.0711680 -0.0711680 0.9288319",
  "NODE 4 COORD 0.0897298 1.0897298 1.0897298",
  "NODE 5 COORD 0.9623662 -0.0376337 -0.0376337",
  "NODE 6 COORD 0.9846652 0.9846652 -0.0153347",
  "NODE 7 COORD 1.0655405 0.0655405 1.0655405",
  "NODE 8 COORD 0.9818398 0.9818398 0.9818398"
],
"STRUCTURE ELEMENTS": [
  "1 SOLID HEX8 1 5 6 2 3 7 8 4 MAT 1 KINEM nonlinear TECH eas_full FIBER1 0.707106 0.707106 0.0"
]

```

4C input file (mesh block)

- Variant 3
 - Rectangular grid internally created
 - VOLUME / SIDE / EDGE / CORNER keywords for boundary conditions
 - Supported element types
 - HEX8 / HEX20 / HEX27
 - WEDGE6 / WEDGE15

```
DESIGN SURF NEUMANN CONDITIONS:
- E: 2
  NUMDOF: 3
  ONOFF: [1, 0, 0]
  VAL: [1, 0, 0]
  FUNCT: [0, 0, 0]
DNODE-NODE TOPOLOGY:
- "CORNER structure x- y- z- DNODE 1"
DSURF-NODE TOPOLOGY:
- "SIDE structure x- DSURFACE 1"
- "SIDE structure x+ DSURFACE 2"
STRUCTURE DOMAIN:
  bottom_corner_point: [0.0, 0.0, 0.0]
  top_corner_point: [4.0, 4.0, 4.0]
  subdivisions: [3, 3, 3]
  rotation_angle: [45.0, 0.0, 30.0]
  auto_partition: false
  elements:
    SOLID:
      HEX8:
        MAT: 1
        KINEM: linear
```

Introduction to the 4C Multiphysics Framework

- What is 4C Multiphysics?
- Capabilities of 4C Multiphysics
- How to build 4C Multiphysics
- The input file
- **Tips and tricks to create a 4C input file**

Input file creation

- Where to find helpful information
 - https://4c-multiphysics.github.io/4C/documentation/input_parameter_reference/parameterreference.html
 - Existing input files (>2000)
 - https://github.com/4C-multiphysics/4C/tree/main/tests/input_files
 - File names are historically grown (f2 → fluid 2-dimensional, elch → electrochemistry)
 - Search for relevant keyword/input parameter in docu
 - Search for files containing the keyword
 - Start with a valid input file

☐ Input Parameter Reference

- ☐ General
- ☐ Element reference
- ☐ Material reference
 - Cloning material reference
- ☐ Prescribed Condition Reference
 - Contact Constitutive Law Reference
 - Result description reference
 - Functions reference
- ☐ Header parameters

Input file creation

- Input file creation with IDE support (e.g. VS Code or CLion) using schema file
 - Auto completion (Ctrl + Space)
 - Only offers not yet set parameters
 - Shows available options and initializes with default value

```
REF_LENGTH: true
ELEMENT_GID: true
IO
  IO/RUNTIME VTK OUTPUT/STRUCTURE
  IO/EVERY ITERATION
  IO/MONITOR STRUCTURE DBC
  IO/RUNTIME VTK OUTPUT/FLUID
  IO/RUNTIME VTP OUTPUT STRUCTURE
  DESIGN NODE 1D ARTERY IN_OUTLET CONDITIONS
  BEAM INTERACTION/BEAM TO SOLID SURFACE/RUNTIME VTK OUTPUT BEAM INT
  BEAM INTERACTION
  FLUID BEAM INTERACTION/BEAM TO FLUID MESHTYING/RUNTIME VTK OUTPUT
  FLUID BEAM INTERACTION
  FLUID BEAM INTERACTION/BEAM TO FLUID MESHTYING
  BEAM INTERACTION/BEAM TO SOLID VOLUME MESHTYING/RUNTIME VTK OUTPUT
```

```
IO/RUNTIME VTK OUTPUT/STRUCTURE:
  STRESS_STRAIN: false
  STRUCTURAL DY
  INT_STRATEG
  DYNAMICTYPE
  true
  false
  Press Ctrl+. to choose the selected (or first) sug
```

Input file creation

- Reuse parts of files via inclusion

periodicCube-base.yaml

```
INCLUDES:
- periodicCube-head.yaml
- periodicCube-bc.yaml
- periodicCube-geo.yaml
```

periodicCube-head.yaml

```
PROBLEM TYPE:
  PROBLEMTYPE: "Structure"
IO:
  STRUCT_STRESS: "Cauchy"
  STRUCT_STRAIN: "GL"
IO/RUNTIME VTK OUTPUT:
  INTERVAL_STEPS: 1
IO/RUNTIME VTK OUTPUT/STRUCTURE:
  OUTPUT_STRUCTURE: true
  DISPLACEMENT: true
  STRESS STRAIN: true
```

periodicCube-bc.yaml

```
FUNCT1:
- COMPONENT: 0
  SYMBOLIC_FUNCTION_OF_SPACE_TIME: "pull"
- VARIABLE: 0
  NAME: "pull"
  TYPE: "linearinterpolation"
  NUMPOINTS: 2
  TIMES: [0,20]
  VALUES: [0,5]
DESIGN SURF DIRICH CONDITIONS:
- E: 1
  NUMDOF: 3
  ONOFF: [1,0,0]
  VAL: [0,0,0]
```

periodicCube-geo.yaml

```
- NODE 238 COORD 3.5857860 5.0000000 5.0000000
- NODE 239 COORD -3.5857860 5.0000000 1.6666670
- NODE 240 COORD -3.5857860 5.0000000 -1.6666670
STRUCTURE ELEMENTS:
- 1 SOLID HEX8 70 69 239 237 4 3 6 7 MAT 1 KINEM linear
- 2 SOLID HEX8 72 71 240 236 70 69 239 237 MAT 1 KINEM linear
- 3 SOLID HEX8 65 66 238 235 72 71 240 236 MAT 1 KINEM linear
- 4 SOLID HEX8 180 152 160 190 37 25 31 43 MAT 1 KINEM linear
- 5 SOLID HEX8 176 174 180 190 36 35 37 43 MAT 1 KINEM linear
- 6 SOLID HEX8 96 82 176 190 12 8 36 43 MAT 1 KINEM linear
- 7 SOLID HEX8 189 95 96 190 45 13 12 43 MAT 1 KINEM linear
```

Input file creation

- https://4c-multiphysics.github.io/4C/documentation/analysis_guide/preprocessing.html
- Several ways to create mesh files
 - Coreform Cubit → *.e (EXODUS II) format
 - <https://pypi.org/project/meshio/> enables mesh conversion
 - Supports widely-used mesh formats (proprietary & open) as input
 - Convert file into *.vtu (or *.msh) format for usage within 4C
 - Please stay tuned, we are currently working on a reliable and versatile solution based on meshio

Growing ecosystem around 4C Multiphysics

- Tools: <https://4c-multiphysics.github.io/4C/documentation/tools/tools.html#tools>
 - QUEENS (<https://github.com/queens-py/queens>)
 - Parameter studies
 - Sensitivity analysis
 - Uncertainty quantification
 - Parameter identification
 - Inverse analysis
 - FourCIPP (<https://github.com/4C-multiphysics/fourcipp>)
 - Python parser to interact with 4C input files
 - 4C-Webviewer (<https://github.com/4C-multiphysics/4C-webviewer>)
 - Visualize and edit 4C input
 - Several input generator scripts (BeamMe, cubitpy, ...)

4C Multiphysics summary

- 4C solves challenging real-world applications
- Extensible modular C++ software structure
- Freely available under LPGL-3.0-or-later
- <https://github.com/4C-multiphysics/4C>
- Join the community 😊
(photo from our this year's workshop July 29th – 31st, 2025)



4C Multiphysics @UKACM&GACM Autumn School 2025



Hands-on-sessions

- 9:30 – 11:00 (1) Solid mechanics with plasticity
- 11:15 – 12:45 (2) Contact mechanics
- 11:15 – 12:45 (3) Fluid-solid interaction
- 15:30 – 16:00 State-of-the-art research with 4C (Overview)

Tomorrow

- QUEENS
- QUEENS & 4C