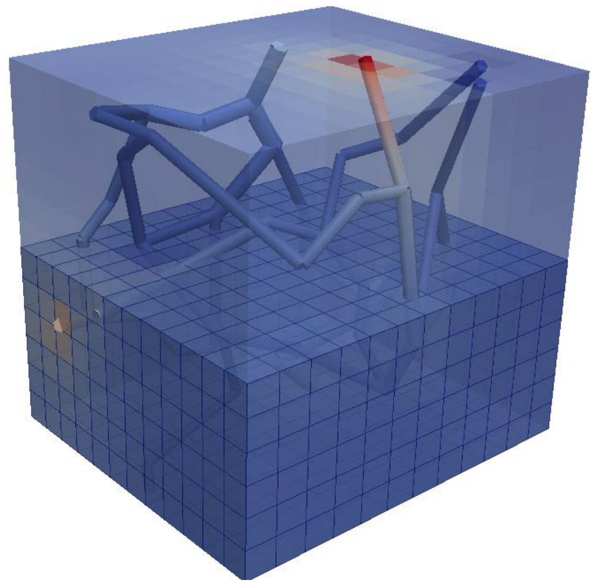
dune-foamgrid

A new implementation of the dune grid interface



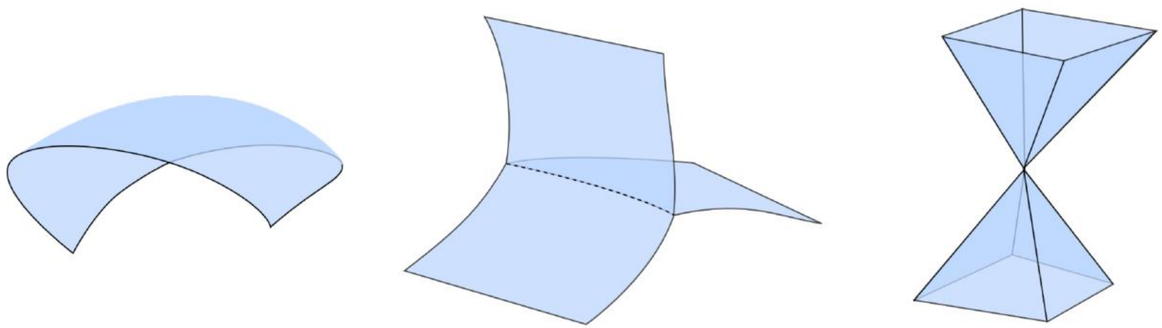
Dune User Meeting 2015
Heidelberg

University of Stuttgart
Germany

Why another implementation?

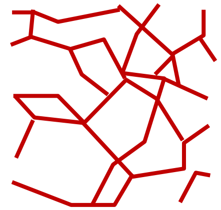
- Grid manager suited for surface grids (low-dimensional grids embedded in higher dimensional grids)

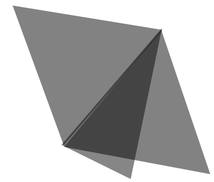
- Non-manifold grids



- Flexible structure
  - disconnected patches
  - Run-time alterations of the grid

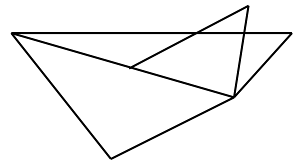
New features

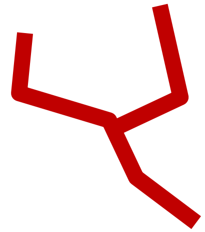




- Branching structures
  - More than two element share a facet
- Element parametrizations
- Run-time growth
  - Adding and removing elements at run-time

Branching points

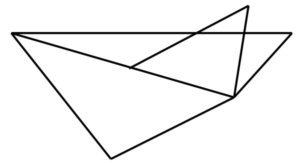


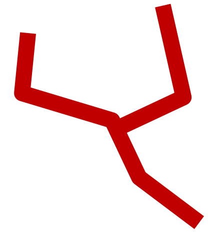


- Intersections at branching points?
  - Multiple intersections with the same `geometryInInside()` …
  - … but different `outside()`-elements
- Flux approximation on facets may require two intersection loops
  - e.g. $f = f(t_1, t_2, t_3)$

→ No simple access during iteration if two intersections share the same `geometryInInside()`

Branching points



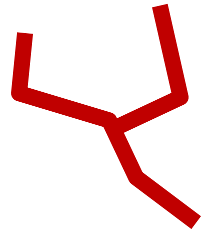


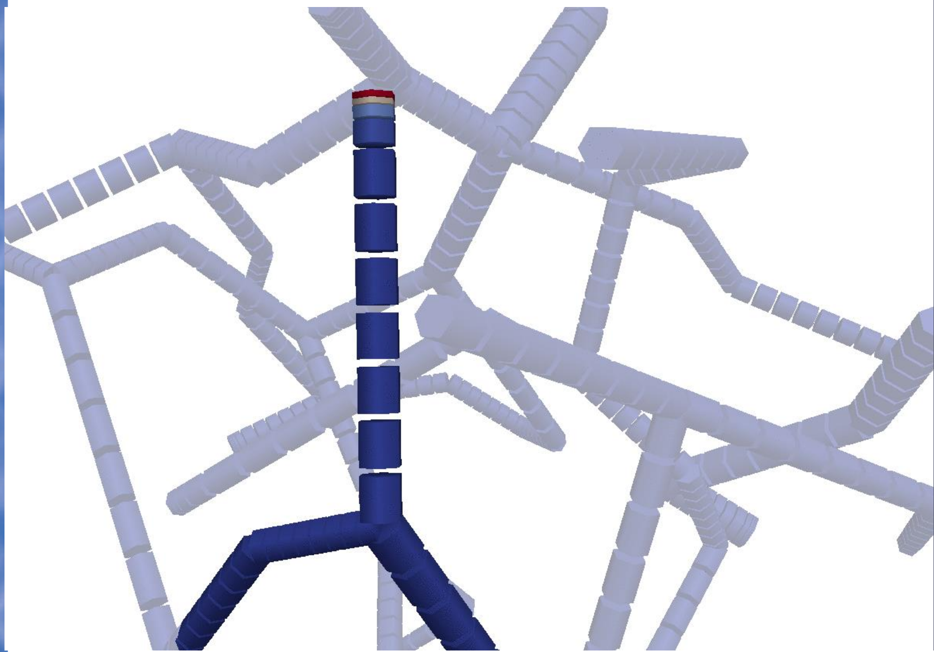
C++ code

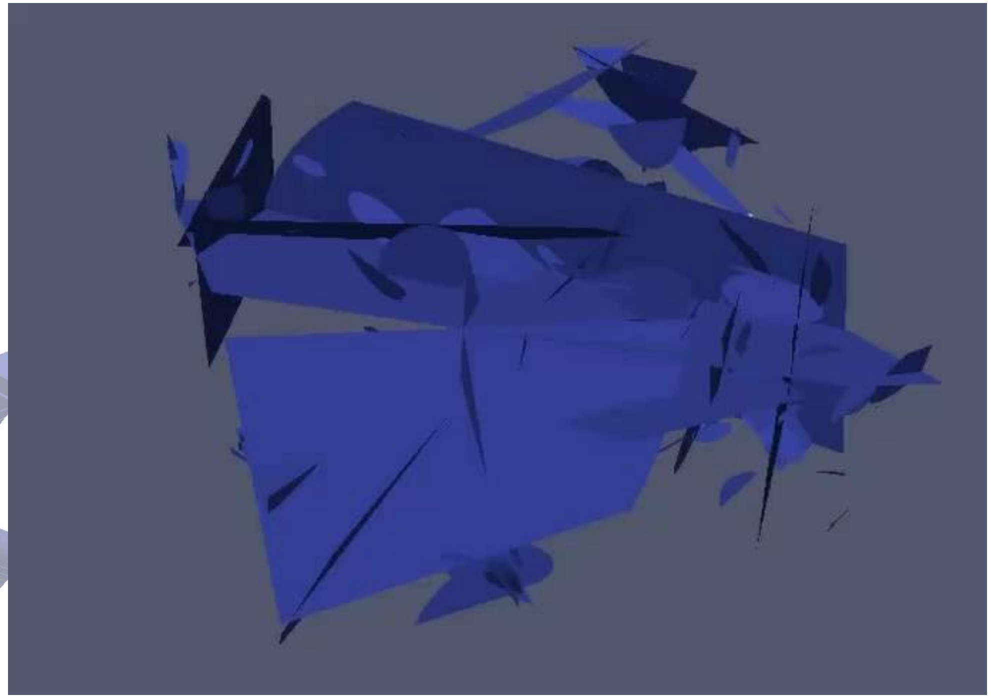
```
std::size_t neighbor() const
```

- Intersections with identical `geometryInInside()` will appear consecutively

Branching points



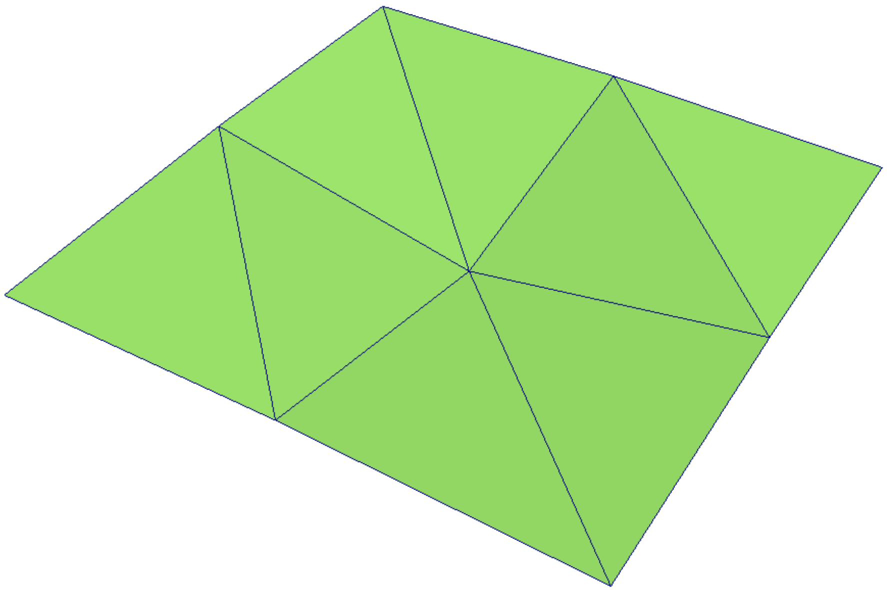


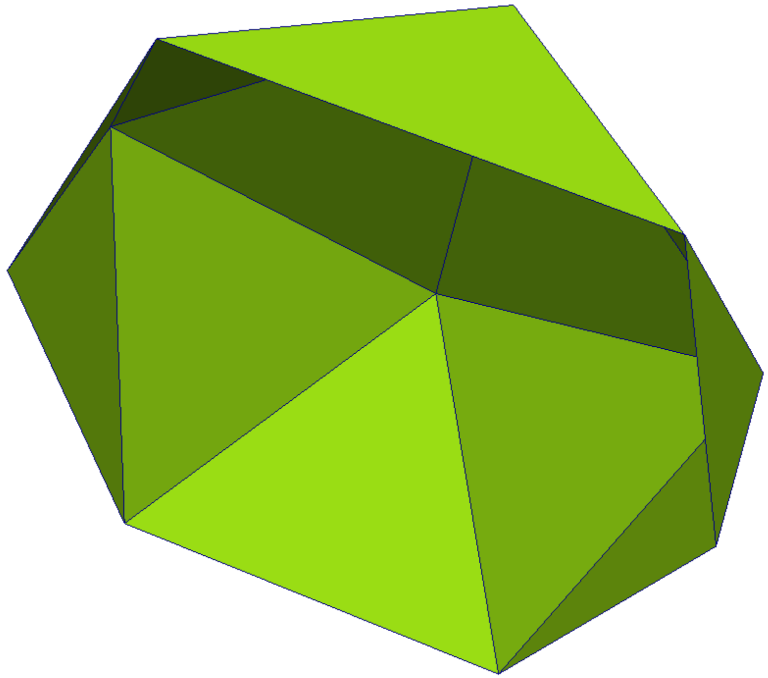


Code from Berninger et al [2011]

Element parametrizations

- Analogously to boundary parametrizations the parametrized shape is approximated better and better with refinement





Element parametrizations interface

$$\varphi_T : T_{\text{ref}} \rightarrow \mathbb{R}^w$$

C++ code

```cpp
VirtualFunction<FieldVector<double,dim>,
                FieldVector<double,dimworld> >;
```

C++ code

```cpp
virtual void evaluate(const FieldVector<double,dim>& x,
                      FieldVector<double,dimworld>& y) const = 0;
```

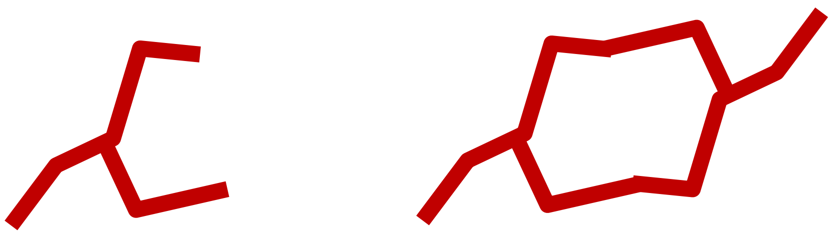
C++ code

```cpp
void insertElement(const GeometryType& type,
                   const std::vector<unsigned int>& vertices,
                   const std::shared_ptr<VirtualFunction<
                                         FieldVector<ctype,dim>,
                                         FieldVector<ctype,dimworld> > >&
                                         elementParametrization);
```

Growth

- Natural structures adapt their geometry
  - Root growth with plants
  - Growth of bacterial colonies
  - Angiogenesis (Blood capillary growth)
  - Fracture propagation

- Growth includes branching and merging



Growth

Challenges:

- Growth and adaptivity want to be combined (e.g. refinement around possible growth point)

  → Keep hierarchic structure intact

- Like for adaptivity: Possibility of user data transfer
  - Element removal creates new boundaries
  - New elements need e.g. data extrapolation

Growth interface

- Inspired by grid factory and adaptivity

C++ code

```
std::size_t insertVertex(const FieldVector<double,dimworld>& x);
```

C++ code

```
void insertElement(const GeometryType& type,
                   const std::vector<std::size_t>& vertices);
```

C++ code

```
void insertElement(const GeometryType& type,
                   const std::vector<std::size_t>& vertices,
                   const std::shared_ptr<VirtualFunction<
                                          FieldVector<ctype,dim>,
                                          FieldVector<ctype,dimworld> > >&
                                          elementParametrization);
```

C++ code

```
void removeElement(const Codim<0>::Entity& element);
```

Growth interface

- Inspired by grid factory and adaptivity

C++ code

```cpp
bool elementsInserted = grid->grow(); // true if at least one element was inserted
```

C++ code

```cpp
bool isNew = element.isNew();    // true if the element was created by growth
grid->postGrow(); // delete markers
```

Growth

- A simple random-based root network generator

courtesy of Natalie Schröder

Where to get?

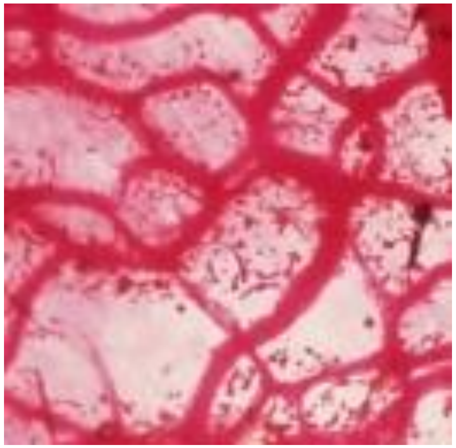
- git clone https://users.dune-project.org/repositories/projects/dune-foamgrid.git
- http://www.dune-project.org/modules/dune-foamgrid/

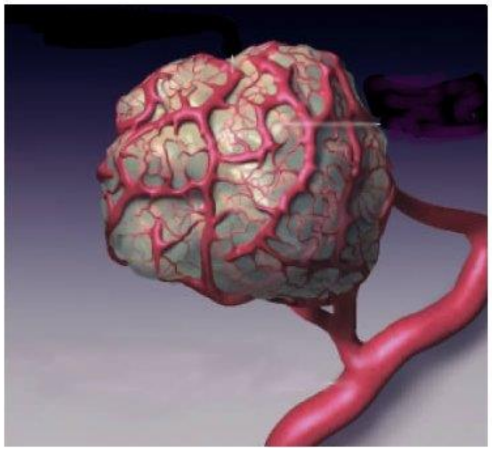
Drawbacks / TODOs

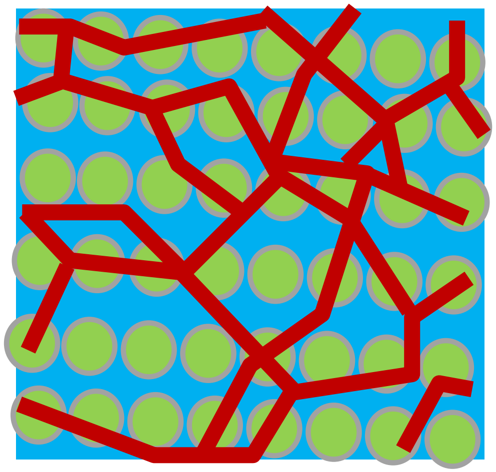
- Only 1d and 2d in nd and simplices
- Only Non-conforming refinement
- Only sequential

→ An application
Coupling flow and transport processes in capillaries and the embedding tissue

Tumor vasculature



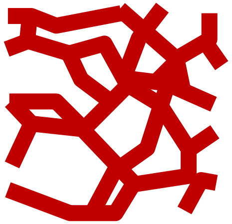




Cells, fibres,..

Interstitial fluid

A 1D-3D coupled flow problem



$$\frac{\partial}{\partial z}\left(\frac{\pi R^4}{2\mu_B(2+\gamma)}\frac{\partial p_v}{\partial z}\right) = \boxed{2\pi R L_p(p_v - \bar{p}_t)} \quad \text{in } \Lambda, \qquad \text{1D}$$

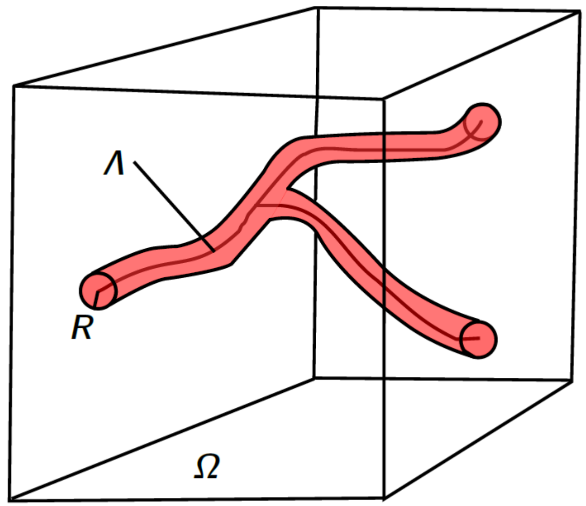
Starling‘s law

$$\nabla \cdot \left(\frac{K}{\mu_l}\nabla p_t\right) = \boxed{2\pi R L_p(\bar{p}_t - p_v)\delta_\Lambda} \quad \text{in } \Omega, \qquad \text{3D}$$

Definitions:

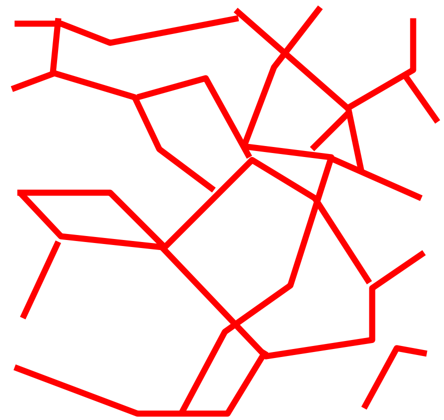
$$\bar{p}_t = \frac{1}{2\pi R}\int_0^{2\pi} p_t\Big|_R R\,d\theta$$

$$\int_\Omega f\delta_\Lambda \, dV = \int_\Lambda f \, ds$$



See also: Cattaneo, L., & Zunino, P. (2014). Computational models for fluid exchange between microcirculation and tissue interstitium. *Netw. Heterog. Media*, *9*(1), 135-159.

A 1D-3D coupled flow problem

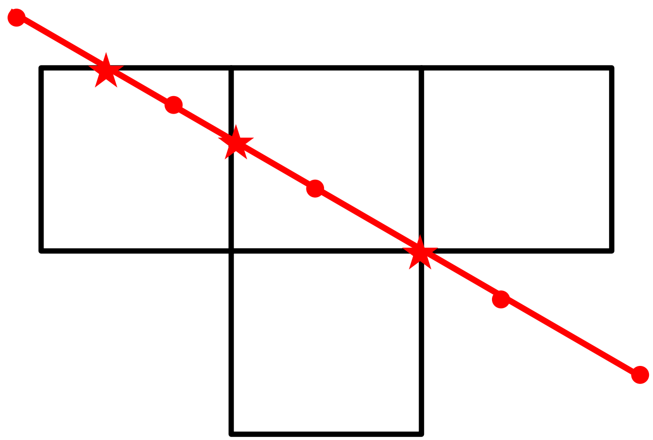


dune-foamgrid

- For the one-dimensional network grid embedded in three-dimensional space

dune-grid-glue (C. Engwer, O. Sander,…)

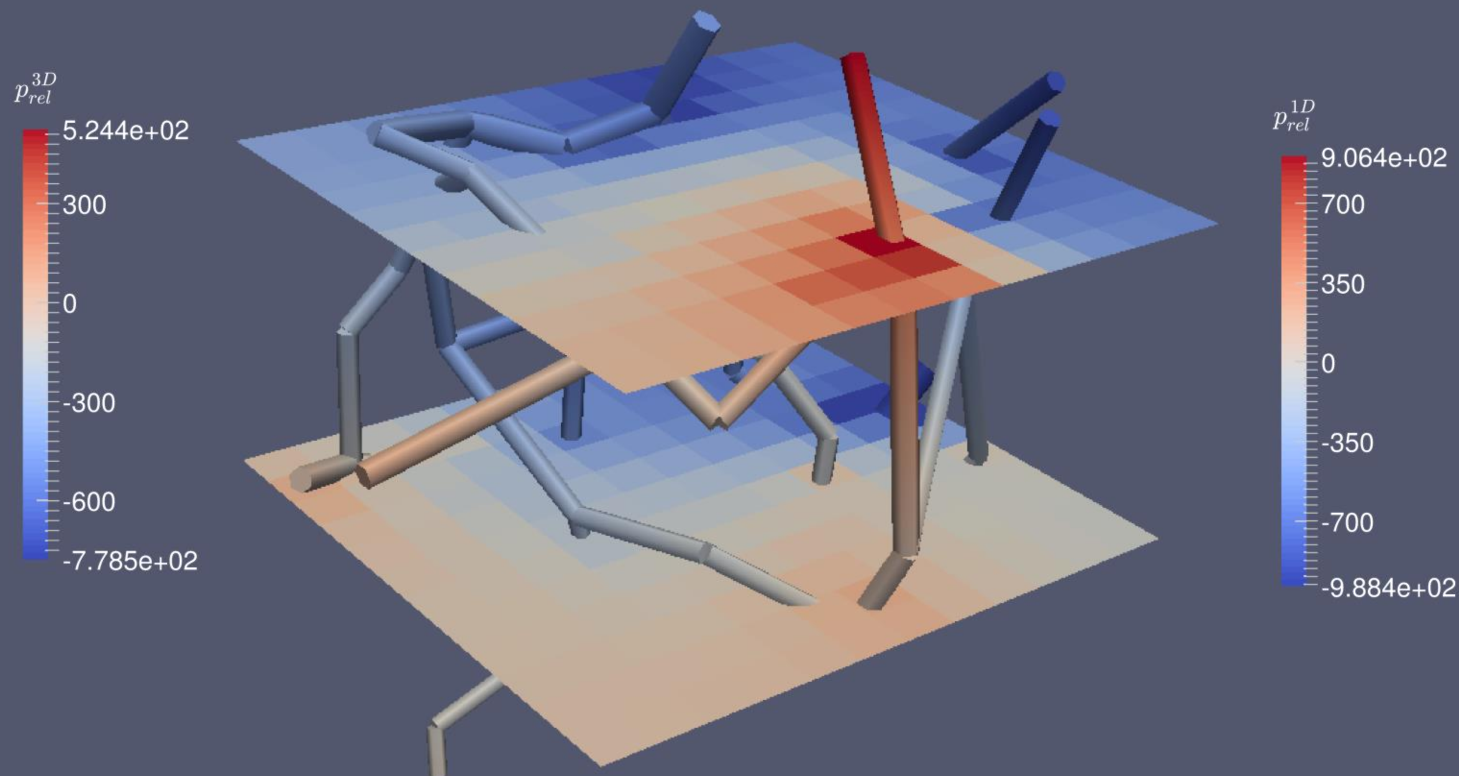
- To calculate intersections between network and three-dimensional bulk grid

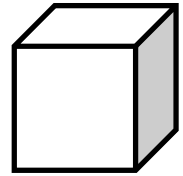


Simulation results

vessel geometry from Secomb, T.W. et al. (2000)

Tumor tissue: Lp = 10^{-10} m/Pas





140 x 160 x 150µm

References

dune-foamgrid

- Sander, O., Koch, T., Schröder, N., Flemisch, B. (2015). Dune-foamgrid a new implementation of the dune grid interface. (in preparation)
- H. Berninger, R. Kornhuber, and O. Sander. Fast and robust numerical solution of the Richardsequation in homogeneous soil. SINUM, 49(6):2576–2597, 2011.

capillary/tissue coupling:

- Cattaneo, L., & Zunino, P. (2014). Computational models for fluid exchange between microcirculation and tissue interstitium. *Netw. Heterog. Media*, *9*(1), 135-159.
- Motti, E.D.F., Imhof, H.G., Yasargil, M.G. (1986) The terminal vascular bed in the superficial cortex of the rat. An SEM study of corrosion casts. J. Neurosurg. 65: 834-846
- Secomb, T.W., Hsu, R., Beamer, N.B. and Coull, B.M. Theoretical simulation of oxygen transport to brain by networks of microvessels: effects of oxygen supply and demand on tissue hypoxia. Microcirculation 7, 237-247 (2000)
- Flemisch, B., Darcis, M., Erbertseder, K., Faigle, B., Lauser, A., Mosthaf, K., Müthing, S., Nuske, P., Tatomir, A., Wolff, M. and R. Helmig (2011): DuMu^x: DUNE for Multi-{Phase, Component, Scale, Physics, ...} Flow and Transport in Porous Media. Advances in Water Resources 34(9): 1102-1112.
- Formaggia, L., Lamponi, D., & Quarteroni, A. (2003). One-dimensional models for blood flow in arteries. *Journal of engineering mathematics*, *47*(3-4), 251-276.
- L.T. Baxter and R.K. Jain (1989). Transport of fluid and macromolecules in tumors. i. role of interstitial pressure and convection. *Microvascular Research*, 37(1):77–104. (…and rest of the series)