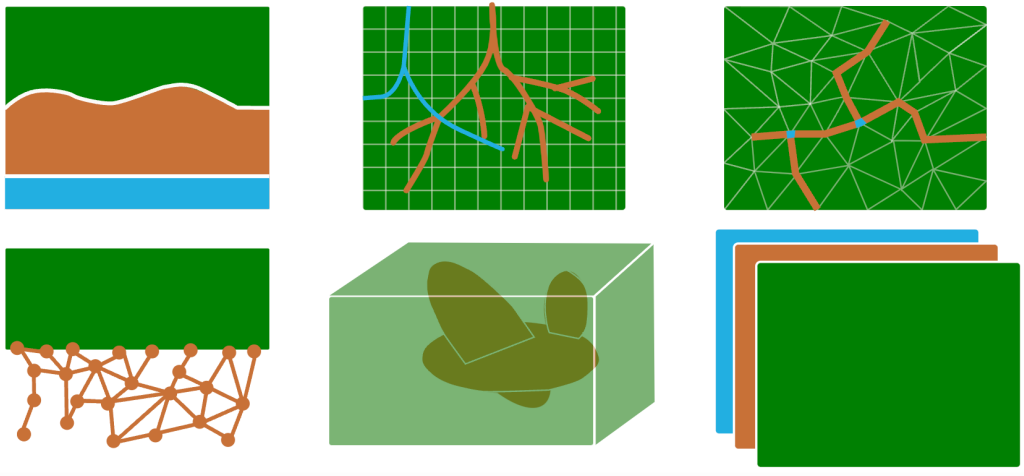
UiO : Department of Mathematics
University of Oslo



DuMu^x

https://dumux.org/

https://timokoch.github.io/

Finite volume schemes and coupled problems in DuMu^x

Timo Koch

DuMu^x is a simulation framework with a focus on **finite volume discretization methods**, **model coupling for multi-physics applications, and flow and transport applications in porous media**.

DuMu^x is based on the DUNE framework from which it uses the versatile grid interface, vector and matrix types, geometry and local basis functions, and linear solvers. DuMu^x then provides

- **Finite volume discretizations** (Tpfa, Mpfa, Staggered) and control-volume finite element discretization schemes
- A (thread-parallel) system matrix assembler (coloring) and approximation of the Jacobian matrix by numeric differentiation
- A customizable Newton method implementation including line search and various stopping criteria
- Many pre-implemented models (Darcy-scale porous media flow, Navier-Stokes, Geomechanics, Pore network models, Shallow water equations) and constitutive models
- A **multi-domain framework** for model coupling suited to couple **subproblems with different discretizations/domains/physics/dimensions**/... and create **monolithic solvers**

DuMu^x - DUNE for Multi-{Phase, Component, Scale, Physics, ...} flow and transport in porous media

An open-source simulator and research code in modern C++

https://dumux.org/

Finite volumes

Vertex-centered, face-centered, cell-centered, …

Control volume finite element schemes

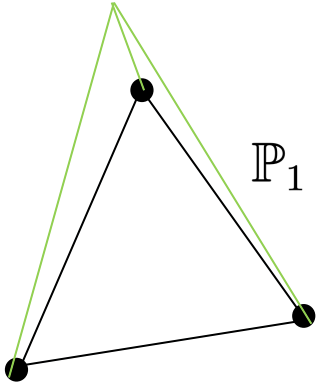
Combine FE functions and control volumes

- **Local mass conservation by construction**
- **Unstructured meshes**

Finite volumes / Control volume finite element schemes

Recipe

Nodal basis function (some FE space)



Construct control volumes “around” nodal degrees of freedom

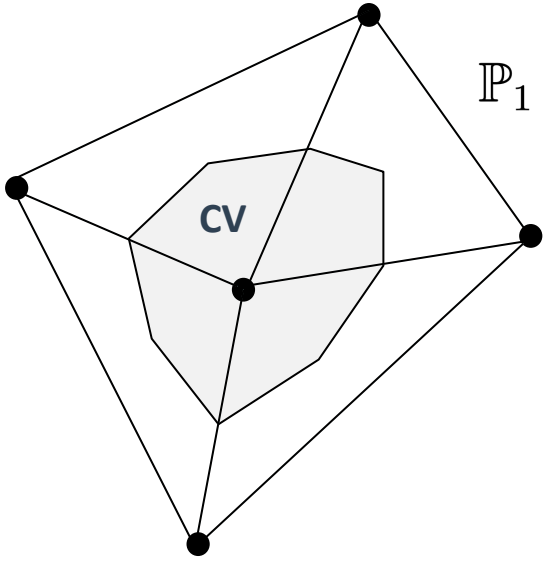
Element-wise assembly

Finite volumes / Control volume finite element schemes

Recipe

Nodal basis function (some FE space)

Construct control volumes “around” nodal degrees of freedom



Element-wise assembly

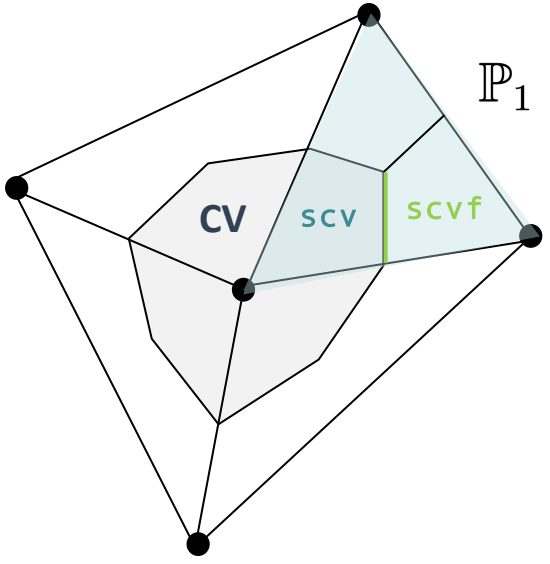
Finite volumes / Control volume finite element schemes

Recipe

Nodal basis function (some FE space)

Construct control volumes “around” nodal degrees of freedom

Element-wise assembly



SCVs are associated with dofs

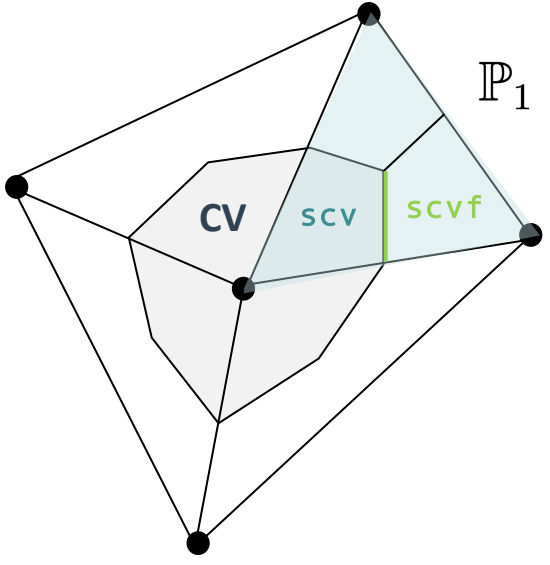
Finite volumes / Control volume finite element schemes

Interpretation as Petrov Galerkin FEM

control volumes $K \in \widetilde{\mathcal{T}}$

$$C_K(\mathbf{x}) := \begin{cases} c_K \in \mathbb{P}^0(K) & \mathbf{x} \in K, \\ 0 & \mathbf{x} \notin K, \end{cases}$$

$$B_h(\widetilde{\mathcal{T}}) = \left\{ q_h \in L^2(\widetilde{\Omega}) : \; q_h = \sum_{K \in \widetilde{\mathcal{T}}} C_K \right\}$$

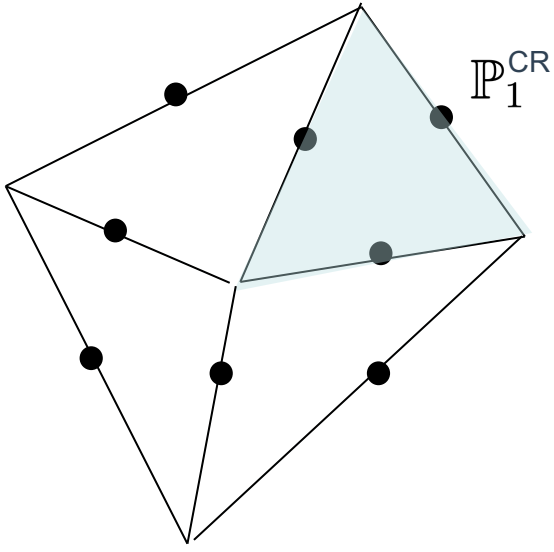


$$\int_\Omega r q_h \, dx = 0 \quad \forall q_h \in B_h(\widetilde{\mathcal{T}}) \quad \Longleftrightarrow \quad \int_K r \, dx = 0 \quad \forall K \in \widetilde{\mathcal{T}}$$

Finite volumes / Control volume finite element schemes

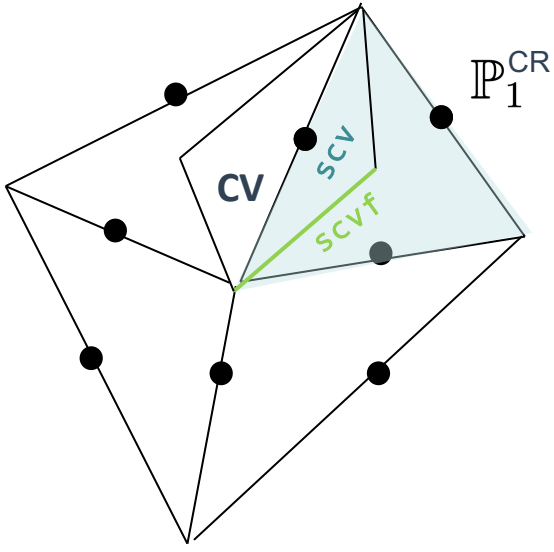
Other trial function spaces

CV
?



Finite volumes / Control volume finite element schemes

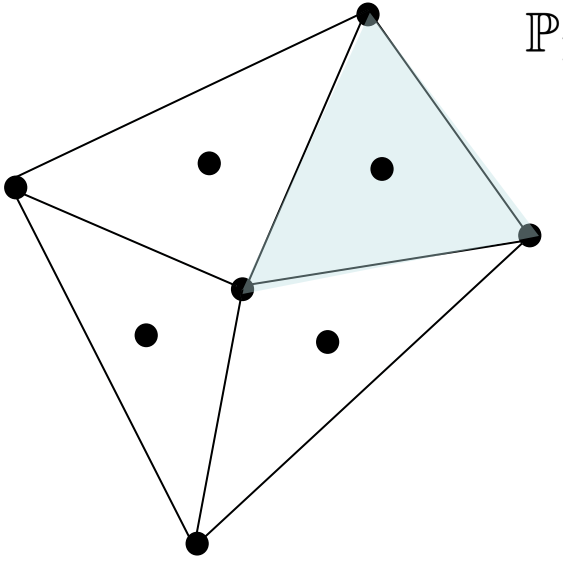
Other trial function spaces



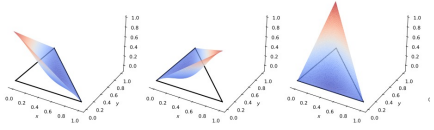
Finite volumes / Control volume finite element schemes

Other trial function spaces

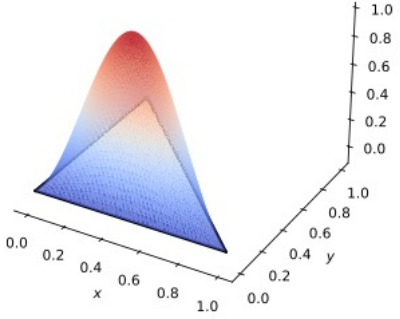
CV
?



$\mathbb{P}_1^{+B}$



$\mathbb{P}_1^{(-B)}$



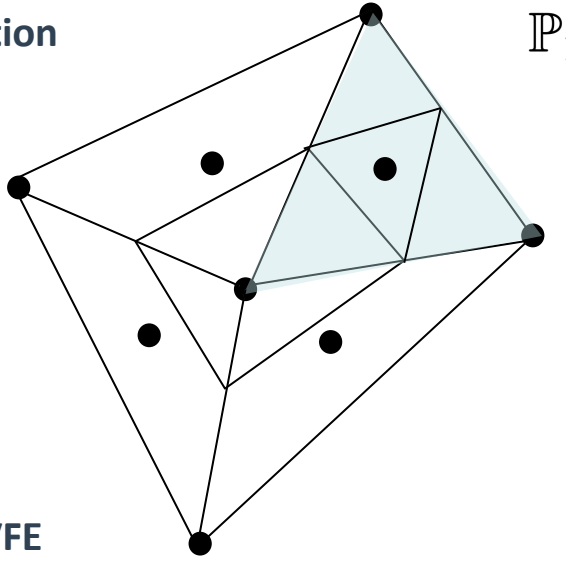
B

cubic bubble functions

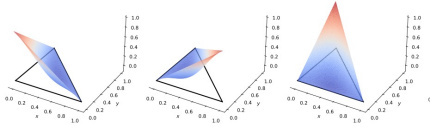
Finite volumes / Control volume finite element schemes

Other trial function spaces

partition

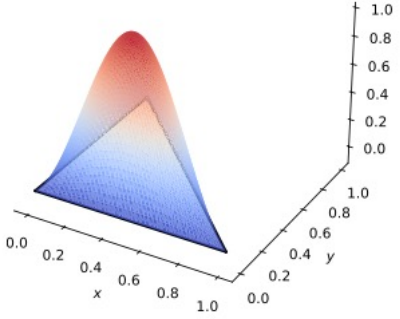


$\mathbb{P}_1^{+B}$



$\mathbb{P}_1^{(-B)}$

3D?
Faces within elements?



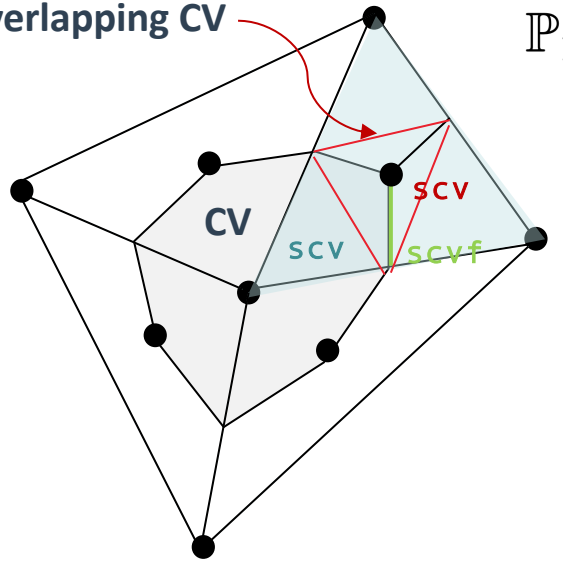
B

Non-overlapping CVFE

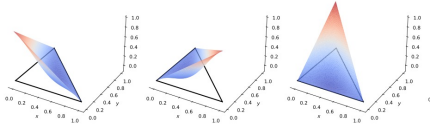
cubic bubble functions

Finite volumes / Control volume finite element schemes

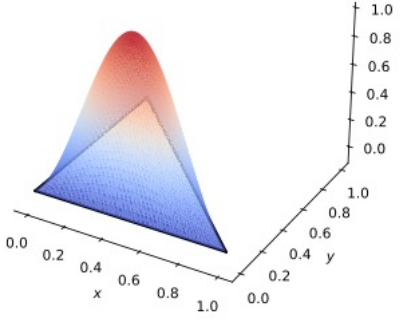
Other trial function spaces



$\mathbb{P}_1^{+B}$



$\mathbb{P}_1^{(-B)}$



B

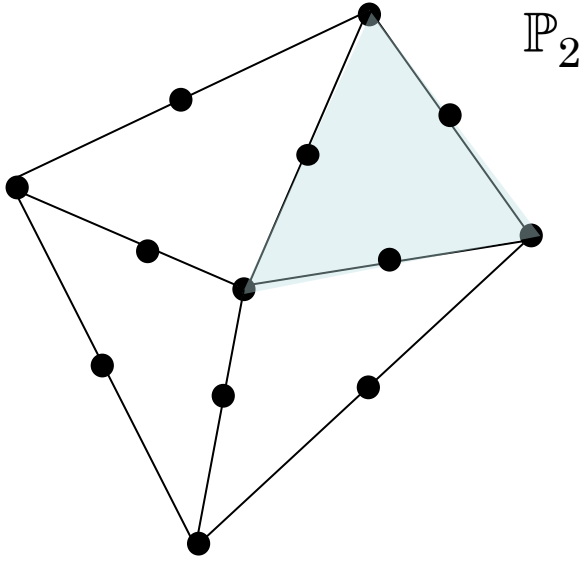
cubic bubble functions

Overlapping CVFE

Finite volumes / Control volume finite element schemes

Other trial function spaces

CV
?



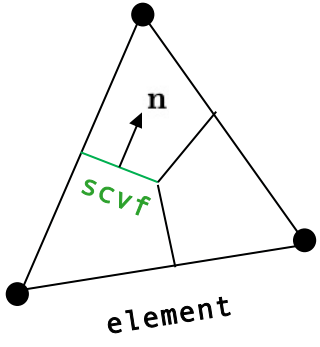
```cpp
template<class Context> NumEqVector stokesMomentumFlux(const Context& context) const
{
    const auto& element = context.element();
    const auto& fvGeometry = context.fvGeometry();
    const auto& elemVolVars = context.elemVolVars();
    const auto& scvf = context.scvFace();
    const auto& fluxVarCache = context.elemFluxVarsCache()[scvf];

    // interpolate velocity gradient at scvf
    Tensor gradV(0.0);
    for (const auto& scv : scvs(fvGeometry))
    {
        const auto& volVars = elemVolVars[scv];
        for (int dir = 0; dir < dim; ++dir)
            gradV[dir].axpy(volVars.velocity(dir), fluxVarCache.gradN(scv.indexInElement()));
    }

    const auto mu = context.problem().effectiveViscosity(element, fvGeometry, scvf);
    const auto pressure = context.problem().pressure(element, fvGeometry, scvf);

    // compute -(mu*(∇v + ∇v^T) - p)*n*dA
    NumEqVector flux = mv(gradV + transpose(gradV), scvf.unitOuterNormal());
    flux *= -mu * Extrusion::area(fvGeometry, scvf);
    flux.axpy(pressure * Extrusion::area(fvGeometry, scvf), scvf.unitOuterNormal());

    return flux;
}
```

Stokes equations

$$\int_{\partial K} [-2\mu \boldsymbol{D}(\boldsymbol{v}_h) + p_h \boldsymbol{I}] \cdot \mathbf{n} \, \mathrm{d}A = \int_K \boldsymbol{f} \, \mathrm{d}x, \quad \forall K \in \mathcal{T}^v,$$

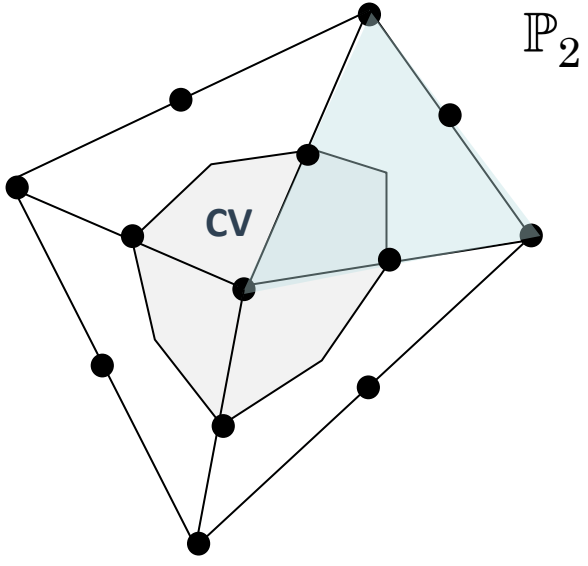
$$\int_{\partial K} \boldsymbol{v}_h \cdot \mathbf{n} \, \mathrm{d}A = \int_K q \, \mathrm{d}x, \quad \forall K \in \mathcal{T}^p,$$

$$\boldsymbol{D}(\boldsymbol{v}) = \frac{1}{2}\left(\nabla \boldsymbol{v} + \nabla^T \boldsymbol{v}\right)$$



Finite volumes / Control volume finite element schemes

Other trial function spaces



Hybrid CVFE

test function space

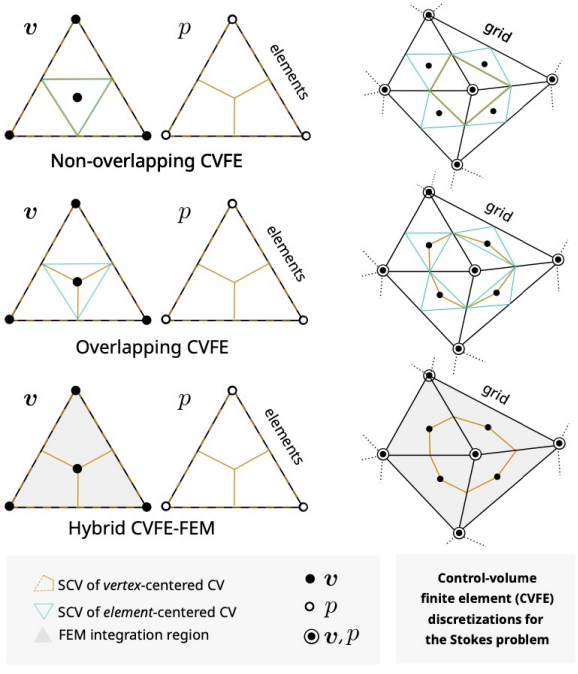
hierachical split

$$\mathbb{P}_2 = \mathbb{P}_1 + \text{rest}$$

CVFE FEM

Local conservation

Finite volumes / Control volume finite element schemes



Stable and locally mass- and momentum-conservative control-volume finite-element schemes for the Stokes problem

Martin Schneider[a], Timo Koch[b]

https://arxiv.org/abs/2309.00321

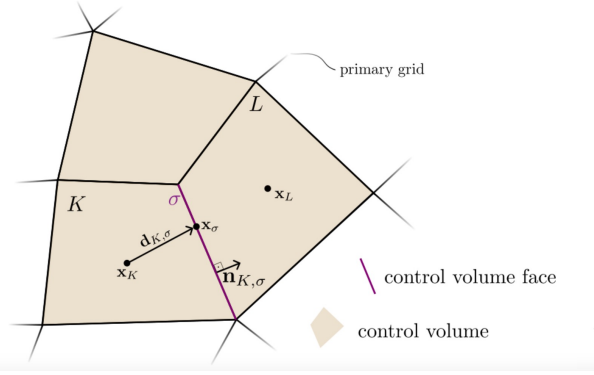
Table 3: **Convergence study**. Bercovier–Engelman test case (2D) on a Delaunay grid.

scheme	h^p	$\|p_h - p\|_{L^2}$	rate	h^v	$\|\boldsymbol{v}_h - \boldsymbol{v}\|_{L^2}$	rate	$\|\boldsymbol{v}_h - \boldsymbol{v}\|_{H^1}$	rate	it
$[\mathbb{P}^{\mathrm{ov}}_{1+b}]^2 \times \mathbb{P}_1$	1.0e-01	6.51e-01	-	6.2e-02	4.73e-02	-	1.74e+00	-	24
	5.7e-02	3.05e-01	1.34	3.4e-02	1.28e-02	2.21	9.49e-01	1.03	26
	3.1e-02	1.09e-01	1.66	1.8e-02	3.62e-03	1.98	4.82e-01	1.06	26
	1.6e-02	4.04e-02	1.51	9.3e-03	9.37e-04	2.04	2.48e-01	1.00	26
	8.1e-03	1.51e-02	1.44	4.7e-03	2.42e-04	1.96	1.24e-01	1.01	26
	4.1e-03	5.48e-03	1.48	2.4e-03	6.16e-05	2.00	6.25e-02	1.00	27
$[\mathbb{P}^{\mathrm{hy}}_{1+b}]^2 \times \mathbb{P}_1$	1.0e-01	8.25e-01	-	6.2e-02	4.74e-02	-	1.76e+00	-	26
	5.7e-02	3.95e-01	1.31	3.4e-02	1.31e-02	2.18	9.60e-01	1.03	28
	3.1e-02	1.41e-01	1.65	1.8e-02	3.61e-03	2.01	4.88e-01	1.06	28
	1.6e-02	5.22e-02	1.52	9.3e-03	9.40e-04	2.03	2.51e-01	1.00	29
	8.1e-03	1.96e-02	1.43	4.7e-03	2.42e-04	1.97	1.25e-01	1.01	29
	4.1e-03	7.08e-03	1.49	2.4e-03	6.17e-05	1.99	6.32e-02	1.00	29
$[\mathbb{P}^{\mathrm{nov}}_{1+b}]^2 \times \mathbb{P}_1$	1.0e-01	6.00e-01	-	6.2e-02	4.83e-02	-	1.73e+00	-	24
	5.7e-02	2.97e-01	1.25	3.4e-02	1.36e-02	2.15	9.48e-01	1.02	26
	3.1e-02	1.07e-01	1.63	1.8e-02	3.45e-03	2.14	4.81e-01	1.06	26
	1.6e-02	4.02e-02	1.50	9.3e-03	8.90e-04	2.05	2.48e-01	1.00	26
	8.1e-03	1.50e-02	1.44	4.7e-03	2.21e-04	2.02	1.24e-01	1.01	26
	4.1e-03	5.46e-03	1.48	2.4e-03	5.64e-05	1.99	6.25e-02	1.00	27
$[\mathbb{P}^{\mathrm{fem}}_{1+b}]^2 \times \mathbb{P}_1$	1.0e-01	1.11e+00	-	6.2e-02	9.81e-02	-	1.82e+00	-	26
	5.7e-02	5.92e-01	1.11	3.4e-02	2.70e-02	2.19	9.82e-01	1.05	28
	3.1e-02	2.19e-01	1.59	1.8e-02	6.37e-03	2.26	4.92e-01	1.08	28
	1.6e-02	8.07e-02	1.53	9.3e-03	1.56e-03	2.12	2.52e-01	1.01	28
	8.1e-03	3.06e-02	1.42	4.7e-03	3.71e-04	2.08	1.25e-01	1.01	28
	4.1e-03	1.10e-02	1.50	2.4e-03	9.34e-05	2.01	6.33e-02	1.00	28

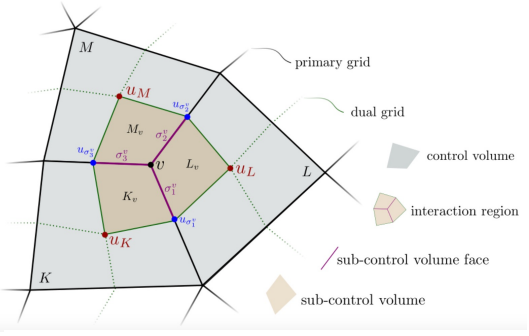
Finite volumes / Cell-centered schemes

`scv = CV ∩ Element`

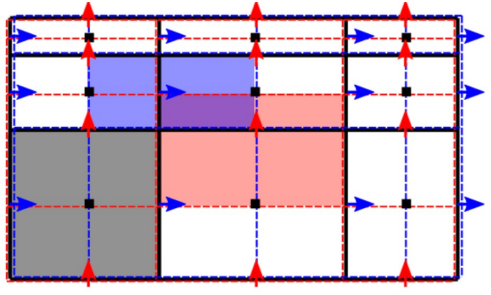
Tpfa



Mpfa

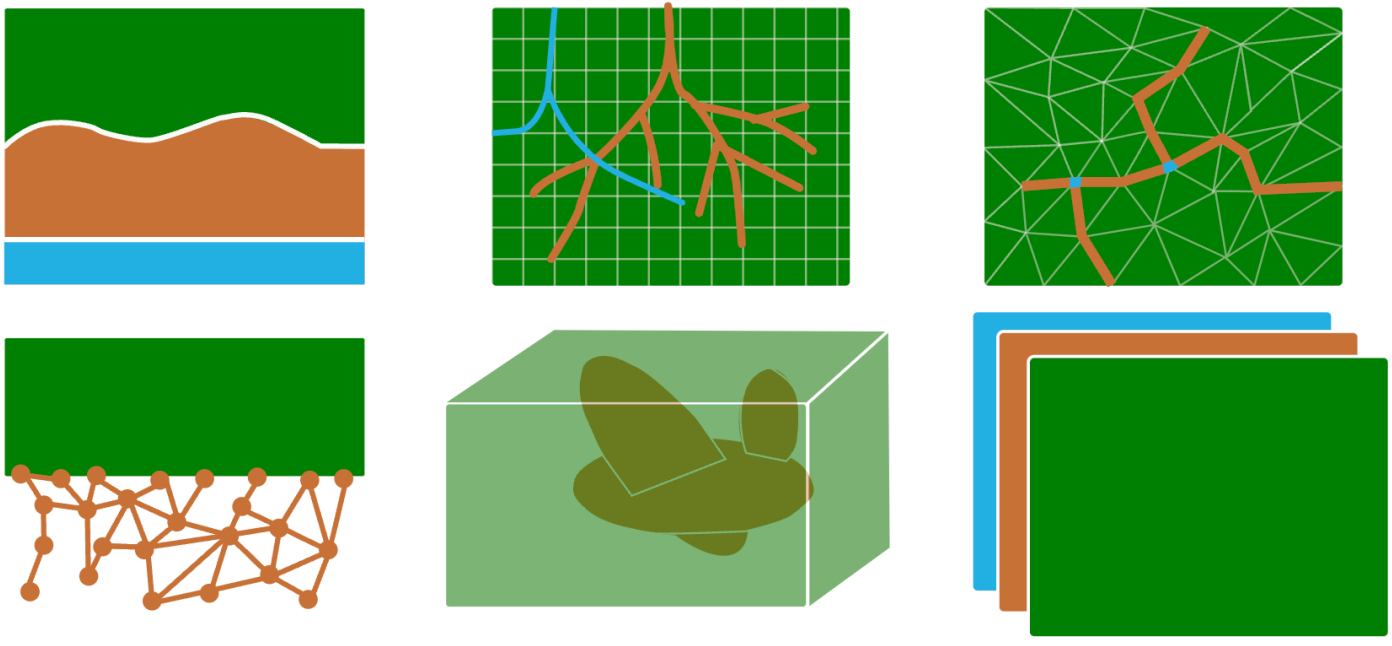


Staggered

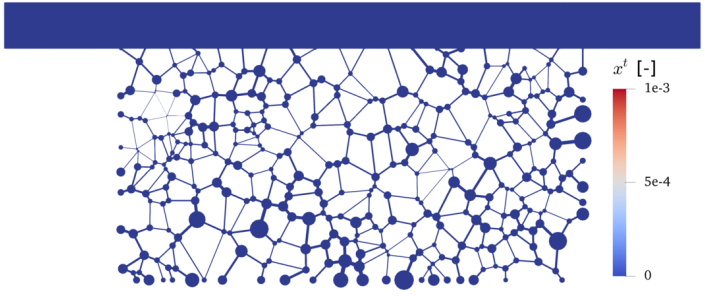


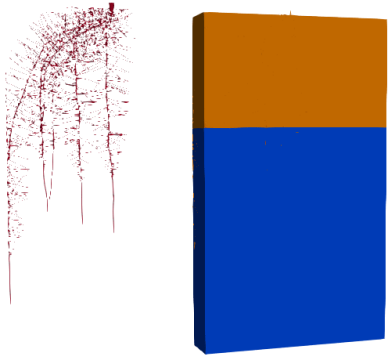
CV = `scv`

Multidomain

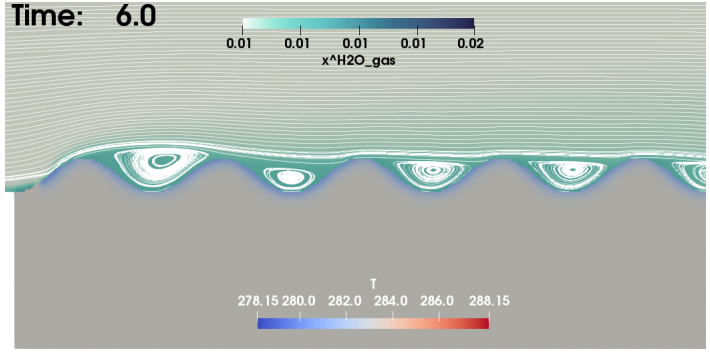


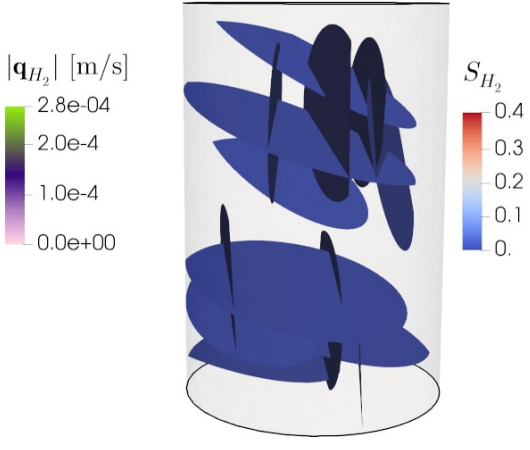
Multidomain





0.0 days



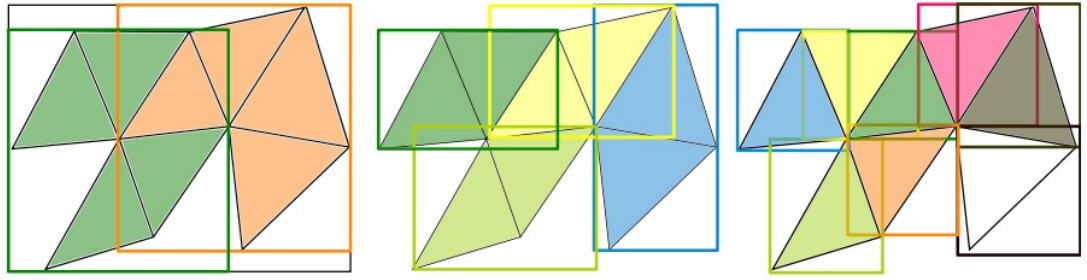


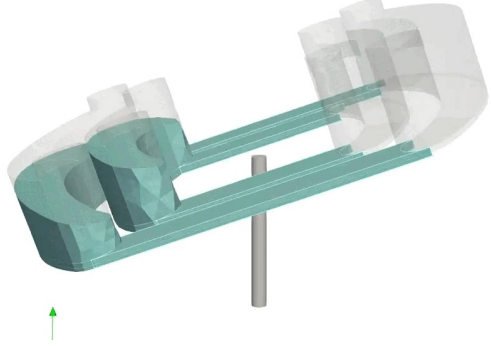
Multidomain "CouplingManager"

```
/*!
 * \brief returns an iterable container of all indices of degrees of freedom of domain j
 *        that couple with / influence the element residual of the given element of domain i
 *
 * \param domainI the domain index of domain i
 * \param elementI the coupled element of domain í
 * \param domainJ the domain index of domain j
 *
 * \note  The element residual definition depends on the discretization scheme of domain i
 *        box: a container of the residuals of all sub control volumes
 *        cc : the residual of the (sub) control volume
 *        fem: the residual of the element
 * \note  This function has to be implemented by all coupling managers for all combinations of i and j
 */
template<std::size_t i, std::size_t j>
const CouplingStencilType<i, j>& couplingStencil(Dune::index_constant<i> domainI,
                                                 const Element<i>& elementI,
                                                 Dune::index_constant<j> domainJ) const
{
    ...
}
```

Multidomain "CouplingManager"

```
/*!
 * \brief returns an iterable container of all indices of degrees of freedom of domain j
 *        that couple with / infl
 *
 * \param domainI the domain inde
 * \param elementI the coupled el
 * \param domainJ the domain inde
 *
 * \note  The element residual de
 *        box: a container of the
 *        cc : the residual of th
 *        fem: the residual of the element
 * \note  This function has to be implemented by all coupling managers for
 */
template<std::size_t i, std::size_t j>
const CouplingStencilType<i, j>& couplingStencil(Dune::index_constant<i> d
                                                 const Element<i>& element
                                                 Dune::index_constant<j> d
{
    ...
}
```



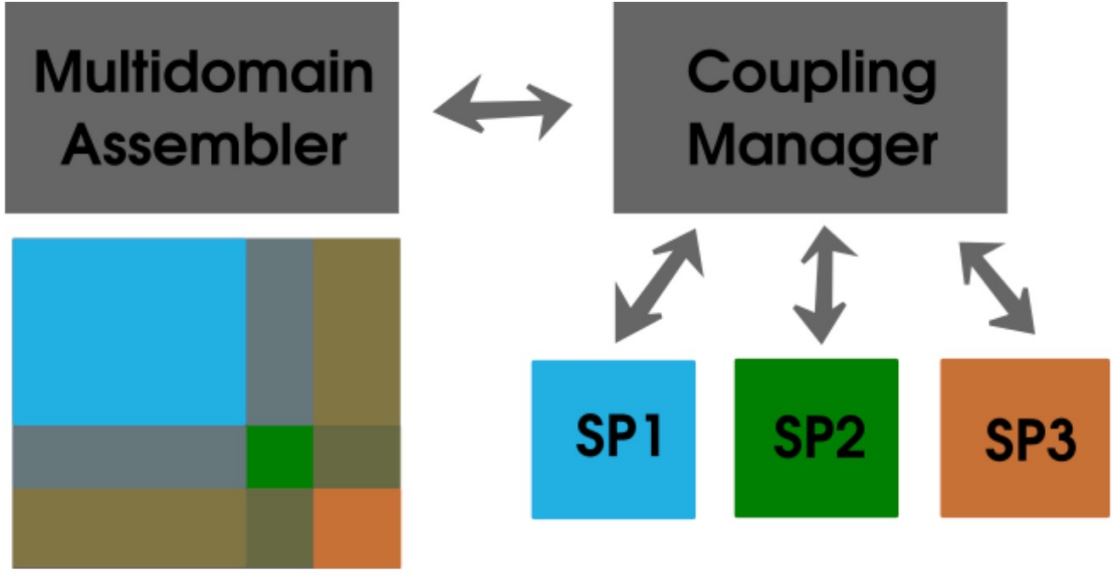


$$R(U) = 0$$

Newton until convergence:

$$U_{n+1} = U_n - (\partial R/\partial U)^{-1} R(U_n)$$

Multidomain “MultiDomainAssembler”



`Dune::MultiTypeBlockMatrix`

Stokes in DuMux multidomain

Prototyping coupled problems

Stokes equations

$$\int_{\partial K} [-2\mu \boldsymbol{D}(\boldsymbol{v}_h) + p_h \boldsymbol{I}] \cdot \mathbf{n} \, \mathrm{d}A = \int_K \boldsymbol{f} \, \mathrm{d}x, \quad \forall K \in \mathcal{T}^{\boldsymbol{v}},$$

$$\int_{\partial K} \boldsymbol{v}_h \cdot \mathbf{n} \, \mathrm{d}A = \int_K q \, \mathrm{d}x, \quad \forall K \in \mathcal{T}^{p},$$

$$\boldsymbol{D}(\boldsymbol{v}) = \frac{1}{2}\left(\nabla \boldsymbol{v} + \nabla^T \boldsymbol{v}\right)$$

Coupled PDE

Multiple discretization schemes

Momentum model	$\mathbb{P}_1^{+B}$	$\mathbb{P}_k$	$\mathbb{P}_1^{CR}$
Mass model	$\mathbb{P}_1$	$\mathbb{P}_{k-1}$	$\mathbb{P}_0$

Multidomain “CouplingManager”

Give me coupled variable at integration point

```
auto divU(typename GridGeometry<flowIdx>::LocalView const& fvGeometry,
          typename GridGeometry<flowIdx>::SubControlVolume const& scv) const
{
    const auto& gg = this->problem(mechanicsIdx).gridGeometry();
    const auto elemSol = elementSolution(fvGeometry.element(), curSol(mechanicsIdx), gg);

    const auto gradU = evalGradients(
        fvGeometry.element(),
        fvGeometry.element().geometry(),
        gg, elemSol,
        scv.center()
    );

    double divU = 0.0;
    for (int i = 0; i < gradU.size(); ++i)
        divU += gradU[i][i];
    return divU;
}
```

```
auto porePressure(typename GridGeometry<mechanicsIdx>::LocalView const& fvGeometry,
                  typename GridGeometry<mechanicsIdx>::SubControlVolumeFace const& scvf) const
{
    const auto& gg = this->problem(flowIdx).gridGeometry();
    const auto elemSol = elementSolution(fvGeometry.element(), curSol(flowIdx), gg);
    return evalSolution(
        fvGeometry.element(),
        fvGeometry.element().geometry(),
        gg, elemSol,
        scvf.ipGlobal()
    )[0];
}
```