Adaptive 2D+t Space-Time Mesh Generation

Tamás Horváth, Emily Smeu
Oakland University

Sander Rhebergen
University of Waterloo

CG Week 2026
Workshop on Triangulations and More General Meshes for Space-Time Applications
June 2-5, 2026

Contents

Time-dependent Navier-Stokes equations

Time dependent incompressible Navier-Stokes

$$
\begin{aligned}
\partial_t u + u \cdot \nabla u + \nabla p - 2\nu\nabla \cdot \varepsilon(u) = f \qquad & \text{in } \Omega(t)\,, \\
\nabla \cdot u = 0 \qquad & \text{in } \Omega(t)\,, \\
+ BC + IC &
\end{aligned}
$$

Time-dependent Navier-Stokes equations

Time dependent incompressible Navier-Stokes

$$
\begin{aligned}
\partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) = f \qquad & \text{in } \Omega(t)\,, \\
\nabla \cdot u = 0 \qquad & \text{in } \Omega(t)\,, \\
+ BC + IC &
\end{aligned}
$$

The domain is time-dependent $\Omega(t), t \in [t_0, t_N]$

Time-dependent Navier-Stokes equations

Time dependent incompressible Navier-Stokes

$$
\begin{aligned}
\partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) = f \qquad & \text{in } \Omega(t)\,, \\
\nabla \cdot u = 0 \qquad & \text{in } \Omega(t)\,, \\
+ BC + IC &
\end{aligned}
$$

The domain is time-dependent $\Omega(t), t \in [t_0, t_N]$

Space-time method

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

- Generate a mesh of the domain $\Omega(t_i)$.

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

- Generate a mesh of the domain $\Omega(t_i)$.
- Generate a mesh of the domain $\Omega(t_{i+1})$

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

- Generate a mesh of the domain $\Omega(t_i)$.
- Generate a mesh of the domain $\Omega(t_{i+1})$
- Generate a space-time mesh using these two meshes

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

- Generate a mesh of the domain $\Omega(t_i)$.
- Generate a mesh of the domain $\Omega(t_{i+1})$
- Generate a space-time mesh using these two meshes
- Solve inside the time slab and evaluate the solution at the upper time level

Space-time algorithm

Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

- Generate a mesh of the domain $\Omega(t_i)$.
- Generate a mesh of the domain $\Omega(t_{i+1})$
- Generate a space-time mesh using these two meshes
- Solve inside the time slab and evaluate the solution at the upper time level
- Use this as an initial condition for the next time slab

Space-time algorithm

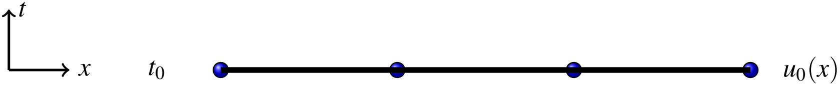
Space-time

Start with $i = 0$ and the initial condition $u(0, x) = u_0(x)$

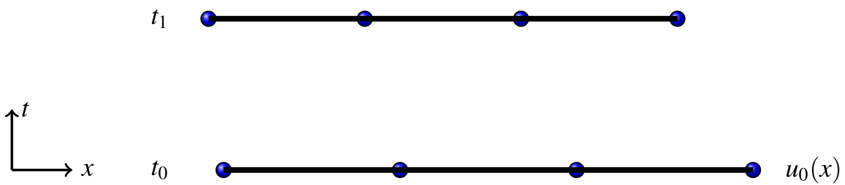
- Generate a mesh of the domain $\Omega(t_i)$.
- Generate a mesh of the domain $\Omega(t_{i+1})$
- Generate a space-time mesh using these two meshes
- Solve inside the time slab and evaluate the solution at the upper time level
- Use this as an initial condition for the next time slab

All we need is u from the previous time slab

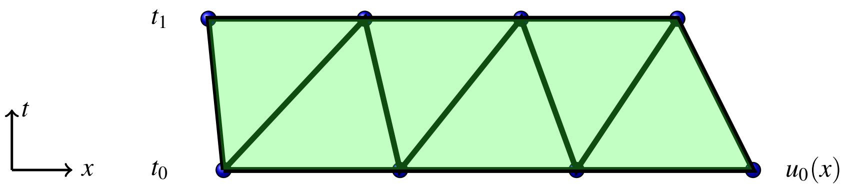
Upwind in time



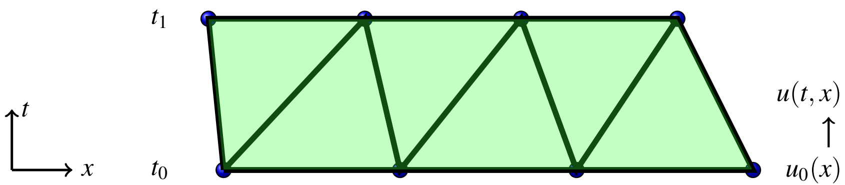
Upwind in time



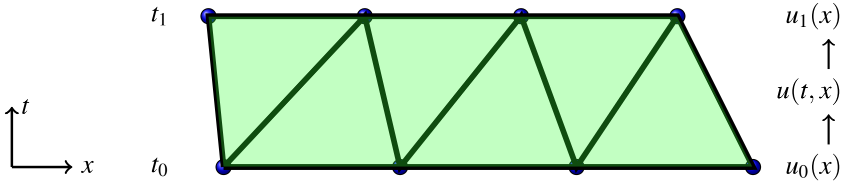
Upwind in time



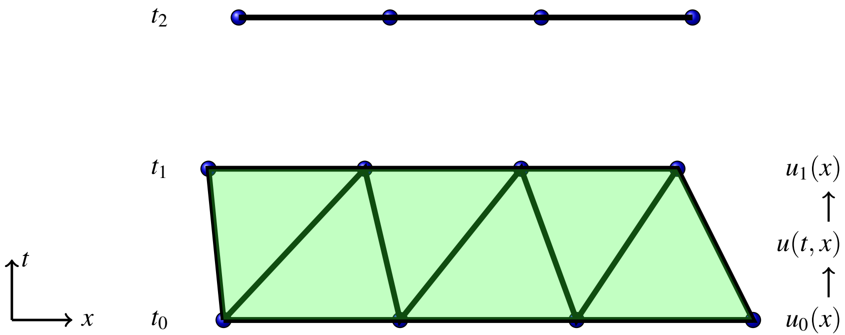
Upwind in time



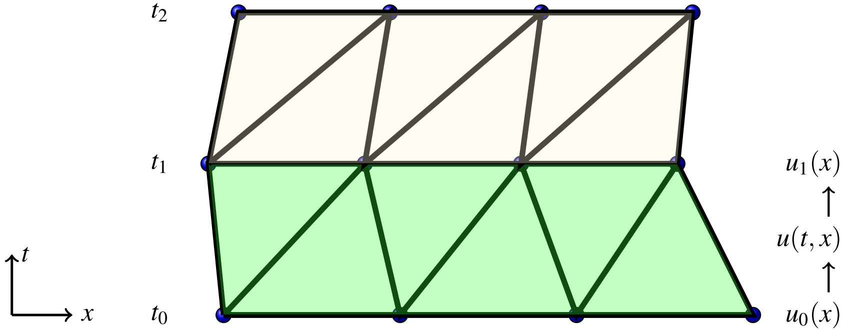
Upwind in time



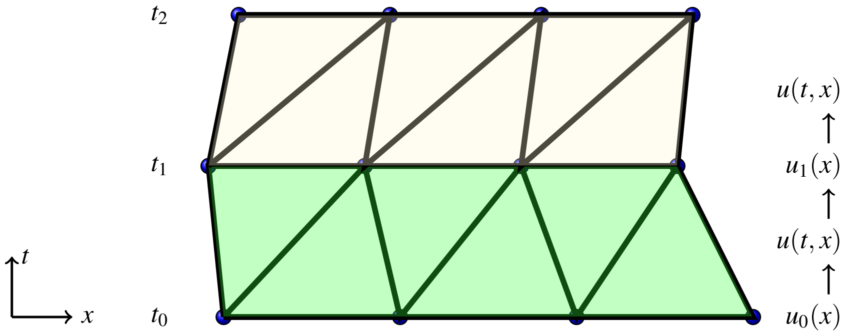
Upwind in time



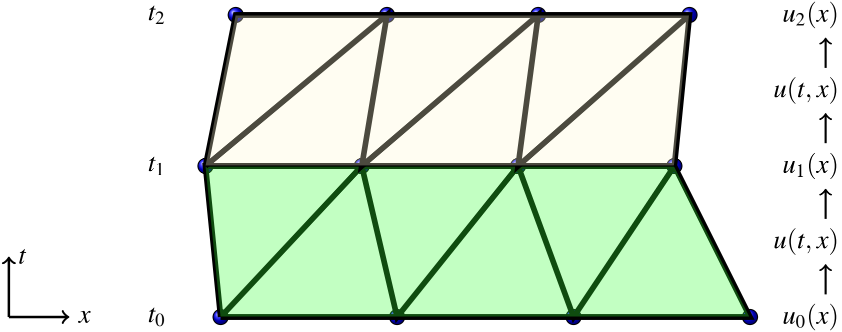
Upwind in time



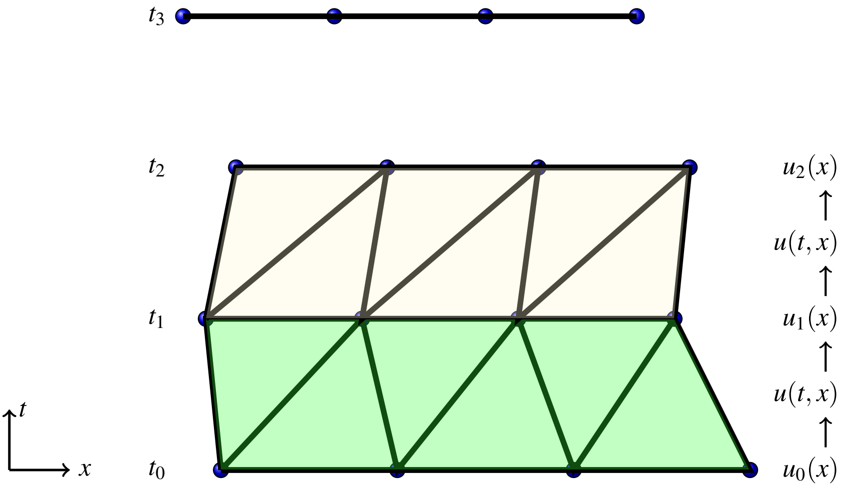
Upwind in time



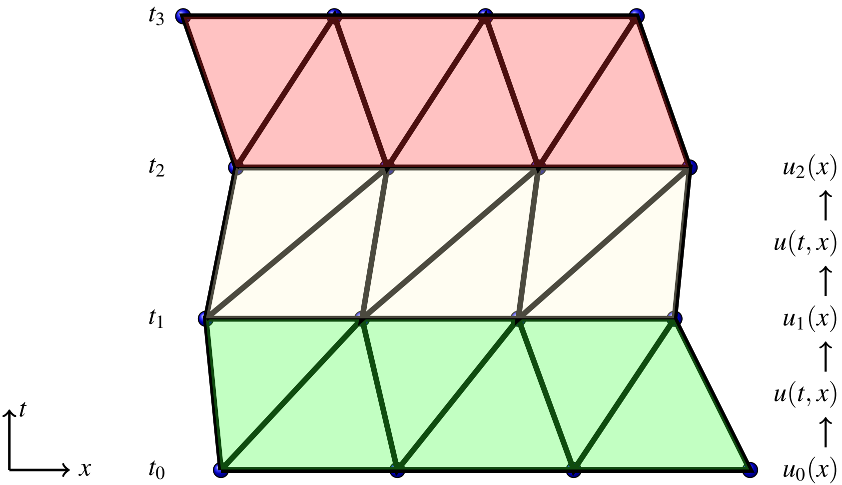
Upwind in time



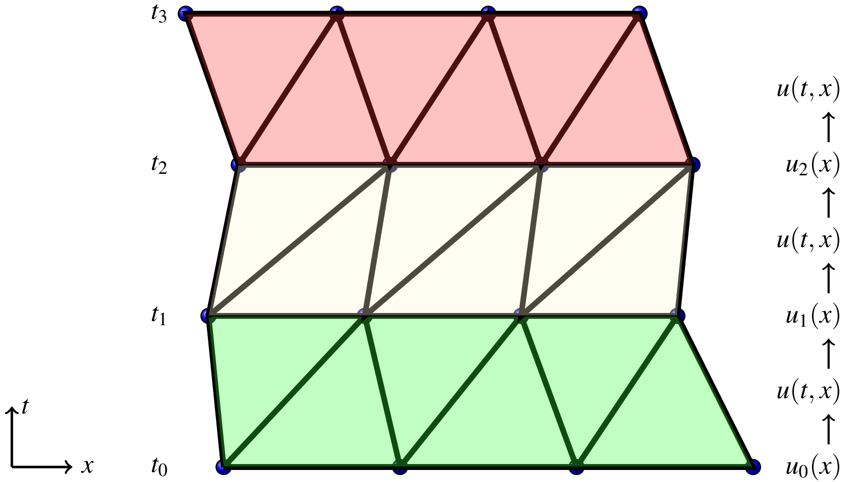
Upwind in time



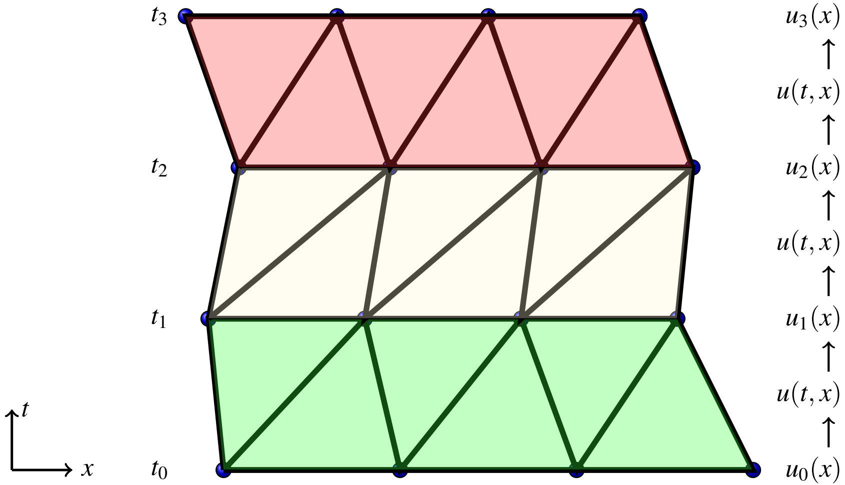
Upwind in time



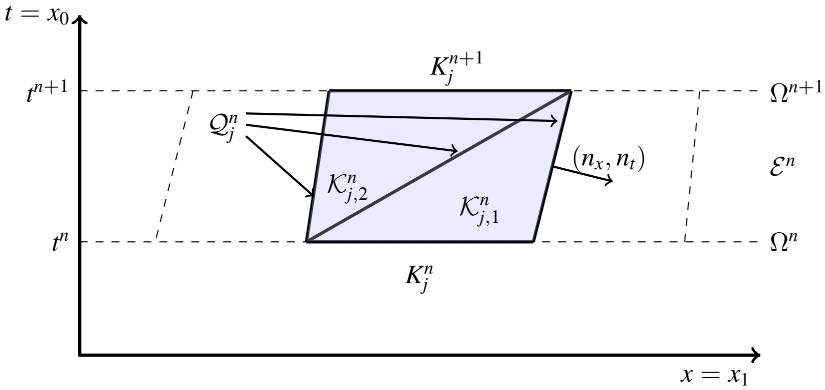
Upwind in time



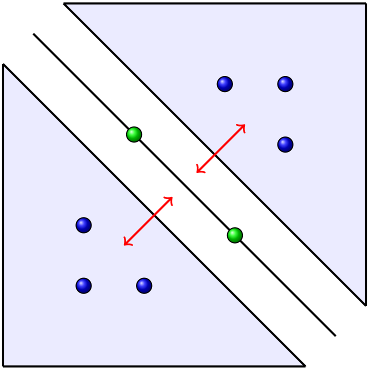
Upwind in time



Space-time notations - simplices



HDG



$$u_h \in V_h = \left\{ v_h \in L^2(\Omega),\ v_h \in P^k(K)\ \forall K \in \mathcal{T}_h \right\}$$

$$\overline{u}_h \in \overline{V}_h = \left\{ \overline{v}_h \in L^2(\mathcal{F}),\ \overline{v}_h \in P^k(F)\ \forall F \in \mathcal{F} \right\}$$

No facet unknowns on time levels

Fluid-rigid body interactions

Time dependent Navier-Stokes

$$\begin{aligned} \partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) &= f && \text{in } \Omega(t) \\ \nabla \cdot u &= 0 && \text{in } \Omega(t) \end{aligned}$$

Fluid-rigid body interactions

Time dependent Navier-Stokes

$$
\begin{aligned}
\partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) &= f \qquad \text{in } \Omega(t) \\
\nabla \cdot u &= 0 \qquad \text{in } \Omega(t)
\end{aligned}
$$

ST-EHDG

Fluid-rigid body interactions

Time dependent Navier-Stokes

$$\begin{aligned} \partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) &= f \qquad \text{in } \Omega(t) \\ \nabla \cdot u &= 0 \qquad \text{in } \Omega(t) \end{aligned}$$

ST-EHDG

Vertical displacement and rotation

$$\begin{aligned} m\ddot{d} + c_y\dot{d} + k_y d &= F_y, \\ I_\theta \ddot{\theta} + c_\theta \dot{\theta} + k_\theta \theta &= M, \end{aligned}$$

where F_y is the lifting force, M is the pitching moment force

Fluid-rigid body interactions

Time dependent Navier-Stokes

$$\begin{aligned} \partial_t u + u \cdot \nabla u + \nabla p - 2\nu \nabla \cdot \varepsilon(u) &= f \qquad \text{in } \Omega(t) \\ \nabla \cdot u &= 0 \qquad \text{in } \Omega(t) \end{aligned}$$

ST-EHDG

Vertical displacement and rotation

$$\begin{aligned} m\ddot{d} + c_y \dot{d} + k_y d &= F_y, \\ I_\theta \ddot{\theta} + c_\theta \dot{\theta} + k_\theta \theta &= M, \end{aligned}$$

where F_y is the lifting force, M is the pitching moment force

Predictor-corrector (BDF2)

Discretization

Viscous term: hybridized IP-DG discretization

Pressure terms: standard hybridization

Convection: nonlinear upwinding

In space-time slab n

Find $(u_h, \bar{u}_h, p_h, \bar{p}_h) \in V_h \times \bar{V}_h \times Q_h \times \bar{Q}_h$ such that

$$t_h^n(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + a_h^n(\mathbf{u}_h, \mathbf{v}_h) + b_h^n(\mathbf{p}_h, \mathbf{v}_h) - b_h^n(\mathbf{q}_h, \mathbf{u}_h) = \sum_{\mathcal{K} \in \mathcal{T}^n} \int_{\mathcal{K}} f \cdot v_h \,\mathrm{d}x - \int_{\partial \mathcal{E}^N \cap I_n} g \cdot \bar{v}_h \,\mathrm{d}s + \int_{\Omega_n} u_h^- \cdot v_h \,\mathrm{d}s$$

Discretization

Viscous term: hybridized IP-DG discretization

Pressure terms: standard hybridization

Convection: nonlinear upwinding

In space-time slab n

Find $(u_h, \bar{u}_h, p_h, \bar{p}_h) \in V_h \times \bar{V}_h \times Q_h \times \bar{Q}_h$ such that

$$t_h^n(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + a_h^n(\mathbf{u}_h, \mathbf{v}_h) + b_h^n(\mathbf{p}_h, \mathbf{v}_h) - b_h^n(\mathbf{q}_h, \mathbf{u}_h) = \sum_{\mathcal{K} \in \mathcal{T}^n} \int_{\mathcal{K}} f \cdot v_h \, \mathrm{d}x - \int_{\partial \mathcal{E}^N \cap I_n} g \cdot \bar{v}_h \, \mathrm{d}s + \int_{\Omega_n} u_h^- \cdot v_h \, \mathrm{d}s$$

Nonlinearity - Picard Iteration

Discretization

Viscous term: hybridized IP-DG discretization

Pressure terms: standard hybridization

Convection: nonlinear upwinding

In space-time slab n

Find $(u_h, \bar{u}_h, p_h, \bar{p}_h) \in V_h \times \bar{V}_h \times Q_h \times \bar{Q}_h$ such that

$$t_h^n(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + a_h^n(\mathbf{u}_h, \mathbf{v}_h) + b_h^n(\mathbf{p}_h, \mathbf{v}_h) - b_h^n(\mathbf{q}_h, \mathbf{u}_h) = \sum_{\mathcal{K} \in \mathcal{T}^n} \int_{\mathcal{K}} f \cdot v_h \,\mathrm{d}x - \int_{\partial\mathcal{E}^N \cap I_n} g \cdot \bar{v}_h \,\mathrm{d}s + \int_{\Omega_n} u_h^- \cdot v_h \,\mathrm{d}s$$

Nonlinearity - Picard Iteration

Linear system - Static condensation

Static Condensation

Using U and $\overline{U}$ for the coefficient vectors

Block system

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ \overline{U} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

Static Condensation

Using U and $\overline{U}$ for the coefficient vectors

Block system

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ \overline{U} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

A is block diagonal

$$U = A^{-1}(F - B\overline{U})$$
$$CA^{-1}(F - B\bar{U}) + D\overline{U} = G$$

Static Condensation

Using U and $\overline{U}$ for the coefficient vectors

Block system

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ \overline{U} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

A is block diagonal

$$\begin{aligned} U &= A^{-1}(F - B\overline{U}) \\ CA^{-1}(F - B\bar{U}) + D\overline{U} &= G \end{aligned}$$

Final problem

$$\begin{aligned} (D - CA^{-1}B)\overline{U} &= G - CA^{-1}F \\ U &= A^{-1}(F - B\overline{U}) \end{aligned}$$

Static Condensation

Using U and $\overline{U}$ for the coefficient vectors

Block system

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ \overline{U} \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

A is block diagonal

$$\begin{aligned} U &= A^{-1}(F - B\overline{U}) \\ CA^{-1}(F - B\bar{U}) + D\overline{U} &= G \end{aligned}$$

Final problem

$$\begin{aligned} (D - CA^{-1}B)\overline{U} &= G - CA^{-1}F \\ U &= A^{-1}(F - B\overline{U}) \end{aligned}$$

Typically, the number of DoFs is smaller for HDG than for DG

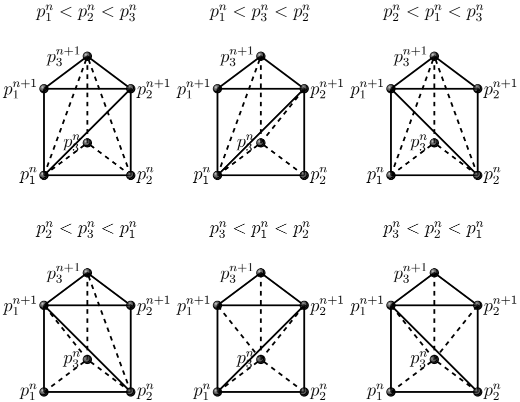
Comparison (on simplices - moving domain)

	ST-HDG*	ST-EDG†	ST-EHDG†
div-free velocity	✓	✓	✓
div-conforming velocity	✓	×	✓
energy-stable	✓	✓	✓
loc. mom. conserving	✓	×	✓
number of degrees-of-freedom	largest	smallest	significantly < ST-HDG slightly > ST-EDG

*T.L. Horvath and S. Rhebergen, A locally conservative and energy-stable finite element method for the Navier–Stokes problem on time-dependent domains, Int. J. Numer. Meth. Fluids, 89/12 (2019), pp 519-532.

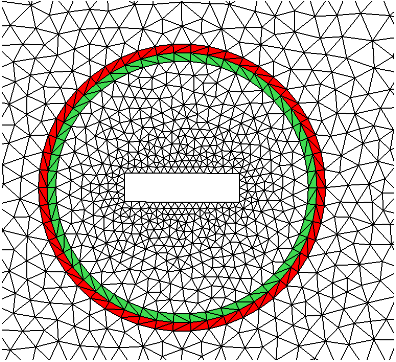
†T.L. Horvath and S. Rhebergen, An exactly mass conserving space-time embedded-hybridized discontinuous Galerkin method for the Navier-Stokes equations on moving domains, J. Comp. Phys. 417 (2020)

Basic meshing idea

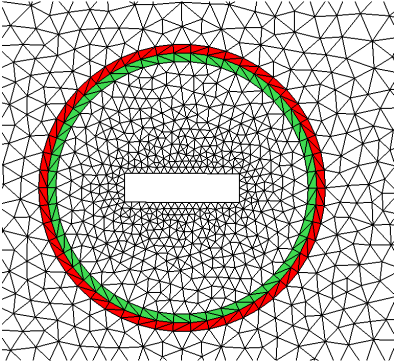


Use the diagonal cut from the node with the smallest index (staircase triangulation)

Sliding mesh generation

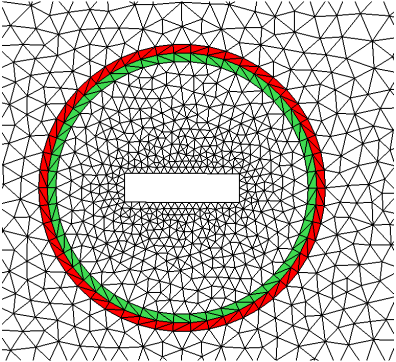


Sliding mesh generation



Inner region rotates with the object

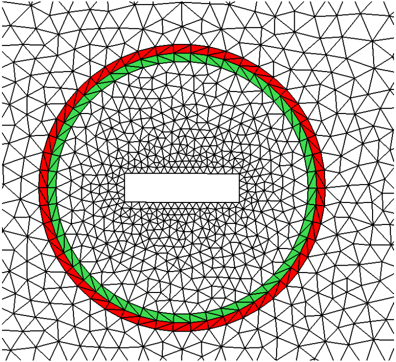
Sliding mesh generation



Inner region rotates with the object

Outer region does not change

Sliding mesh generation

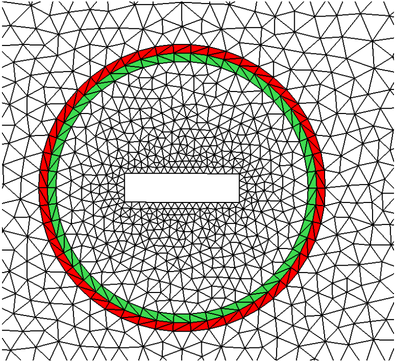


Inner region rotates with the object

Outer region does not change

In between: sliding layer

Sliding mesh generation



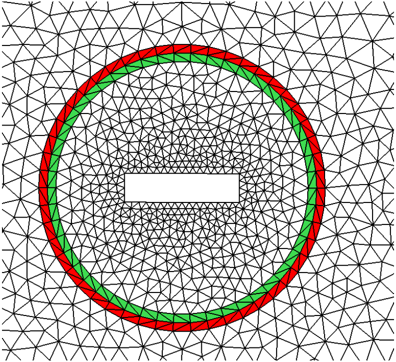
Inner region rotates with the object

Outer region does not change

In between: sliding layer

Buffer layer allows precomputable meshes

Sliding mesh generation



Inner region rotates with the object

Outer region does not change

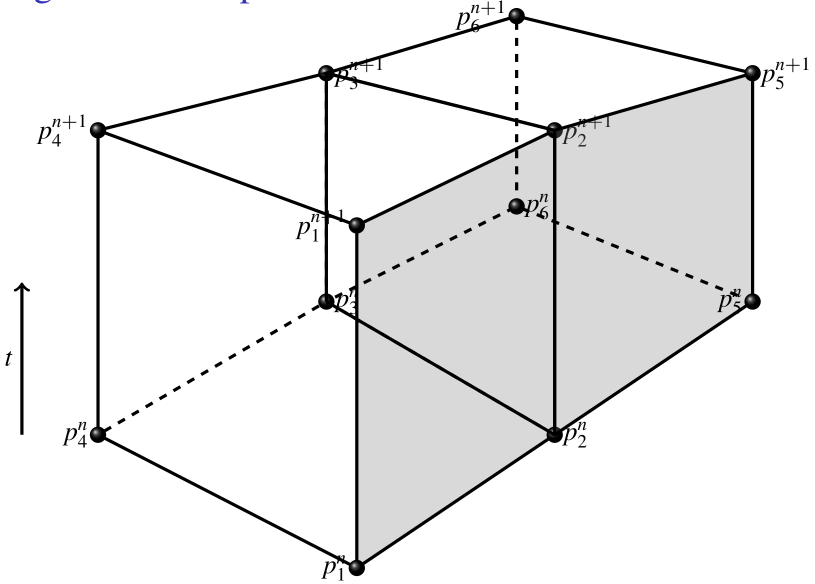
In between: sliding layer

Buffer layer allows precomputable meshes

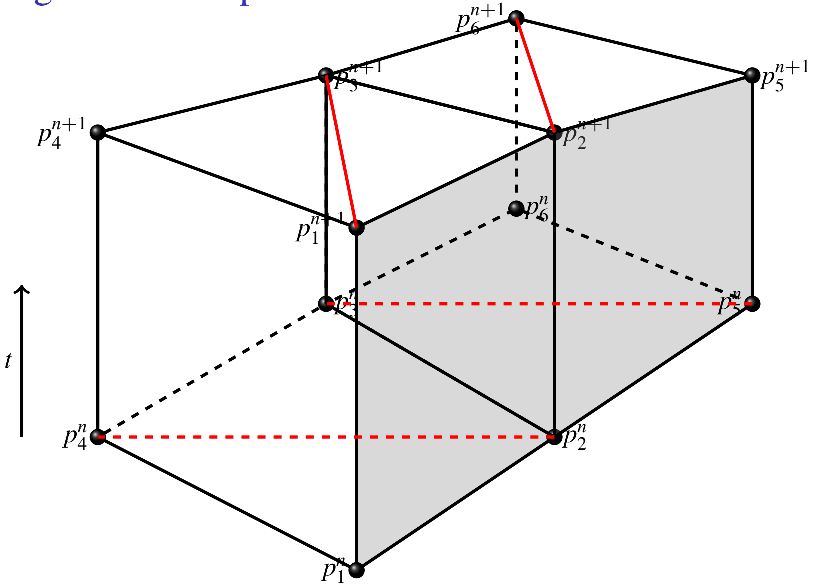
"Build your own mesh"[a]

[a] T.L. Horvath and S. Rhebergen, A conforming sliding mesh technique for an embedded-hybridized discontinuous Galerkin discretization for fluid-rigid body interaction (2022), Int. J. Numer. Meth. Fluids, 94/11 (2022), pp. 1784-1809

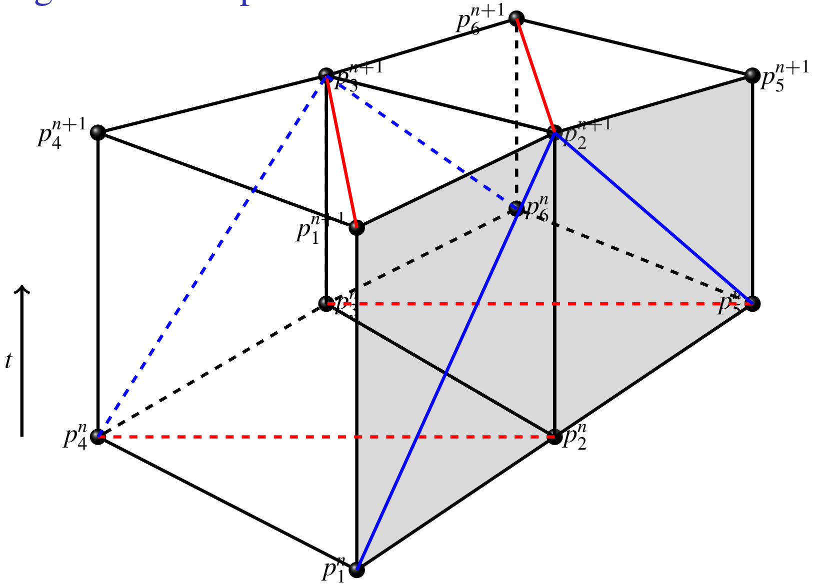
Sliding block example



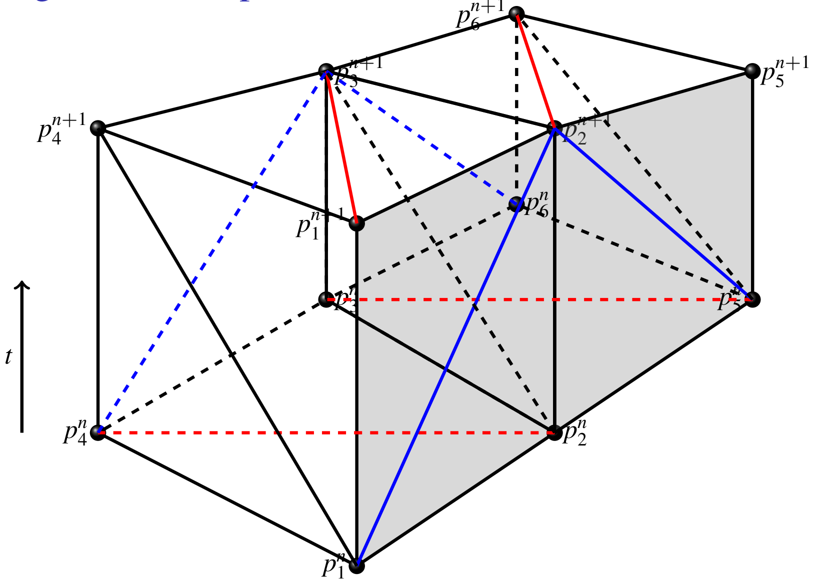
Sliding block example



Sliding block example

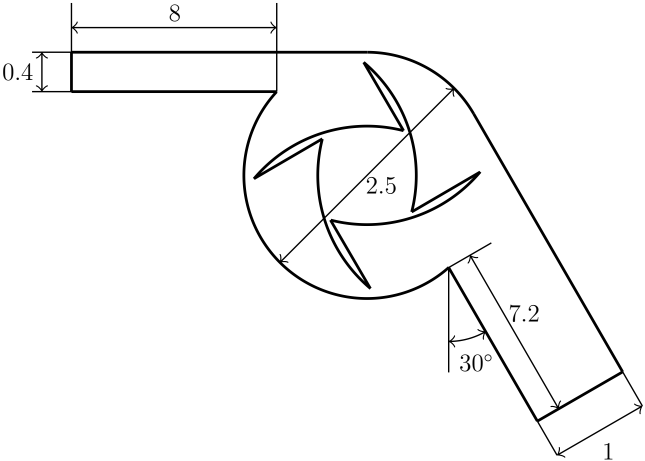


Sliding block example

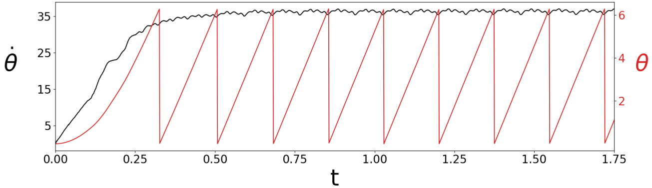


Rotating turbine

- Parabolic inflow, $\max |u_{in}| = 50$



Rotating turbine



Adaptive mesh refinement

AMR with Derefinement (coarsening)

Only refine the mesh where it is needed

Adaptive mesh refinement

AMR with Derefinement (coarsening)

Only refine the mesh where it is needed

Dictated by either an error estimator or some properties of the solution

Adaptive mesh refinement

AMR with Derefinement (coarsening)

Only refine the mesh where it is needed

Dictated by either an error estimator or some properties of the solution

Refinement only for time-dependent problems can be too expensive

Adaptive mesh refinement

AMR with Derefinement (coarsening)

Only refine the mesh where it is needed

Dictated by either an error estimator or some properties of the solution

Refinement only for time-dependent problems can be too expensive

It needs to be combined with derefinement (coarsening)

Space-time adaptivity[‡]

Space and time adaptivity

We can "eliminate" or create new spatial elements

[‡] **TLH, E. Smeu** A conforming space-time mesh adaptation with local coarsening, in preparation

Space-time adaptivity[‡]

Space and time adaptivity

We can "eliminate" or create new spatial elements

Slab-by-slab approach: simplices allow local time refinement

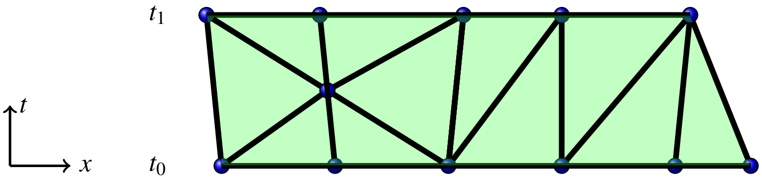
[‡] **TLH, E. Smeu** A conforming space-time mesh adaptation with local coarsening, in preparation

Space-time adaptivity‡

Space and time adaptivity

We can "eliminate" or create new spatial elements

Slab-by-slab approach: simplices allow local time refinement

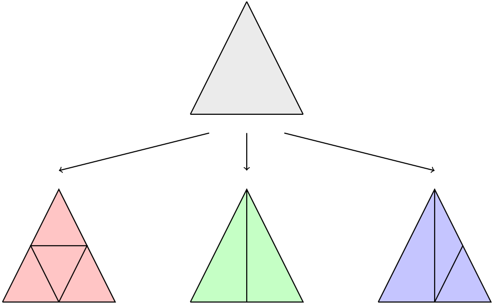


‡**TLH, E. Smeu** A conforming space-time mesh adaptation with local coarsening, in preparation

2D RGB refinement[§]

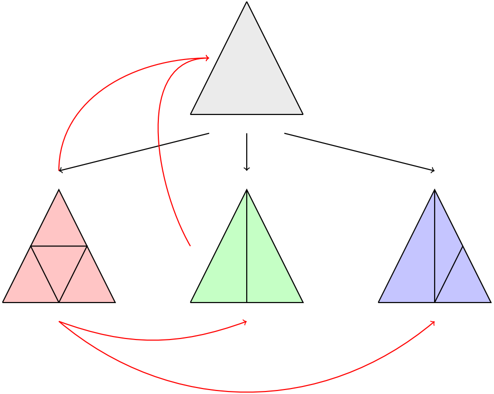
[§] Funken, S. A., Schmidt, A. (2021). A coarsening algorithm on adaptive red-green-blue refined meshes. Numerical Algorithms, 87, 1147-1176.

2D RGB refinement[§]



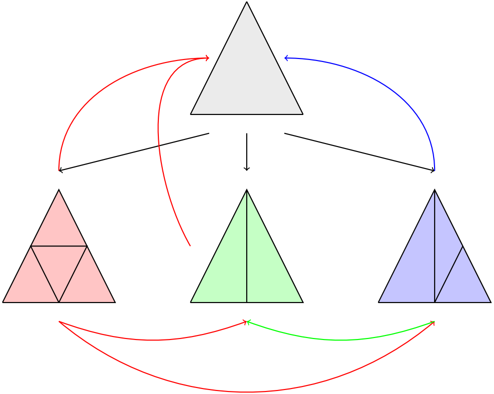
§Funken, S. A., Schmidt, A. (2021). A coarsening algorithm on adaptive red-green-blue refined meshes. Numerical Algorithms, 87, 1147-1176.

2D RGB refinement[§]



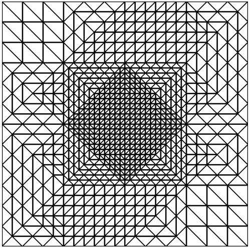
§ Funken, S. A., Schmidt, A. (2021). A coarsening algorithm on adaptive red-green-blue refined meshes. Numerical Algorithms, 87, 1147-1176.

2D RGB refinement§

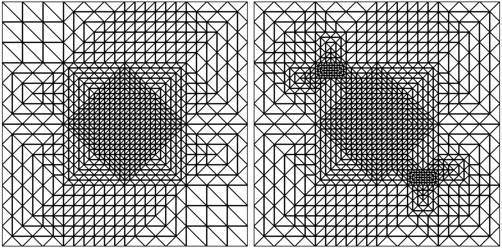


§ Funken, S. A., Schmidt, A. (2021). A coarsening algorithm on adaptive red-green-blue refined meshes. Numerical Algorithms, 87, 1147-1176.

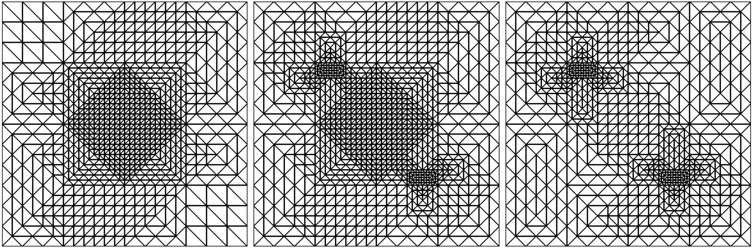
Example



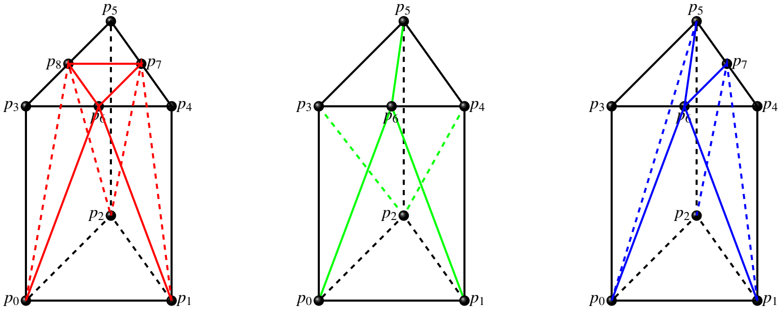
Example



Example

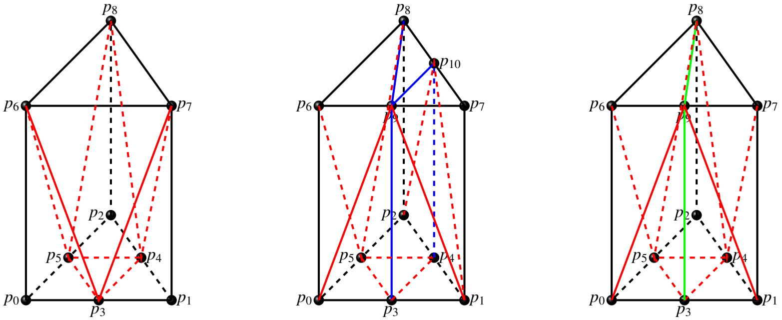


Extension to ST I - no need of projection[¶]



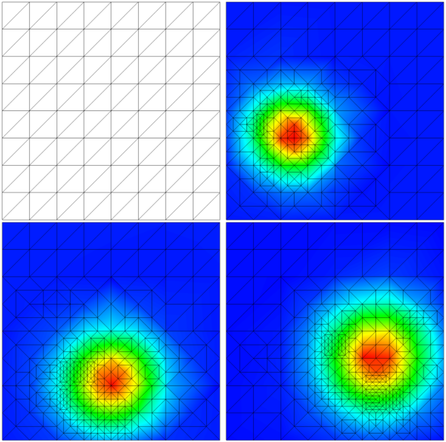
¶ 9 different space-time elements

Extension to ST II - no need of projection‖

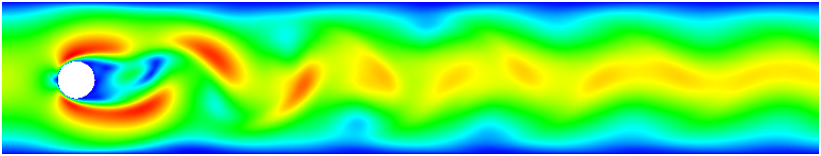


‖8 different space-time elements

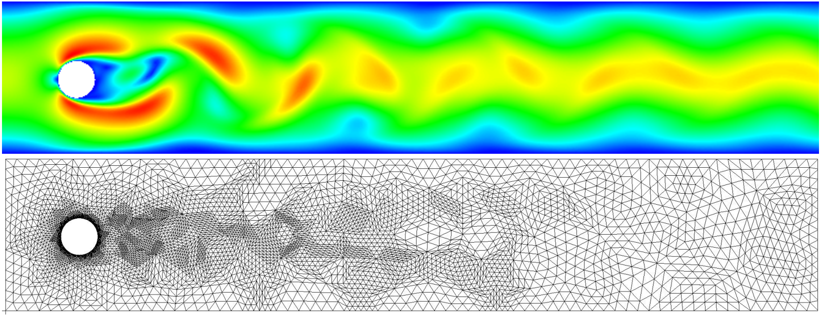
Rotation Gaussian pulse - advection-diffusion



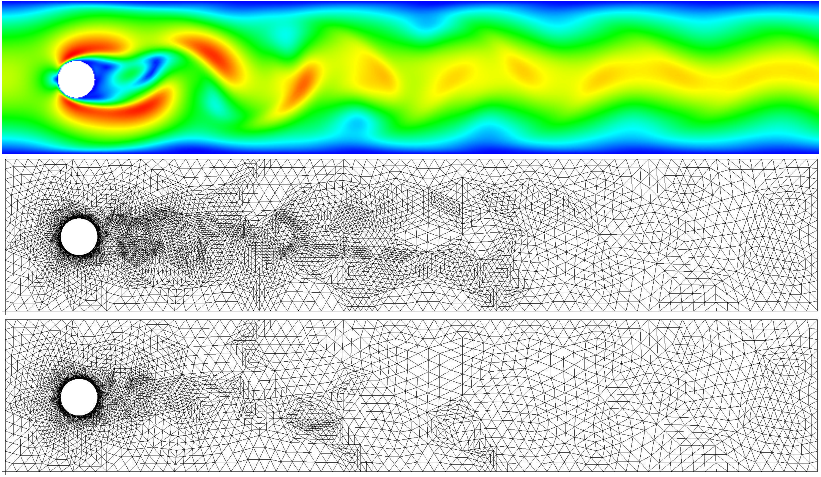
Flow past the cylinder example



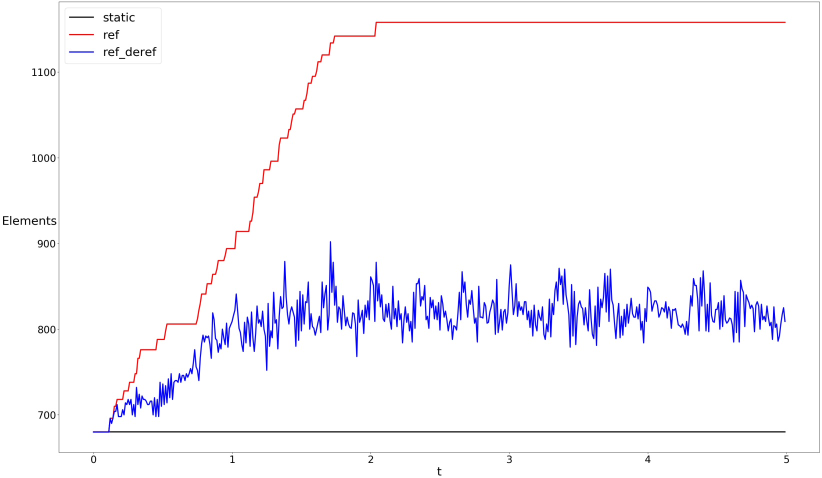
Flow past the cylinder example



Flow past the cylinder example

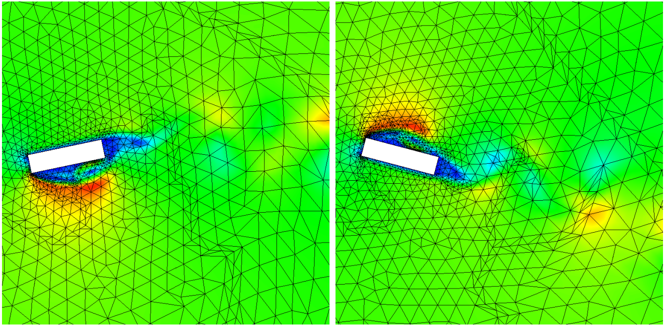


Number of mesh elements



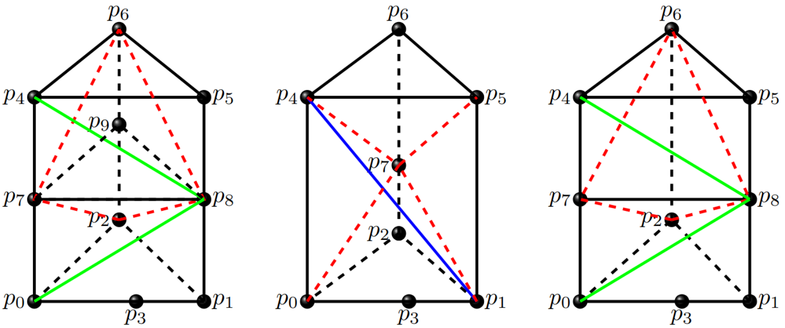
FRBI with mesh adaptivity

Rotational galloping

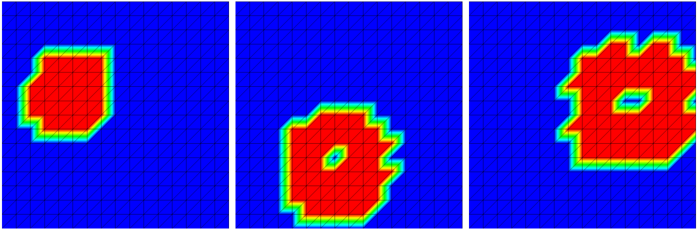


Maximum angle and frequency of the oscillation is better matching the literature then in our previous work while using fewer elements

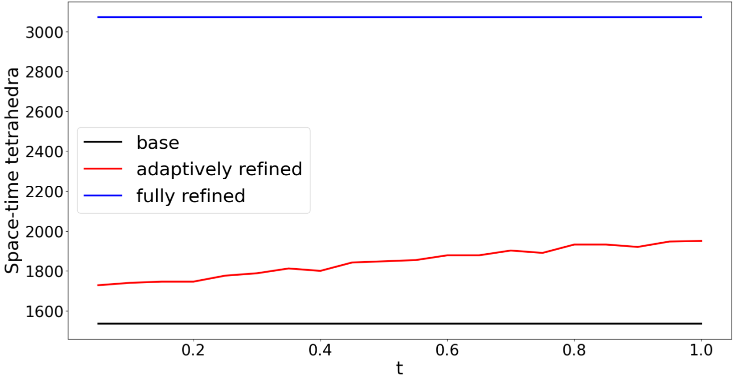
Time adaptivity



Rotation Gaussian pulse - advection-diffusion



Time adaptivity



mesh	unrefined	fully refined	adaptively refined
L^2 error	$8.99 \cdot 10^{-5}$	$6.2 \cdot 10^{-5}$	$6.2 \cdot 10^{-5}$

Extension to 4D

MFEM 4D branch

Extension to 4D

MFEM 4D branch

Base case: tetrahedra to 4D space-time "wedge" to 4D pentatope

Extension to 4D

MFEM 4D branch

Base case: tetrahedra to 4D space-time "wedge" to 4D pentatope

The diagonal from the node with the lowest index works (staircase triangulation) - 24 options

Extension to 4D

MFEM 4D branch

Base case: tetrahedra to 4D space-time "wedge" to 4D pentatope

The diagonal from the node with the lowest index works (staircase triangulation) - 24 options

Temporal adaptivity should extend

Extension to 4D

MFEM 4D branch

Base case: tetrahedra to 4D space-time "wedge" to 4D pentatope

The diagonal from the node with the lowest index works (staircase triangulation) - 24 options

Temporal adaptivity should extend

AMR with coarsening without mesh history: not with RGB

Extension to 4D

MFEM 4D branch

Base case: tetrahedra to 4D space-time "wedge" to 4D pentatope

The diagonal from the node with the lowest index works (staircase triangulation) - 24 options

Temporal adaptivity should extend

AMR with coarsening without mesh history: not with RGB

Predecessor of the RGB paper: an NVB paper, might generalize to 3D

Conclusion & Future work

Conclusion

- Refinement and coarsening without projecting the solution
- Local temporal refinement

Future work

- Time refinment next to the obtacle (high order ODE solver)
- Extend to 3+1 D (currently working on it)
- Fluid-Structure Interaction
- Cahn-Hilliard equation (phase separations: sharp moving discontinuities)

https://thorvath12.github.io

Thank you!

Thank you!

Midwest Numerical Analysis Day 2027
May 21-22, 2027 at Oakland University