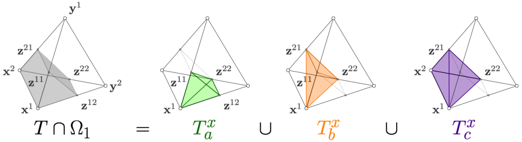
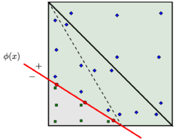
ngsxfem: Add-on to NGSolve for unfitted discretizations

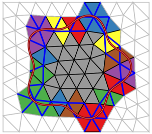
F. Heimann, Christoph Lehrenfeld, J. Preuß, H. v. Wahl,
P. Stocker, T. v. Beeck, T. Ludescher, M. Zienecker

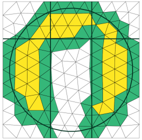
Institute for Numerical and Applied Mathematics, Georg-August University

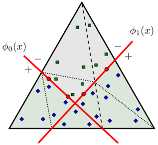
NGSolve User Meeting, 2024, Vienna

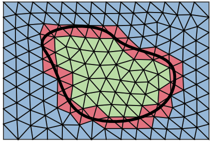


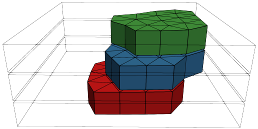


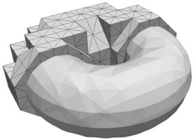


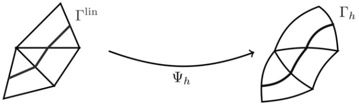


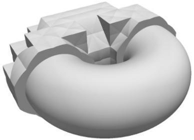






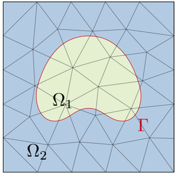


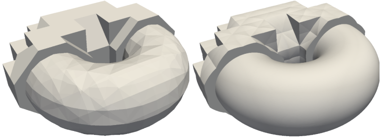


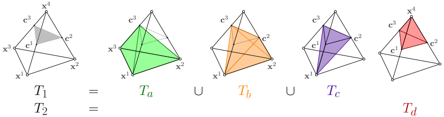


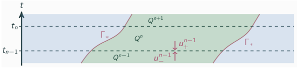
Content

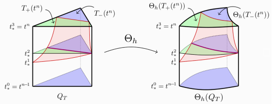
- Motivation
  - Field of research (the setting)
  - What is `ngsxfem`?
- Features of `ngsxfem`
  - Working on submeshes
  - Numerical integration on level set domains
  - higher order (iso param)
  - Multiple level set geometries
  - Cell aggregation
  - Space-time FEM
- Where `ngsxfem` is used

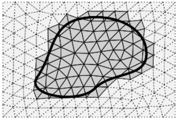


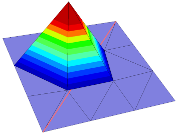


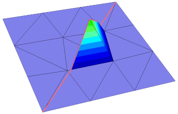








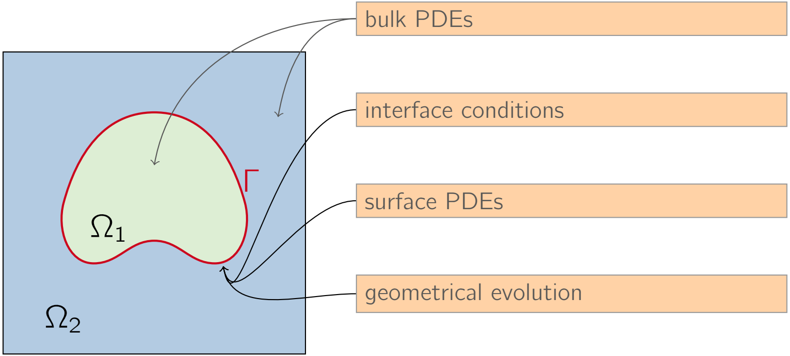




Section 1

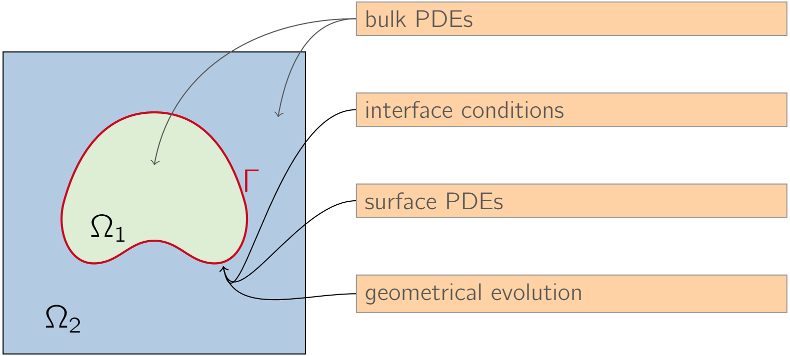
Motivation

Considered problems



- **Solution of one of the above problems (stationary)**

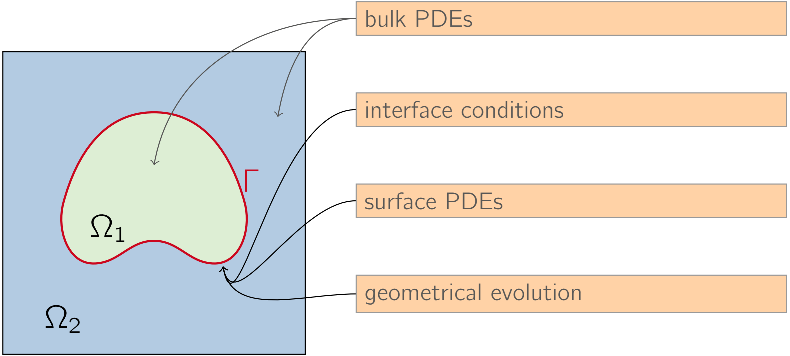
Considered problems



- **Solution of one of the above problems (stationary)**
- **Coupling of two or more of the above problems (stationary)**

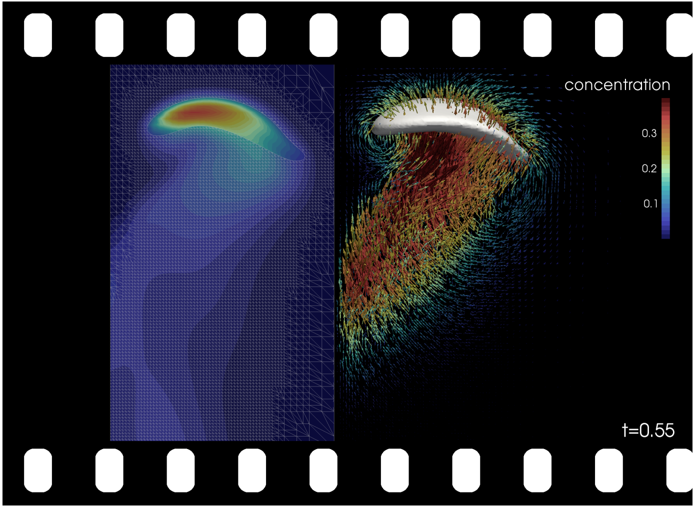
Considered problems

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737



- **Solution of one of the above problems** (stationary)
- **Coupling of two or more of the above problems** (stationary)
- **± Change of geometry configuration (e.g. physical motion or within optimization loop)**

Prototypical examples: Two-phase flows



Geometrically unfitted discretizations

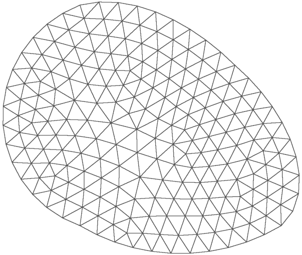
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

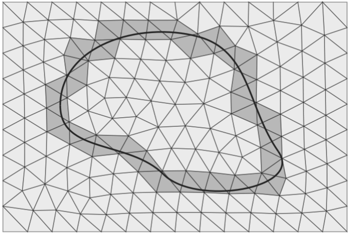
Geometrically fitted vs. unfitted discretizations

Above problems can be solved using **body-fitted** discretizations, but :

meshing, mesh tracking and re-meshing can become a burden

(complex geometries, large deformations, topology changes, ...)





Geometrically unfitted discretizations

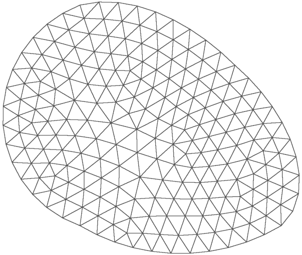
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Geometrically fitted vs. unfitted discretizations

Above problems can be solved using **body-fitted** discretizations, but :

meshing, mesh tracking and re-meshing can become a burden

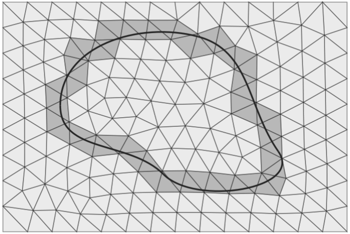
(complex geometries, large deformations, topology changes, ...)



Idea of (geometrically) unfitted discretizations

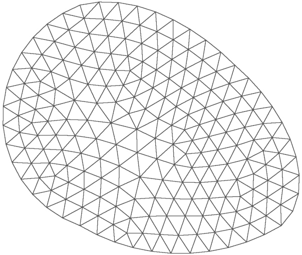
decouple mesh and geometry (e.g. level set)

⇝ **flexible geometry handling,**



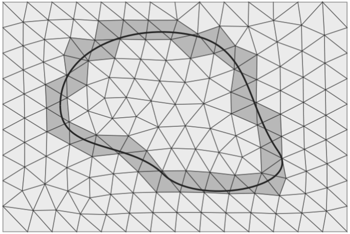
Geometrically fitted vs. unfitted discretizations

Above problems can be solved using **body-fitted** discretizations, but :
meshing, mesh tracking and re-meshing can become a burden
(complex geometries, large deformations, topology changes, ...)



Idea of (geometrically) unfitted discretizations

decouple mesh and geometry (e.g. level set)
$\rightsquigarrow$ **flexible geometry handling,**



No free lunch. New challenges:

- shape irregular cuts (stability, conditioning),
- numerical integration,
- imposition of boundary conditions,
- time integration,
- ...

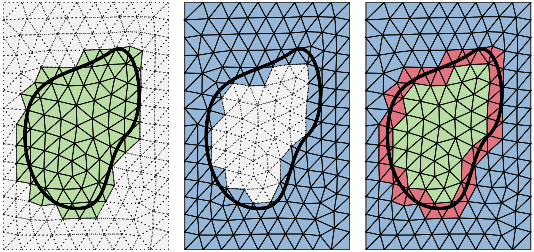
Types of methods

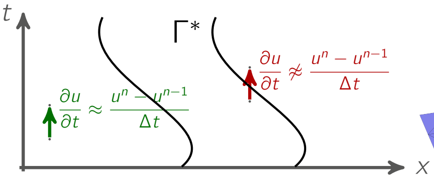
Unfitted FE spaces (based on bg. space V_h^{bg}):

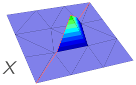
CutFEM: $V_h = V_h^{bg}|_\Omega$ or $V_h^{bg}|_{\mathcal{T}_h^{active}}$

XFEM: $V_h = V_h^{bg} \oplus V_h^x$

TraceFEM: $V_h = V_h^{bg}|_\Gamma$ or $V_h^{bg}|_{\mathcal{T}_h^{cut}}$







Types of methods

Unfitted FE spaces (based on bg. space V_h^{bg}):

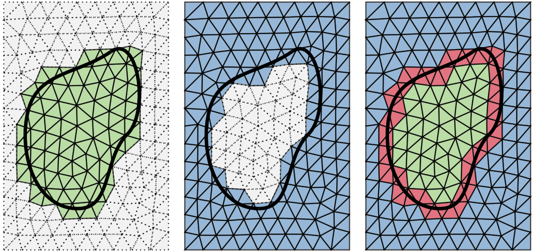
CutFEM: $V_h = V_h^{\text{bg}}|_\Omega$ or $V_h^{\text{bg}}|_{\mathcal{T}_h^{\text{active}}}$

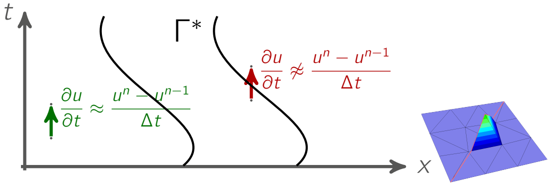
XFEM: $V_h = V_h^{\text{bg}} \oplus V_h^x$

TraceFEM: $V_h = V_h^{\text{bg}}|_\Gamma$ or $V_h^{\text{bg}}|_{\mathcal{T}_h^{\text{cut}}}$

Components of unfitted formulations:

- Nitsche's method, stabilized Lagrange multipliers, ...
- Ghost penalty stabilization / FEM aggregation
- Extension or Space-time based time integration





Types of methods

Unfitted FE spaces (based on bg. space V_h^{bg}):

CutFEM: $V_h = V_h^{bg}|_\Omega$ or $V_h^{bg}|_{\mathcal{T}_h^{active}}$

XFEM: $V_h = V_h^{bg} \oplus V_h^x$

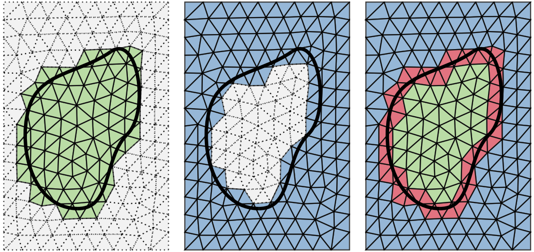
TraceFEM: $V_h = V_h^{bg}|_\Gamma$ or $V_h^{bg}|_{\mathcal{T}_h^{cut}}$

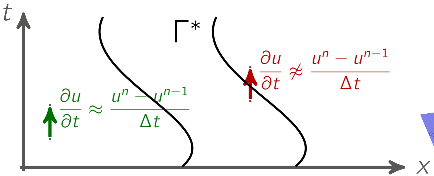
Components of unfitted formulations:

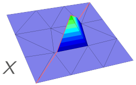
- Nitsche's method, stabilized Lagrange multipliers, ...
- Ghost penalty stabilization / FEM aggregation
- Extension or Space-time based time integration

Similar approaches, different names:

CutFEM, XFEM, TraceFEM, XDG, UDG, Finite Cell, Immersed FEM (≈), ...







What is ngsxfem?

ngsxfem ...

- ... aims at providing tools for **unfitted discretizations**
- ... is a C++ library for NGSolve ($\approx 14K$ lines of code)
- ... has a python interface
- ... is **open source** (LGPL), hosted on **github**, PyPi package (`pip install xfem`)

Dev. team: C. Lehrenfeld, J. Preuß (UCL), H. v. Wahl (U Jena), F. Heimann (U Göttingen), contributors: T. Ludescher, P. Stocker, T. v. Beeck, M. Zienecker

documentation, demos, material:

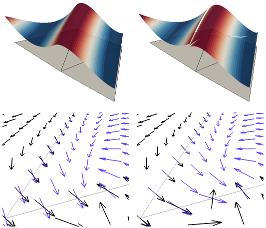
- short paper in JOSS https://joss.theoj.org/papers/10.21105/joss.03237 (2021)
- jupyter tutorials (unit-8.x in NGSolve) and on **/ngsxfem/ngsxfem-jupyter**
- python-demos in repo: **/ngsxfem/ngsxfem/demos**
- API and further documentation: https://lehrenfeld.pages.gwdg.de/ngsxfem/

Section 2

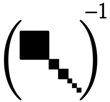
Features of `ngsxfem`

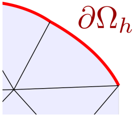
Standard `NGSolve` features

- Background finite element spaces
  (scalar/vector, continuous/discontinuous)
- Convenient integral forms with `DifferentialSymbols`
- Finite element assembly, static condensation
- Handling of dofs: `UNUSED_DOF`s
- Easy set up of preconditioners
- linear solvers
- **mesh deformation** supported (isoparametric FE)
- **Restriction of integration on set of elements**
- webgui, netgen gui, `VTK` output, ...
- ...



`ds(mesh.Boundaries(".."))`





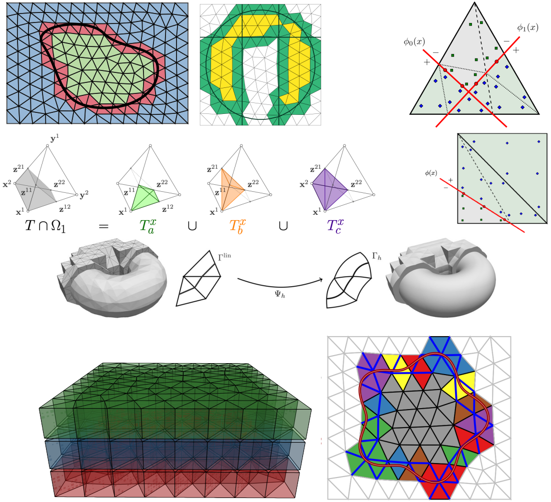

```
dx(definedonelements=...,
   deformation=...)
```

`ngsxfem` **features**

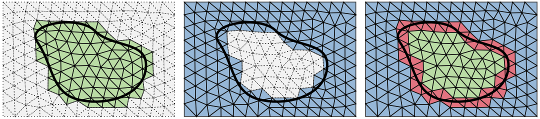
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

- **2.1** Tools to work on **submeshes**
- **2.2** **Higher order accurate numerical integration** on unfitted geometries described
  - **2.2a** by one level set function or
  - **2.2b** by multiple level sets
- **2.3** **Stability patches**, **FE Aggregation** and ghost penalties
- **2.4** **Space-Time FEM** for moving domain problems

(selection, not all details)



ngsxfem features: Working on submeshes



Submesh selection based on one level set fct.

BitArrays for elements:

- NEG elements ("inside", $\phi(x) < 0 \ \forall x \in T$),
- POS elements ("outside", $\phi(x) > 0 \ \forall x \in T$),
- IF elements ("cut", $\exists x \in T, \phi(x) = 0$),
- HASNEG: NEG or IF,
- ...

BitArrays for facets:

based on **neighbor** elements

```
from xfem import *
...

lsetp1 = GridFunction(H1(mesh,order=1))
InterpolateToP1(levelset,lsetp1)
...

ci = CutInfo(mesh, lsetp1)
...

hasneg = ci.GetElementsOfType(HASNEG)
hascut = ci.GetElementsOfType(IF)
...

facets
  = GetFacetsWithNeighborTypes(mesh,hasneg,hascut)
```

ngsxfem features: Working on submeshes

Restricting FESpaces to submesh (dofs)

- Compress[a]: based on dof-BitArray
- Restrict : based on element-BitArray

```
Vhbg = H1(mesh, order=1, dirichlet=[])
active_dofs = GetDofsOfElements(Vhbg,hasneg)
VhC = Compress(Vhbg,active_dofs)
# or
active_els = ci.GetElementsOfType(HASNEG)
VhR = Restrict(Vhbg,active_els)
```

Integrals on submesh[b]:

$$\sum_{T \in \mathcal{T}_h^*} \int_T f(x)\, dx, \quad \sum_{T \in \mathcal{T}_h^*} \int_{\partial T} f(x)\, dx, \ldots$$

```
u,v = VhR.TnT()
dxbar = dx(definedonelements=active_els)
integrator = u * v * dxbar
```

BilinearForms on submesh:

- Reserve only non-zero entries for **active elements** and **facets** (reduce dgjumps couplings)

```
a = RestrictedBilinearForm(VhR,
                           element_restriction=hasneg,
                           facet_restriction=facets)
```

[a] exists in NGSolve already

[b] exists in NGSolve already

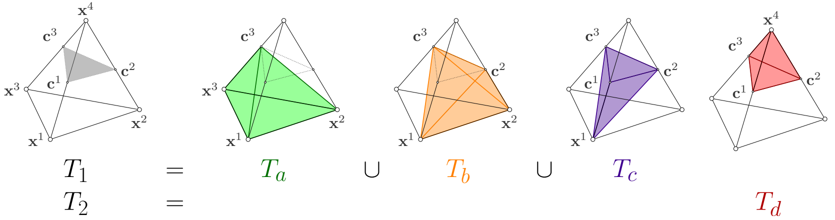
`ngsxfem` features: Numerical integration

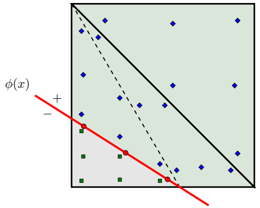
Unfitted formulations require numerical integration on **level set domains**:

$$\int_{\Omega} f(x)\, dx = \sum_{T \in \mathcal{T}_h^*} \int_{T \cap \Omega} f(x)\, dx$$

```
...

lsetp1 = GridFunction(H1(mesh,order=1))
InterpolateToP1(levelset,lsetp1)
dX = dCut(levelset=lsetp1,
          domain_type=NEG, order=order)
...
```

Description of **domain** through $\phi^{\text{lin}} \in \mathbb{P}^1(\mathcal{T}_h)$





Isoparametric unfitted FEM for stationary geometries

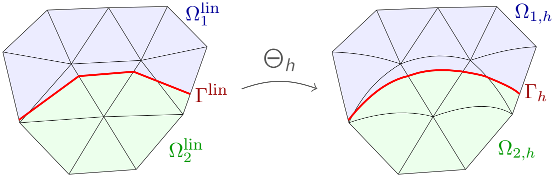
$\phi^{\text{lin}} \in \mathbb{P}^1 \rightsquigarrow$ geom. accuracy $\mathcal{O}(h^2)$

[1] C.L., High order unfitted finite element methods on level set domains using isoparametric mappings. CMAME, 2016

$\phi^{\text{lin}} \in \mathbb{P}^1 \rightsquigarrow$ geom. accuracy $\mathcal{O}(h^2)$

Idea isoparametric unfitted FEM:

Apply a mesh deformation



with Θ_h a FE function such that

$$\phi^{\text{lin}} \approx \phi \circ \Theta_h$$

```
from xfem import *
from xfem.lsetcurv import *
...

lsetMA = LevelSetMeshAdaptation(mesh,order=2)
deform = lsetMA.CalcDeformation(levelset)
lsetp1 = lsetMA.lset_p1
...

ds = dCut(levelset=lsetp1, domain_type=IF,
          definedonelements=hascut,
          deformation=deform)
...
```

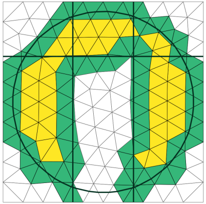
[1] C.L., High order unfitted finite element methods on level set domains using isoparametric mappings. CMAME, 2016

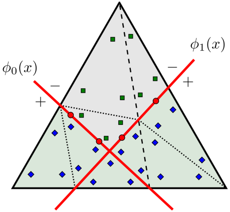
Geometries described by multiple level sets

Multiple level sets, unions and intersections:

Example:

$$\Omega = \{\phi_1 < 0\} \cap \{\phi_2 < 0\} \cap \{\phi_3 < 0\} \cap \{\phi_4 < 0\}$$





Cut integration is setup consecutively:

- subdivide element according to ϕ_0
- take new subelement, divide according to ϕ_1
- ...

```
from xfem import *
from xfem.mlset import *
...
# list of P1 fct.
lsetp1 = [lsetp1a, lsetp1b, lsetp1c, lsetp1d]
mlci = MultiLevelsetCutInfo(mesh, lsetp1)

sq = DomainTypeArray((NEG, NEG, NEG, NEG))
bnd = sq.Boundary()

hasneg = mlci.GetElementsWithContribution(sq)
hascut = mlci.GetElementsWithContribution(bnd)

dX = dCut(lsetp1, sq, definedonelements=hasneg)
dS = dCut(lsetp1, bnd, definedonelements=hascut)
```

ngsxfem features: Stability patches and FE aggregation

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Idea ghost penalties:

Borrow stability on (bad) cut elements from good/uncut **neighbor elements** by adding integrals of the form

$$\sum_{F \in \mathcal{F}_h^*} \gamma \int_{\omega_F} [\![u]\!]_\omega [\![v]\!]_\omega \, dx$$

where ω_F : facet patch, $[\![u]\!]_\omega = \mathcal{E}^{\mathbb{P}} u|_{T_1} - \mathcal{E}^{\mathbb{P}} u|_{T_2}$.

```
from xfem import *
...

dw = dFacetPatch(definedonelements=facets)
a += (u - u.Other())*(v - v.Other())*dw
...
```

Ghost penalties are an established technique in unfitted FEM to re-enable tools for stability (inverse inequalities, ...).

Alternative formuations exist.

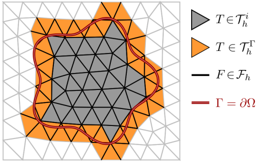
ngsxfem features: Stability patches and FE aggregation

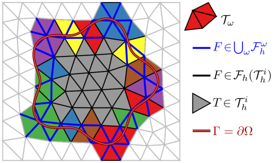
Stability patches:

Group every **(bad) cut** element into a patch with a **(good) uncut** element. Purposes:

- GP stabilization only within stability patches
- **Solve patchwise** problems

```
roots = ci.GetElementsOfType(NEG)
bads = ci.GetElementsOfType(IF)
EA = ElementAggregation(mesh, roots, bads)
...
gfu.vec.data = PatchwiseSolve(EA,Vh,lhs,rhs)
```






```
...
a = BilinearForm(VhR) # includes also "bad" dofs
...
E = AggEmbedding(EA, VhR)
ET = E.CreateTranspose()
A = ET @ a.mat @ E # sparse matrix mult
gfu.vec.data = A.Inverse() * (PT * f.vec)
```

ngsxfem features: Stability patches and FE aggregation

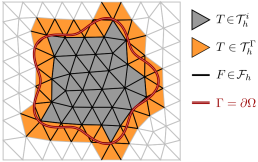
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

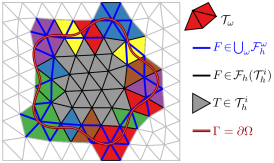
Stability patches:

Group every **(bad) cut** element into a patch with a **(good) uncut** element. Purposes:

- GP stabilization only within stability patches
- **Solve patchwise** problems
- or apply **FE aggregation**

```
roots = ci.GetElementsOfType(NEG)
bads = ci.GetElementsOfType(IF)
EA = ElementAggregation(mesh, roots, bads)
...
gfu.vec.data = PatchwiseSolve(EA,Vh,lhs,rhs)
```

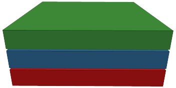




Idea FE aggregation

Discard `dofs` with degenerated support (only on cut elements) and bound them to interior `dofs` as smooth **extensions**.

```
...
a = BilinearForm(VhR) # includes also "bad" dofs
...
E = AggEmbedding(EA, VhR)
ET = E.CreateTranspose()
A = ET @ a.mat @ E # sparse matrix mult
gfu.vec.data = A.Inverse() * (PT * f.vec)
```



Space-Time FE discretization in ngsxfem:

- **Tensor product FE spaces** (per slab): $W_h = V_h \otimes \mathcal{P}_k([t_{n-1}, t_n])$
- We pull back to ref. time interval $[0, 1]$
- $\hat{Q} = \Omega \times [0, 1] \mapsto Q = \Omega \times [t_{n-1}, t_n]$

Example: **DG**-in-time for heat equation:

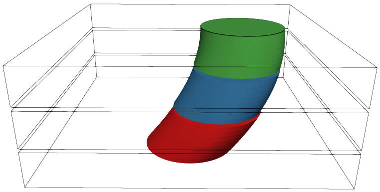
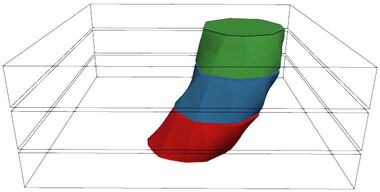
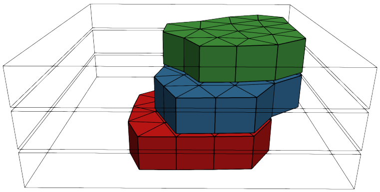
Find $u \in W_h$ such that for all $v \in W_h$ there is

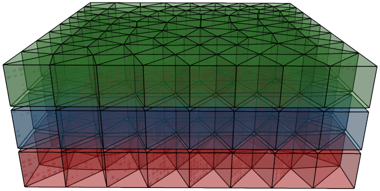
$$\int_Q \partial_t u v + \nabla u \cdot \nabla v d(x,t) + \int_\Omega u_-(\cdot, t_{n-1}) v_-(\cdot, t_{n-1}) dx$$

$$= \int_Q f v d(x,t) + \int_\Omega u_+(\cdot, t_{n-1}) v_-(\cdot, t_{n-1}) dx$$

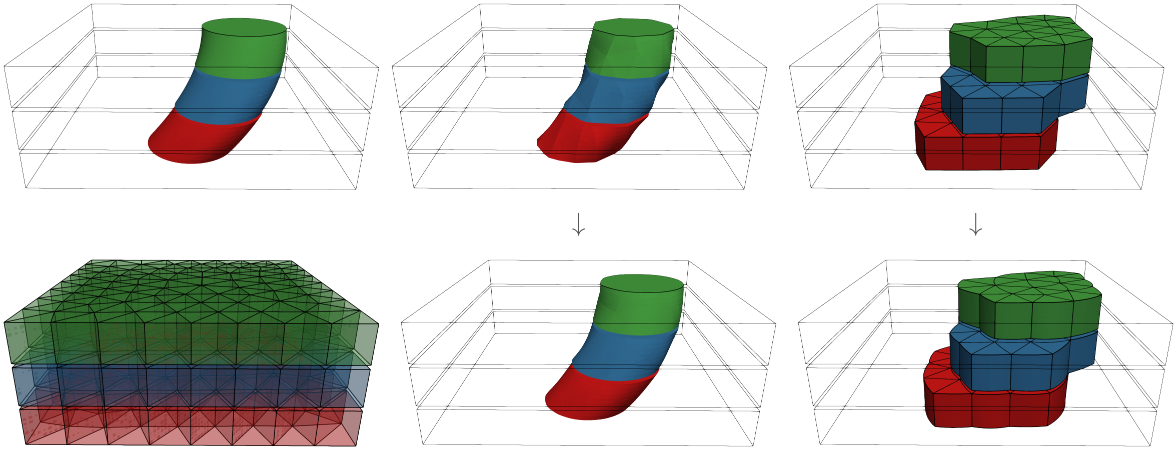
```
tfe = ScalarTimeFE(k_t)
Wh = tfe * Vh
gfu = GridFunction(Wh)
...

dxt = tau * dxtref(mesh, time_order=2)
dt = lambda u: 1.0 / tau * dtref(u)
...

a = BilinearForm(Wh, symmetric=False)
a += (dt(u) * v + grad(u) * grad(v)) * dxt
a += u * v * dmesh(mesh, tref=0)
f = LinearForm(Wh)
f += f * v * dxt
f += u_last * v *dmesh(mesh, tref=0)
```

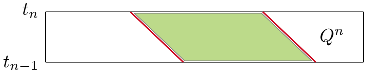
ngsxfem features: Space-Time FEM (unfitted)



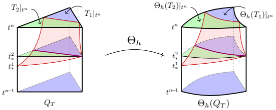
ngsxfem features: Space-Time FEM (unfitted)





Unfitted Space-Time FE discretization in ngsxfem:

- level set approx. $\phi^{\text{lin}} \in \mathbb{P}^1 \otimes \mathbb{P}^k$ as reference
- CutInfo and LevelSetMeshAdaptation $\rightsquigarrow$ space-time
- dFacetPatch can do space-time
- Mesh deformation in space-time version




```
from xfem.lset_spacetime import *
lsetMA_ST = LevelSetMeshAdaptation_Spacetime
lsetadap = lsetMA_ST(mesh,k_s,q_t)
# lset depends on space and time
lsetadap.CalcDeformation(lset)
lsetp1 = lsetadap.levelsetp1 # space-time
deform = lsetadap.deformation # space-time
ci = CutInfo(mesh, time_order=0)
ci.Update(lsetp1[INTERVAL], time_order=0)
dQ = tau*dCut(lsetp1[INTERVAL], NEG,
              time_order=time_order,
              deformation=deform[INTERVAL],
              definedonelements=hasneg)
```

Section 3

Where `ngsxfem` is used

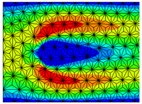
Literature using `ngsxfem`

Fictitious domain

C. Lehrenfeld *"A higher order isoparametric fict. domain method for level set domains"* in "Geom. Unf. [FEM] & Applic."(book ch.; 2018)

A. Aretaki, E. N. Karatzas, and G. Katsouleas, *"Equal Higher Order Analysis of an Unf. [DG] Method for Stokes..."* J. Sci. Comput. (2022)

H. Liu *"Unf. [FEM] for .. Stokes .. using .. Scott-Vogelius .."*, PhD thesis, Pittsburgh (2022)



C. Lehrenfeld, T. van Beeck, I. Voulis, *"Analysis of divergence-preserving unfitted [FEM] for the mixed Poisson problem"*, arXiv:2306.12722 (2023)

Interface problems

C. Lehrenfeld, A. Reusken *"Optimal preconditioners for Nitsche-XFEM .. of interface problems"*, Numer. Math. (2016)

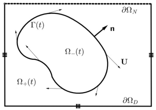
C. Lehrenfeld *"High order unf. [FEM] on level set domains using isop[..] mappings"*, CMAME (2016)

C. Lehrenfeld *"Removing the stabilization parameter in fitted and unfitted symmetric Nitsche formulations"*, ECCOMAS Proc. (2016)

C. Lehrenfeld, A. Reusken *"L^2-error analysis of an isop[..] unf. [FEM] for elliptic interface problems"*, J. Numer. Math. (2019)

P.Lederer, C.Pfeiler, C.Wintersteiger, C.Lehrenfeld *"Higher order unfitted FEM for Stokes [..]"*, PAMM (2016)

M. Olshanskii, A.Quaini, Q. Sun *"An unfitted [FEM] for two-phase Stokes problems with slip ..."*, J. Sci. Comp. (2021)

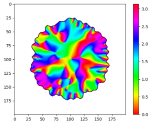


M. Olshanskii, A.Quaini, Q. Sun *"A [FEM] for two-phase flow with material viscous interface"*, CMAM (2021)

T. Ludescher *"Multilevel preconditioning of stabilized unfitted finite element discretizations"*, PhD thesis, RWTH Aachen (2020)

S. Groß, A. Reusken *"Analysis of optimal Preconditioners for CutFEM"*, Num. Lin. Alg. w. Appl. (2022)

J. Smith-Roberge *"Microcolony Dynamics: Motion from Growth, Order, and Incompressibility"*, PhD thesis, Univ. Waterloo (2023)

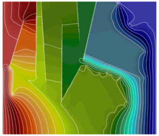


Literature using `ngsxfem` (cont'd)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

fractured porous media

G. Fu, Y. Yang, *A [HDG] method on unfitted meshes for [..] Darcy flow in fractured porous media*, Adv. Water Resources (2023)

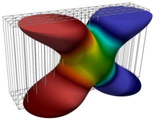


Space-time FEM (fitted)

G. Fu, Z. Xu *"High-order space-time [FEM] for the Poisson-Nernst-Planck equations: Positivity and unconditional energy stability"*, CMAME (2022)

Moving domains; space-time

F. Heimann, C. Lehrenfeld, J. Preuß, *"Geometrically higher order unfitted space-time methods for PDEs on moving domains"*, SISC (2023)

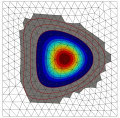


F. Heimann, C. Lehrenfeld, *" Geometrically higher order unfitted space-time methods for pdes on moving domains: Geometry error analysis"*, arXiv (2023)

F. Heimann, *"A higher order unfitted space-time finite element method for coupled surface-bulk problems"*, arXiv (2024)

Moving domains; time-stepping

Y. Lou, C. Lehrenfeld, *"Isoparametric unfitted BDF – finite element method for PDEs on evolving domains"*, SINUM (2022)

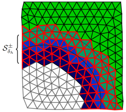


C. Lehrenfeld, M. A. Olshanskii *"An Eulerian [FEM] for PDEs in time-dependent domains"*, ESAIM:M2AN (2019)

H. von Wahl, T. Richter, C. Lehrenfeld *"An unfitted Eulerian [FEM] for the time-dependent Stokes problem on moving domains"*, IMAJNA (2021)

H. von Wahl, T. Richter, *"An Eulerian time-stepping scheme for a coupled parabolic moving domain problem using equal order unfitted finite elements"*, PAMM (2023)

H. von Wahl, T. Richter *"Error analysis for a parabolic PDE model problem on a coupled moving domain in a fully Eulerian framework"*, SINUM (2023)



M. Olshanskii, H. von Wahl *"A conservative Eulerian finite element method for transport and diffusion in moving domains"*, arXiv (2024)

Literature using ngsxfem (cont'd)

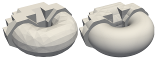
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Surface PDEs

E. Bachini, P. Brandner, T. Jankuhn, M. Nestler, S. Praetorius, A. Reusken, A. Voigt, *"Diffusion of tangential tensor fields: numerical issues and influence of geometric properties"*, arXiv:2205.12581

P. Brandner, T. Jankuhn, S. Praetorius, A. Reusken, A. Voigt, *"Finite element discretization methods for velocity-pressure and stream function formulations of surface Stokes equations"*, SISC (2022)

J. Grande, C. Lehrenfeld, A. Reusken, *"Analysis of a High-Order Trace [FEM] for PDEs on Level Set Surfaces"*, SINUM (2018)



T. Jankuhn, A. Reusken, *"Trace [FEM] for surface vector-Laplace equations"*, IMAJNA (2019)

A. Reusken, *"Analysis of finite element methods for surface vector-Laplace eigenproblems"*, Math. Comp. (2022)

P. Brandner, A. Reusken, *"Finite element error analysis of surface Stokes equations in stream function formulation"*, ESAIM:M2AN (2020)

H. Sass, A. Reusken, *"An Accurate and Robust Eulerian Finite Element Method for Partial Differential Equations on Evolving Surfaces"*, Comp. & Math., 2023

P. Brandner, *"Numerical Methods for Surface Navier-Stokes Equations in Stream Function Formulation"*, PhD thesis, RWTH Aachen (2022)

S. Lu, X. Xu, *"A Geometrically Consistent Trace Finite Element Method For The Laplace-Beltrami Eigenvalue Problem"*, arXiv:2108.02434

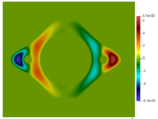
M. Olshanskii, A. Reusken, P. Schwering, *"An Eulerian Finite Element Method for Tangential Navier-Stokes Equations on Evolving Surfaces"*, Math. Comp., 2023

Literature using `ngsxfem` (cont'd)

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

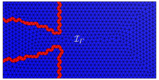
Model order reduction

A. Aretaki, E. N. Karatzas, *Random geom[..] for optimal control PDE problems based on … cut elements*, J. Comp. Appl. Math. (2022)



(c) Computed control u^h

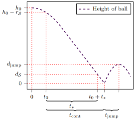
E. N. Karatzas, M. Nonino, F. Ballarin, G. Rozza *A Reduced Order Cut [FEM] for geometrically parameterized … Navier-Stokes problems*, CAMWA (2022)



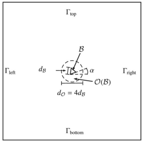
G. Katsouleas, E. N. Karatzas, F. Travlopanos *Discrete empirical interpolation and unfitted mesh FEMs: application in PDE-constrained optimization*, Optimization (2022)

Fluid structure interaction

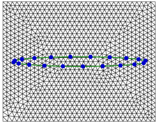
H. von Wahl, T. Richter, S. Frei, T. Hagemeier, *"Falling balls in a viscous fluid with contact: Comparing numerical simulations with experimental data"*, Phys. Fluids (2021)



H. von Wahl, T. Richter, *"Using a deep neural network to predict the motion of under-resolved triangular rigid bodies in an incompressible flow"*, IJNMF (2021)

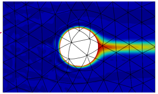


H. von Wahl, T. Wick, *"A high-precision framework for phase-field fracture interface reconstr. with appl. to Stokes fluid-filled fracture …"*, CMA-ME (2023)



Trefftz

F.Heimann, C.Lehrenfeld, P.Stocker, H.v.Wahl *"Unf. Trefftz [DG] methods for elliptic [bvp].*
,
ESAIM:M2AN (2023)



Thanks for your attention!

Time for jupyter (?)

More at the hands-on-session

Further material:

- NGSolve i-tutorials unit-8.x
- github.com/ngsxfem/ngsxfem-jupyter
- github.com/ngsxfem/ngsxfem $\rightsquigarrow$ `demos/...`

Feature matrix of ngsxfem

Features ⇲	CFE	XFE	DG	Iso	MLS	ST	Gh	Ag	Hex	Tet	MPI
CFE : CutFEM form.	/	/	y	y	y	y	y	y	y	y	y
XFE : XFEM formulation	/	/	y	y	n	n	y	n	y	y	y
DG : Discont. Galerkin	y	y	/	y	n	y	y	y	y	y	n
Iso : isoparametric map	y	y	y	/	n	y	y	y	y	y	y
MLS : multiple level set	y	n	n	n	/	n	y	y	n	y	y
ST : space-time FEM	y	n	y	y	n	/	y	n	y	y	y
Gh : Ghost penalty	y	y	y	y	y	y	/	/	y	y	n
Ag : Agg. FEM	y	n	y	y	y	n	/	/	y	y	n
Hex : quads / hexes	y	y	y	y	n	y	y	y	/	/	y
Tet : trigs./tets	y	y	y	y	y	y	y	y	/	/	y
MPI : MPI	y	y	n	y	y	y	n	n	y	y	/