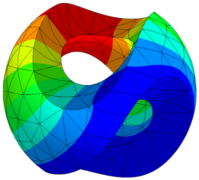
NGSTrefftz: Add-on to NGSolve for Trefftz methods

Paul Stocker

Georg-August-Universität, Göttingen, Germany

NGSolve User Meeting 2023, Portland

UNIVERSITÄT
GÖTTINGEN



Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

- Trefftz-DG methods on unfitted geometries

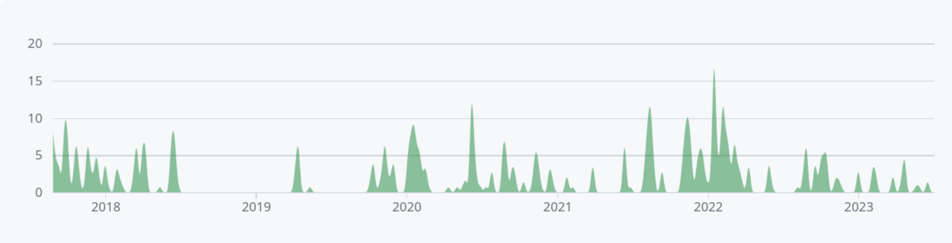
The ngstrefftz package

- Started in 2018 for Trefftz-DG methods for the acoustic wave equation
- pip install available on linux and mac

```
pip install ngstrefftz
```

- source code available on github

```
https://github.com/PaulSt/NGSTrefftz
```



Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

- Trefftz-DG methods on unfitted geometries

(polynomial) Trefftz-DG methods

$$\begin{cases} \mathcal{L}u = 0 \text{ in } \Omega, \\ + \text{ bndc. on } \partial\Omega, \end{cases}$$

$$\text{Find } u_h \in \mathbb{T}^p(\mathcal{T}_h), \quad \text{s.t. } a_h^{\mathsf{DG}}(u_h, v_h) = \ell_h^{\mathsf{DG}}(v_h) \qquad \forall v_h \in \mathbb{T}^p(\mathcal{T}_h) \text{ with}$$

$$\mathbb{T}^p(\mathcal{T}_h) := \Pi_{K\in\mathcal{T}_h} \mathbb{T}^p(K), \quad \mathbb{T}^p(K) := \{v_h \in \mathbb{P}^p(K) \text{ s.t. } \mathcal{L}u = 0 \text{ on } K\}$$

$$\dim \mathbb{T}^p(\mathcal{T}_h) = \mathcal{O}(p^{n-1}) \ll \dim \mathbb{P}^p(\mathcal{T}_h) = \mathcal{O}(p^n)$$

Assumption: Trefftz space is 'rich enough' ($\mathcal{L} = \sum_{l=1}^{d} \alpha_l \partial_{x_l}^{\beta_l}$ for $\alpha_l \in \mathbb{R}$ and $\beta_l \in \mathbb{N}$)

E. Trefftz, *Ein Gegenstück zum Ritzschen Verfahren*, Proc. 2nd Int. Cong. Appl. Mech., Zurich, 1926

Example: Laplace equation

$$\mathcal{L}u = -\Delta u = 0 \text{ in } \Omega, \quad u = g \text{ on } \partial\Omega.$$

$$\mathbb{T}^p(K) = \{1, x, y, xy, x^2 - y^2, x^3 - 3xy^2, \dots\}$$

$$a_h(u,v) = \sum_K \int_K \nabla u \nabla v \, dx + \sum_{F \in \mathcal{F}_h^{\text{int}}} \int_F \overbrace{-\{\!\{\partial_{\mathbf{n}} u\}\!\}[\![v]\!]}^{\text{consistency}} \; \overbrace{-\{\!\{\partial_{\mathbf{n}} v\}\!\}[\![u]\!]}^{\text{symmetry}} \; \overbrace{+ \, \alpha p^2 h^{-1} [\![u]\!][\![v]\!]}^{\text{stability}} \, ds + \text{bnd}$$

$\{\!\{\cdot\}\!\}$: average across facets, $[\![\cdot]\!]$: jump across facets. $\rightsquigarrow$ communication between neighbors.

F. Li, C.-W. Shu, *A local-structure-preserving local discontinuous Galerkin method for the Laplace equation*, Methods Appl. Anal., 2006

R. Hiptmair, A. Moiola, I. Perugia, C. Schwab, *Approximation by harmonic polynomials [...] of Trefftz hp-dGFEM*, ESAIM Math. Model. Num. Anal., 2014

O. Cessenat, B. Despres *Application of an Ultra Weak Variational Formulation of Elliptic PDES to the Two-Dimensional Helmholtz Problem* SIAM J. Num. Anal. 2015

TrefftzFESpace

In `python`:

```python
from ngstrefftz import *
fes = trefftzfespace(mesh,order=order,eq="laplace",dgjumps=True)
```

- `mesh`: `ngs.Mesh` object
- `order`: polynomial order p
- `eq`: equation to solve, e.g. `"laplace"`, `"helmholtz"`, `"wave"`
- `dgjumps`: include DG jumps in bilinear form (always True)

TrefftzFESpace: Under the hood

On the C++ side:

```
ScalarMappedElement (int ndof, int ord, CSR localmat,
                     ELEMENT_TYPE eltype, Vec<D> elcenter = 0,
                     double elsize = 1, double c = 1.0)
```

- extends NGSolve `FiniteElement` class
- evaluation of basis functions on the physical element
- `localmat`: coefficients of the Trefftz polynomial basis

Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

- Trefftz-DG methods on unfitted geometries

Quasi-Trefftz DG for the acoustic wave equation

- Extension of the Trefftz-DG method to smooth coefficients
- Polynomial basis functions are elementwise approximate PDE solutions
- We consider the problem

$$-\Delta u + G(x)\partial_{tt} u = 0 \quad \text{with} \quad G(x) := \sum_{\ell=0}^{\infty} \gamma_\ell (\boldsymbol{x} - \boldsymbol{x}_K)^\ell$$

Definition (Quasi-Trefftz space)

$$\mathbb{QT}^p(K) := \left\{ f \in \mathbb{P}^p(K) \mid D^{\boldsymbol{i}}(-\Delta + G(x)\partial_{tt}) f(\mathbf{x}_K, t_K) = 0,\ |\boldsymbol{i}| < p-1 \right\}$$

L.M. Imbert-Gérard, A. Moiola, PS, *A space-time quasi-Trefftz DG method for the wave eq. with piecewise-smooth coefficients*, Math. Comp., 2020

L.M. Imbert-Gérard, B. Despres, *A generalized plane wave numerical method for smooth non constant coefficients*, IMA J. Num. Anal., 2014

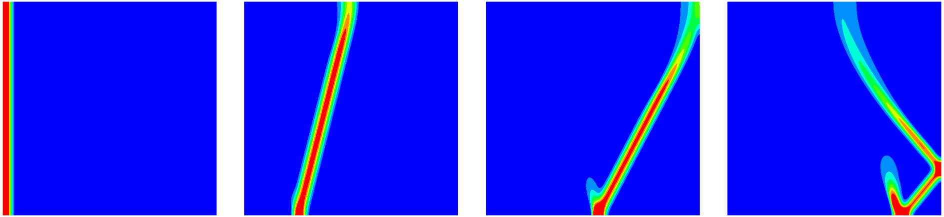
J. Yang, M. Potier-Ferry, K. Akpama, H. Hu, Y. Koutsawa, H. Tian, and D. S. Zézé, *Trefftz methods and Taylor series*, Arch. Comput. Methods Eng., 2020

Quasi-Trefftz DG

In `python`:

```
fes = trefftzfespace(mesh,order=order,eq="qtwave",dgjumps=True)
fes.SetCoeff(y+1)
```

- eq: for the acoustic wave equation use `"qtwave"`
- `fes.SetCoeff`: set the coefficient $G(x)$



Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

- Trefftz-DG methods on unfitted geometries

Embedded Trefftz-DG method

- Goal: Represent Trefftz basis $\{\psi_j\}_M$ in the standard DG basis $\{\phi_i\}_N$:

$$\psi_j = \sum_{j=1}^{N} \mathbf{T}_{ij}\phi_i, \; j = 1, .., M, \text{ for } \mathbf{T} \in \mathbb{R}^{N\times M}.$$

Then instead of solving $\mathbf{A}\mathbf{u}_{\mathbb{P}} = \mathbf{b}$ we solve $\mathbf{T}^T\mathbf{A}\mathbf{T}\,\mathbf{u}_{\mathbb{T}} = \mathbf{T}^T\mathbf{b}$ where $\mathbf{T}^T\mathbf{A}\mathbf{T} \in \mathbb{R}^{M\times M}$ with

$$\dim \mathbb{T}^p(\mathcal{T}_h) = M \ll N = \dim \mathbb{P}^p(\mathcal{T}_h)$$

Christoph Lehrenfeld, PS, *Embedded Trefftz discontinuous Galerkin methods*, Int. J. Numer. Methods Eng., 2022

Embedded Trefftz-DG method

- Goal: Represent Trefftz basis $\{\psi_j\}_M$ in the standard DG basis $\{\phi_i\}_N$:

$$\psi_j = \sum_{j=1}^{N} \mathbf{T}_{ij}\phi_i, \; j = 1, .., M, \text{ for } \mathbf{T} \in \mathbb{R}^{N\times M}.$$

Then instead of solving $\mathbf{A}\mathbf{u}_{\mathbb{P}} = \mathbf{b}$ we solve $\mathbf{T}^T\mathbf{A}\mathbf{T}\;\mathbf{u}_{\mathbb{T}} = \mathbf{T}^T\mathbf{b}$ where $\mathbf{T}^T\mathbf{A}\mathbf{T} \in \mathbb{R}^{M\times M}$ with

$$\dim \mathbb{T}^p(\mathcal{T}_h) = M \ll N = \dim \mathbb{P}^p(\mathcal{T}_h)$$

- Recipe: Find u for which $\|\mathcal{L}u\|_{0,h} = 0 \Leftrightarrow \langle \mathcal{L}u, \mathcal{L}v\rangle_{0,h} = 0, \; \forall v \in V_h$

$$\mathbf{T} = \ker(\mathbf{W}) \text{ with } \mathbf{W} = \sum_{K\in\mathcal{T}_h} \langle \mathcal{L}\phi_i, \mathcal{L}\phi_j\rangle_K$$

Christoph Lehrenfeld, PS, *Embedded Trefftz discontinuous Galerkin methods*, Int. J. Numer. Methods Eng., 2022

Embedded Trefftz - inhomogeneous problem

What to do for inhomogeneous problems $\mathcal{L}u = f$

On each element we can construct a **local particular solution** using the pseudo-inverse

$$\mathbf{W}^\dagger = \begin{pmatrix} | & & | & | & & | \\ \mathbf{v}_1 & \dots & \mathbf{v}_L & \mathbf{v}_{L+1} & \dots & \mathbf{v}_N \\ | & & | & | & & | \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sigma_1} & & & & \\ & \ddots & & & \\ & & \frac{1}{\sigma_L} & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{pmatrix} \cdot \begin{pmatrix} \text{---} & \mathbf{u}_1^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{u}_L^T & \text{---} \\ \text{---} & \mathbf{u}_{L+1}^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{u}_N^T & \text{---} \end{pmatrix}$$

For $u_{h,f}$ a particular solution, we are looking for a solution $u_{\mathbb{T}} \in \mathbb{T}^p(\mathcal{T}_h)$ that (uniquely) solves

$$a_h(u_{\mathbb{T}}, v_{\mathbb{T}}) = \ell(v_{\mathbb{T}}) - a_h(u_{h,f}, v_{\mathbb{T}}) \quad \forall\ v_{\mathbb{T}} \in \mathbb{T}^p(\mathcal{T}_h).$$

Embedded Trefftz - no polynomial Trefftz space

What to do for operators like $-\Delta \pm \mathrm{id}$, $-\operatorname{div}(\alpha(\mathbf{x})\nabla\cdot)$, ...

- Idea: Instead of $\langle \mathcal{L}u, \mathcal{L}v\rangle = 0,\ \forall v \in V_h$ we use the relaxed condition

$$\langle \mathcal{L}u, w\rangle_{0,h} = 0,\ \forall w \in W_h \subset V_h$$

  ($W_h := \mathcal{L}\mathbb{P}^p(\mathcal{T}_h)$ recovers the previous embedding)

- Introduce a **weak Trefftz space** for the embedding

$$\mathbb{T}^p(\mathcal{T}_h) = \{v \in \mathbb{P}^p(\mathcal{T}_h),\ \Pi_W \mathcal{L}v = 0 \text{ on each } K \in \mathcal{T}_h\}.$$

- Proceed with the embedding

$$\mathbf{T} = \ker(\mathbf{W}) \text{ with } \mathbf{W} = \sum_{K\in\mathcal{T}_h} \langle \mathcal{L}\phi_i, \psi_j\rangle_K, \quad \forall \psi_j \in W_h$$

Example code

function DG MATRIX
  $(\mathbf{A})_{ij} = a_h(\phi_j, \phi_i)$
  $(\mathbf{l})_i = \ell(\phi_i)$
for $K \in \mathcal{T}_h$ **do**
  $(\mathbf{W}_K)_{ij} = \langle \mathcal{L}\phi_j, \mathcal{L}\phi_i \rangle_{0,h}$
  $\mathbf{T}_K = \ker_h(\varepsilon; \mathbf{W}_K)$
  $(\mathbf{w}_K)_i = \langle f, \mathcal{L}\phi_i \rangle_{0,h}$
  $(\mathbf{u}_f)_K = \mathbf{W}_K^{\dagger} \mathbf{w}_K$
Solve $\mathbf{T}^T \mathbf{A} \mathbf{T}\, \mathbf{u}_{\mathbb{T}} = \mathbf{T}^T (\mathbf{l} - \mathbf{A}\mathbf{u}_f)$
$\mathbf{u}_h = \mathbf{T}\mathbf{u}_{\mathbb{T}} + \mathbf{u}_f$

```
fes = L2(mesh,order=order,dgjumps=True)
uh = GridFunction(fes)
a,f = dgscheme(fes)
u,v = fes.TnT()
W = L(u)*L(v)*dx
w = rhs*L(v)*dx
T, uf = TrefftzEmbedding(W,fes,eps,w)
Tt = T.CreateTranspose()
TA = Tt@a.mat@T
ut = TA.Inverse()*(Tt*(f.vec-a.mat*uf))
uh.vec.data = T*ut + uf
```

Example: Laplace equation

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega. \end{cases}$$

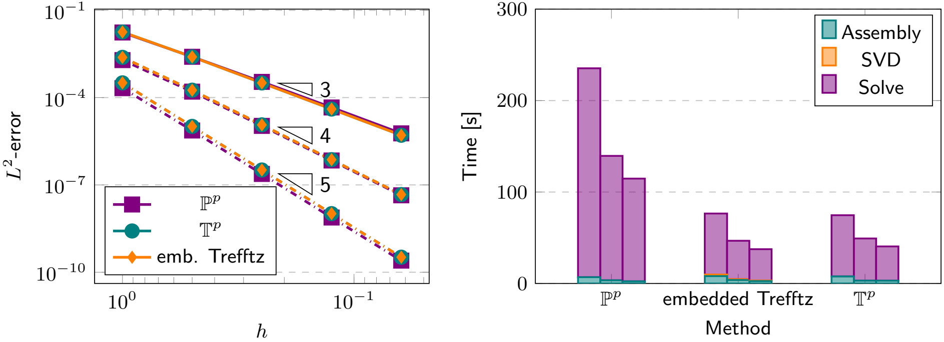


Figure: Results for the Laplace problem in 3 dimensions. Left: h-convergence. Right: Timings of the different steps of each method, for $p = 5$ on a fixed mesh with $h = 2^{-3}$. The bars from left to right correspond to computations using $4, 8, 12$ threads for each method.

Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

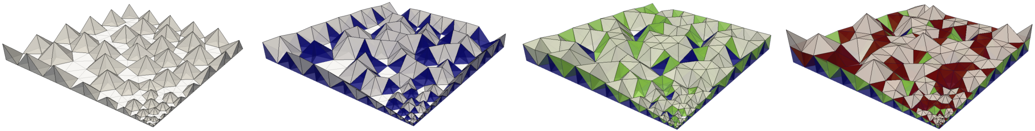
- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

- Trefftz-DG methods on unfitted geometries

Trefftz + ngstents

- space–time mesh using tent-shaped elements conforming to the cone of dependence
- allow to solve the equation elementwise with locally optimal advances in time
- independent tents can be solved in parallel



In python:

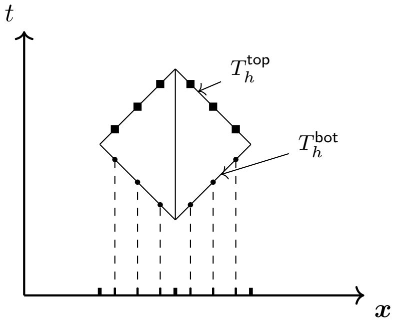
```
from ngstents._pytents import TentSlab
ts = TentSlab(mesh, method="edge")
ts.SetMaxWavespeed(wavespeed)
ts.PitchTents(dt=dt, local_ct=True)
```

J. Gopalakrishnan, J. Schöberl, C. Wintersteiger, *Mapped tent pitching schemes for hyperbolic systems*, SIAM J. Sci. Comp., 2017

https://github.com/jayggg/ngstents

Trefftz + ngstents

- (quasi-)Trefftz DG method for the acoustic wave equation on tent-pitching mesh
- only on facets of the mesh due to ultra-weak formulation



In python:

```
from ngstrefftz import TWave
TT=TWave(order,ts,CoefficientFunction(wavespeed))
TT.SetInitial(initc)
TT.SetBoundaryCF(bdd)
while t < 1.5:
    TT.Propagate()
```

I. Perugia, J. Schöberl, PS, C. Wintersteiger, *Tent pitching and Trefftz-DG method for the acoustic wave equation*, Comput. Math. Appl., 2020

Overview

Trefftz discontinuous Galerkin methods

- Trefftz spaces
- quasi-Trefftz methods (for the acoustic wave equation)
- embedded Trefftz methods

Trefftz + `ngstents`

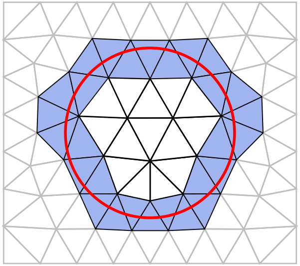
- space–time DG method for the acoustic wave equation on tent-pitching mesh

Trefftz + `ngsxfem`

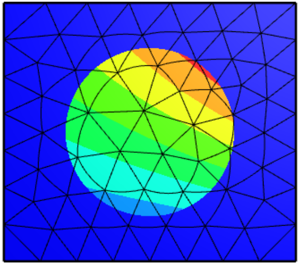
- Trefftz-DG methods on unfitted geometries

Trefftz + ngsxfem

- works with ngsxfem - a package for unfitted finite element discretizations
- Trefftz methods can be used with unfitted DG method in a natural way



→



In python:

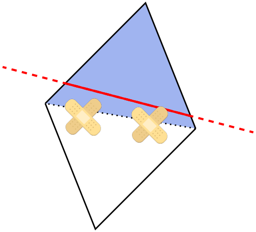
```
Vhbase = trefftzfespace(mesh,order=order, eq="laplace", dgjumps=True)
Vh = Restrict(Vhbase, els_hasneg)
```

Fabian Heimann, Christoph Lehrenfeld, PS, Henry von Wahl, *Unfitted Trefftz discontinuous Galerkin methods for elliptic boundary value problems*, arXiv:2212.12236, 2022

Trefftz + `ngsxfem`: Stability for small cuts

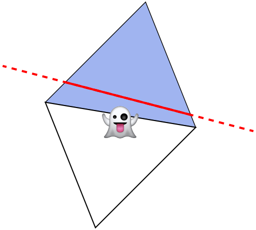
- cuts can lead to shape irregular elements require stabilization
- different stabilization techniques can be easily used with Trefftz methods

Element aggregation:



🩹 elements with small cuts are merged with interior elements

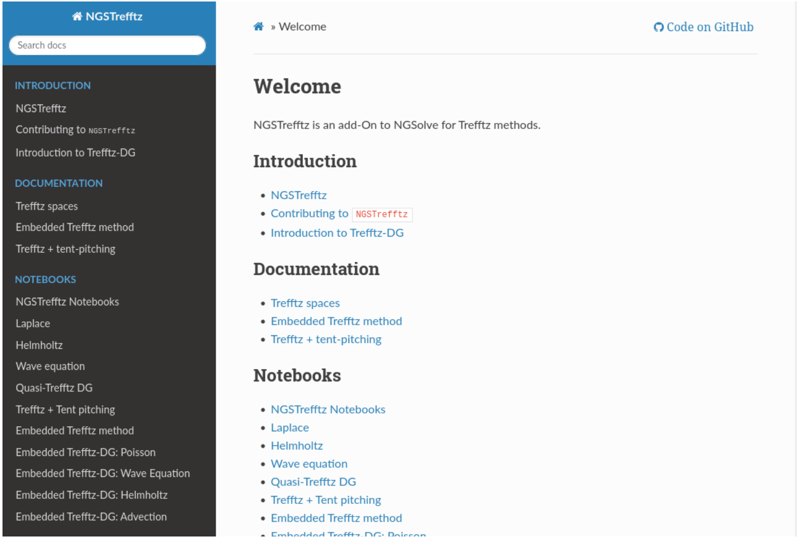
Ghost penalty:



$$👻(u,v) = \sum_{\ell=1}^{k} (h_F^{2\ell-1} [\![\partial_{\boldsymbol{n}_F}^{\ell} u]\!], [\![\partial_{\boldsymbol{n}_F}^{\ell} v]\!])_F$$

Fabian Heimann, Christoph Lehrenfeld, PS, Henry von Wahl, *Unfitted Trefftz discontinuous Galerkin methods for elliptic boundary value problems*, arXiv:2212.12236, 2022

Conclusion



paulst.github.io/NGSTrefftz

Conclusion

References

F. Heimann, C. Lehrenfeld, P. Stocker, and H. von Wahl.
Unfitted Trefftz discontinuous Galerkin methods for elliptic boundary value problems.
arXiv preprint arxiv:2212.12236, 2022.

L.-M. Imbert-Gérard, A. Moiola, and P. Stocker.
A space-time quasi-Trefftz DG method for the wave equation with piecewise-smooth coefficients.
arXiv preprint, arXiv:2011.04617, 2021.

P. L. Lederer, C. Lehrenfeld, and P. Stocker.
Trefftz discontinuous Galerkin discretization for the Stokes problem.
arXiv preprint arxiv:2306.14600, 2023.

C. Lehrenfeld and P. Stocker.
Embedded trefftz discontinuous galerkin methods.
International Journal for Numerical Methods in Engineering, n/a(n/a).

I. Perugia, J. Schöberl, P. Stocker, and C. Wintersteiger.
Tent pitching and Trefftz-DG method for the acoustic wave equation.
Comput. Math. Appl., 79(10):2987–3000, 2020.

P. Stocker.
NGSTrefftz: Add-on to NGSolve for Trefftz methods.
Journal of Open Source Software, 7(71):4135, 2022.

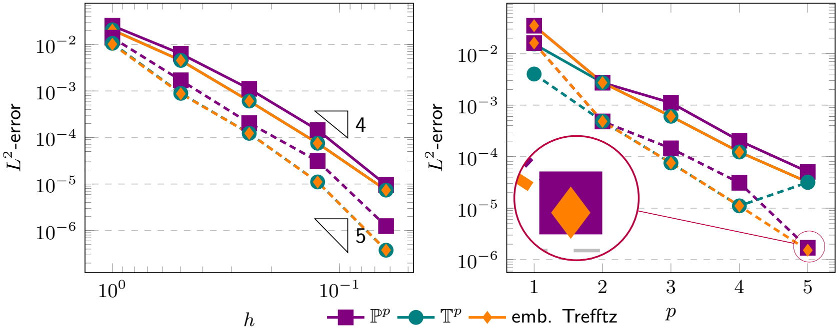
paulst.github.io/NGSTrefftz

Example: Helmholtz

$$\begin{cases} -\Delta u - \omega^2 u = 0 & \text{in } \Omega, \\ \frac{\partial u}{\partial n_{\mathbf{x}}} + iu = g & \text{on } \partial\Omega. \end{cases}$$

$$\mathbb{T}^p = \{e^{-i\omega(d_j \cdot \mathbf{x})} \text{ s.t. } j = 0, \ldots, k\}$$

$$\mathbb{ET}^p = \{v \in \mathbb{P}^p(\mathcal{T}_h),\ \Pi_{\mathbb{P}^{p-2}}(-\Delta - \omega^2)v = 0 \text{ on each } K \in \mathcal{T}_h\}$$



Algorithmic complexity: A rough comparison

- direct solver
- $N_{\text{el}} := \#\mathcal{T}_h \sim h^{-d}$
- p-scaling (no constants)

Costs:	Standard DG	Trefftz DG	Embedded Trefftz DG	Hybrid DG
Vector representation:				
total `ndofs` stored	$\sim N_{\text{el}} p^d$	$\sim N_{\text{el}} p^{d-1}$	$\sim N_{\text{el}} p^d$	$\sim N_{\text{el}} p^d$
globally coupled `ndofs`	$\sim N_{\text{el}} p^d$	$\sim N_{\text{el}} p^{d-1}$	$\sim N_{\text{el}} p^{d-1}$	$\sim N_{\text{el}} p^{d-1}$
Setup linear systems:				
`nnzes` $\mathbf{A}$	$\sim N_{\text{el}} p^{2d}$	$\sim N_{\text{el}} p^{2d-2}$	$\sim N_{\text{el}} p^{2d}$	$\sim N_{\text{el}} p^{2d}$
Additional costs:			Setup $\mathbf{T}$:	static cond.:
	—	—	$\sim N_{\text{el}} p^{3d}$	$\sim N_{\text{el}} p^{3d}$
Solving linear systems:				
global matrix	$\mathbf{A}$	$\mathbf{A}$	$\mathbf{T}^T \mathbf{A} \mathbf{T}$	$\mathbf{S}$
`nnzes`	$\sim N_{\text{el}} p^{2d}$	$\sim N_{\text{el}} p^{2d-2}$	$\sim N_{\text{el}} p^{2d-2}$	$\sim N_{\text{el}} p^{2d-2}$
arithmetic ops. $(\mathcal{O}(N^\alpha))$	$\sim (N_{\text{el}} p^{2d})^\alpha$	$\sim (N_{\text{el}} p^{2d-2})^\alpha$	$\sim (N_{\text{el}} p^{2d-2})^\alpha$	$\sim (N_{\text{el}} p^{2d-2})^\alpha$