Simple and sharp: Error estimates of Bank–Weiser type in the FEniCS Project

Jack S. Hale

Raphaël Bulle, Alexei Lozinski

Stéphane P. A. Bordas, Franz Chouly

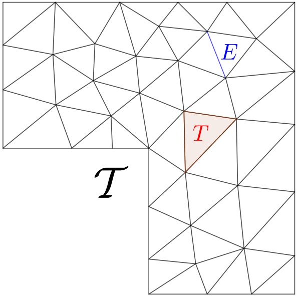
University of Luxembourg
Université de Bourgogne Franche-Comté

March 24, 2021

We would like to acknowledge the support of the ASSIST research project of the University of Luxembourg. This presentation has been prepared in the framework of the DRIVEN project funded by the European Union's Horizon 2020 Research and Innovation

- The problem.
- Estimates of Bank-Weiser type.
- Implementation.
- Results.

Problem setting



Find u_k in V^k such that

$$\int_\Omega \nabla u_k \cdot \nabla v_k = \int_\Omega f v_k \quad \forall v_k \in V^k. \tag{1}$$

Error

We quantify the discretization error $e := u_k - u$ using the energy norm $\eta_{\mathrm{err}} := \|\nabla e\|_\Omega = \|\nabla u_k - \nabla u\|_\Omega$.

Error

We quantify the discretization error $e := u_k - u$ using the energy norm $\eta_{\text{err}} := \|\nabla e\|_\Omega = \|\nabla u_k - \nabla u\|_\Omega$.

Goal: estimate η i.e. find a computable quantity η_{bw} such that

$$\eta_{\text{bw}} \approx \eta_{\text{err}}.$$

Contributions

- A high-level way of expressing Bank–Weiser type error estimators in DOLFIN and DOLFINx [Bank and Weiser, 1985].
- A simple dual-weighted error estimation and marking strategy originally proposed in [Becker et al., 2011].
- A proof of the reliability of the Bank–Weiser estimator in dimension three [Bulle et al., 2020].
- arXiv: `https://arxiv.org/abs/2102.04360`
- Code: `https://github.com/rbulle/fenics-error-estimation`

The Bank–Weiser Estimator

The restriction e_T of e to any cell T of the mesh satisfies the equation

$$\int_T \nabla e_T \cdot \nabla v_T := \int_T (f - \Delta u_k) v_T + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T \quad \forall v \in H^1_0(T).$$

The Bank–Weiser Estimator

The restriction e_T of e to any cell T of the mesh satisfies the equation

$$\int_T \nabla e_T \cdot \nabla v_T := \int_T (f - \Delta u_k) v_T + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T \quad \forall v \in H^1_0(T).$$

On a cell T, the Bank–Weiser problem is given by:
find e_T^{bw} in V_T^{bw} such that

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

The Bank–Weiser Estimator

The restriction e_T of e to any cell T of the mesh satisfies the equation

$$\int_T \nabla e_T \cdot \nabla v_T := \int_T (f - \Delta u_k) v_T + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T \quad \forall v \in H^1_0(T).$$

On a cell T, the Bank–Weiser problem is given by:
find e_T^{bw} in V_T^{bw} such that

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

The Bank–Weiser estimator is defined as

$$\eta_{\mathrm{bw}}^2 := \sum_{T \in \mathcal{T}} \eta_{\mathrm{bw},T}^2, \quad \eta_{\mathrm{bw},T} := \|\nabla e_T^{\mathrm{bw}}\|_T.$$

The space V_T^{bw}

The space V_T^{bw}

- Different definitions of V_T^{bw} lead to different variants of the estimator.

The space V_T^{bw}

- Different definitions of V_T^{bw} lead to different variants of the estimator.
- General principle: let $V_T^- \subsetneq V_T^+$ be two finite element spaces and

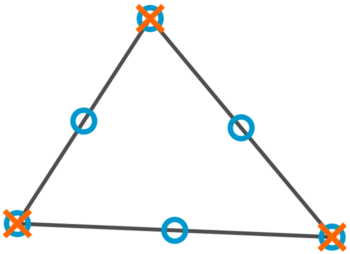
$$\mathcal{L}_T : V_T^+ \longrightarrow V_T^-,$$

be the local Lagrange interpolation operator,

$$V_T^{\mathrm{bw}} := \ker(\mathcal{L}_T) = \{v_T^+ \in V_T^+,\ \mathcal{L}_T(v_T^+) = 0\}.$$

Example

For $V_T^+ = V_T^2$ and $V_T^- = V_T^1$



V_T^2

V_T^1

Implementation

We need to compute the matrix A_T^{bw} and vector b_T^{bw} from

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

Implementation

We need to compute the matrix A_T^{bw} and vector b_T^{bw} from

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

Problem: the space V_T^{bw} is not provided by DOLFIN.

Implementation

We need to compute the matrix A_T^{bw} and vector b_T^{bw} from

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

Problem: the space V_T^{bw} is not provided by DOLFIN.
Idea: we rely on the matrix A_T^+ and vector b_T^+ from

$$\int_T \nabla e_T^+ \cdot \nabla v_T^+ = \int_T (f - \Delta u_k) v_T^+ + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^+ \quad \forall v_T^+ \in V_T^+,$$

since V_T^+ is provided by DOLFIN.

Implementation

We need to compute the matrix A_T^{bw} and vector b_T^{bw} from

$$\int_T \nabla e_T^{\mathrm{bw}} \cdot \nabla v_T^{\mathrm{bw}} = \int_T (f - \Delta u_k) v_T^{\mathrm{bw}} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{\mathrm{bw}} \quad \forall v_T^{\mathrm{bw}} \in V_T^{\mathrm{bw}}.$$

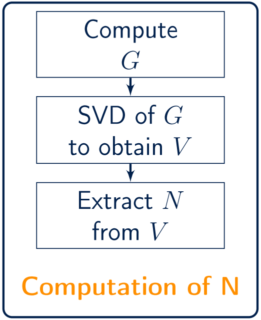
Problem: the space V_T^{bw} is not provided by DOLFIN.
Idea: we rely on the matrix A_T^+ and vector b_T^+ from

$$\int_T \nabla e_T^{+} \cdot \nabla v_T^{+} = \int_T (f - \Delta u_k) v_T^{+} + \sum_{E \in \partial T} \frac{1}{2} \int_E [\![\partial_n u_k]\!]_E \, v_T^{+} \quad \forall v_T^{+} \in V_T^{+},$$

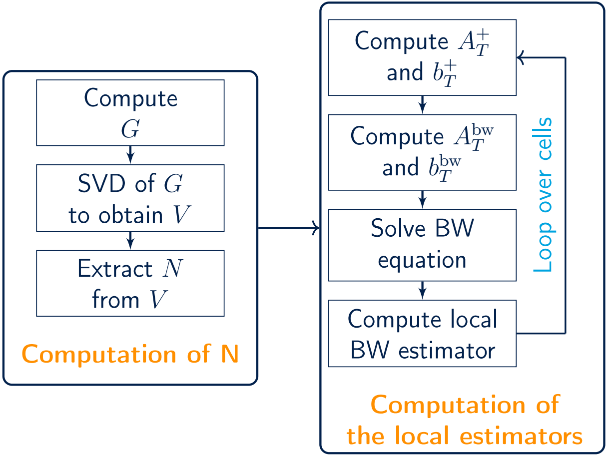
since V_T^+ is provided by DOLFIN. and we look for a matrix N such that:

$$A_T^{\mathrm{bw}} = N^{\mathsf{t}} A_T^+ N, \quad \text{and} \quad b_T^{\mathrm{bw}} = N^{\mathsf{t}} b_T^+.$$

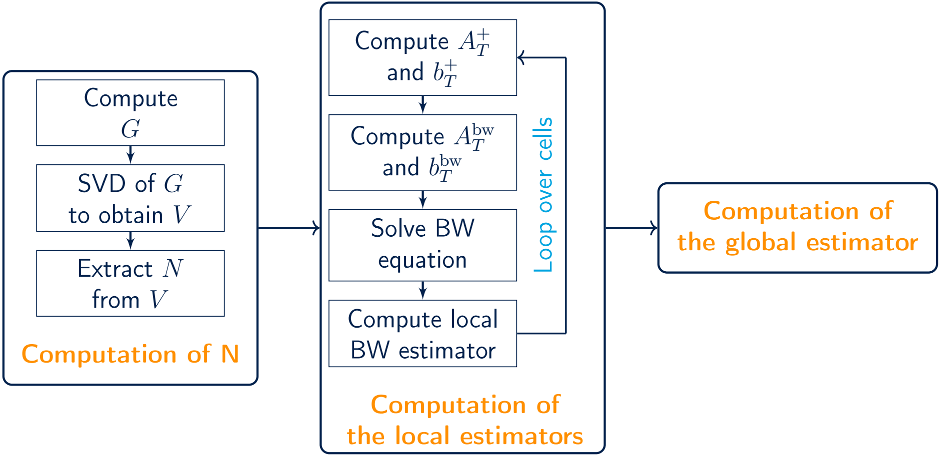
Algorithm


Compute
G
SVD of G
to obtain V
Extract N
from V
Computation of N

Algorithm


Compute
G
SVD of G
to obtain V
Extract N
from V
Computation of N
Compute A_T^+
and b_T^+
Compute A_T^{bw}
and b_T^{bw}
Solve BW
equation
Compute local
BW estimator
Loop over cells
Computation of
the local estimators

Algorithm



Compute G
SVD of G to obtain V
Extract N from V
Computation of N
Compute A_T^+ and b_T^+
Compute A_T^{bw} and b_T^{bw}
Solve BW equation
Compute local BW estimator
Loop over cells
Computation of the local estimators
Computation of the global estimator

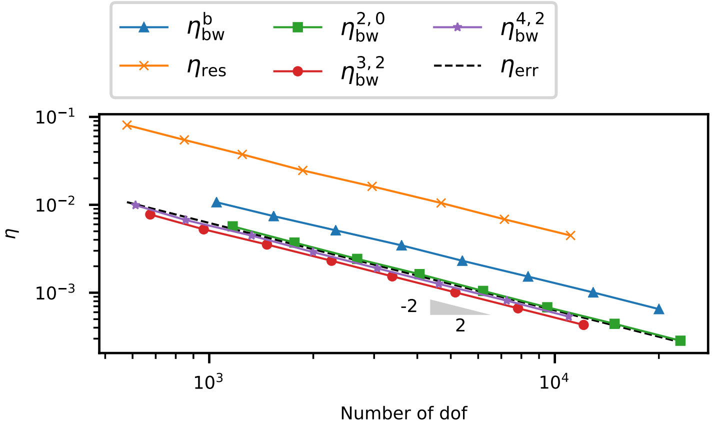
In code

```
def estimate(u_h):
    mesh = u_h.function_space().mesh()
    element_f = FiniteElement("DG", triangle, 2)
    element_g = FiniteElement("DG", triangle, 1)

    N = fenics_error_estimation.create_interpolation(element_f, element_g)

    V_f = FunctionSpace(mesh, element_f)
    e = TrialFunction(V_f)
    v = TestFunction(V_f)
    f = Constant(0.0)
    bcs = DirichletBC(V_f, Constant(0.0), "on_boundary", "geometric")

    n = FacetNormal(mesh)
    a_e = inner(grad(e), grad(v))*dx
    L_e = inner(f + div(grad(u_h)), v)*dx + \
            inner(jump(grad(u_h), -n), avg(v))*dS

    e_h = fenics_error_estimation.estimate(a_e, L_e, N, bcs)
    error = norm(e_h, "H10")

    V_e = FunctionSpace(mesh, "DG", 0)
    v = TestFunction(V_e)
    eta_h = Function(V_e, name="eta_h")
    eta = assemble(inner(inner(grad(e_h), grad(e_h)), v)*dx)
    eta_h.vector()[:] = eta

    return eta_h
```

Results I

Adaptive finite elements for a Poisson problem:

$-\Delta u = 0$ in Ω, $\quad u = u_D$ on Γ. Quadratic finite elements.



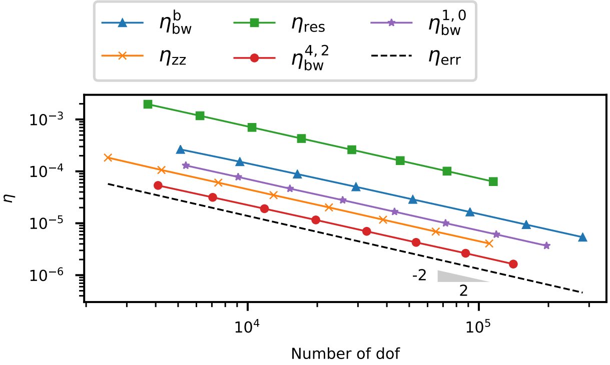
Notation	V_T^+	V_T^-
$\eta_{\text{bw}}^{k_+,k_-}$	V_T^{k+}	V_T^{k-}
η_{bw}^{b}	$V_T^2 + \text{bubble}$	V_T^1

Results II

Goal oriented adaptive finite elements for a Poisson problem:

$-\Delta u = 0$ in Ω, $\quad u = u_D$ on Γ. $\qquad \eta_{\text{err}} := J(u - u_1) = \int_\Omega (u - u_h)c$, where c is a smooth weight function.

The estimators are computed using the WGO method from [Becker et al., 2011].

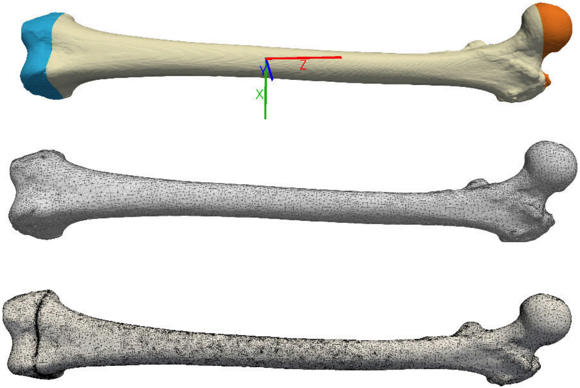


Notation	V_T^+	V_T^-
$\eta_{\text{bw}}^{k_+,k_-}$	$V_T^{k_+}$	$V_T^{k_-}$
η_{bw}^{b}	$V_T^2 + \text{bubble}$	V_T^1

Results II

GO AFEM for a linear elasticity problem:

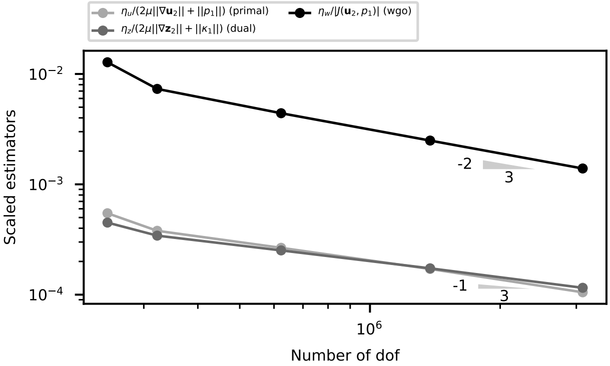
we used a technique from [Khan et al., 2019] to compute the estimators. The goal functional is defined by $J(\mathbf{u_2}, p_1) := \int_\Gamma \mathbf{u_2} \cdot \mathbf{n} c$.



Results II

GO AFEM for a linear elasticity problem:

we used a technique from [Khan et al., 2019] to compute the estimators. The goal functional is defined by $J(\mathbf{u_2}, p_1) := \int_\Gamma \mathbf{u_2} \cdot \mathbf{n} c$.



Thank you for your attention!

I would like to acknowledge the support of the ASSIST research project of the University of Luxembourg. This presentation has been prepared in the framework of the DRIVEN project funded by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 811099.

References I

Bank, R. E. and Weiser, A. (1985).
Some A Posteriori Error Estimators for Elliptic Partial Differential Equations.
Math. Comput., 44(170):283–301.

Becker, R., Estecahandy, E., and Trujillo, D. (2011).
Weighted Marking for Goal-oriented Adaptive Finite Element Methods.
SIAM J. Numer. Anal., 49(6):2451–2469.

Bulle, R., Chouly, F., Hale, J. S., and Lozinski, A. (2020).
Removing the saturation assumption in Bank–Weiser error estimator analysis in dimension three.
Appl. Math. Lett., 107(1):106429.

Khan, A., Powell, C. E., and Silvester, D. J. (2019).
Robust a posteriori error estimators for mixed approximation of nearly incompressible elasticity.
Int. J. Numer. Methods Eng., 119(1):18–37.