JAX-DIPS: Differentiable Inverse Free Boundary PDE Solver

Pouria A. Mistani* [†,a], Samira Pakravan[†,b], Rajesh Ilango[a], Frederic G. Gibou[b]

[a] *NVIDIA Corporation, Santa Clara, CA 95051, USA*
[b] *Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, USA*

Abstract

We present an end-to-end differentiable free boundary PDE solver based on the level-set method. We implemented this framework using JAX, offering support for CPU/GPU/TPU platforms. Algorithmically, our proposed framework builds on the blended inverse PDE solver architecure (BiPDE) that authors have proposed earlier [3]. JAX-DIPS is an open-source software package published under MIT license and is available at https://github.com/JAX-DIPS/JAX-DIPS.

Keywords: level-set method, free boundary problems, inverse problems, jump conditions, differentiable programming

1. Introduction

2. Numerical Scheme for Free Boundary Problems

Consider a closed irregular interface (Γ) that partitions the computational domain (Ω) into interior (Ω^-) and exterior (Ω^+) subdomains; *i.e.*, $\Omega = \Omega^- \cup \Gamma \cup \Omega^+$. We are interested in the solutions $\mathrm{u}^\pm \in \Omega^\pm$ to the following class of linear elliptic problems in $\mathbf{x} \in \Omega^\pm$:

$$\begin{aligned} &k^\pm u^\pm - \nabla \cdot (\mu^\pm \nabla u^\pm) = f^\pm, && \mathbf{x} \in \Omega^\pm \\ &[u] = \alpha, && \mathbf{x} \in \Gamma \\ &[\mu \partial_{\mathbf{n}} u] = \beta, && \mathbf{x} \in \Gamma \end{aligned}$$

Here $f^\pm = f(\mathbf{x} \in \Omega^\pm)$ is the spatially varying source term, $\mu^\pm = \mu(\mathbf{x} \in \Omega^\pm)$ are the diffusion coefficients, and $k^\pm$ are the reaction coefficients in the two domains. We consider Dirichlet boundary conditions in a cubic domain $\Omega = [0, L]^3$.

2.1. The level-set method

2.2. Finite discretization method

For spatial discretizations at the presence of jump conditions we employ the numerical algorithm proposed by Bochkov and Gibou (2020) [1] (BG20) on Cartesian grids. BG20 produces second-order accurate solutions and first-order accuracte gradients in the L^∞-norm, while having a compact stencil that makes it a good candidate for parallelization. Moreover, treatment of the interface jump conditions do not introduce any augmented variables, this preserves the homogeneous structure of the linear system.

Here we use a finite volume discretization equation uniformly for all grid points. At grid points where the finite volumes are crossed by Γ we have

$$\sum_{s=-,+} \int_{\Omega^s \cap \mathcal{V}_{i,j}} k^s u^s d\Omega - \sum_{s=-,+} \int_{\Omega^s \cap \partial\mathcal{V}_{i,j}} \mu^s \partial_{\mathbf{n}^s} u^s d\Gamma = \sum_{s=-,+} \int_{\Omega^s \cap \mathcal{V}_{i,j}} f^s d\Omega + \int_{\Gamma \cap \mathcal{V}_{i,j}} [\mu \partial_{\mathbf{n}} u] d\Gamma$$

*Corresponding author: p.a.mistani@gmail.com
†These authors contributed equally to this work

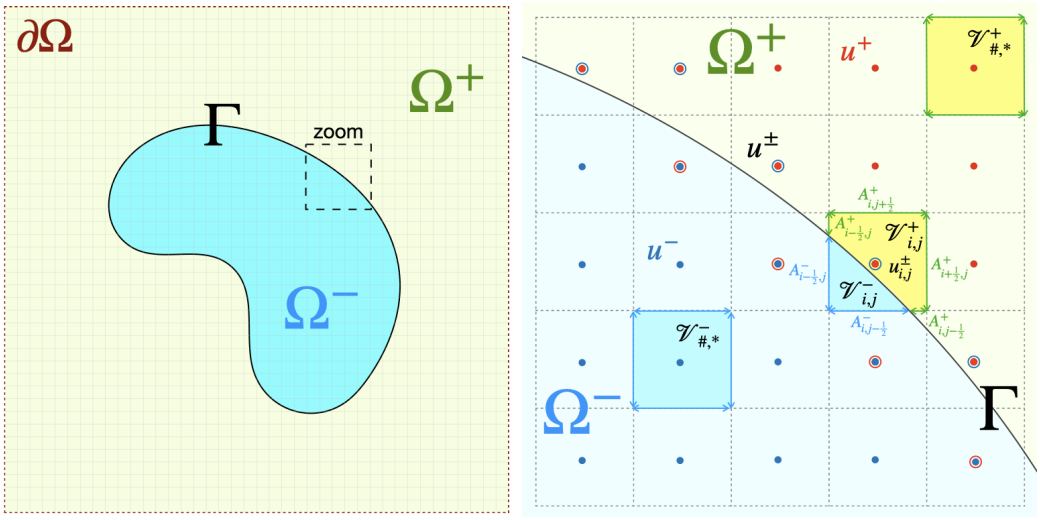

Figure 1: Notation used in this paper. Close to the interface where finite volumes are crossed by the interface, there are extra degrees of freedom (open circles) that are extrapolations of solutions from each domain to the opposite domain. Jump conditions are implicitly encoded in these extrapolated values.

following standard treatment of volumetric integrals and using central differencing for derivatives we obtain in 2D (with trivial 3D extension)

$$\sum_{s=-,+} k^s u^s_{i,j} |\mathcal{V}^s_{i,j}| - \sum_{s=-,+} \left(\mu^s_{i-\frac{1}{2},j} A^s_{i-\frac{1}{2},j} \frac{u^s_{i-1,j} - u^s_{i,j}}{\Delta x} + \mu^s_{i+\frac{1}{2},j} A^s_{i+\frac{1}{2},j} \frac{u^s_{i+1,j} - u^s_{i,j}}{\Delta x} + \right.$$
$$\left. \mu^s_{i,j-\frac{1}{2}} A^s_{i,j-\frac{1}{2}} \frac{u^s_{i,j-1} - u^s_{i,j}}{\Delta y} + \mu^s_{i,j+\frac{1}{2}} A^s_{i,j+\frac{1}{2}} \frac{u^s_{i,j+1} - u^s_{i,j}}{\Delta y} \right)$$
$$= \sum_{s=-,+} f^s_{i,j} |\mathcal{V}^s_{i,j}| + \int_{\Gamma \cap \mathcal{V}_{i,j}} \beta d\Gamma + \mathcal{O}(\max(\Delta x, \Delta y)^{\mathcal{D}})$$

where $\mathcal{D}$ is the problem dimensionality.

Note that far from interface either $s = -$ (for $\mathbf{x} \in \Omega^-$) or $s = +$ (for $\mathbf{x} \in \Omega^+$) is retained. This is automatically considered through zero values for sub-volumes $|\mathcal{V}^+_{i,j}|$ and $|\mathcal{V}^-_{i,j}|$ as well as their face areas. Note that $\mu^-_{i-1/2,j}$ (or $\mu^+_{i-1/2,j}$) corresponds to the value of diffusion coefficient at the middle of segment $A^-_{i-1/2,j}$ (or $A^+_{i-1/2,j}$) respectively, same is true for other edges as well. However, there are extra degrees of freedom on grid points whose finite volumes are crossed by the interface; *i.e.*, see double circles in figure 1. Bochkov and Gibou derived analytical expressions for the extra degrees of freedom (u^+ in Ω^- and u^- in Ω^+) in terms of the original degrees of freedom (u^- in Ω^- and u^+ in Ω^+) as well as the jump conditions, this preserves the original $\mathrm{N_x} \times \mathrm{N_y}$ system size.

In this scheme the basic idea is to extrapolate the jump at grid point from jump condition at the projected point onto the interface using a Taylor expansion: $u^+_{i,j} - u^-_{i,j} = [u]_{\mathbf{r}^{pr}_{i,j}} + \delta_{i,j}(\partial_{\mathbf{n}} u^+(\mathbf{r}^{pr}_{i,j}) - \partial_{\mathbf{n}} u^-(\mathbf{r}^{pr}_{i,j}))$. The unknown value ($u^-_{i,j}$ or $u^+_{i,j}$) is obtained based on approximation of the normal derivatives (*i.e.* $\partial_{\mathbf{n}} u^{\pm}(\mathbf{r}^{pr}_{i,j})$) which are computed using a least squares calculation on neighboring grid points that are in the fast-diffusion region (referred to as "Bias Fast") or in the slow diffusion region (referred to as "Bias Slow"). This makes two sets of ruls for unknown values $u^{\pm}_{i,j}$.

In two dimensions and on uniform grids, the gradient operator at the grid cell (i, j) that is

crossed by an interface is estimated by a least squares solution given by

$$(\nabla u^{\pm})_{i,j} = \mathbf{D}^{\pm}_{i,j} \begin{bmatrix} u_{i-1,j-1} - u^{\pm}_{i,j} \\ u_{i,j-1} - u^{\pm}_{i,j} \\ \vdots \\ u_{i+1,j+1} - u^{\pm}_{i,j} \end{bmatrix} \qquad \mathbf{D}^{\pm}_{i,j} = \left(X^T_{i,j} W^{\pm}_{i,j} X_{i,j}\right)^{-1} \left(W^{\pm}_{i,j} X_{i,j}\right)^T$$

and

$$W^{\pm}_{i,j} = \begin{bmatrix} \omega^{\pm}_{i,j}(-1,-1) & & & \\ & \omega^{\pm}_{i,j}(0,-1) & & \\ & & \ddots & \\ & & & \omega^{\pm}_{i,j}(1,1) \end{bmatrix} \qquad X_{i,j} = \begin{bmatrix} -h_x & -h_y \\ 0 & -h_y \\ h_x & -h_y \\ -h_x & 0 \\ 0 & 0 \\ h_x & 0 \\ -h_x & h_y \\ 0 & h_y \\ h_x & h_y \end{bmatrix}$$

and

$$\omega^{\pm}_{i,j}(p,q) = \begin{cases} 1 & (p,q) \in N^{\pm}_{i,j} \\ 0 & \text{else} \end{cases} \tag{1}$$

In this case, $D^{\pm}_{i,j}$ is a 2×9 matrix and we denote each of its 2×1 columns with $d^{\pm}_{i,j,p,q}$

$$\mathbf{D}^{\pm}_{i,j} = \begin{bmatrix} d^{\pm}_{i,j,-1,-1} & d^{\pm}_{i,j,0,-1} & d^{\pm}_{i,j,1,-1} & d^{\pm}_{i,j,-1,0} & d^{\pm}_{i,j,0,0} & d^{\pm}_{i,j,1,0} & d^{\pm}_{i,j,-1,1} & d^{\pm}_{i,j,0,1} & d^{\pm}_{i,j,1,1} \end{bmatrix}$$

The least square coefficients are then obtained by dot product of normal vector with these columns

$$c^{\pm}_{i,j,p,q} = \mathbf{n}^T_{i,j} d^{\pm}_{i,j,p,q}$$

At this point we can define a few intermediate variables at each grid point to simplify the presentation of the method,

$$\zeta^{\pm}_{i,j,p,q} := \delta_{i,j} \frac{[\mu]}{\mu^{\mp}} c^{\pm}_{i,j,p,q} \qquad \zeta^{\pm}_{i,j} := -\sum_{(p,q) \in N^{\pm}_{i,j}} \zeta^{\pm}_{i,j,p,q}$$

$$\gamma^{\pm}_{i,j,p,q} := \frac{\zeta^{\pm}_{i,j,p,q}}{1 \pm \zeta^{\pm}_{i,j}} \qquad \gamma^{\pm}_{i,j} := -\sum_{(p,q) \in N^{\pm}_{i,j}} \gamma^{\pm}_{i,j,p,q}$$

where the set of neighboring grid points are

$$N^{\pm}_{i,j} = \{(p,q) : \quad p = -1,0,1, \quad q = -1,0,1, \quad (p,q) \neq (0,0), \quad \mathbf{x}_{i+p,j+q} \in \Omega^{\pm}\}$$

and $\delta_{i,j}$ is the signed distance from $\mathbf{x}_{i,j}$ that is computed from the level-set function $\phi(\mathbf{x})$

$$\delta_{i,j} = \frac{\phi(\mathbf{x}_{i,j})}{|\nabla \phi(\mathbf{x}_{i,j})|}$$

- Rules based on approximating $\partial_{\mathbf{n}} u^+(\mathbf{r}^{pr}_{i,j})$:

$$u^-_{i,j} = \begin{cases} u_{i,j} & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j}(1 - \gamma^-_{i,j}) - \sum_{(p,q) \in N^-_{i,j}} \gamma^-_{i,j,p,q} u_{i+p,j+q} - (\alpha + \frac{\delta_{i,j}\beta}{\mu^+})(1 - \gamma^-_{i,j}) & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{2}$$

$$u^+_{i,j} = \begin{cases} u_{i,j}(1-\zeta^-_{i,j}) - \sum_{(p,q)\in N^-_{i,j}} \zeta^-_{i,j,p,q} u_{i+p,j+q} + \alpha + \delta_{i,j}\frac{\beta}{\mu^+} & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{3}$$

It is useful to cast this in the form of matrix operations through defining intermediate tensors:

$$\mathbf{\Gamma}_{i,j} := \begin{bmatrix} \gamma^-_{i-1,j+1} & \gamma^-_{i,j+1} & \gamma^-_{i+1,j+1} \\ \gamma^-_{i-1,j} & \gamma^-_{i,j} & \gamma^-_{i+1,j} \\ \gamma^-_{i-1,j-1} & \gamma^-_{i,j-1} & \gamma^-_{i+1,j-1} \end{bmatrix}, \qquad \boldsymbol{\zeta}_{i,j} := \begin{bmatrix} \zeta^-_{i-1,j+1} & \zeta^-_{i,j+1} & \zeta^-_{i+1,j+1} \\ \zeta^-_{i-1,j} & \zeta^-_{i,j} & \zeta^-_{i+1,j} \\ \zeta^-_{i-1,j-1} & \zeta^-_{i,j-1} & \zeta^-_{i+1,j-1} \end{bmatrix}$$

$$\mathbf{U}_{i,j} := \begin{bmatrix} u_{i-1,j+1} & u_{i,j+1} & u_{i+1,j+1} \\ u_{i-1,j} & u_{i,j} & u_{i+1,j} \\ u_{i-1,j-1} & u_{i,j-1} & u_{i+1,j-1} \end{bmatrix}, \quad \mathbf{N}^\pm_{i,j} := \begin{bmatrix} \omega^\pm_{i,j}(-1,1) & \omega^\pm_{i,j}(0,1) & \omega^\pm_{i,j}(1,1) \\ \omega^\pm_{i,j}(-1,0) & 0 & \omega^\pm_{i,j}(1,0) \\ \omega^\pm_{i,j}(-1,-1) & \omega^\pm_{i,j}(0,-1) & \omega^\pm_{i,j}(1,-1) \end{bmatrix}$$

where $\mathbf{N}^-$ is a masking filter that passes the values in the negative neighborhood of node (i,j).

We also introduce the Hadamard product $\odot$ between two identical matrices that creates another identical matrix with each entry being elementwise products. Moreover, double contraction of two tensors A and B is defined by $A : B = \sum A \odot B$ which is a scalar value and equals the sum of all entries of the Hadamard product of the tensors; *i.e.*, note $A : A$ is square of Frobenius norm of A. Using these notations, the substitution rules read

$$u^-_{i,j} = \begin{cases} u_{i,j} & \mathbf{x}_{i,j} \in \Omega^- \\ \big(1 + \mathbf{\Gamma}^-_{i,j} : \mathbf{N}^-_{i,j}\big) u_{i,j} - \big(\mathbf{\Gamma}^-_{i,j} \odot \mathbf{N}^-_{i,j}\big) : \mathbf{U}_{i,j} - (\alpha + \delta_{i,j}\frac{\beta}{\mu^+})\big(1 + \mathbf{\Gamma}^-_{i,j} : \mathbf{N}^-_{i,j}\big) & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{4}$$

$$u^+_{i,j} = \begin{cases} \big(1 + \boldsymbol{\zeta}^-_{i,j} : \mathbf{N}^-_{i,j}\big) u_{i,j} - \big(\boldsymbol{\zeta}^-_{i,j} \odot \mathbf{N}^-_{i,j}\big) : \mathbf{U}_{i,j} + \alpha + \delta_{i,j}\frac{\beta}{\mu^+} & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{5}$$

- Rules based on approximating $\partial_{\mathbf{n}} u^-(\mathbf{r}^{pr}_{i,j})$:

$$u^-_{i,j} = \begin{cases} u_{i,j} & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j}(1-\zeta^+_{i,j}) - \sum_{(p,q)\in N^+_{i,j}} \zeta^+_{i,j,p,q} u_{i+p,j+q} - \alpha - \delta_{i,j}\frac{\beta}{\mu^-} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{6}$$

$$u^+_{i,j} = \begin{cases} u_{i,j}(1-\gamma^+_{i,j}) - \sum_{(p,q)\in N^+_{i,j}} \gamma^+_{i,j,p,q} u_{i+p,j+q} + (\alpha + \delta_{i,j}\frac{\beta}{\mu^-})(1-\gamma^+_{i,j}) & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{7}$$

in matrix notation we have

$$u^-_{i,j} = \begin{cases} u_{i,j} & \mathbf{x}_{i,j} \in \Omega^- \\ \big(1 + \boldsymbol{\zeta}^+_{i,j} : \mathbf{N}^+_{i,j}\big) u_{i,j} - \big(\boldsymbol{\zeta}^+_{i,j} \odot \mathbf{N}^+_{i,j}\big) : \mathbf{U}_{i,j} - \alpha - \delta_{i,j}\frac{\beta}{\mu^-} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{8}$$

$$u^+_{i,j} = \begin{cases} \big(1 + \mathbf{\Gamma}^+_{i,j} : \mathbf{N}^+_{i,j}\big) u_{i,j} - \big(\mathbf{\Gamma}^+_{i,j} \odot \mathbf{N}^+_{i,j}\big) : \mathbf{U}_{i,j} + (\alpha + \delta_{i,j}\frac{\beta}{\mu^-})\big(1 + \mathbf{\Gamma}^+_{i,j} : \mathbf{N}^+_{i,j}\big) & \mathbf{x}_{i,j} \in \Omega^- \\ u_{i,j} & \mathbf{x}_{i,j} \in \Omega^+ \end{cases} \tag{9}$$

2.3. 3D geometric integrations

We use uniform Cartesian grids. For computational cells that are crossed by the interface, *i.e.* $\mathcal{V}_{i,j,k} \cap \Gamma \neq 0$, we use the geometric integrations proposed by Min & Gibou (2007) [2]. In this scheme each grid cell is decomposed into five tetrahedra by the middle-cut triangulation [4].

procedure BIAS SLOW

if $\Gamma \cap \mathcal{C}_{i,j} = \emptyset$ **then**

$$B^{\pm}_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad r^{\pm}_{i,j} = 0$$

else

if $\mu^{-}_{i,j} > \mu^{+}_{i,j}$ **then**

if $\phi_{i,j} \geq 0$ **then**

$$B^{+}_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad r^{+}_{i,j} = 0$$

$$B^{-}_{i,j} = \begin{bmatrix} -\gamma^{-}_{i,j,-1,1} & -\gamma^{-}_{i,j,0,1} & -\gamma^{-}_{i,j,1,1} \\ -\gamma^{-}_{i,j,-1,0} & 1-\gamma^{-}_{i,j} & -\gamma^{-}_{i,j,1,0} \\ -\gamma^{-}_{i,j,-1,-1} & -\gamma^{-}_{i,j,0,-1} & -\gamma^{-}_{i,j,1,-1} \end{bmatrix}; \quad r^{-}_{i,j} = -(\alpha^{proj}_{i,j} + \delta_{i,j}\frac{\beta^{proj}_{i,j}}{\mu^{+}_{proj}})(1-\gamma^{-}_{i,j})$$

else

$$B^{+}_{i,j} = \begin{bmatrix} -\zeta^{-}_{i,j,-1,1} & -\zeta^{-}_{i,j,0,1} & -\zeta^{-}_{i,j,1,1} \\ -\zeta^{-}_{i,j,-1,0} & 1-\zeta^{-}_{i,j} & -\zeta^{-}_{i,j,1,0} \\ -\zeta^{-}_{i,j,-1,-1} & -\zeta^{-}_{i,j,0,-1} & -\zeta^{-}_{i,j,1,-1} \end{bmatrix}; \quad r^{+}_{i,j} = \alpha^{proj}_{i,j} + \delta_{i,j}\frac{\beta^{proj}_{i,j}}{\mu^{+}_{proj}}$$

$$B^{-}_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad r^{-}_{i,j} = 0$$

else

if $\phi_{i,j} \geq 0$ **then**

$$B^{+}_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad r^{+}_{i,j} = 0$$

$$B^{-}_{i,j} = \begin{bmatrix} -\zeta^{+}_{i,j,-1,1} & -\zeta^{+}_{i,j,0,1} & -\zeta^{+}_{i,j,1,1} \\ -\zeta^{+}_{i,j,-1,0} & 1-\zeta^{+}_{i,j} & -\zeta^{+}_{i,j,1,0} \\ -\zeta^{+}_{i,j,-1,-1} & -\zeta^{+}_{i,j,0,-1} & -\zeta^{+}_{i,j,1,-1} \end{bmatrix}; \quad r^{-}_{i,j} = \alpha^{proj}_{i,j} + \delta_{i,j}\frac{\beta^{proj}_{i,j}}{\mu^{-}_{proj}}$$

else

$$B^{+}_{i,j} = \begin{bmatrix} -\gamma^{+}_{i,j,-1,1} & -\gamma^{+}_{i,j,0,1} & -\gamma^{+}_{i,j,1,1} \\ -\gamma^{+}_{i,j,-1,0} & 1-\gamma^{+}_{i,j} & -\gamma^{+}_{i,j,1,0} \\ -\gamma^{+}_{i,j,-1,-1} & -\gamma^{+}_{i,j,0,-1} & -\gamma^{+}_{i,j,1,-1} \end{bmatrix}; \quad r^{+}_{i,j} = (\alpha^{proj}_{i,j} + \delta_{i,j}\frac{\beta^{proj}_{i,j}}{\mu^{-}_{proj}})(1-\gamma^{+}_{i,j})$$

$$B^{-}_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad r^{-}_{i,j} = 0$$

Algorithm 1: Bias Slow approximation of the non-existing solution value on a grid point based on existing solution values in its neighborhood. The notation is used for $u^{\pm}_{i,j} = B^{\pm}_{i,j} : \mathbf{U}_{i,j} + r^{\pm}_{i,j}$.

3. Numerical Examples

3.1. Stationary linear case

$$
\begin{aligned}
k^{\pm}u^{\pm} - \nabla \cdot (\mu^{\pm}\nabla u^{\pm}) &= f^{\pm}, && \mathbf{x} \in \Omega^{\pm} \\
[u] &= \alpha, && \mathbf{x} \in \Gamma \\
[\mu \partial_{\mathbf{n}} u] &= \beta, && \mathbf{x} \in \Gamma
\end{aligned}
$$

Acknowledgement

References

[1] D. Bochkov and F. Gibou. Solving elliptic interface problems with jump conditions on cartesian grids. *Journal of Computational Physics*, 407:109269, 2020.

[2] C. Min and F. Gibou. Geometric integration over irregular domains with application to level-set methods. *Journal of Computational Physics*, 226(2):1432–1443, 2007.

[3] S. Pakravan, P. A. Mistani, M. A. Aragon-Calvo, and F. Gibou. Solving inverse-pde problems with physics-aware neural networks. *Journal of Computational Physics*, 440:110414, 2021.

[4] J. F. Sallee. The middle-cut triangulations of the n-cube. *SIAM Journal on Algebraic Discrete Methods*, 5(3):407–419, 1984.