論文紹介

FastVPINNs: Tensor-Driven Acceleration of VPINNs for Complex Geometries

2024年5月18日

紹介する論文

- タイトル

FastVPINNs: Tensor-Driven Acceleration of VPINNs for Complex Geometries

- 著者

Thivin Anandh, Divij Ghose, Himanshu Jain, Sashikumaar Ganesan

- GitHub

https://github.com/cmgcds/fastvpinns

論文目次

論文概要

- VPINNs（Variational-PINNs）は、変分損失関数を用いて偏微分方程式を解く手法である。
- 従来のhp-VPINNsは、高周波問題に効果的な一方で、計算が集約的かつスケール性が貧弱であり、複雑な幾何学形状に対して限界があった。
- FastVPINNsでは、テンソルに基づいた発展をさせることにより、計算の間接負荷を大幅に軽減し、かつ安定性を向上させている。
- 最適化されたテンソル演算を用いることにより、従来のhp-VPINNs と比較して、1エポック当たりの計算時間を（中央値で）100倍軽減することに成功した。
- 適切なハイパーパラメータを選択すれば、特に高周波問題において、速度および精度の両面でFastVPINNsは従来のPINNsを凌駕した。

PINNsの発展について

紹介論文に登場するアーキテクチャ

- PINNs
- VPINNs
- hp-VPINNs
- cv-PINNs

紹介論文のFastVPINNsは、hp-VPINNsを改良したもの

PINNs

- コンピュータシミュレーション等の代わりに、機械学習によって物理量の入出力関係を再現させる。
- 支配方程式による拘束を表現した損失関数を導入することにより、機械学習を使って物理的な要請を満たす出力を得る。
- Raissi.et al(2019)[1)]によって人気に火が付いた印象があるが、1990年代後半の時点で種となるアイディア[2)]は登場していたらしい。

1) Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378:686–707, 2019.

2) Lagaris, A. Likas, and D. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5):987–1000, 1998.

VPINNs

- Neural Networkを用いた非線形近似を表す解法。
- 変分法による弱形式を組み込み、変分損失関数を使う。
- ガラーキン法[3)]の代わりにペトロフ・ガラーキン法[4)]を適用する。
- ガラーキン法を用いると、試行関数と検査関数は一致する。一方、ペトロフ・ガラーキン法を用いる場合は、試行空間と検査空間が異なれば試行関数と検査関数も異なる。

3) ガラーキン法についてのわかりやすいweb記事 ⇒ https://qiita.com/Sego-don/items/e9f46960ff30452ea140

4) 有限要素法などで偏微分方程式を近似的に解く場合，重み関数として補間（評価）関数と同一の関数を選ぶガラーキン法もしくはバブノフ・ガラーキン法に対して，流体計算における移流項や混合型の有限要素に起因する不安定性を制御するために，両者に異なる関数を使う場合，ペトロフ・ガラーキン法という．(https://www.jsme.or.jp/jsme-medwiki/doku.php?id=01:1011846)

VPINNs

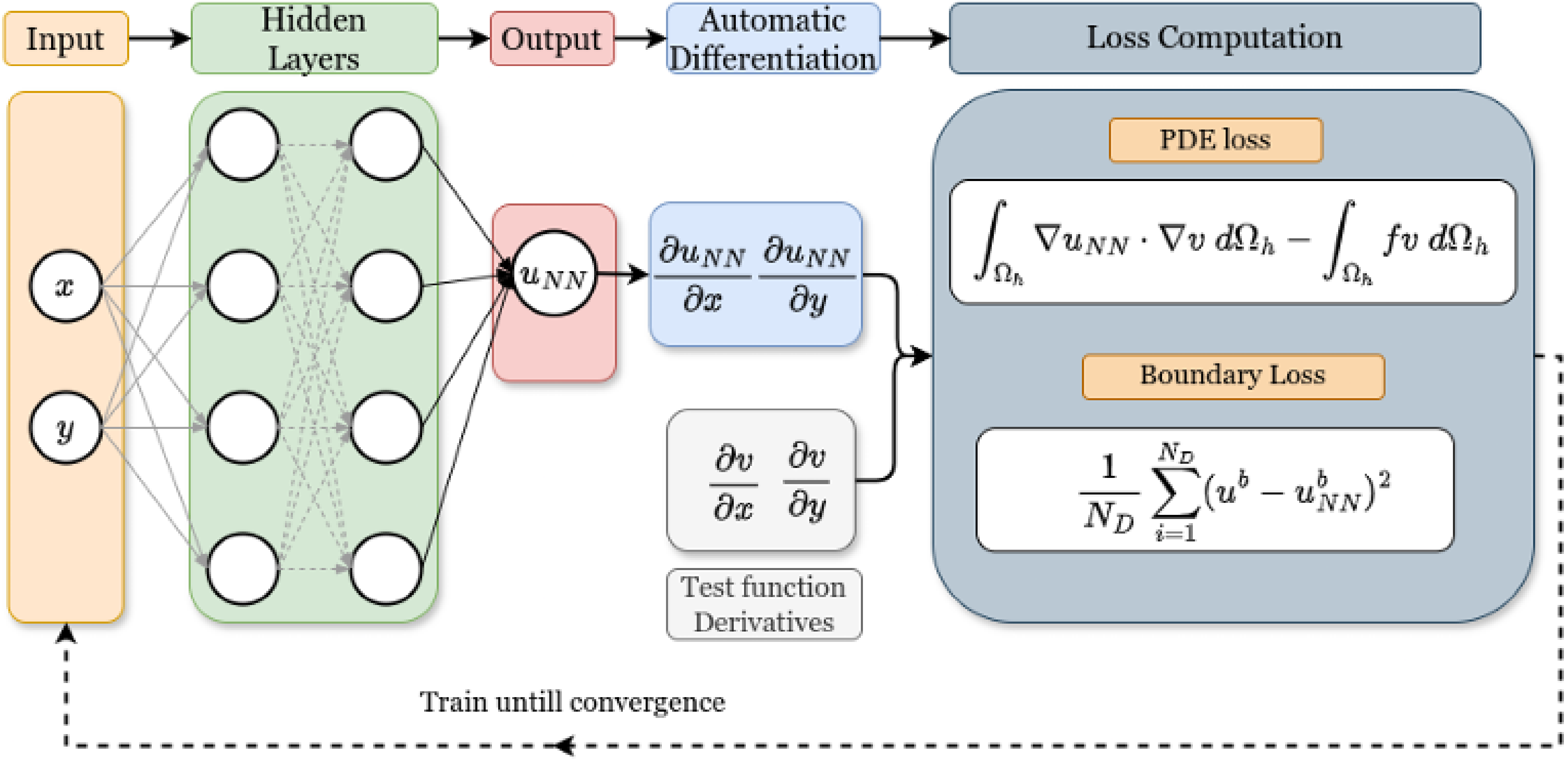
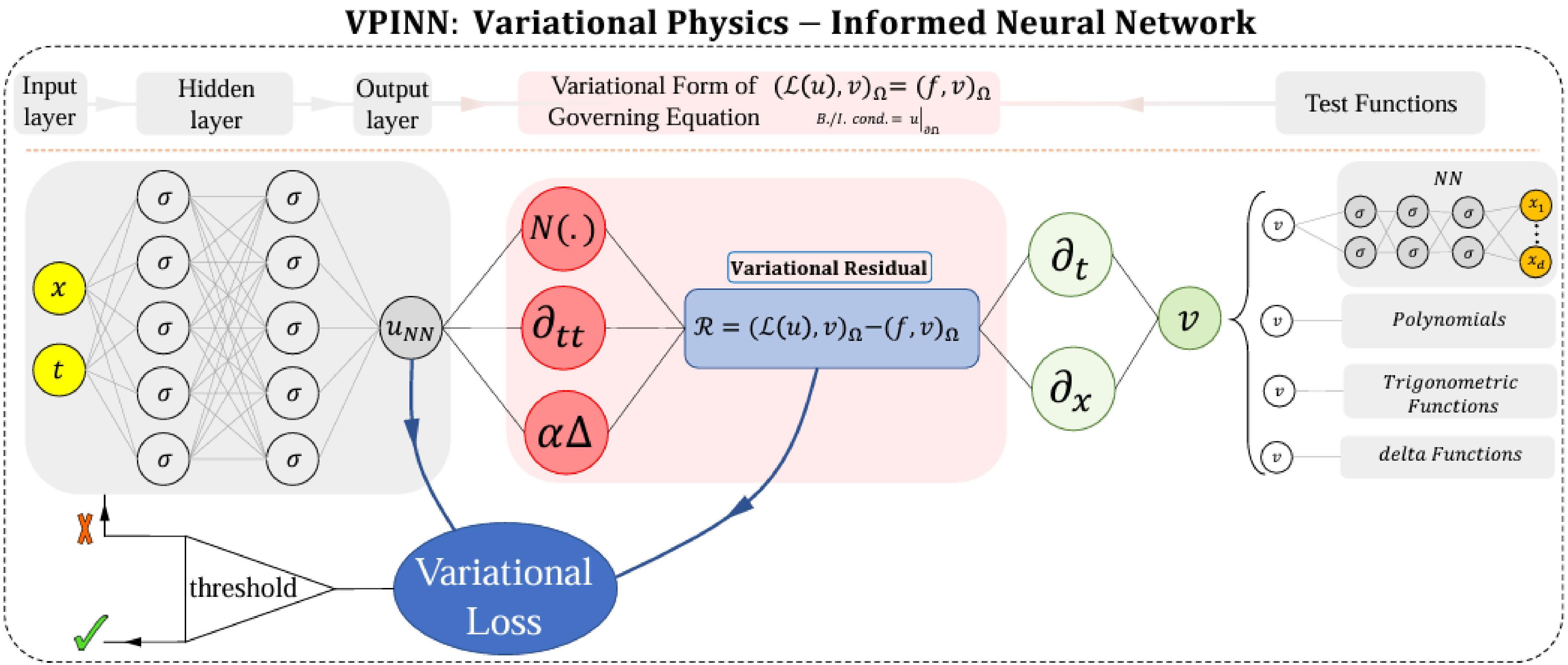


Figure 1: Schematics of Variational PINNs for a 2D Poisson problem.

VPINNs



E. Kharazmi, Z. Zhang, G. E. Karniadakis：Variational Physics-Informed Neural Networks For Solving Partial Differential Equations
https://arxiv.org/abs/1912.00873

VPINNs

- Burgers方程式の理論解との比較

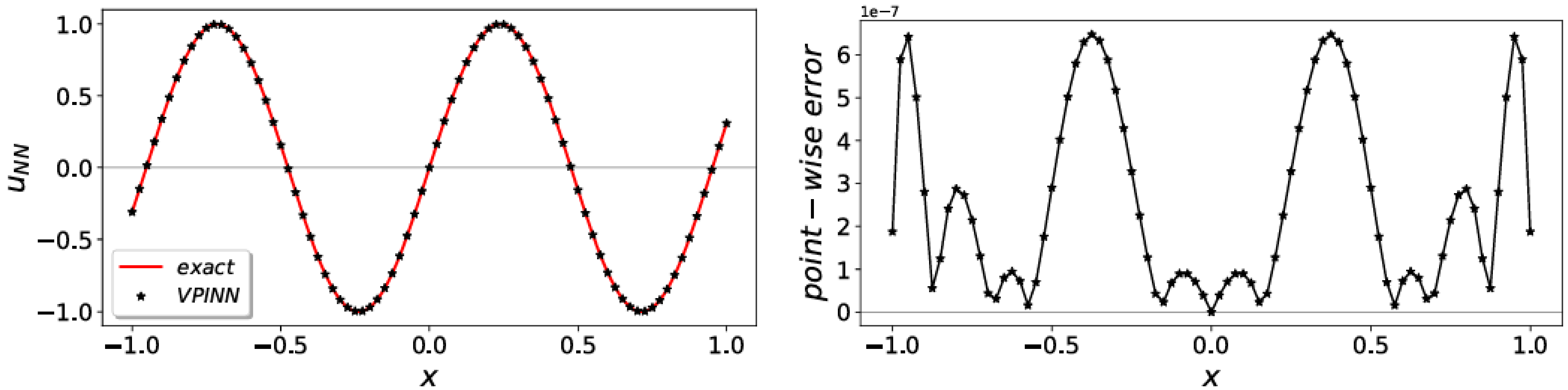


Fig. 4: One-dimensional steady state Burger's equation: VPINN with $\mathcal{R}^{(1)} = \mathcal{R}^{(2)}$ formulation. Left: exact solution $\sin(2.1\pi x)$ and VPINN approximation. Right: point-wise error averaged over several random network initializations. See Table 1 for VPINN hyperparameters.

FastVPINNsの仕組み

- テンソル演算の適用により、個別の要素に対する繰り返し計算の必要性を取り除く。
- テンソル演算は、BLAS[5] を用いたGPU計算に適している。BLASを使えば、GPUのコアが持つハードウェア能力を有効に利用することができる。
- 従来の手法では勾配計算のために複数の逆伝播計算を実行する必要がある。FastVPINNsではこれを克服し、複雑な形状でも1ステップのみ逆伝播計算で済ませられる。

5) Basic Linear Algebra Subprograms: 線型代数の基本的な計算コードを集めたパッケージ。Level 1, Level 2, および Level 3 にカテゴライズされている。https://netlib.org/blas/

FastVPINNsの仕組み

(a)求積方向、テスト方向、セル方向の3階テンソル演算

(b)四角形への双線形変換（有限要素法の手法を取り入れる）

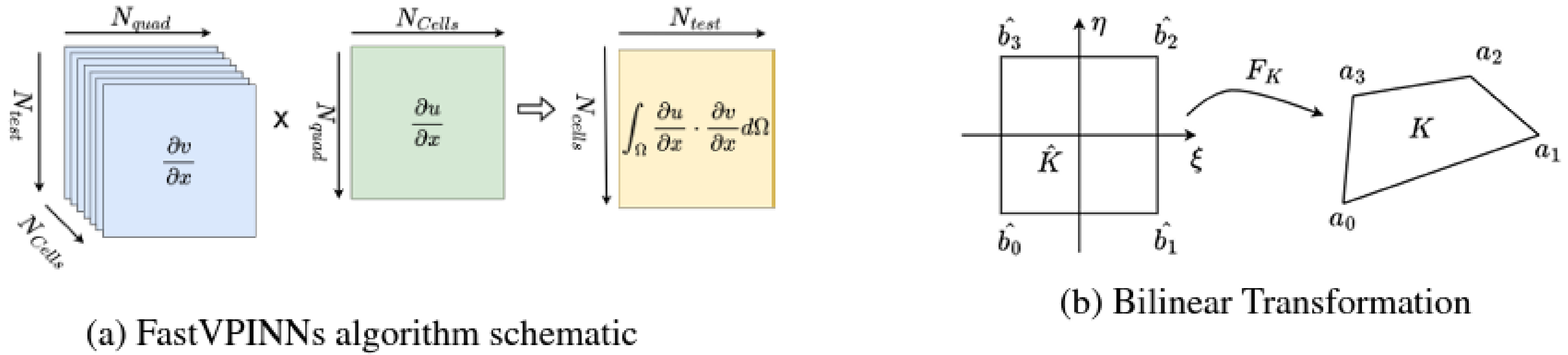


Figure 2: **(a)** Tensor-based computation of the variational loss. **(b)** Bilinear transformations to handle reference transformations for quadrilateral cell.

従来のPINNsとFastVPINNsの比較

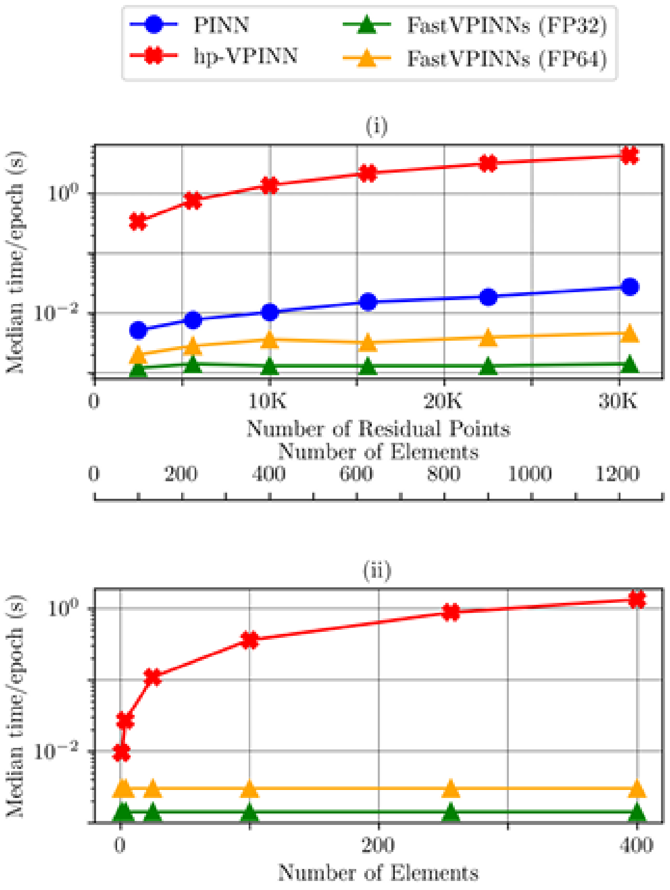
- 訓練環境：NVIDIA RTX A6000 GPU with 48GB
- PINNsの結果：The NVIDIA-Modulus library(NVIDIA Modulus)
- まずはシンプルなポアソン方程式で比較

$$\begin{aligned} -\Delta u(x,y) &= -2\omega^2 \sin(\omega x)\sin(\omega y) \quad (x,y) \in \Omega = [0,1]^2, \\ u(0,\cdot) &= u(\cdot,0) = 0 \end{aligned} \tag{1}$$

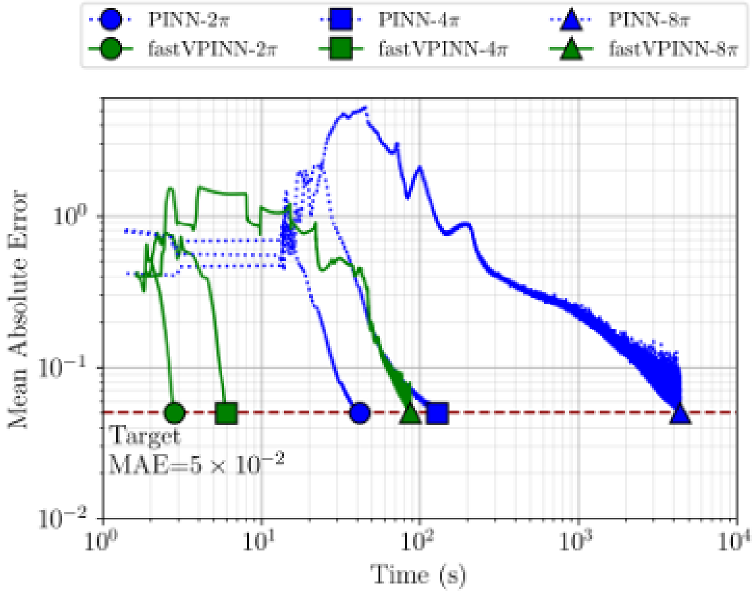
This problem has the exact solution

$$u(x,y) = -\sin(\omega x)\sin(\omega y) \tag{2}$$

結果



(a)



(b)

Figure 4: **(a)** (i) Variation of computational time with the number of quadrature (residual) points, plotted against the median time taken per epoch; (ii) Comparison of computational time between hp-VPINNs and fastVPINNs for varying numbers of cells. **(b)** Comparison between PINNs and fastVPINNS of total time taken to reach a target mean absolute error for different solution frequencies.

複雑な幾何学形状でのFastVPINNs

- 幾何学形状：２次元の歯車形状を約15,000セルにメッシュ分割
- 支配方程式は移流拡散方程式

$$
\begin{aligned}
-\varepsilon \Delta u + \mathbf{b} \cdot \nabla u &= f, \quad \mathbf{x} \in \Omega_{\text{gear}}, \\
u &= 0, \quad \mathbf{x} \in \partial\Omega_{\text{gear}},
\end{aligned}
\tag{12}
$$

where

$$
f = 50 \times \sin(x) + \cos(x); \quad \epsilon = 1, \quad \mathbf{b} = [0.1, \ \ 0]^T.
$$

複雑な幾何学形状でのFastVPINNs

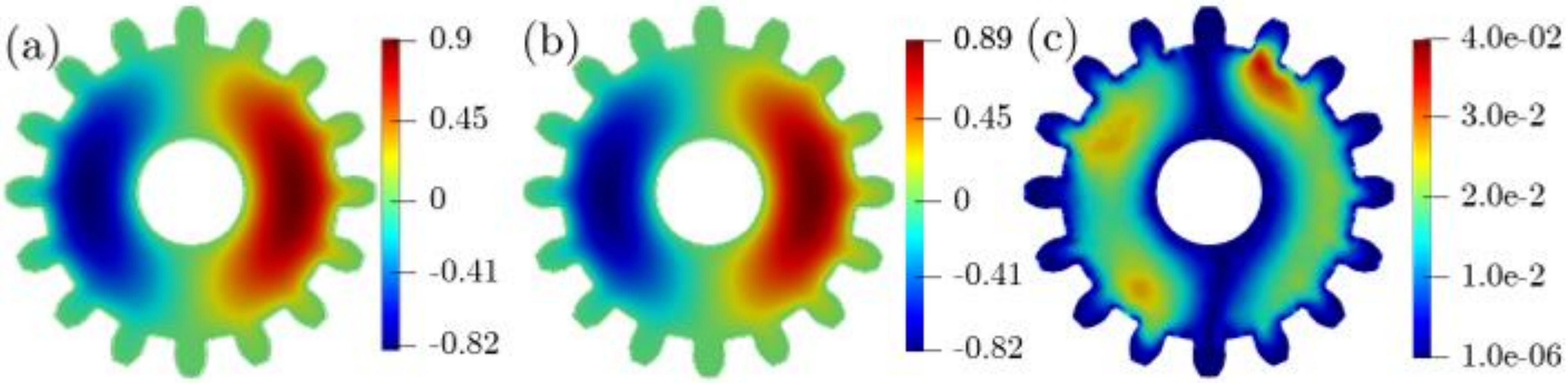


Figure 12: (a) Exact Solution obtained using FEM. (b) Predicted solution-FastVPINNs (c) Pointwise absolute error.

結論

- hp-VPINN は、高周波数解を捉えることに優れているが、要素数が多く複雑な形状を含む問題に対しては、学習時間の延長に苦戦している。
- 本論文では、テンソルベースの計算を採用することで、要素数に依存する学習時間を大幅に短縮し、複雑なメッシュを効率的に扱う新しいフレームワークであるFastVPINNsを導入した。その結果、既存のhp-VPINNsの実装と比較して100倍のスピードアップを実証した。
- さらに、適切なハイパーパラメータを選択することで、FastVPINNsは速度と精度の両方で最先端のPINNコードを凌駕した。
- このハイパーパラメータ分析は、特定の問題に最も適したハイパーパラメータを選択するための貴重な洞察結果である。
- FastVPINNs の多用途性は、流体力学のような分野での実世界での応用の可能性を広げる。将来的には、FastVPINNs の機能を三角形要素や３次元領域に拡張し、科学・工学分野での影響力をさらに広げることを目指している。

議論～疑問に思ったこと～

- 数値解析の手法（ガラーキン法など）をPINNsに適用する発想は面白い。しかし、アーキテクチャ図をみると、数値解析に似た収束計算を行っているように見える。PINNsの利点を損なってしまっていないか？

- 支配方程式に非等方性（例えば重力とか）がなければ、上下左右対称形状のシミュレーションは周期境界条件を使ってスケールダウンすることができる。そのため、上下左右対称の歯車形状を「複雑な幾何学形状」と表現することに違和感がある。非対称性の強い形状か、支配方程式に非等方性を入れてみる必要があるのではないか（そのかわり、評価が難しくなる）。

- 残差の空間分布をみると非等方性（左右も上下も）がみられるが、これは何に起因するのか？

議論～疑問に思ったこと～

- 結局「Fast」はテンソル演算を工夫することにミソがあった？BLASの利用や逆伝播計算の回数削減はどれくらい効果があったのか？
- 変分法を取り入れた仕組みをまだ完全に理解できていないため、PINNsに変分法を取り入れた目的・利点がはっきりわからない。論文で検討している支配方程式（移流拡散方程式）や幾何学形状（上下左右対称の歯車）は、変分法を取り入れるうまみが本当にあるのか？
- 損失関数の表現を変分法にする点については、より詳細に仕組みを調べたい。本当に変分法表現に利点があれば、他のアーキテクチャにも応用できないか。