

## 第5回とめ研究所若手研究者懸賞論文

流体力学シミュレーションの進展: 物理インフォームド  
ニューラルネットワークの最適化への包括的アプローチ

周文<sup>1</sup>, 三輪修一郎<sup>1</sup>, 岡本孝司<sup>1</sup>

<sup>1</sup> 東京大学大学院 工学系研究科 原子力国際専攻

## 1. はじめに

近年、有限要素法やスペクトル法、メッシュレス法により非圧縮および圧縮ナビエ-ストークス方程式 (NSE) を解くための計算流体力学 (CFD) の大きな進展が見られている [1-4]. しかしながら、従来の数値および解析手法には、依然として解決すべき課題がいくつか存在する. CFD で使用される数値手法は通常、境界条件の変動などの現実的な現象を考慮しない理想化された条件下で適用され、既存のアルゴリズムにシームレスかつ多様な精度のデータを統合することができない. さらに、非常に複雑な流体流れの問題では、グリッド生成が計算コストを高くする懸念がある. 同様に、逆問題 (例: 不明な境界条件) を解く場合にも、グリッド生成は計算コストが高く、実際の解析を行うためには異なるアルゴリズムやコードの開発が必要不可欠である.

AI を活用した複雑な流れのシミュレーションは、直接数値シミュレーション (DNS) の加速 [5-7], 低次元流モデル [8-10], 改良されたレイノルズ平均化ナビエ-ストークスモデルおよびサブグリッドスケールモデル [11-14], 流れのパターン分類 [15, 16] など、さまざまな応用技術を駆使して驚異的な進歩を遂げている. この新しい AI およびデータ駆動型の CFD アプローチのパラダイムは、グラフィックプロセッシングユニット, テンソルプロセッシングユニット, および機械学習技術の進展により、上述の問題を解決する可能性が示唆されている.

物理インフォームドニューラルネットワーク (PINNs) は、Raissi らによって提案され [17], 既知の物理法則や原理をネットワーク構造に組み込むタイプの AI フレームワークであるため、通常、非線形偏微分方程式 (PDE) の形で実装される. 標準的な AI モデル (例えば、畳み込みニューラルネットワークやリカレントニューラルネットワーク) は、予測と実際のデータの違いを損失関数を介して最小化することによって訓練することができるが、これらの従来のモデルはしばしば解釈可能性に欠け、ドメイン固有の知識を使用しない. 一方 PINNs は、損失関数に物理法則を表す微分方程式として対応する項を含めることによって生成される. その結果、疎なデータやノイズの多いデータでも、与えられた物理法則と一致する結果を予測できるニューラルネットワークを得ることが可能になる. したがって、PINNs は流れのモデリングおよび予測に新しい解釈可能性を提供しうることが期待されている.

従来の PINNs は、さまざまな流体動力問題に提案および適用されている. Cai ら [18] は、トモグラフィック背景指向シュリーレン (Tomo-BOS) イメージングデータから連続した 3D 速度および圧力場を推論するための PINNs を使用する方法を提案した. 彼らは、Tomo-BOS イメージングでキャプチャされた温度場を使用してニューラルネットワークを訓練し、隠れた速度および圧力場を推論することで、エスプレッソカップ上の流れを調べた. Jin ら [19] は、非圧縮流の問題を速度-圧力 (VP) および渦度-速度 (VV) 形式を含むナビエ-ストークス方程式 (NSE) に基づいてカバーする PINNs ベースの Navier-Stokes フローネット (NSFnet) を

開発した。彼らは、解析解および DNS データベースを使用して NSFnet シミュレーションの適切な初期および境界条件を設定した。結果は、NSFnet が VP または VV 形式のいずれかを使用して非圧縮流を効果的にシミュレートできることを示している。Eivazi ら[20]による別の研究は、Falkner-Skan 境界層を例として使用し、乱流および層流の両方に対する NSEs の解決における PINNs の有効性を示した。複雑な乱流条件にもかかわらず、彼らの PINNs ベースのシミュレーションは Reynolds-stress コンポーネントに対しても優れた精度を達成した。

Ji ら[21]は、硬い化学反応速度問題に取り組むための PINNs の有効性を調査した。常微分方程式 (ODE) の文脈では、「硬さ」という用語は解の成分が大きく異なるシステムを指し、通常直接数値シミュレーションでの数値的不安定性につながる。これは、ヤコビアン行列の最大の絶対固有値によってしばしば定量化され、この場合反応源項から種濃度に関連して導かれる。Ji らは、これらの ODE システムに固有の硬さを緩和するための戦略的アプローチとして準定常状態仮定を使用し、PINNs の成功裏の展開を促進した。Laubscher[22]は、単純な二次元矩形領域内での乾燥空気の加湿シナリオにおける運動量、種、および温度分布を予測するための PINNs の適用を探求した。その研究は、分離ネットワーク PINNs のパラメータ化されたバージョンを訓練し、異なる水蒸気入口速度に対する結果を生成できることを示した。Aliakbari ら[23]は、流体流れ、熱伝達、ポーラスメディア輸送問題を含むさまざまなマルチフィジックス偏微分方程式を解くためのマルチフィジティ PINNs アプローチを発表した。彼らは低フィジティデータを使用してマルチフィジックス CFD ソルバーによる低フィジティシミュレーションを実行し、PINNs を使用してデータに適合させた。その後、対応する初期化を使用して PINNs を初期化するために転移学習法を利用した。

近年、多くの研究が PINNs の改良に寄与し、それによって流体力学問題の広範なスペクトルにわたる有用性と適用性を拡大し、新しい挑戦的な条件下での PINNs の堅牢性と安定性を強化した。Liu ら[24]は、3 種類の PDE、すなわち楕円型、双曲型、および放物型の二次元 NSE に対する適応勾配降下アルゴリズムを備えた改良された PINNs を開発した。Cheng ら[25]は、流体流れ問題を解決するための ResNet ブロックを備えた改良された PINNs を提案した。Wong ら[26]は、流体力学システムの代理モデルを作成するための PINNs の適用を調査し、従来の Xavier 初期化と比較して収束速度が 3 倍に向上した新しい転移最適化スキームを提示した。Zhang ら[27]は、フーリエ特徴、すなわち周波数と位相のオフセットパラメータを PINNs に導入し、シミュレーション出力、基礎となる幾何学、境界条件との関係を効果的に記述することに成功している。

しかしながら、流体力学への応用における PINNs の改良を体系的に探求した研究はこれまでに実例がない。既存の文献は主に PINNs の特定の側面 (例: 最適化アルゴリズムやニューラルネットワークの内部構造) を最適化することに焦点を当

ている。さらに、これらの改良の多くはクローズドソースとして報告されており、十分な一般化性に欠けている。

上記の問題を背景として、本研究は、非圧縮ナビエ-ストークス流体をシミュレートするための改良された汎用 PINNs を体系的に開発することを目的としている。そのために、次の主要な改良を備えた改良された PINNs を提案する：(1) 残差に基づく適応サンプリングを採用し、データ処理フェーズでNSEの残差が大きい領域に自動的に多くの点をサンプリングするように改善した。(2) データ訓練フェーズで各サブ損失項をバランスさせるために適応損失を導入し、改善した。(3) 訓練フェーズでPINNsの最適なハイパーパラメータを見つけるために微分進化 (DE) 最適化アルゴリズムを適用した。

改良された PINNs の効率を評価するために、低レイノルズ数での3つの検証ケースを選択して検証を行った。結果は、改良された PINNs がNSEの流体動力解析において速度と圧力場を正確に記述できることを示し、改良された PINNs の効率を示している。さらに、各改良が最終シミュレーション結果に与える貢献の調査と定量化も行った。

本論文の構成は以下の通りである：第2節では、本研究で使用された PINNs のフレームワーク、NSEの支配方程式、およびすべての改良を紹介する。第3節では、3つのケースの検証結果を示し、改良された PINNs のシミュレーション性能を確認する。最後に、第4節では結論をまとめ、今後の研究課題を提案する。

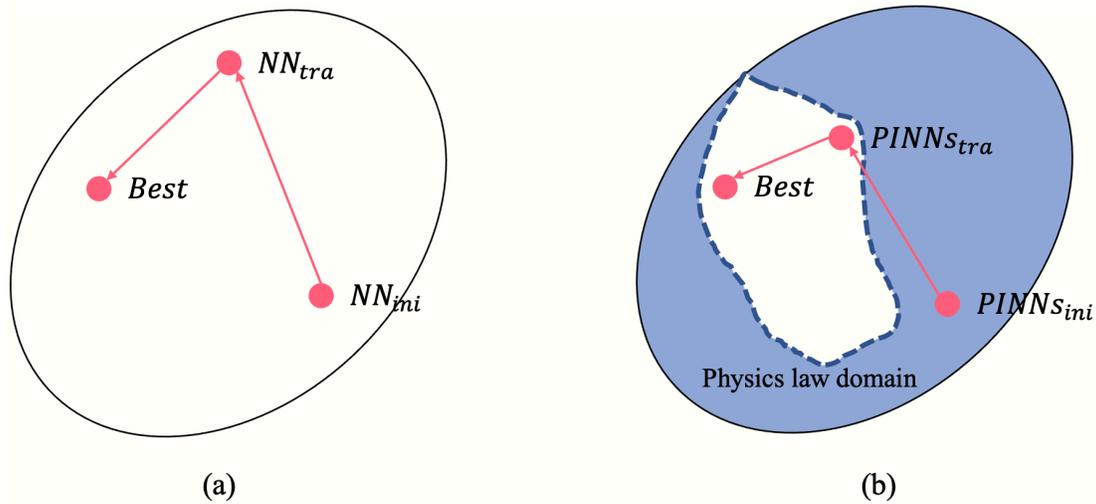
## 2 計算方法

### 2.1 物理インフォームドニューラルネットワーク (PINNs)

本質的に、ニューラルネットワークは普遍的な関数近似として説明することができ、画像認識や音声認識を含むさまざまな分野で成功を収めていることが証明されている。第3次AI時代、つまりディープラーニング技術の隆盛によって特徴付けられる時代まで、ニューラルネットワークは主にデータ駆動型のコンテキストで使用されていた。しかし、実際の科学的および工学的な応用においては、常微分方程式 (ODE) や偏微分方程式 (PDE) の形で物理知識が豊富に存在している。

一方、PINNs はこれらの物理法則をネットワーク構造に組み込むことで、この知識を活用する。これは、図1に示されるように、深層ニューラルネットワークを使用して解を近似することによって、与えられたPDEを解くことを目的としている。図において、NNini と NNtra はそれぞれ従来のニューラルネットワークの初期モデルと訓練およびチューニングによって得られたモデルを表している。同様に、PINNsini と PINNstra はそれぞれ PINNs の初期モデルと訓練およびチューニ

ングによって得られたモデルを表している。図 1(a) の楕円領域はすべてのモデルの解空間を示す。「Best」は、実際の問題設定における最適なモデル、すなわち無限のデータで得られる最適解を表す。図 1(b) の点線で囲まれた領域は、物理法則の下でのモデルの解空間を示しており、物理法則で制約されたモデルの解空間が小さくなり、訓練されたモデルが最適モデルからの距離が短いことを示す。したがって、PINNs は物理学とデータサイエンスの分野の概念を統合する新しいパラダイムを表している。



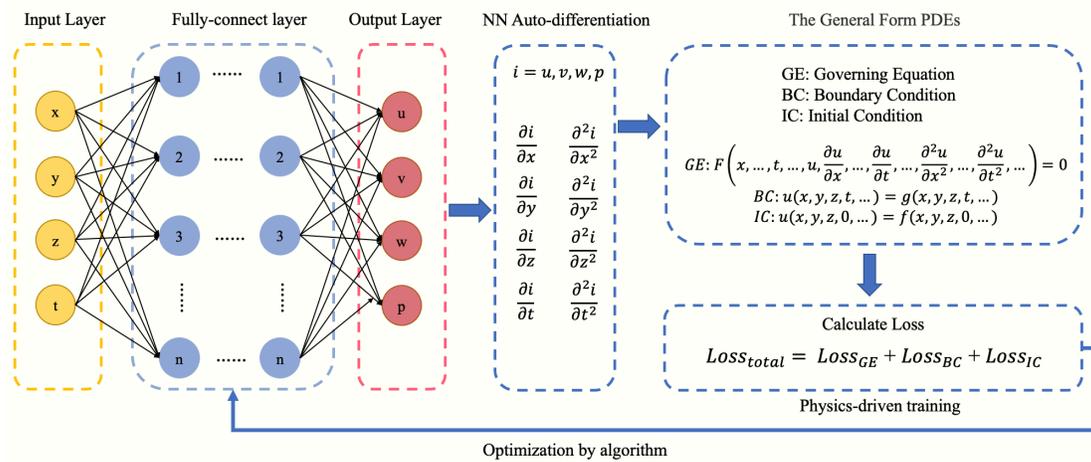
**Figure 1.** Illustration of a solution space for a NN model and a PINNs model. (a) Pure data-driven based NN model, (b) Physics law and data-driven based PINNs model.

図 2 は、PINNs のフローチャートと一般的な構造を示している。ここで、 $x$ ,  $y$ ,  $z$ ,  $t$  は座標と時間の入力であり、 $u$ ,  $v$ ,  $w$ ,  $p$  は PINNs からの三次元の速度と圧力の出力である。ニューラルネットワークの自動微分技術により、出力の微分形態を導き出し、それを偏微分方程式 (PDE) を表すために組み込むことが可能になる。PDE は、複数の変数とその偏導関数を含む数学的方程式であり、関数  $u(x, y, z, t, \dots)$  の基本的な PDE の形は、図 2 の PDE の一般形の GE (一般形方程式) として記述する。ここで、 $F$  は独立変数 ( $x, y, z, t, \dots$  など)、従属変数  $u$ , およびその一定次数までの偏導関数を含む関数である。

境界条件 (BC) は、PDE を解くために必要な制約である。これらは、方程式が定義されている領域の境界上での解  $u$  の振る舞いを指定する (図 2 の PDE の一般形の BC を参照)。初期条件 (IC) は、時間に依存する PDE で使用される。これらは初期時点でのシステムの状態を指定しており、関数  $u(x, y, z, t)$  についての初期条件は、図 2 の PDE の一般形の IC に示す。

PINNs は、GE, BC, および IC から計算された損失を利用してネットワークのパラメータを調整する。損失関数を定義すると、これは PINN の予測と既知の物理法則との間の不一致を定量化するものであり、最適化アルゴリズムは確率的勾配降下法やその変種（例：Adam オプティマイザー）などの勾配に基づく手法を使用する。このアルゴリズムは、各パラメータの損失勾配を計算することにより、逐次的に PINN の重みを調整し、それによってモデルの精度を向上させる。

PINNs のパラメータを勾配の逆方向に反復的に更新することで、PINN は徐々に GE で表される基礎的な物理法則に適合するように学習し、指定された BC と IC も満たす。この反復的な改良は、損失が最小化されるまで続き、ネットワークの予測が問題の物理的制約と調和していることを示す。したがって、訓練プロセス中、PDE は本質的に損失項として機能し、最終的にニューラルネットワークを物理的に妥当な解へと導く。



**Figure 2.** Flow chart and structure of the PINNs.

## 2.2 支配方程式

本研究では、非圧縮、等温、および不混和の流れを考慮する。連続の式と運動方程式はそれぞれ以下のように表される：

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = f - \frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{u} \quad (2)$$

ここで、 $\rho$ ,  $u$ ,  $p$ ,  $f$ , および  $\mu$  はそれぞれ流体の密度, 速度ベクトル ( $u$ ,  $v$ ,  $w$ ), 圧力, 流体単位体積あたりの外力, および動粘性係数を示す. 式 (1) で表される連続の式は, 質量保存の法則を表現している. 非圧縮流体の場合, 速度ベクトル  $\mathbf{u}$  の発散はゼロである. これは, 流体の密度が一定であり, 流体の流れの中のどの微小体積内にも質量の蓄積や不足がないことを意味する. これにより, 流体が圧縮も膨張もしていないことが保証される. 式 (2) で表される運動方程式は, 流体粒子の速度が時間と共にどのように変化するかを表している.  $\partial \mathbf{u} / \partial t$  の項は局所加速度を示し, 特定の場所で流体粒子の速度が時間と共にどのように変化するかを反映している. 対流加速度項  $(\mathbf{u} \cdot \nabla) \mathbf{u}$  は流体の非線形的な側面を捉え, 流体自身の動きによる速度の変化を表している. 外力項  $f$  は, 重力や電磁力などの流体単位体積あたりに作用する外力を考慮する.  $-1/\rho \nabla p$  の項は, 流体に作用する圧力勾配による力を表し, 高圧から低圧への流れを促す. 最後に, 粘性散逸項  $\mu/\rho \nabla^2 \mathbf{u}$  は, 速度場のラプラシアンを含み, 流体の粘性による内部摩擦を反映している.

本研究では, 改良された PINNs を用いて NSEs (ナビエ-ストークス方程式) を解く効率を探求する. つまり, この PINNs は空間的および時間的座標を入力として, 対応する速度および圧力場を予測することにより, NSEs の解を近似するために利用される (図 2 参照).

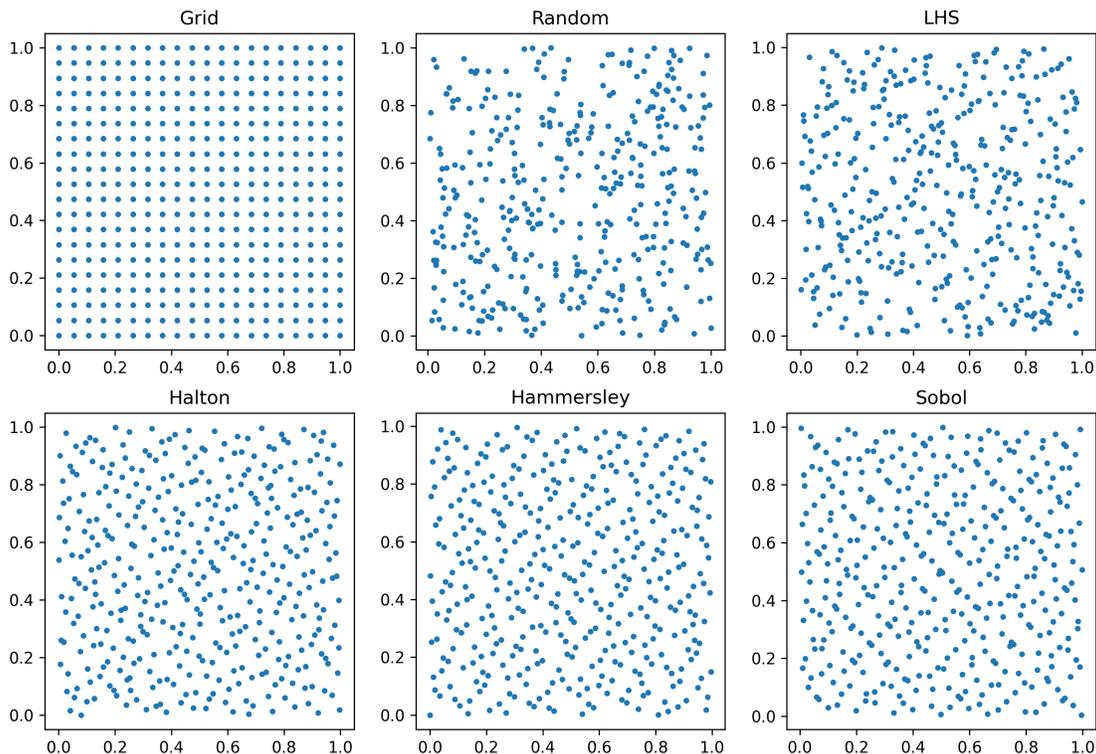
## 2.3 ネットワークの改良

### 2.3.1 残差に基づく適応サンプリング

データセットの品質は, 機械学習モデルの予測性能を決定する上で重要な役割を果たす. PINNs のコンテキストでは, データセットの品質はサンプリングポイントの選択に依存する. 初期条件および境界条件のサンプリングポイントは教師あり学習と見なされ, 既知のデータ量と計算資源に依存して選択される.

一方で, PDEs のサンプリングポイントは教師なし学習と見なされる. 直感的には, PDEs のサンプリングポイントの効果は有限要素法におけるメッシュポイントの効果に似ている. なぜなら, PINNs は主に PDE の損失に基づいて最適化されており, 訓練されたネットワークが PDEs に一致するようにしているからである. したがって, PDEs のサンプリングポイントの選択は, PINNs の性能に大きな影響を与える. 図 3 は, 一般的な 6 つのサンプリング方法を示している. その中で,

グリッドサンプリングは多次元空間を規則的なグリッドに離散化する決定論的な方法である。このアプローチは、ポイント間の間隔が一定であるため、空間の均一なカバレッジを保証する。ランダムサンプリングは、空間からランダムに独立してサンプルポイントが選ばれる確率的な特性を持つ。ラテンハイパーキューブサンプリング (LHS) は、空間の各パーティションがサンプリングされることを保証することでランダムサンプリングを強化する。各次元  $i$  について、その範囲を等確率の  $N$  区間に分割する。そして、各区間にはランダムに1つのサンプルしか含まれないように、次元  $i$  の  $j$  番目の区間内に  $x_i$  が落ちるようにサンプリングする。ハルトン列は、準ランダムサンプリングに使用される低分散列であり、各次元に異なる素数を使用し、再帰的に列を生成する。与えられた素数  $p$  について、基数  $p$  でのハルトン列の  $n$  番目の要素は  $\sum_{k=0}^{\infty} a_k p^{-k-1}$  であり、 $n = \sum_{k=0}^{\infty} a_k p^k$  かつ  $a_k$  は  $n$  の基数  $p$  表現での係数である。ハマースレーセットはハルトン列に似ており、より均一なポイント分布を保証する。d 次元の  $N$  ポイントのうち  $i$  番目について、最初の次元は  $i/N$  であり、他の次元は異なる基数のハルトン列で与えられる。ソボル列は、方向数の概念に基づいた別の低分散の準ランダム列であり、二進行列を使用して列を生成し、単位超立方体内で均等に分布したポイントを生産するために使用される。これら6つのランダムサンプリング方法はサンプルの均一性を保証するが、その適応性は保証しない。言い換えれば、どこでより多くのサンプリングを行うべきか、どこで少ないサンプリングが十分であるかを決定するものではない。



**Figure 3.** Six uniform sampling methods for 400 sampling points generated in  $[0,1]^2$ .

残差に基づく適応サンプリングは、急勾配を持つ解の特定の偏微分方程式 (PDE) を解くための PINNs の性能を向上させる [28]. これは、すべてのサンプリングポイントが PDE の残差  $\varepsilon(x)$  に比例する確率密度関数 (PDF)  $p(x)$  に従って再サンプリングされることを意味する.

$$p(\mathbf{x}) \propto \varepsilon(\mathbf{x}) \quad (3)$$

具体的には、任意の時空間点  $\mathbf{x}(x, y, z; t)$  に対して、PDE の残差は次のようになる.

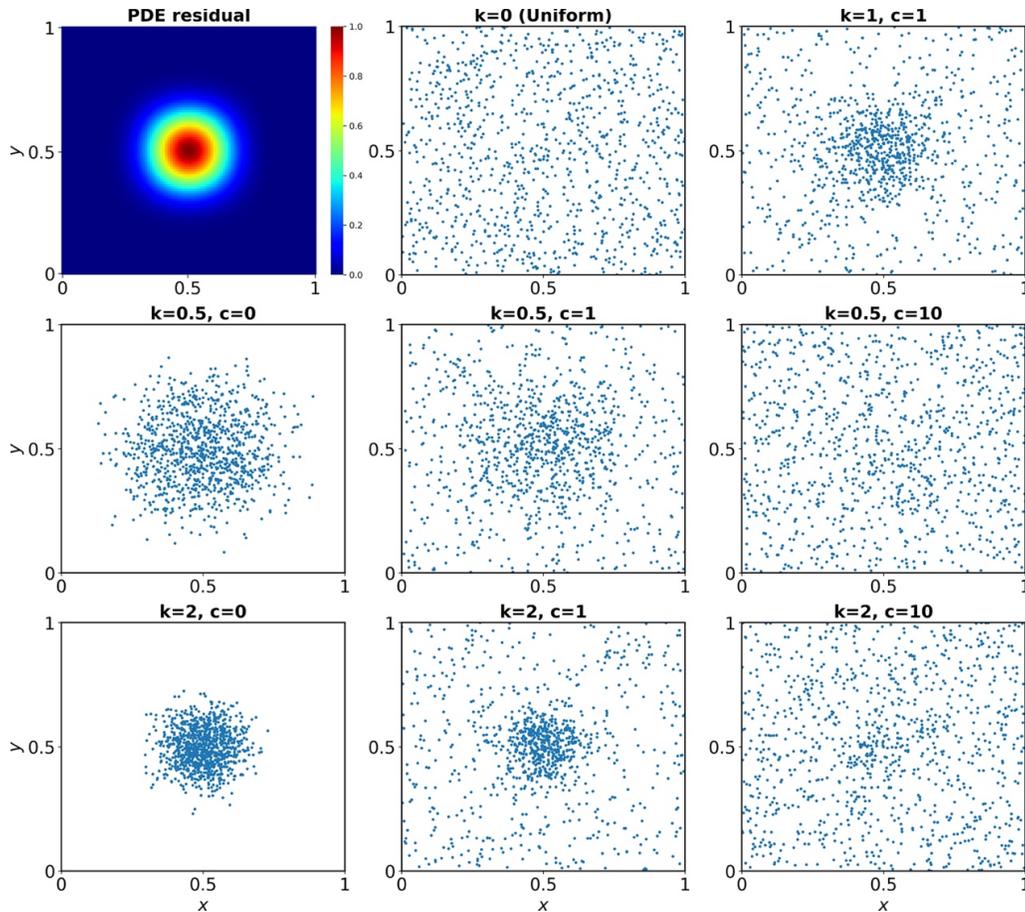
$$\varepsilon(\mathbf{x}) = |f(\mathbf{x}; \hat{u}(\mathbf{x}))| \quad (4)$$

ここで、 $f(x)$  は PDE と訓練された PINNs  $\hat{u}(x)$  によって計算された時空間点  $\mathbf{x}$  の残差値を示す.

多数の残差に基づく適応サンプリング方法が提示され、比較されており、新しい PDF が定義されている [29] :

$$p(\mathbf{x}) \propto \frac{\varepsilon^k(\mathbf{x})}{\mathbb{E}[\varepsilon^k(\mathbf{x})]} + c \quad (5)$$

ここで、 $k \geq 0$  および  $c \geq 0$  は 2 つのハイパーパラメータである.  $\mathbb{E}[\varepsilon^k(\mathbf{x})]$  はモンテカルロ積分などの数値積分によって近似できる. 図 4 は  $k$  と  $c$  の効果を示している. 本研究では、 $k=0.5$  と  $c=1$  が使用されている.



**Figure 4.** Example plots with different values of  $k$  and  $c$  obtained using Eq. (5).

一部のサンプリングポイントは大きな残差を示すことがあるにもかかわらず、多くの既存データポイントが存在するため、選択される確率が低いままである。その結果、Wu ら[29]は、サンプリングプロセスの収束を確実にするために、複数回の再サンプリングを推奨した。

PINNs を用いてナビエ-ストークス方程式 (NSE) をシミュレーションする場合、2 つの課題に対処する必要がある。(1) 必要なサンプリングポイントの数が多いため、上述のプロセスは計算コストが高くなる可能性がある。したがって、残差に対して指数関数を使用することを提案し、サンプリングプロセスを加速させる。(2) NSE には複数の PDE セットが存在するため、これらのセットの残差の大きさは大幅に異なる可能性がある。そのため、PDE セット内の各方程式が適切にサンプリングされるようにするために、セットベースの残差に基づく適応サンプリングを提案する。

したがって、PINNs に基づく NSE のシミュレーションに適用可能なサンプリング方法を次のように提案する： ステップ 1：ソボルサンプリングに基づいてサンプリングポイントのセット  $S_0$  を生成，ステップ 2： $S_0$  に基づいて PINNs を訓練，ステップ 3：ステップ 2 で訓練されたモデルを使用して，ソボルサンプリングに基づいて密なサンプリングポイントセット  $S_1$  をサンプリングし，PDE の残差を計算，ステップ 4：ステップ 3 の残差に基づいて，指数関数と PDF サンプリングのグループ化手法を使用して再サンプリングポイント  $S_2$  を取得，ステップ 5：最終的なサンプルポイント  $S_2 \cup S_0$  を生成し，以降の PINNs で使用する。

提案された残差に基づく適応サンプリング手法は，サンプリングの均一性を確保するだけでなく，残差が大きい領域でサンプルポイントを優先し，サンプリングの収束プロセスを加速する。

### 2.3.2 適応損失

PINNs ベースの NSE の損失は，以下のように，支配方程式 (GE)，境界条件 (BC)，初期条件 (IC) で表される 3 つのサブロスに分けられる：

$$\begin{aligned}
 Loss_{total} &= Loss_{GE} + Loss_{BC} + Loss_{IC} \\
 Loss_{GE}(\theta; N_{GE}) &= \frac{1}{|N_{GE}|} \sum_{j=1}^{N_{GE}} \|f(x_{GE}^j, t_{GE}^j; \theta)\|^2 \\
 Loss_{BC}(\theta; N_{BC}) &= \frac{1}{|N_{BC}|} \sum_{j=1}^{N_{BC}} \|\hat{u}(x_{BC}^j, t_{BC}^j; \theta) - u(\mathbf{x})_{BC}^j\|^2 \quad (6) \\
 Loss_{IC}(\theta; N_{IC}) &= \frac{1}{|N_{IC}|} \sum_{j=1}^{N_{IC}} \|\hat{u}(x_{IC}^j, 0; \theta) - u(\mathbf{x})_{IC}^j\|^2
 \end{aligned}$$

ここで， $\theta$  は PINNs のパラメータを表し， $u_{BC}$  と  $u_{IC}$  は境界条件および初期条件の真の値を表し， $N_{GE}$ ,  $N_{BC}$ ,  $N_{IC}$  はそれぞれ PDE，境界条件，および初期条件のサンプリングポイントを表す。

異なる損失項の大きさの違いに対処するために，これらをバランスするための重みパラメータが導入される。

$$Loss_{total} = W_{GE} Loss_{GE} + W_{BC} Loss_{BC} + W_{IC} Loss_{IC} \quad (7)$$

しかし、PINNs の訓練が進むにつれて、固定された重み値を使用する従来のアプローチでは、依然として異なる大きさの損失項が発生する可能性がある。したがって、損失の重みを自動的に学習し、PINNs の精度とロバスト性を向上させるより効率的な方法を開発することが望ましい。

適応損失バランス法は、多タスク深層学習シナリオにおける不確実性のガウスの尤度を最大化することで複数の損失関数を適用した研究から着想を得た[30, 31]. PINNs では、さまざまな損失項が複数の最適化タスクと見なされ、それらはすべて総損失の最小化を目指す。PINNs はデータの変動性やモデルアーキテクチャの制限などの要因による不確実性を内在して予測するため、適応損失バランス戦略が採用される。PINNs の適応損失バランスでは、ガウスの尤度が PINNs の近似  $\hat{u}(x)$  と不確実性パラメータ  $\varepsilon$  によって与えられる平均として定義される。

$$p(u(\mathbf{x}) | \hat{u}(\mathbf{x})) = N(\hat{u}(\mathbf{x}), \varepsilon^2) \quad (8)$$

$\varepsilon$  は PINNs の訓練されたパラメータとして導入および調整され、総損失を最小化することを最終目標としている。モデルの負の対数尤度は式(9)を使用して最小化される[30]：

$$\begin{aligned} -\log p(u(\mathbf{x}) | \hat{u}(\mathbf{x})) &\propto \frac{1}{2\varepsilon^2} \|u(\mathbf{x}) - \hat{u}(\mathbf{x})\|^2 + \log \varepsilon \\ &= \frac{1}{2\varepsilon^2} \text{Loss} + \log \varepsilon = W(\text{Loss}) \end{aligned} \quad (9)$$

さらに、ガウス確率モデルの出力は、各々がガウス分布に従う 2 つのベクトル  $u_{BC}$ ,  $u_{ICuBC}$ ,  $u_{IC}$  で構成される：

$$\begin{aligned} p(u(\mathbf{x})_{BC}, u(\mathbf{x})_{IC} | \hat{u}(\mathbf{x})) &= p(u(\mathbf{x})_{BC} | \hat{u}(\mathbf{x})) \cdot p(u(\mathbf{x})_{IC} | \hat{u}(\mathbf{x})) \\ &= N(\hat{u}(\mathbf{x}), \varepsilon_{BC}^2) \cdot N(\hat{u}(\mathbf{x}), \varepsilon_{IC}^2) \end{aligned} \quad (10)$$

これは、境界条件および初期条件に対応する  $u(x)_{BC}$ ,  $u(x)_{ICuBC}$  および  $u(x)_{IC}$  の最小化目標をもたらす：

$$\begin{aligned}
-\log p(u(\mathbf{x})_{BC}, u(\mathbf{x})_{IC} | \hat{u}(\mathbf{x})) &\propto \frac{1}{2\varepsilon_{BC}^2} \|u(\mathbf{x})_{BC} - \hat{u}(\mathbf{x})\|^2 + \frac{1}{2\varepsilon_{IC}^2} \|u(\mathbf{x})_{IC} - \hat{u}(\mathbf{x})\|^2 + \log \varepsilon_{BC} \varepsilon_{IC} \\
&= \frac{1}{2\varepsilon_{BC}^2} Loss_{BC} + \frac{1}{2\varepsilon_{IC}^2} Loss_{IC} + \log \varepsilon_{BC} \varepsilon_{IC} = W_{BC}(Loss_{BC}) + W_{IC}(Loss_{IC}) \quad (11)
\end{aligned}$$

前述の研究は式(11)の正規項をさらに最適化し、式(12)で示される[30, 31]. したがって、改良されたPINNsの現在の目標は、すべての損失を最小化するための最適なパラメータ  $\theta$  と適切な適応重み  $\frac{1}{2\varepsilon^2}$  を見つけることである.

$$\begin{aligned}
-\log p(u_{BC}, u_{IC} | \hat{u}(\mathbf{x})) &\propto \frac{1}{2\varepsilon_{BC}^2} Loss_{BC} + \frac{1}{2\varepsilon_{IC}^2} Loss_{IC} + \log(\varepsilon_{BC}^2 + 1) + \log(\varepsilon_{IC}^2 + 1) \quad ( \\
&= W_{BC}(Loss_{BC}) + W_{IC}(Loss_{IC}) \quad (12)
\end{aligned}$$

NSEs に適応損失を適用するには、次の要素を考慮する：これらの 3 つの損失 (LossGE, LossBC, LossIC) は同じ桁数でないかもしれないだけでなく、速度ベクトル場 ( $u, v, w$ )、質量 ( $m$ )、および圧力場 ( $p$ ) の損失も異なる条件では異なる桁数である可能性がある。したがって、NSEs の各サブロスおよび残差に対して適応重みが導入される。

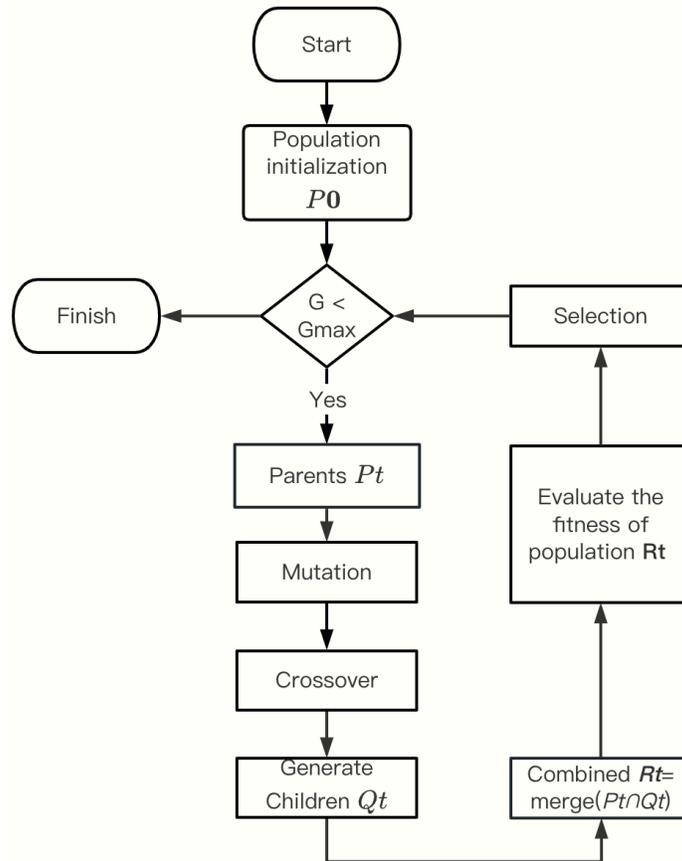
$$\begin{aligned}
Loss_{total} &= Loss_{GE} + Loss_{BC} + Loss_{IC} \\
Loss_{GE} &= \sum_{i=1}^{S_i} W_{S_i} (Loss_{GE_{S_i}}), S_i = \{u, v, w, m\} \\
Loss_{BC} &= \sum_{i=1}^{S_i} W_{S_i} (Loss_{BC_{S_i}}), S_i = \{u, v, w, p\} \\
Loss_{IC} &= \sum_{i=1}^{S_i} W_{S_i} (Loss_{IC_{S_i}}), S_i = \{u, v, w, p\}
\end{aligned} \quad (13)$$

式(13)は、適応重み  $W$  が増加すると、LossGE, LossBC, LossIC に対する罰則効果が大きくなることを示す。不確実性パラメータ  $\varepsilon$  は保護手段として機能し、 $W$  が過度に増加するのを防ぐ。このメカニズムは、NSEs に基づく PINNs の訓練中に各損失項の重みを自動的に調整し、数値安定性を促進する。

### 2.3.3 PINNs のパラメータを最適化するための DE 最適化アルゴリズム

PINNs は本質的に完全に接続された構造を含んでいるため、隠れ層、ノード、活性化関数、学習率など、多くのパラメータの調整が必要である。これは計算コストが高く、時間がかかるプロセスである。これに対する潜在的な解決策は、差分進化 (DE) 最適化アルゴリズムの適応である。

Storn と Price によって最初に提案された [32] DE アルゴリズムは、複雑なブラックボックス最適化問題を解くための最も効果的な方法の 1 つである。これは、与えられた集団の質の尺度に基づいて候補解を改善することを反復的に試みることによって機能する。DE アルゴリズムは、生物進化の原理をエミュレートし、適合度関数に最も適した個体を連続する反復を通じて保持することによって最適化を達成する。このアルゴリズムは、突然変異、交叉、および選択操作を介して新しい候補解を生成することによって動作する。その堅牢性と適応性がさまざまな領域で示されているため、DE アルゴリズムを使用して PINN パラメータを最適化することは有望であると判断される。図 5 は、DE アルゴリズムのフローチャートを示す。



**Figure 5.** Flow chart of the DE algorithm.

DE の主な強みは、解空間を効果的に探索し、最適なパラメータセットを特定する能力にあり、それにより PINNs モデルの性能を大幅に向上させることができる。DE 最適化アルゴリズムを活用することで、パラメータ選択を自動化し、調整時間を短縮し、したがって PINNs モデリングプロセス全体の効率を向上させることができる。

本研究では、隠れ層の数、各層のノード数、活性化関数、および初期学習率が最適化変数として考慮されている。一方、PINNs の損失が最適化プロセスでの最小化目標として使用される。DE アルゴリズムの設定は、表 1 に詳細を示す。

**Table 1.** Configuration of the DE algorithm.

Item	Value
------	-------

---

Population size	10
Maximum number of generations	10
Initial differential scale factor F	0.5
Crossover probability CR	0.5
Domain of optimization variables	hidden layers $\in$ [3, 4, 5, 6, 7, 8, 9, 10, 11, 12] nodes $\in$ [30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110, 105, 120] activation function $\in$ [Softsign, Softplus, Tanh, Tanhshrink, Tanhshrink, RReLU] initial learning rate $\in$ [0.0001, 0.001, 0.01, 0.1, 1]

---

### 2.3.4 追加の改良

改良された PINNs には、いくつかの追加の改良が適用されている。

#### (1) 適応的な学習率減衰

予備実験中に、固定学習率がデータ学習に対する PINNs の性能に大きな影響を与えることがわかった。文献レビューにより、以前の研究の大半が学習率減衰法を使用していたことが示されたが、それらは主に固定エポック内で固定学習率を持つ方法に焦点を当てていた。改良された PINNs の目的は、学習率に対する人工的な介入を最小限に抑えることである。そのため、新しい適応的な学習率減衰法が使用され、損失が特定のステップ内で 0.0001 以上減少しない場合、学習率が 0.1 倍される。

#### (2) 自動混合精度 (AMP)

AMP は、NVIDIA が PyTorch ディープラーニングライブラリと提携して導入した革新的な技術であり、ディープニューラルネットワークの訓練における計算効率を向上させるためのものである [33]。AMP は、マスターウェイトを float32 形式で保持しながら、順伝播および逆伝播のテンソルを計算のために float16 に変換して動作する。この混合精度戦略により、メモリ使用量が削減され、float16 データ型が小さいため計算が高速化される。

#### (3) PINNs のマルチプロセッシング

DE 最適化アルゴリズムと PINNs の統合は、必然的に計算時間の増加をもたらす。この増加を軽減するために、Python のマルチプロセスモジュールを使用して PINNs のマルチプロセスバージョンが開発された。

上記のすべての改良は Python と Pytorch に基づいて実装され、詳細な説明とともにすべてのケースを訓練および視覚化するためのフルコードは Github で利用可能.

( <https://github.com/zhouzhouwen/An-improved-PINNs-with-the-adaptive-weight-sampling-and-DE-algorithm.git>)

### 3 検証

このセクションでは、従来の PINNs と新たに改良された PINNs を低レイノルズ数での 3 つの異なるケースを使用して比較した。これらのケースは、改良された PINNs が NSE を処理する能力を検証し強調するために選ばれた。精度は、正確な値  $y_i$  と従来および改良された PINNs によって推測された予測値  $\hat{y}_i$  の間の平均絶対パーセンテージ誤差 (MAPE) によって評価された。

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (14)$$

Linux システムには、NVIDIA RTX A6000 48GB GPU メモリと Intel i9-10980XE CPU × 36 が使用された。

#### 3.1 ケース 1 : 2D 定常 Kovasznay 流

2D 定常 Kovasznay 流は、2 次元 NSE (式 (15) および (16) ) を解く数値コードの精度を検証するために頻繁に使用される。Kovasznay 流は、式 (17) , (18) , (19) , および (20) に示されるように、非圧縮性 NSE の正確な解を持つ[34].

$$\nabla \cdot \mathbf{u} = 0 \quad (15)$$

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (16)$$

$$u(x, y) = 1 - e^{\lambda x} \cos(2\pi y) \quad (17)$$

$$v(x, y) = \frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y) \quad (18)$$

$$p(x, y) = \frac{1}{2} (1 - e^{2\lambda x}) \quad (19)$$

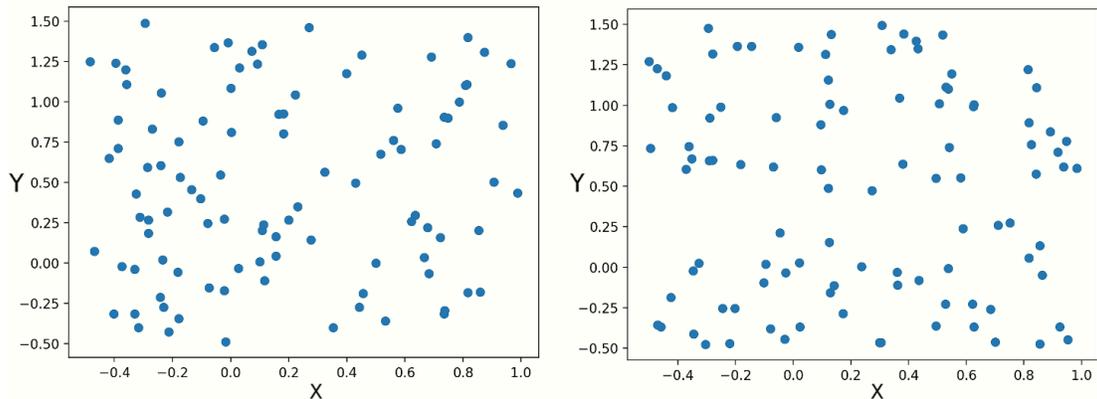
$$\lambda = \frac{1}{2V} - \sqrt{\frac{1}{4V^2} + 4\pi^2}, V = \frac{1}{\text{Re}} = \frac{1}{40} \quad (20)$$

このケースの設定は表 2 に示す。解析にはレイノルズ数 40 が使用した。

**Table 2. The configuration for 2D steady Kovaszny flow.**

Item	Value
Computational domain	$X \in [-0.5, 1], Y \in [-0.5, 1.5]$ , continuous value
Re	40
Number of BC sampling points	40 (10 points equally spaced on each side)
Number of PDEs sampling points	100
Number of training data	100+40 = 140
Number of testing data	300 x 300 = 90,000
Maximum epochs	10,000

図 6 に示すように、従来の PINNs では、ケース 1 のポイント生成にランダムサンプリングが使用されたが、改良された PINNs では、残差に基づく適応サンプリングが生成された。



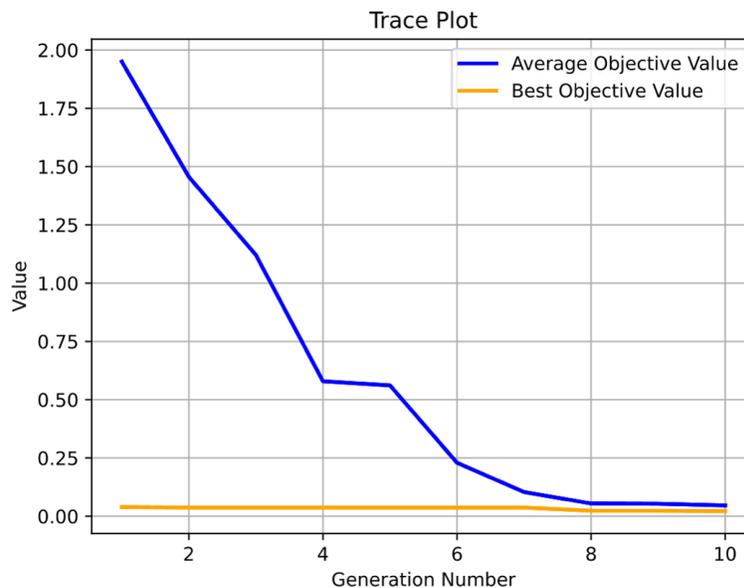
**Figure 6.** Sampling points for Case 1: residual-based adaptive sampling (left), and random sampling (right).

従来の PINNs では、全結合ニューラルネットワーク構造は 8 層、各層に 100 ノード、学習率は 0.001、活性化関数として双曲線正接 (Tanh) が設定されていた。

一方、改良された PINNs では DE 最適化アルゴリズムを使用して、表 1 で示された範囲内で最適なパラメータを見つけた。最適化プロセスでは、集団サイズを 10 に設定し、最大世代数を 10 に設定し、全体の探索プロセスで 100 の PINNs モデルを計算した。

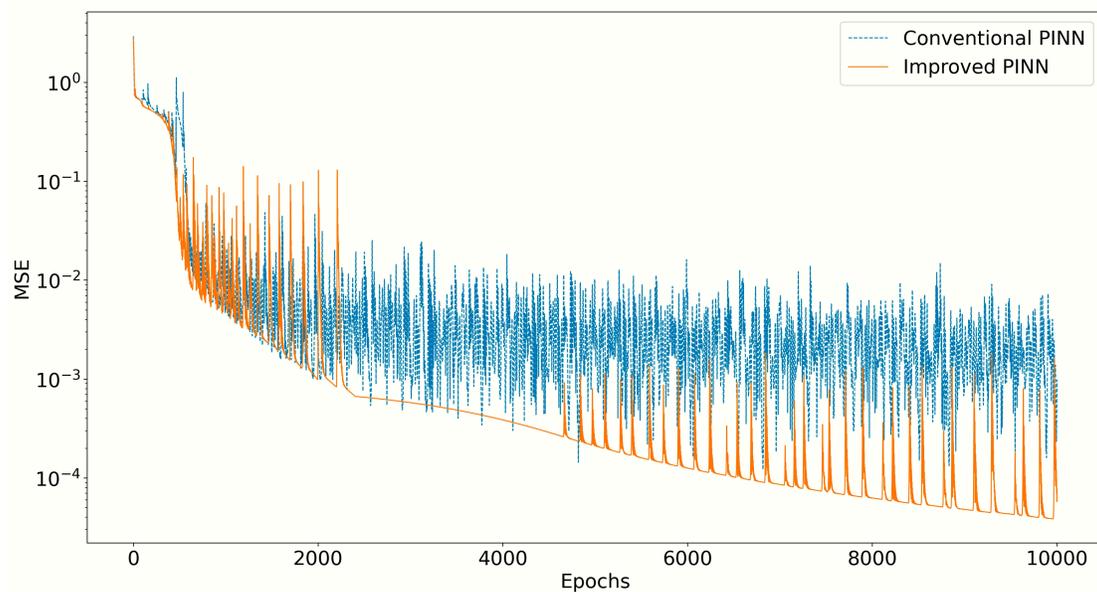
最適化プロセスは図 7 に示されており、x 軸は最適化の世代数を表し、y 軸は最適化の目的値、具体的には改良された PINNs の損失関数を示す。世代を重ねるごとに平均値と最良の目的値が一貫して減少していることから、PINNs フレームワーク内での DE 最適化アルゴリズムの有効性が示されている。8 世代目以降、最適化アルゴリズムが収束する傾向があることがわかる。最適化の結果、第 10 世代の 92 番目の進化モデル（6 層、各層に 60 ノード、学習率 0.001、Tanh 活性化関数）が最も低い損失を示し、したがってケース 1 の最適なパラメータと見なされる。

その他のハイパーパラメータは、従来の PINNs と改良された PINNs で一貫している。例えば、両モデルとも Xavier 初期化と Adam オプティマイザを使用した。ケース 1 のこれらのプロセスと設定は、ケース 2 およびケース 3 の設定とも一致している。



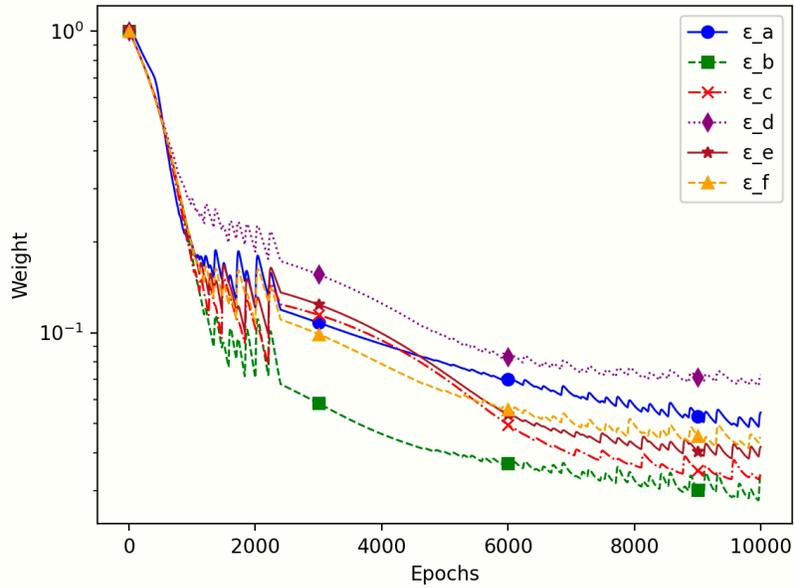
**Figure 7.** Optimization process of DE algorithm to find the optimal parameter in PINNs for Case 1.

図 8 は、ケース 1 における従来の PINNs と改良された PINNs の平均二乗誤差 (MSE) を示す。改良された PINNs はより小さな誤差を示し、優れた学習能力とナビエ-ストークス方程式 (NSE) を解く能力を持っている。改良された PINNs と従来の PINNs は、それぞれ 10,000 エポックの計算に 356 秒と 347 秒を要した。



**Figure 8.** Training loss curves of the conventional and improved PINNs for Case 1.

図 9 は、ケース 1 における各サブロスに対応する適応的な重みの変化を示す。モデルによって重み項が学習され、より低い訓練誤差を達成していることが観察される。ここで、 $\epsilon_a$ ,  $\epsilon_b$ ,  $\epsilon_c$ ,  $\epsilon_d$ ,  $\epsilon_e$ , および  $\epsilon_f$  はそれぞれのサブロスに対応し、境界条件の  $u, v, p$  と PDE の  $u, v, m$  を表す。訓練の終了時点で、 $\epsilon_d$  が最も大きな値を持ち、PDE の  $u$  に最も小さな重みが適用されることを意味し、ペナルティと制約が少なくなる。対照的に、 $\epsilon_b$  は最も小さな値を持ち、境界条件の  $v$  に最も大きな重みが適用され、大きなペナルティと制約を示す。適応的な重みの変化により、改良された PINNs は各訓練段階で異なるロスに適応的な重みを割り当てる能力を持つため、より効率的に収束することができると解釈される。



**Figure 9.** Variation of the adaptive weights for Case 1.

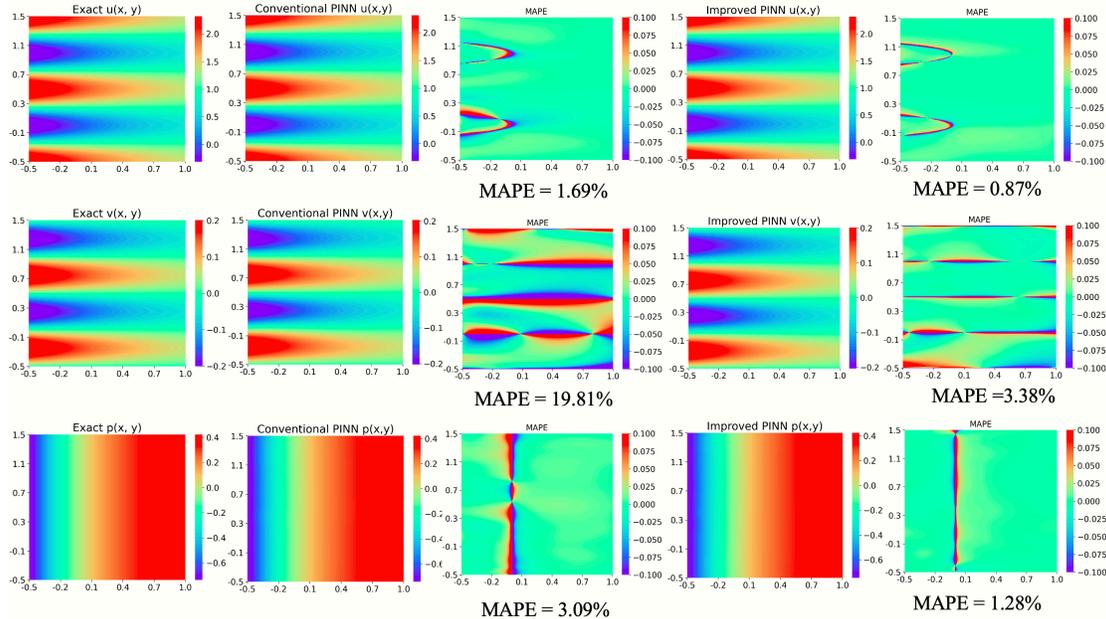
表 3 は、ケース 1 における従来の PINNs と改良された PINNs の最良モデルのエポックおよび対応するサブロスの訓練結果をまとめたものである。改良された PINNs の各サブロスは、従来の PINNs のサブロスよりも小さい。

**Table 3. Training results of the best model of the conventional and improved PINNs for Case 1.**

	Epoch	Total	pde_u	pde_v	pde_m	bc_u	bc_v	bc_p
Conventional PINNs	6551	1.09E-04	5.16E-05	1.47E-05	1.41E-05	1.85E-05	4.18E-06	6.06E-06
Improved PINNs	9952	3.86E-05	1.91E-05	3.64E-06	4.63E-06	8.06E-06	1.10E-06	2.09E-06

Kovasznyay 流のケースは、 $300 \times 300$  のグリッドに分割してテストされた。従来の PINNs と改良された PINNs のシミュレーション結果は図 10 に示す。ケース 1 における改良された PINNs のシミュレーション性能は従来の PINNs を上回り、各

MAPE も従来の PINNs より小さいことが示されており，改良された PINNs の効率の高さを強調している。



**Figure 10.** Simulation results for conventional and improved PINNs for Case 1.

Images in the first, second, and fourth columns are the velocity and pressure fields obtained for the exact solution, the conventional PINNs, and the improved PINNs, respectively. Images in the third and fifth columns are the MAPEs of the conventional PINNs and the improved PINNs, respectively.

最終結果に対する各改良の寄与を定量化するために，各改良を個別にケース 1 のシミュレーションに含め，定量化の結果を表 4 に示した．表の結果は，改良された PINNs に組み込まれた各改良の有効性を明確に示す．DE アルゴリズムのみに基づく PINNs の場合， $v$  の誤差は  $u$  の誤差よりもはるかに大きく，これはランダムなサンプリングポイントの選択が  $x$  および  $y$  方向で異なる誤差伝播を引き起こす可能性があるためかもしれない．サンプリングポイントが最も強い速度勾配の方向に整列していない場合，誤差が増加する可能性がある．残差に基づく適応サンプリングと適応損失は，PINNs の構造パラメータを最適化するために DE アルゴリズムを使用するよりも効果的であるように見える．しかし，改良された PINNs で最良の性能を得るには，これら 3 つの改良すべてが不可欠である．

Kovasznyai 流のシナリオでは、改良された PINNs は低レイノルズ数の特徴である層流パターンを巧みに捉えた。モデルは、このようなシナリオにおける流体の挙動の重要な側面である流れの剥離を正確に予測した。これは特に、従来のモデルが苦勞することが多い高い勾配変化の領域で顕著であった。

**Table 4. Quantification of the contribution of each improvement to the final results for Case 1.**

Type	$u$	$v$	$p$	Average
Conventional PINNs	1.69%	19.81%	3.09%	8.20%
PINNs with residual-based adaptive sampling	0.90%	4.96%	2.58%	2.81%
PINNs with adaptive loss	1.18%	4.17%	2.14%	2.50%
PINNs with DE optimization algorithm	2.61%	19.57%	2.37%	8.18%
PINNs with all improvements	0.87%	3.38%	1.28%	1.84%

### 3.2 ケース 2 : 円柱後方の 2 次元非定常渦放出

このセクションでは、円柱後方の 2 次元非定常渦放出のケースを使用して、改良された PINNs を検証した。解析には文献からの DNS 結果を使用した [17]。図 11 は全体の計算領域を示しており、赤枠で囲まれた後流領域の領域がシミュレーションの対象となっている。このケースの設定は表 5 に要約されており、対応する支配方程式は式 (21) および (22) で示す。

**Table 5. Configuration for 2D unsteady vortex shedding past a circular cylinder.**

Item	Value
Computational domain	$X \in [1, 8]$ with 100 equal parts, $Y \in [-2, 2]$ with 50 equal parts, discrete value $T \in [0, 1.9]$ , discrete values, interval = 0.1s, 20 time slices in total
Re	100
Center of cylinder	(0, 0)
Velocity of inlet	1
Pressure of outlet	0
Upper and lower boundaries	Periodic boundary conditions
Number of BC sampling point	6,000 ((2 x 100+2 x 50) x 20), 20 time slices in total
Number of IC sampling point	5,000 (50 x 100)

Number of PDEs sampling point	20,000 in total (1000 points in each second)
Number of training data	6000+5000+20,000 = 31,000
Number of testing data	100 x 50 x 20 = 100,000
Maximum epochs	300,000

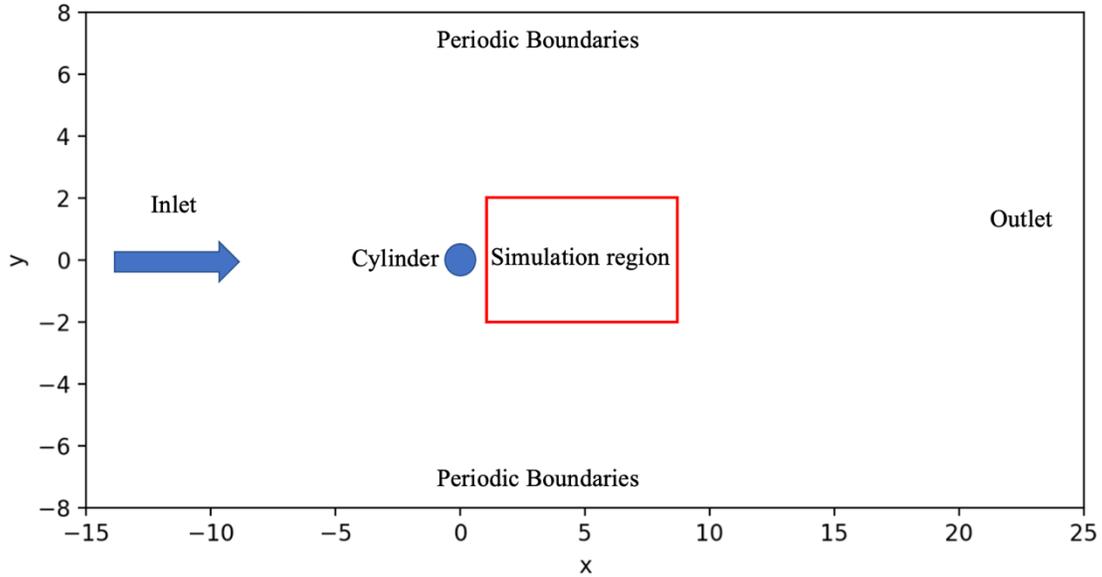
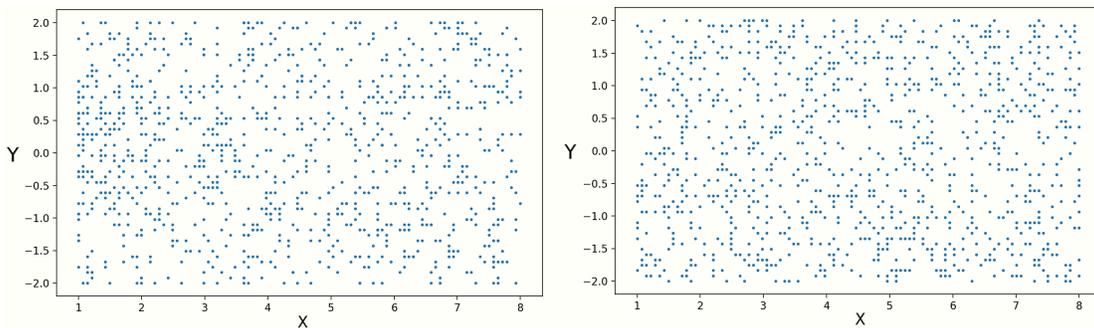


Figure 11. Domain of the 2D unsteady vortex shedding past a circular cylinder.

$$\nabla \cdot \mathbf{u} = 0 \quad (21)$$

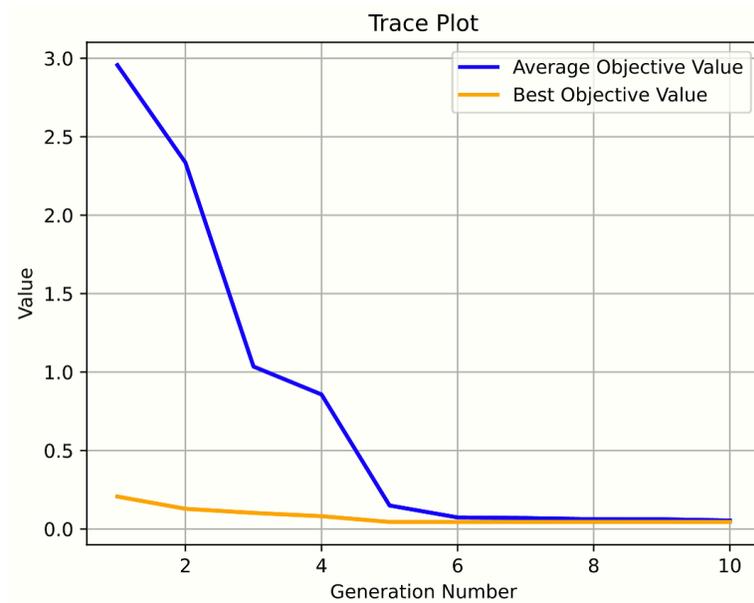
$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (22)$$

図 12 は、ケース 2 における改良された PINNs と従来の PINNs のサンプリングポイントを示す。



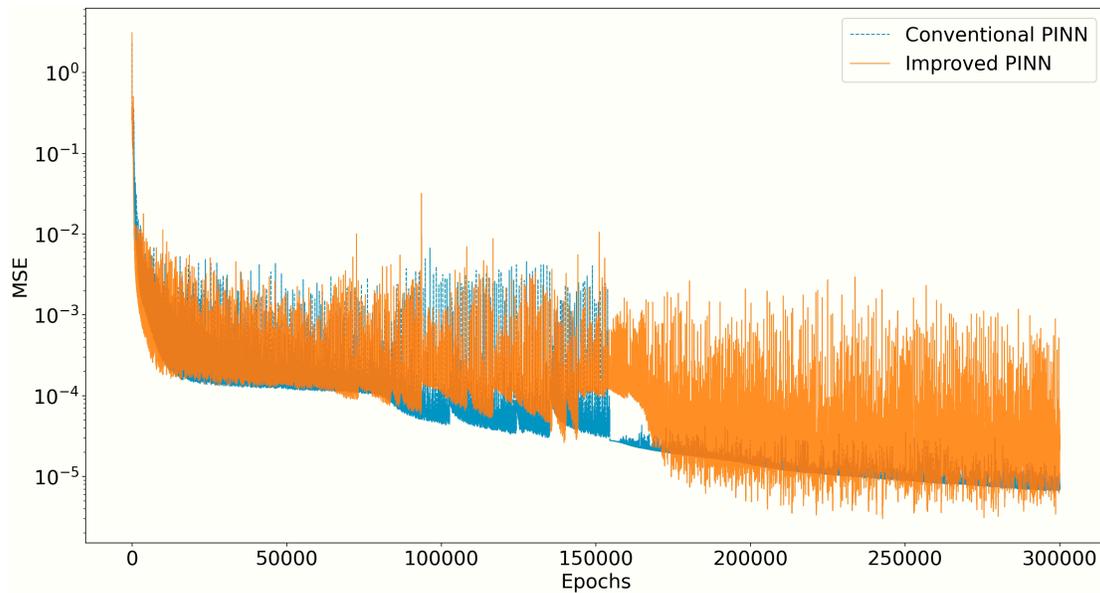
**Figure 12.** Sampling points for Case 2: residual-based adaptive sampling (left), and random sampling (right).

The DE algorithm was employed to discover the optimal structure and parameters of the PINNs. Figure 13 shows the optimization process for Case 2. After the 6<sup>th</sup> generation, the optimization algorithm tends to converge. The optimization results show that the model at the 92<sup>nd</sup> evolution model in the 10<sup>th</sup> generation, comprising 10 layers with 100 nodes per layer, a learning rate of 0.001, and the Tanh activation function, generates the lowest loss. Hence, these parameters are considered optimal for Case 2.



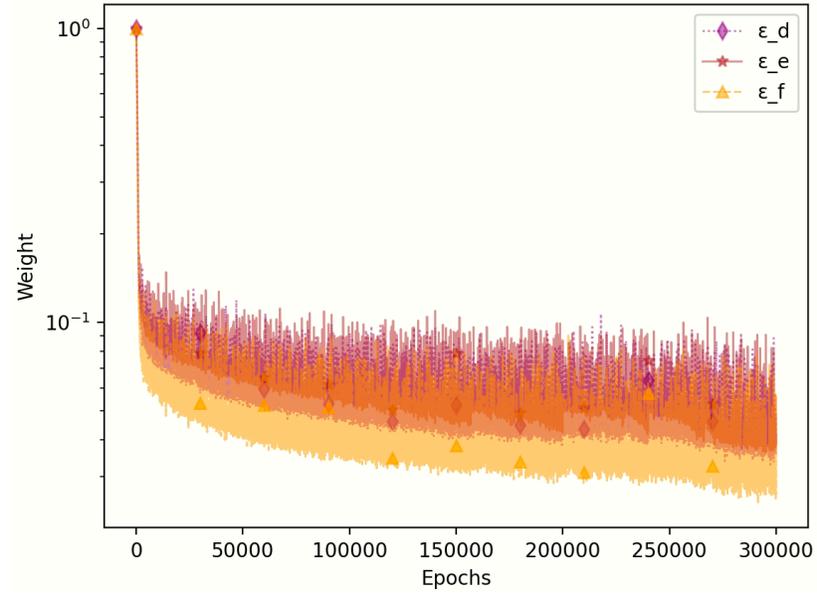
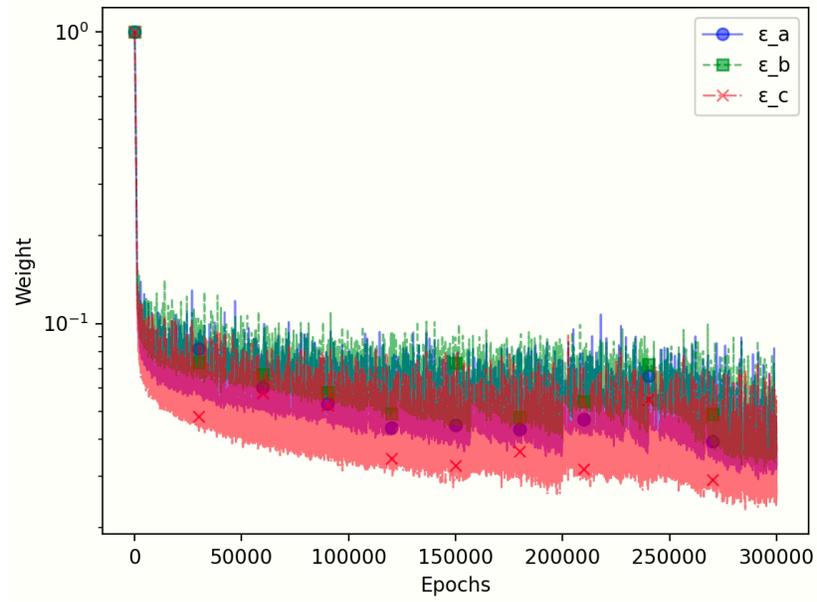
**Figure 13.** Optimization process of DE algorithm to find the optimal parameter in PINNs for Case 2.

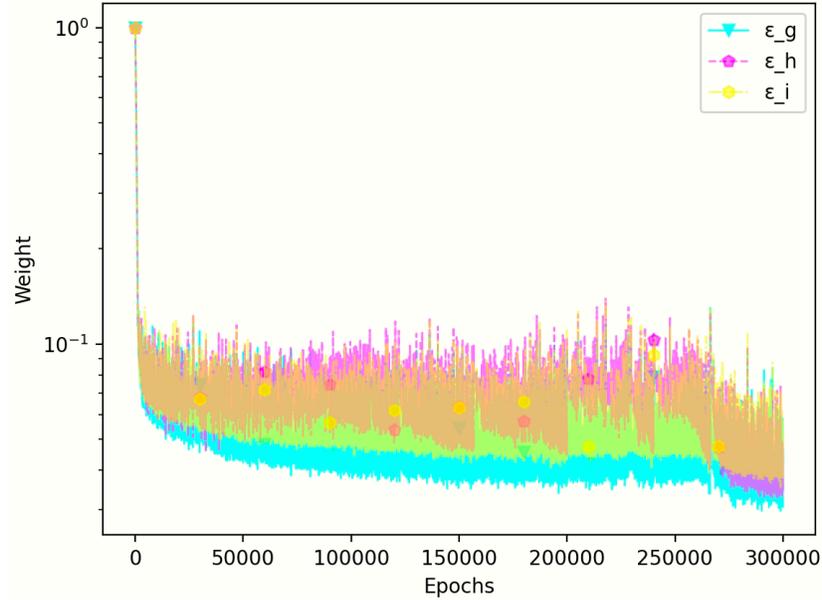
図 14 は、ケース 2 における改良された PINNs と従来の PINNs の平均二乗誤差 (MSE) を示す。改良された PINNs の誤差の変動範囲は比較的大きいものの、訓練誤差は低く、より早く最小誤差に達している。改良された PINNs と従来の PINNs は、それぞれ 300,000 エポックの計算に 42,664 秒と 33,229 秒を要した。



**Figure 14.** Training loss curves of the conventional and improved PINNs for Case 2.

図 15 は、ケース 2 における適応的な重みの変動を示す。ここで、 $\varepsilon a, \varepsilon b, \varepsilon c, \varepsilon d, \varepsilon e, \varepsilon f, \varepsilon g, \varepsilon h$ , および  $\varepsilon i$  はそれぞれのサブロスに対応し、境界条件の  $u, v, p$  初期条件の  $u, v, p$  および PDE の  $u, v, m$  を表す。訓練の終了時点で、 $\varepsilon i$  の値が最も大きくなり、これは PDE の  $m$  に最小の重みが適用されていることを示し、より小さなペナルティと制約を意味する。対照的に、 $\varepsilon a$  は最も小さな値を持ち、境界条件の  $u$  に最大の重みが適用され、したがってより大きなペナルティと制約があることを意味する。





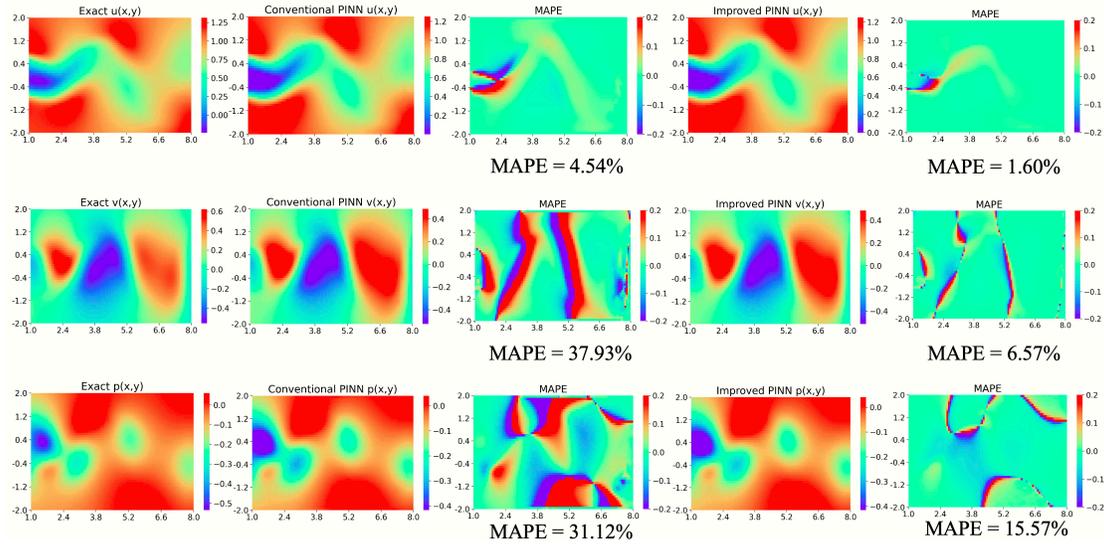
**Figure 15.** Variation of the adaptive weights for Case 2.

表 6 は、ケース 2 における従来の PINNs と改良された PINNs の最良モデルのエポックおよび対応するサブロスの訓練結果を示す。改良された PINNs の総損失は、従来の PINNs の総損失よりも小さくなっている。

**Table 6. Training results of the best model of the conventional and improved PINNs for Case 2.**

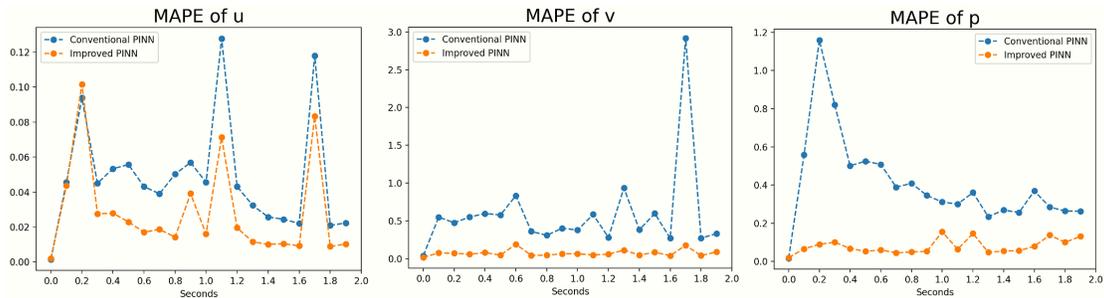
	Epoch	Total	pde_u	pde_v	pde_m	bc_u	bc_v	bc_p	ic_u	ic_v	ic_p
Conventional PINNs	299,491	6.63E-06	7.05E-07	9.41E-07	1.06E-06	1.06E-06	1.42E-06	5.82E-07	3.74E-07	3.96E-07	9.19E-08
Improved PINNs	242,713	3.02E-06	5.14E-07	5.76E-07	7.56E-07	1.98E-07	2.05E-07	1.06E-07	2.50E-07	2.99E-07	1.13E-07

円柱後方の 2 次元非定常渦放出のケースは、テストのために 0 秒から 1.9 秒までの 20 のタイムスライスを持つ  $100 \times 50$  のグリッドに分割された。従来の PINNs と改良された PINNs における XY 平面でのシミュレーション結果 ( $t = 1.0$  秒) を図 16 に示す。



**Figure 16.** Simulation results at the XY plane,  $t=1s$ , for conventional and improved PINNs. Images in the first, second, and fourth columns are the velocity and pressure fields obtained for the exact solution, the conventional PINNs, and the improved PINNs, respectively. Images in the third and fifth columns are the MAPEs of the conventional PINNs and the improved PINNs, respectively.

テスト期間中の  $u$ ,  $v$ , および  $p$  の MAPE は、従来の PINNs ではそれぞれ 4.83%, 58.29%, 40.73%, 改良された PINNs では各タイムスライスで 2.82%, 7.39%, 7.90%であった (図 17 および表 7 参照) . 改良された PINNs の各時点での MAPE は、従来の PINNs の MAPE よりも優れている。



**Figure 17.** The MAPE of  $u$ ,  $v$ , and  $p$  during the test time for the conventional and improved PINNs.

最終結果に対する各改良の寄与を定量化するために、各改良を個別にテストした。結果は表 7 に示す。表の定量化結果は、各改良がケース 2 の問題に対しても有効であることを示唆している。残差に基づく適応サンプリングと適応損失は、DE アルゴリズムを利用して PINNs の構造を最適化するよりも効果的である傾向がある。それにもかかわらず、最適な PINNs を達成するためには、3 つの改良すべてを統合することが推奨される。円柱後方の渦放出のシミュレーションは、改良された PINNs が非定常で複雑な流れパターンを処理する能力を強調した。改良された PINNs は、シリンダーから放出される渦の周波数とサイズを成功裏に予測した。これは、流体の挙動が流れ条件のわずかな変化に敏感な低レイノルズ数での挑戦的な側面である。

**Table 7. Quantification of the contribution of each improvement to the final results for Case 2.**

Type	$u$	$v$	$p$	Average
Conventional PINNs	4.83%	58.29%	40.73%	34.62%
PINNs with residual-based adaptive sampling	2.51%	14.73%	11.13%	9.45%
PINNs with adaptive loss	3.36%	14.92%	10.37%	9.55%
PINNs with DE optimization algorithm	4.86%	42.65%	21.11%	22.87%
PINNs with all improvements	2.82%	7.39%	7.90%	6.04%

### 3.3 ケース 3 : 3D 非定常ベルトラミ流

流体力学の分野では、ベルトラミ流（ベルトラミ場とも呼ばれます）は速度ベクトル場が渦度ベクトル場と平行である流れ場を指す。言い換えると、流体の運動が渦度の線に沿って行われる流体の流れである。渦度は、流れ中の小さな流体要素の回転の度合いを示す。したがって、ベルトラミ流は流体要素がその回転軸と同じ方向に移動する独特のタイプの渦流である。

このため、3D ベルトラミ流は、流体力学の数値シミュレーションのテストケースとして頻繁に使用される（式 (23) および (24) 参照）。ベルトラミ流はまた、

非圧縮性NSEの正確な解を持ち、式(25)、(26)、(27)、および(28)に示されている[35].

$$\nabla \cdot \mathbf{u} = 0 \quad (23)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (24)$$

$$u(x, y, z, t) = -a[e^{ax} \sin(ay + dz) + e^{az} \cos(ax + dy)]e^{-d^2t} \quad (25)$$

$$v(x, y, z, t) = -a[e^{ay} \sin(az + dx) + e^{ax} \cos(ay + dz)]e^{-d^2t} \quad (26)$$

$$w(x, y, z, t) = -a[e^{az} \sin(ax + dy) + e^{ay} \cos(az + dx)]e^{-d^2t} \quad (27)$$

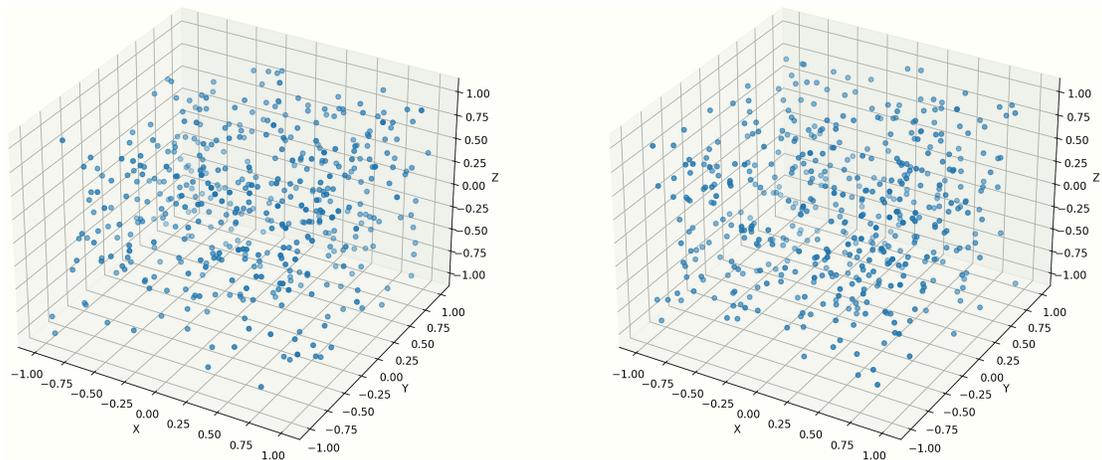
$$p(x, y, z, t) = -\frac{1}{2}a^2[e^{2ax} + e^{2ay} + e^{2az} + 2 \sin(ax + dy) \cos(az + dx) e^{a(y+z)} + 2 \sin(ay + dz) \cos(ax + dy) e^{a(z+x)} + 2 \sin(az + dx) \cos(ay + dz) e^{a(x+y)}]e^{-2d^2t} \quad (28)$$

ケース3で使用される設定は表8に示されている。ここでは、 $Re=1$ ,  $a=1$ ,  $d=1$ と仮定されている。

**Table 8. Configuration for 3D unsteady Beltrami flow.**

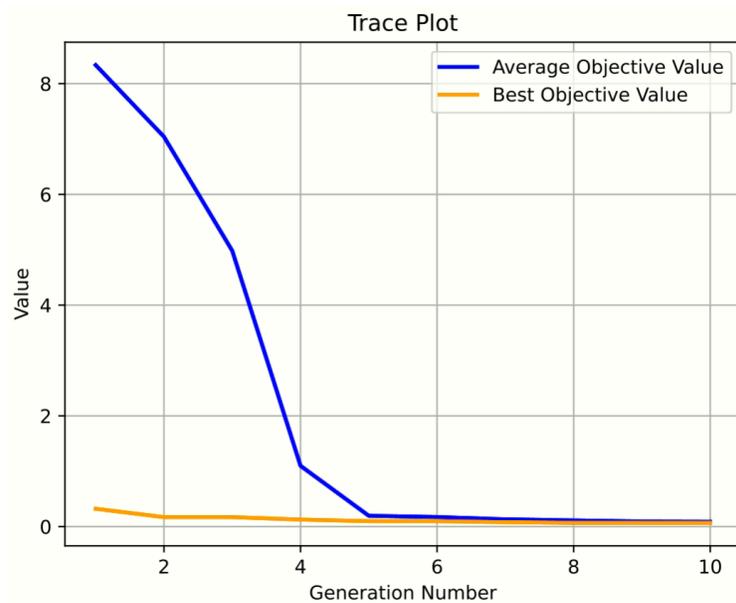
Item	Value
Computational domain	$X \in [-1, 1], Y \in [-1, 1], Z \in [-1, 1]$ , continuous value $T \in [0, 1]$ , discrete values, interval = 0.1s, 11 time slices in total
Re	1
$a, d$	1, 1
Number of BC sampling points	Divide the every line into 10 equal parts, generate 11 points, 6600 (6 x 10 x 10 x 11) in total
Number of IC sampling points	1331 (11 x 11 x 11)
Number of PDEs sampling points	5500 in total (500 points in each second)
Number of training data	6600+1331+5500 = 13,431
Number of testing data	31 x 31 x 31 x 11 = 327,710
Maximum epochs	30,000

図18は、ケース3における提案されたPINNsと従来のPINNsのサンプリングポイントを示す。



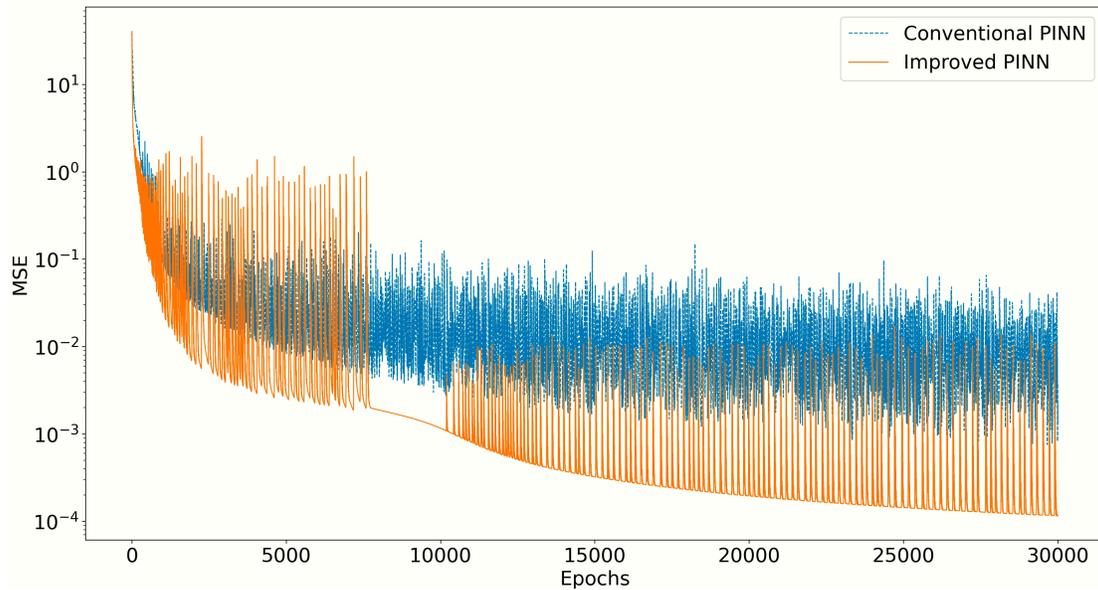
**Figure 18.** Sampling points for Case 3: residual-based adaptive sampling (left), and random sampling (right).

DE アルゴリズムは、PINNs の最適な構造とパラメータを特定するために使用された。図 19 は、ケース 3 の最適化プロセスを示す。5 世代目以降、最適化アルゴリズムが収束する傾向があることが確認できる。最適化の結果、第 10 世代の 94 番目の進化モデル（10 層、各層に 120 ノード、学習率 0.001、Tanh 活性化関数）が最も低い訓練誤差を生成し、ケース 3 の最適なパラメータセットと見なされる。



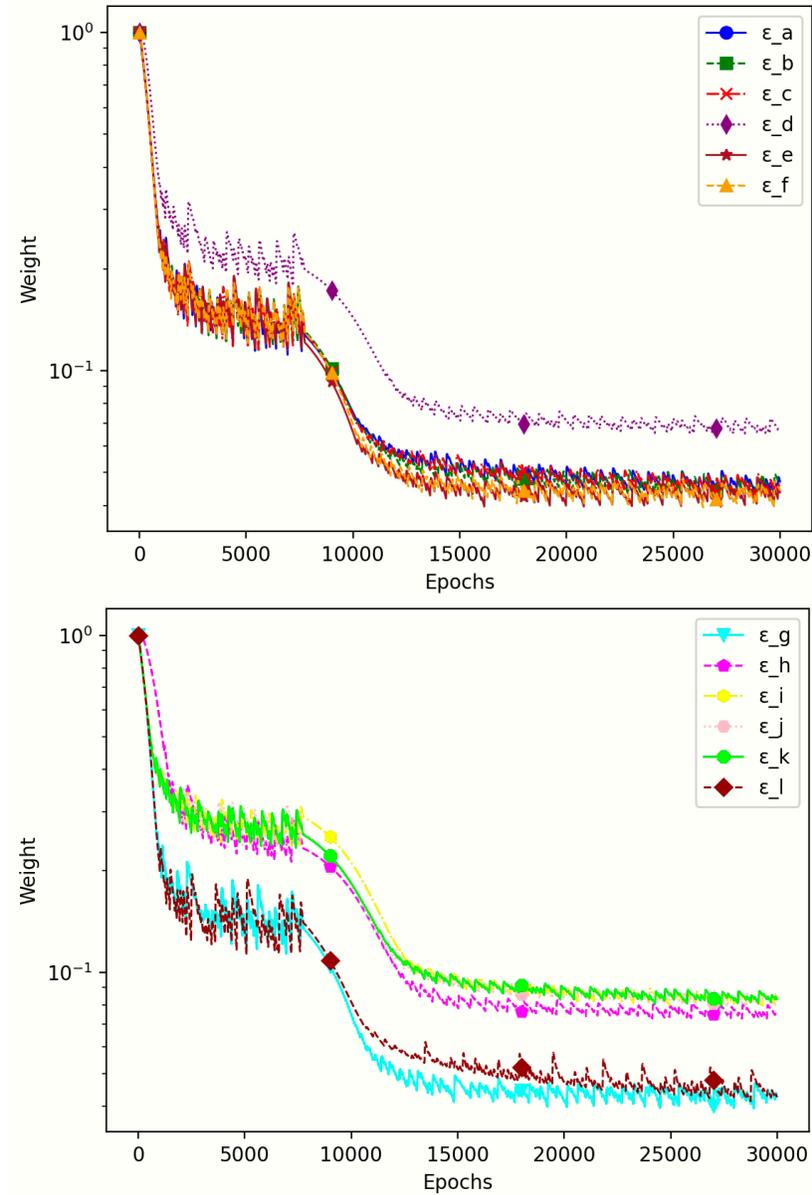
**Figure 19.** Optimization process of DE algorithm to find the optimal parameter in PINNs for Case 3.

図 20 は、ケース 3 における提案された PINNs と従来の PINNs の平均二乗誤差 (MSE) を示す。改良された PINNs は一貫して誤差が低く、優れた学習能力と NSE を解く能力を持っていることが確認された。改良された PINNs と従来の PINNs は、それぞれ 30,000 エポックの計算に 3219 秒と 2714 秒を要した。



**Figure 20.** Training loss curves of the conventional and improved PINNs for Case 3.

図 21 は、ケース 3 における適応的な重みの変動を示す。ここで、 $\epsilon_a, \epsilon_b, \epsilon_c, \epsilon_d, \epsilon_e, \epsilon_f, \epsilon_g, \epsilon_h, \epsilon_i, \epsilon_j, \epsilon_k$  および  $\epsilon_l$  はそれぞれのサブロスに対応し、境界条件の  $u, v, w, p$  初期条件の  $u, v, w, p$  および PDE の  $u, v, w, m$  を表す。訓練の終了時点で、 $\epsilon_j$  の値が最も大きく、PDE の  $v$  に最小の重みが適用されていることを示し、より小さなペナルティと制約を意味する。一方、 $\epsilon_l$  は最も小さな値を持ち、PDE の  $m$  に最大の重みが適用され、大きなペナルティと制約を示す。



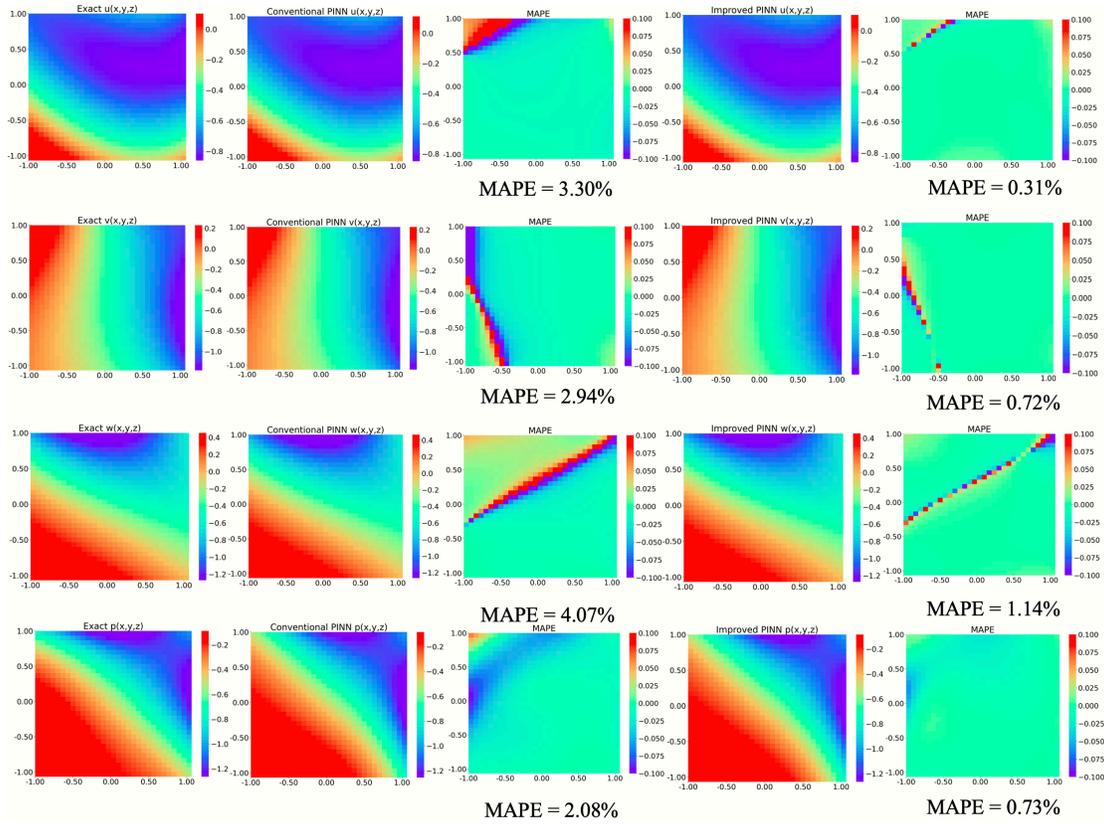
**Figure 21.** Variation of the adaptive weights for Case 3.

表 9 は、ケース 3 における従来の PINNs と改良された PINNs の最良モデルのエポックおよび対応するサブロスの訓練結果を示す。改良された PINNs のサブロスは、一貫して従来の PINNs のサブロスよりも小さい。

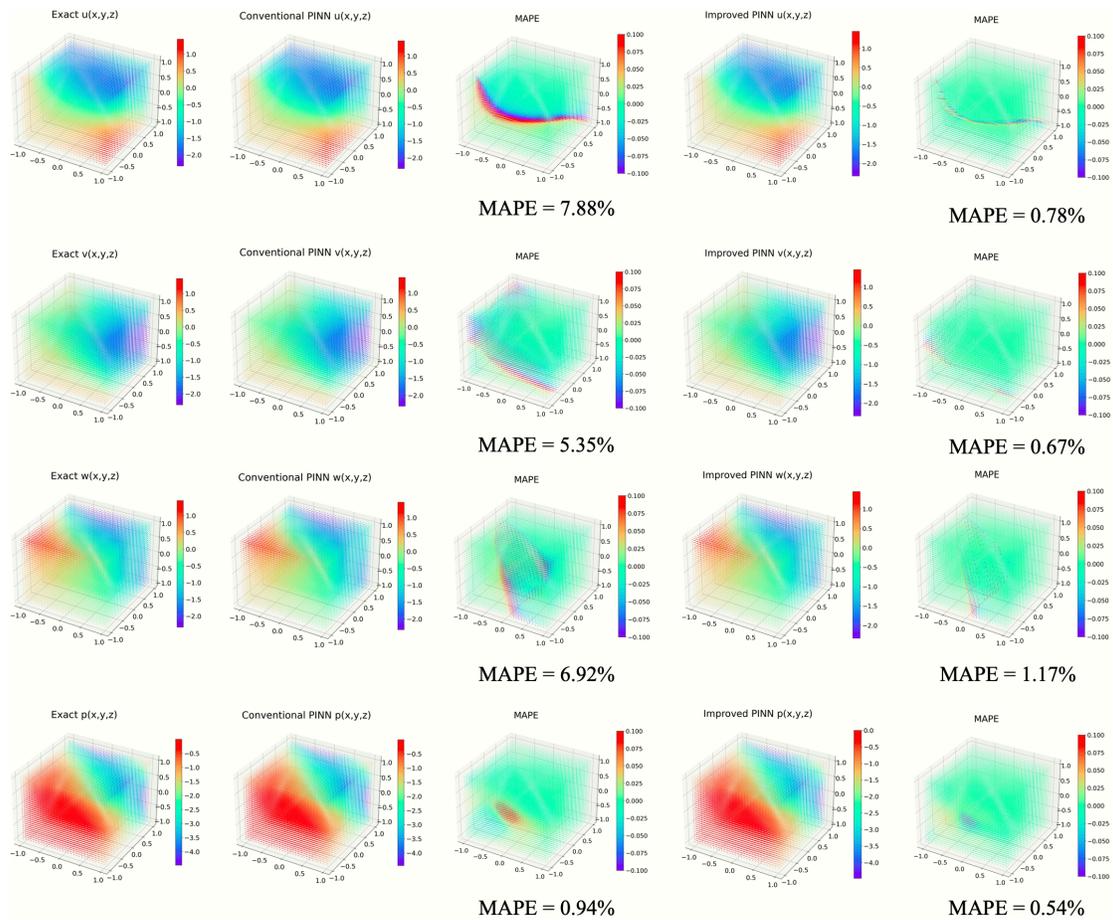
**Table 9.** Training results of the best model of the conventional and improved PINNs for Case 3.

	Epoch	Total	pde_u	pde_v	pde_w	pde_m	bc_u	bc_v
Conventional PINNs	29,655	5.38E-04	1.42E-04	1.40E-04	1.45E-04	3.97E-05	3.60E-05	3.44E-05
Improved PINNs	29,702	9.03E-05	2.63E-05	2.56E-05	2.69E-05	5.17E-06	3.41E-06	2.93E-06
	Epoch	Total	bc_w	bc_p	ic_u	ic_v	ic_w	ic_p
Conventional PINNs	29655		3.40E-05	5.12E-05	2.19E-05	2.16E-05	2.43E-05	6.66E-05
Improved PINNs	29702		3.19E-06	1.16E-05	7.00E-07	8.58E-07	7.47E-07	9.80E-06

3D 非定常ベルトラミ流のシミュレーションケースは、テストのために 0 秒から 1.0 秒までの 11 のタイムスライスを持つ  $31 \times 31 \times 31$  のグリッドに分割された。従来の PINNs と改良された PINNs における XY 平面でのシミュレーション結果 ( $Z=0.5\text{m}$ 、 $t=1$  秒) を図 22 に示す。従来の PINNs と改良された PINNs における全 XYZ 空間でのシミュレーション結果 ( $t=0.5$  秒) は図 23 に示されている。

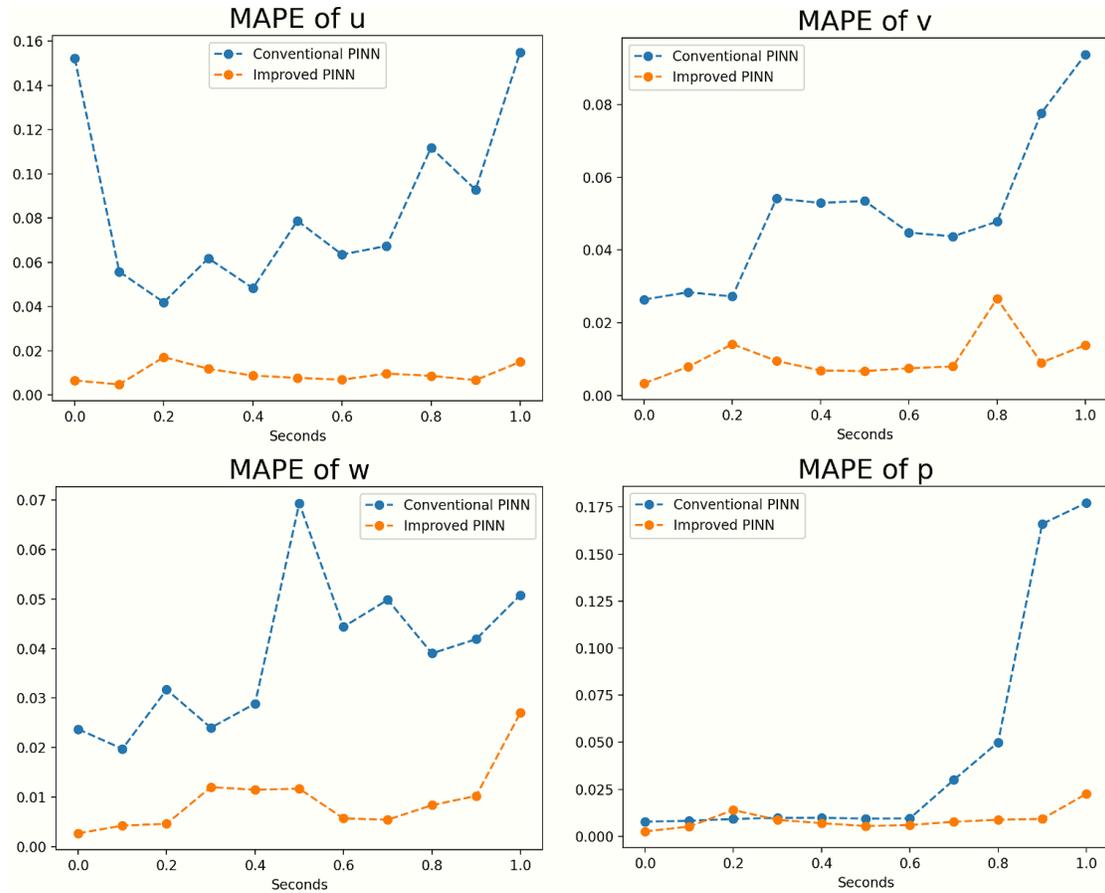


**Figure 22.** Simulation results at XY plane,  $Z=0.5\text{m}$ ,  $t=1\text{s}$  between conventional PINNs and improved PINNs. Images in the first, second, and fourth columns are the velocity and pressure fields obtained for the exact solution, the conventional PINNs, and the improved PINNs, respectively. Images in the third and fifth columns are the MAPEs of the conventional PINNs and the improved PINNs, respectively.



**Figure 23.** Simulation results for the whole XYZ space,  $t=0.5s$ , for conventional and improved PINNs.

テスト期間中の  $u$ ,  $v$ ,  $w$ , および  $p$  の MAPE は、従来の PINNs ではそれぞれ 8.45%, 5.00%, 3.85%, 4.43%, 改良された PINNs では各タイムスライスで 0.95%, 1.03%, 0.95%, 0.88%であった (図 24 および表 10 参照) . 改良された PINNs は、従来の PINNs と比較してすべての時点で優れている。



**Figure 24.** MAPE of  $u$ ,  $v$ ,  $w$ , and  $p$  during the test time for the conventional and improved PINNs.

最終結果に対する各改良の寄与を定量化するために、各改良を個別に実装してケース 3 をシミュレートし、その結果を表 10 に示した。表の結果は、PINNs の性能向上において最も効果的な単一の改良が、適応損失を使用した PINNs であることを示唆している。この優位性は、おそらくNSEにおけるさまざまな損失項（支配方程式、初期条件、および速度、質量、圧力損失に関連する境界条件を含む）をバランスさせる能力から来していると考えられる。残差に基づく適応サンプリングを用いたPINNsは、個別の改良の中で2番目に効果的であり、サンプリングポイントの選択が適応的な重み付けと比較して結果に与える影響が少ないことを示す。この観察結果は、現在のPINNsの改良がNSEに埋め込まれた知識を十分に学習しており、この段階でサンプリングポイントが十分に包括的であることを示唆しているかもしれない。DE アルゴリズムは個別の改良の中で最下位にランクされており、ネットワーク構造が最終的なシミュレーション性能に最も影響を与えないことを示している。これは、現在のPINNsの改良がNSの埋め込まれた知識を効果的に学習していることをさらに強調している。しかし、最適なシミュレー

シミュレーション結果を達成するためには、これら3つの方法を統合することが不可欠である。

ベルトラミ流の場合、改良された PINNs は三次元の渦構造を効果的に捉え、複雑な流れのダイナミクスをシミュレートする上でのロバスト性を示した。これは、低レイノルズ数での流体挙動を理解するために、そのような複雑なパターンを正確に表現することが重要である場合に特に重要である。

**Table 10. Quantification of the contribution of each improvement to the final results for Case 3.**

Type	$u$	$v$	$w$	$p$	Average
Conventional PINNs	8.45%	5.00%	3.85%	4.43%	5.43%
PINNs with residual-based adaptive sampling	5.42%	3.27%	3.98%	1.90%	3.64%
PINNs with adaptive loss	1.54%	2.29%	1.36%	1.17%	1.59%
PINNs with DE optimization algorithm	5.83%	6.60%	5.35%	1.92%	4.93%
PINNs with all improvements	0.95%	1.03%	0.95%	0.88%	0.95%

## 5 結論と今後の課題

本研究では、流体力学システムを記述するための改良された汎用的な PINNs を提案した。改良された PINNs のシミュレーション性能は、次の3つの主要な改良によって強化された：残差が大きい領域により多くのサンプリングポイントを自動的に割り当てるための残差に基づく適応サンプリング、損失項目間のバランスの取れた分配を達成するための適応損失重み付け、および PINNs の構造とパラメータを最適化するための DE 最適化アルゴリズムの利用である。改良された PINNs のシミュレーション性能は、低レイノルズ数の3つのケース（Kovaszny 流、円柱後方の渦放出、ベルトラミ流）によって検証された。Kovaszny 流では、モデルは流れの剥離を正確に予測し、重要な流体力学を捉える精度を示した。円柱後方の渦放出においては、改良された PINNs は流れの非定常性を巧みに処理し、円柱から放出される渦を正確にシミュレートし、動的な流体パターンに対する対応力を示した。最後に、ベルトラミ流では改良された PINNs が複雑な三次元渦構造

を効果的に捉え、複雑で精緻な流れのダイナミクスをシミュレートする際のモデルのロバスト性を強調した。ケース 1, 2, および 3 の各ケースでの改良された PINNs のシミュレーション精度はそれぞれ 1.84%, 6.04%, および 0.95%であり、従来の PINNs の精度 (8.20%, 34.62%, および 5.43%) よりもはるかに低い結果となった。

個別の改良に関しては、適応損失を持つ PINNs が一般的に最高の性能を示し、次に残差に基づく適応サンプリングを持つ PINNs, 最後に DE 最適化アルゴリズムを持つ PINNs が続いた。これは、PINNs 内のさまざまな損失項目のバランスを取ることが最も重要であり、それが PINNs が基礎的な物理法則をより完全に学習するのを助けることを示す。残差に基づく適応サンプリングを持つ PINNs は、残差が大きい領域での物理法則の学習を強化することに寄与する。一方で、DE 最適化アルゴリズムは、現在の物理法則により適応するように PINNs の結果を微調整する。総じて、各改良はそれ自体で効果的であるが、最良のシミュレーション結果を達成するためには、3つの方法を組み合わせることが不可欠である。

本研究は、流体力学の分野を前進させ、他のシステムにも適用可能な優れた性能を持つ堅牢で柔軟な PINNs モデルを提示した。今後注力すべき研究項目を以下に挙げる。(1) 改良された PINNs の性能をさらに検証するために、逆問題、二相流ケース、および熱流体力学のシミュレーションが進行中である。(2) 本研究で見られた精度は満足できるものであったが、PINNs の訓練時間は比較的長いことが課題に挙げられる。PINNs の性能と訓練時間はトレードオフの関係にあるため、訓練時間をさらに短縮する必要がある。

## Nomenclature

---

$\hat{u}$	Predicted value of trained pinns
$c$	The control parameter of residual-based adaptive sampling
$P_0$	Initial population
$P_t$	Parent population
$Q_t$	Children population

$R_t$	Combined population
$\mathbf{u}$	Velocity vector ( $u, v, w$ )
$\mathbf{x}$	Space-time point ( $x, y, z; t$ )
$N$	Number of sampling points
$Re$	Reynolds number
$S$	Collection of sampling points
$W$	Adaptive weights
$f$	External force per unit volume of fluid
$f(\mathbf{x})$	Calculated residual value of space-time point $\mathbf{x}$
$m$	Quality
$p$	Pressure
$t$	Time
$u(\mathbf{x})$	Real value
$u$	Velocity in x direction
$v$	Velocity in y direction
$w$	Velocity in z direction

*Greek symbol*

$\varepsilon$	Uncertainty, as trained parameter of the PINNs
$\theta$	Parameters of the PINNs
$\mu$	Dynamic viscosity
$\rho$	Density

## Subscripts

<i>BC</i>	Boundary condition
<i>GE</i>	Governing equation
<i>IC</i>	Initial conditions
<i>ini</i>	Initial model
<i>Tra</i>	Trained model

## Superscripts

<i>k</i>	To the k-th power
----------	-------------------

## Mathematical symbol

$\mathbb{E}[\ ]$	Numerical integration
------------------	-----------------------

## Acronyms

CFD	Computational fluid dynamics
DE	Distribution parameter
DNS	Direct numerical simulation
MAPE	Mean absolute percentage error
MSE	Mean square error
NN	Neural network
NSE	Navier-Stokes equation

ODE	Ordinary differential equation
PDE	Partial differential equation
PDF	Probability density function
PINNs	Physics-informed neural networks
VP	Velocity-pressure
VV	Vorticity-velocity

## References

- [1] Pendar, M. R., Cândido, S., & Páscoa, J. C. (2023). Optimization of painting efficiency applying unique techniques of high-voltage conductors and nitrotherm spray: Developing deep learning models using computational fluid dynamics dataset. *Physics of Fluids*, 35(7).
- [2] Xue, K., Zhang, Z., Hao, S., Luo, P., & Wang, Y. (2022). On the onset of nonlinear fluid flow transition in rock fracture network: Theoretical and computational fluid dynamic investigation. *Physics of Fluids*, 34(12).
- [3] Zamolo, R., & Parussini, L. (2020). Analysis of geometric uncertainties in CFD problems solved by RBF-FD meshless method. *Journal of Computational Physics*, 421, 109730.
- [4] Kalli, M., Pico, P., Chagot, L., Kahouadji, L., Shin, S., Chergui, J., ... & Angeli, P. (2023). Effect of surfactants during drop formation in a microfluidic channel: a combined experimental and computational fluid dynamics approach. *Journal of Fluid Mechanics*, 961, A15.
- [5] Jiang, Y., Guo, Y., Yu, Z., Hua, X., Lin, J., Wassgren, C. R., & Curtis, J. S. (2021). Discrete element method–computational fluid dynamics analyses of flexible fibre fluidization. *Journal of Fluid Mechanics*, 910, A8.
- [6] Dang, G., Liu, S., Guo, T., Duan, J., & Li, X. (2022). Direct numerical simulation of compressible turbulence accelerated by graphics processing unit: An open-source high accuracy accelerated computational fluid dynamic software. *Physics of Fluids*, 34(12).

- [7] Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), e2101784118.
- [8] Fouladi, K., Wemhoff, A. P., Silva-Llanca, L., Abbasi, K., & Ortega, A. (2017). Optimization of data center cooling efficiency using reduced order flow modeling within a flow network modeling approach. *Applied Thermal Engineering*, 124, 929-939.
- [9] Heaney, C. E., Wolffs, Z., Tómasson, J. A., Kahouadji, L., Salinas, P., Nicolle, A., ... & Pain, C. C. (2022). An AI-based non-intrusive reduced-order model for extended domains applied to multiphase flow in pipes. *Physics of Fluids*, 34(5), 055111.
- [10] Lee, K., & Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404, 108973.
- [11] Yousif, M. Z., Yu, L., & Lim, H. (2022). Physics-guided deep learning for generating turbulent inflow conditions. *Journal of Fluid Mechanics*, 936, A21.
- [12] Wu, J. L., Xiao, H., & Paterson, E. (2018). Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7), 074602.
- [13] Gupta, R., & Jaiman, R. (2022). Three-dimensional deep learning-based reduced order model for unsteady flow dynamics with variable Reynolds number. *Physics of Fluids*, 34(3).
- [14] Novati, G., de Laroussilhe, H. L., & Koumoutsakos, P. (2021). Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, 3(1), 87-96.
- [15] Li, B., Yang, Z., Zhang, X., He, G., Deng, B. Q., & Shen, L. (2020). Using machine learning to detect the turbulent region in flow past a circular cylinder. *Journal of Fluid Mechanics*, 905, A10.
- [16] Brantson, E. T., Abdulkadir, M., Akwensi, P. H., Osei, H., Appiah, T. F., Assie, K. R., & Samuel, S. (2022). Gas-liquid vertical pipe flow patterns convolutional neural network classification using experimental advanced wire mesh sensor images. *Journal of Natural Gas Science and Engineering*, 99, 104406.
- [17] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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.

- [18] Cai, S., Wang, Z., Fuest, F., Jeon, Y. J., Gray, C., & Karniadakis, G. E. (2021). Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks. *Journal of Fluid Mechanics*, 915, A102.
- [19] Jin, X., Cai, S., Li, H., & Karniadakis, G. E. (2021). NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426, 109951.
- [20] Eivazi, H., Tahani, M., Schlatter, P., & Vinuesa, R. (2022). Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. *Physics of Fluids*, 34(7), 075117.
- [21] Ji, W., Qiu, W., Shi, Z., Pan, S., & Deng, S. (2021). Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36), 8098-8106.
- [22] Laubscher, R. (2021). Simulation of multi-species flow and heat transfer using physics-informed neural networks. *Physics of Fluids*, 33(8), 087101.
- [23] Aliakbari, M., Mahmoudi, M., Vadasz, P., & Arzani, A. (2022). Predicting high-fidelity multiphysics data from low-fidelity fluid flow and transport solvers using physics-informed neural networks. *International Journal of Heat and Fluid Flow*, 96, 109002.
- [24] Liu, Y., Li, X., & Liu, Z. (2022). An Improved Physics-Informed Neural Network Based on a New Adaptive Gradient Descent Algorithm for Solving Partial Differential Equations of Nonlinear Systems.
- [25] Cheng, C., & Zhang, G. T. (2021). Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. *Water*, 13(4), 423.
- [26] Wong, J. C., Ooi, C., Chiu, P. H., & Dao, M. H. (2021). Improved surrogate modeling of fluid dynamics with physics-informed neural networks. *arXiv preprint arXiv:2105.01838*.
- [27] Zhang, T., Dey, B., Kakkar, P., Dasgupta, A., & Chakraborty, A. (2020). Frequency-compensated pinns for fluid-dynamic design problems. *arXiv preprint arXiv:2011.01456*.
- [28] Yu, J., Lu, L., Meng, X., & Karniadakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393, 114823.

- [29] Wu, C., Zhu, M., Tan, Q., Kartha, Y., & Lu, L. (2023). A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403, 115671.
- [30] Liebel, L., & Körner, M. (2018). Auxiliary tasks in multi-task learning. arXiv preprint arXiv:1805.06334.
- [31] Xiang, Z., Peng, W., Liu, X., & Yao, W. (2022). Self-adaptive loss balanced Physics-informed neural networks. *Neurocomputing*, 496, 11-34.
- [32] Storn, R. and Price, K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), pp.341-359.
- [33] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- [34] Kovasznay, L. I. G. (1948, January). Laminar flow behind a two-dimensional grid. In *Mathematical Proceedings of the Cambridge Philosophical Society* (Vol. 44, No. 1, pp. 58-62). Cambridge University Press.
- [35] Ethier, C. R., & Steinman, D. A. (1994). Exact fully 3D Navier–Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19(5), 369-375.