Article

Data Analysis Using Manifold Learning: The RDSF Algorithm

Mehdi Nadjafikhah and Mohammad Nasiri *

Posted Date: 26 May 2025

doi: 10.20944/preprints202505.1528.v2

Keywords: RDSF algorithm; manifold learning; dimensionality reduction; data-driven mathematics; partial differential equations; prime number topology; scientific visualization

Article

Data Analysis Using Manifold Learning: The RDSF Algorithm

Mehdi Nadjafikhah and Mohammad Nasiri *

Dept. Math., UST Iran
* Correspondence: m_nasiri56@mathdep.iust.ac.ir

Abstract: In modern information systems, data analysis is pivotal for uncovering hidden patterns and extracting meaningful insights. Visualizing the behavior of real-valued functions over high-dimensional domains enables researchers to gain intuitive understanding of complex systems. Traditionally, mathematical problems are tackled analytically; however, a data-driven approach—where synthetic samples are explored systematically—can reveal new directions and solution spaces. In this paper, we introduce the RDSF algorithm, short for *Reducing the Dimension of the Space of Independent and Dependent Variables of Real-Valued Functions*. This algorithm leverages manifold learning, specifically Multidimensional Scaling (MDS), to reduce the dimensionality of both input and output spaces, allowing effective visualization and analysis of function behaviors. We present a unified framework for analytical and visual exploration of mathematical and engineering problems using RDSF. The utility of RDSF is demonstrated through multiple case studies, including approximate solutions to partial differential equations (PDEs) and topological analysis of the distribution of prime numbers. These applications reveal that even abstract mathematical domains can benefit significantly from visual, data-oriented perspectives. The proposed framework is general, adaptable, and opens new avenues for exploration across disciplines.

Keywords: RDSF algorithm; manifold learning; dimensionality reduction; data-driven mathematics; partial differential equations; prime number topology; scientific visualization

1. Introduction

Visual exploration of high-dimensional datasets often reveals subtle patterns, structural anomalies, or deviations from expected objective function behaviors. These insights can uncover new solution pathways or isolate subspaces of particular interest within the data landscape [1].

Manifold learning, situated at the intersection of geometry, computation, and statistics, has become a pivotal framework in modern data analysis [2]. It aims to uncover low-dimensional structures embedded within high-dimensional spaces [3]. Classic linear techniques such as Principal Component Analysis (PCA) [4]) and Multidimensional Scaling (MDS) [5]) provide interpretable projections by preserving variance or pairwise distances, respectively.

Understanding the behavior of functions involving more than three variables remains a critical challenge. Visualizing such high-dimensional relationships enhances the interpretability of model dynamics and variable interactions [6].

In this article, we introduce the **Reducing the Dimension of the Space of Independent and Dependent Variables of Real-Valued Functions (RDSF)**, a novel computational algorithm that builds on MDS for dimensionality reduction and leverages random sampling to approximate objective landscapes. RDSF is particularly well-suited for cases where analytical formulations are unavailable or incomplete but a computable objective function exists.

We demonstrate the power and versatility of RDSF through three representative applications: (1) the approximation of solutions to partial differential equations (PDEs), (2) analysis of the distribution of prime numbers in multi-dimensional lattices, and (3) topological characterization of function behaviors.

In each case, RDSF serves as a tool for converting computationally intractable problems into visually and numerically analyzable forms [7,8].

2. Algorithm for Reducing the Dimension of the Space of Independent and Dependent Variables of Real-Valued Functions (RDSF)

Let $M \subseteq \mathbb{R}^n$ be an information space with dimension n, and let $f : M \to \mathbb{R}$ be a real-valued function defined over M. The goal of the RDSF algorithm is to analyze and visualize the behavior of f across M by constructing an interpretable geometric representation in $\mathbb{R}^3$, where the third dimension corresponds to the functional value of f at each point.

To achieve this, we employ **Multidimensional Scaling (MDS)**, a manifold learning technique that projects high-dimensional data into lower dimensions while preserving pairwise distances as closely as possible [9]. Preserving these distances ensures that the relative geometric and topological relationships among points in M are reflected in the reduced space.

The algorithm consists of the following steps:

1. For each point $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in M$, compute the function value $f(\mathbf{x})$.
2. Apply MDS to the point cloud $\{\mathbf{x}_i\}_{i=1}^N$ in M to obtain a 2D embedding $\{\mathbf{y}_i = (y_1^{(i)}, y_2^{(i)})\}_{i=1}^N$ in a new space $N \subseteq \mathbb{R}^2$, such that the Euclidean distances are preserved:
$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\| \quad \forall i, j.$$
3. For each embedded point $\mathbf{y}_i$, associate the corresponding function value $f(\mathbf{x}_i)$, forming a new 3D representation:
$$\mathbf{z}_i = (y_1^{(i)}, y_2^{(i)}, f(\mathbf{x}_i)) \in \mathbb{R}^3.$$
4. Visualize the set $\{\mathbf{z}_i\}_{i=1}^N$ in $\mathbb{R}^3$. This 3D embedding enables intuitive analysis of the functional landscape, highlighting patterns such as gradients, clusters, local extrema, and other structural insights into the original function f.

This method is particularly useful for exploring the behavior of high-dimensional functions in optimization, approximation, and data analysis, especially where interpretability and visualization are important.

3. Applications of the RDSF Algorithm

The RDSF algorithm provides a powerful computational framework for analyzing a wide variety of mathematical and engineering problems. In many real-world scenarios, sufficient data can be collected or derived to construct an informative feature space. This allows for systematic evaluation of an objective function over varying parameters, forming what we call the M-space.

For theoretical problems, however, such data may not be readily available or might need to be generated synthetically. In such cases, the RDSF algorithm leverages random sampling and repeated simulations to iteratively refine the understanding of the objective landscape. This ability to explore solution spaces with minimal initial assumptions makes RDSF particularly valuable for high-dimensional, underdetermined, or nonlinear problems.

In the following subsection, we demonstrate the utility of the RDSF algorithm in approximating solutions to Partial Differential Equations (PDEs), a fundamental class of problems in mathematics and applied sciences.

3.1. Solving Partial Differential Equations Using RDSF

Partial Differential Equations (PDEs) are ubiquitous in modeling physical systems, from heat transfer to fluid dynamics [10]. However, analytical solutions are often intractable, necessitating approximate and numerical methods [11]. Here, we introduce a theoretical result that justifies the use of RDSF in approximating solutions to a general class of PDEs using randomly generated functional forms.

Definition 1. *A real-valued function $f(x_1, \dots, x_n)$ is called a Finitely and Analytically Differentiable Function (FADF) if the function and all its partial derivatives of any order exist and remain finite over the domain of interest.*

Example 1. *The functions* $\sin(x)$ *and* $\cos(x)$ *are examples of FADF functions.*

Theorem 1. *Let $\Delta(X, U) = 0$ be a PDE with finite coefficients over the domain of its independent variables $X = (x_1, \dots, x_n)$. Then there exists a FADF function $f(X)$ and a small constant $\varepsilon > 0$ such that the function*

$$U(X) = \varepsilon\Big(f(X) + G^k(X)\Big)$$

approximately satisfies the equation $\Delta(X, U) \approx 0$, where $G(X)$ is a known continuous function and k represents the degree of the equation.

Proof. The detailed proof constructs an approximate solution by inserting a perturbed FADF-based function into the PDE. Due to the boundedness of derivatives and the finiteness of coefficients, the residual $\Delta(X, U)$ can be bounded above and made arbitrarily small by choosing an appropriate ε. For brevity, we refer the reader to the appendix for a full derivation. □

Corollary 1. *In practice, due to the complexity of high-dimensional PDEs, the exact value of ε is unknown. The RDSF algorithm enables us to estimate a near-optimal value of ε through random sampling and numerical evaluation of the residual $\Delta(X, U)$ over many realizations of X.*

Example 2 (Heat Equation in Four Variables). *Consider the heat equation:*

$$\Delta(x, y, z, t) = \frac{\partial u}{\partial t} - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0$$

Let the trial solution be:

$$u(x, y, z, t) = \varepsilon \cdot \sin(x)\sin(y)\sin(z)\sin(t)$$

with initial choice $\varepsilon = 0.01$.

Using the RDSF algorithm, we generate 500 random samples of (x, y, z, t) in a bounded domain and compute $\Delta(x, y, z, t, u)$. The residuals are visualized in Figure 1.

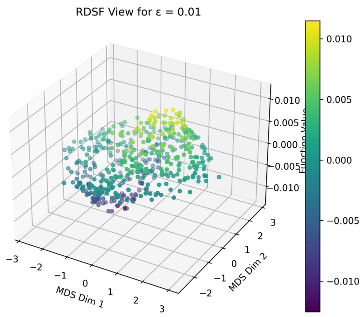

Figure 1. Residuals for $\varepsilon = 0.01$.

To improve accuracy, we repeat this process for $\varepsilon = 0.001$ and $\varepsilon = 0.0001$ (Figures 2, 3). The dispersion reduces as ε decreases.

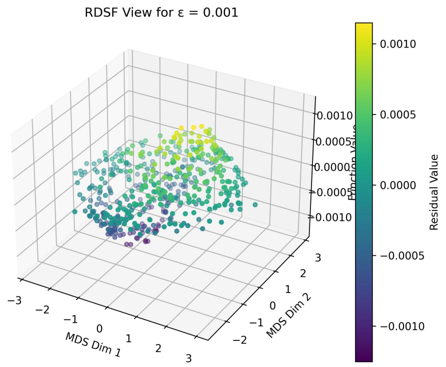

Figure 2. Residuals for $\varepsilon = 0.001$.

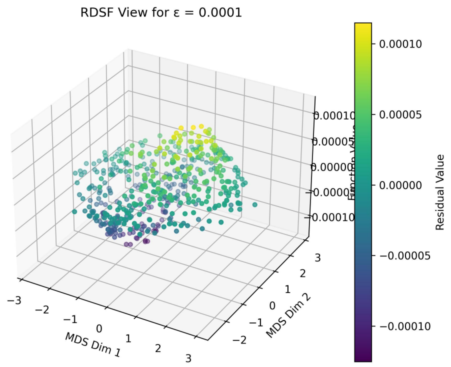

Figure 3. Residuals for $\varepsilon = 0.0001$.

From these results, we determine that the function

$$u(x, y, z, t) = 0.0001 \cdot \sin(x) \sin(y) \sin(z) \sin(t)$$

is a highly accurate approximate solution to the given PDE under the examined conditions.

3.2. Analysis and Investigation of the Dispersion of Prime Numbers

The distribution of prime numbers has long been a central theme in number theory and remains one of the most intriguing subjects in mathematics [12,13]. In this section, we demonstrate how the RDSF algorithm can be applied to analyze the dispersion pattern of prime numbers from a topological perspective, using synthetic integer spaces.

We construct an n-dimensional grid M, where each coordinate can take an integer value from 1 to 9. Each point $x \in M$ represents a vector $(x_1, x_2, \ldots, x_n)$, and the product $P(x) = x_1 \cdot x_2 \cdots x_n$ defines a non-prime candidate. We define the objective function $f : M \to \mathbb{N}$ as follows:

$$f(x) = \min\{t \in \mathbb{N} : P(x) + t \text{ is a prime number}\}.$$

This function measures the “distance” from each constructed number to the next prime [14]. We apply the RDSF algorithm to build the corresponding topological spaces and visualize the distribution of $f(x)$ across the grid. The results for 2D, 3D, 4D, and 5D spaces are combined in a unified visualization shown in Fig. 4.

In the two-dimensional case, the values of $f(x)$ range from 1 to 5 with frequencies of 14, 9, 5, 4, and 4, respectively. The highest concentration occurs at $f(x) = 1$ and $f(x) = 2$, indicating that most products lie very close to a prime number.

In the three-dimensional case, $f(x)$ spans a broader range: 1 through 13. The most frequent values are again within the range 1 to 5, with frequencies of 37, 18, 11, 9, and 13. A few outlier values like

11 and 13 appear with low frequency, illustrating the increased variability of prime gaps in higher dimensions.

In the four-dimensional case, $f(x)$ takes values between 1 and 20. Notably, the frequency remains concentrated within $f(x) = 1$ to $f(x) = 7$, with the highest at $f(x) = 1$ (75 instances). A few new values like $f(x) = 10, 13, 14, 16, 17, 19, 20$ emerge as dimensionality increases.

In the five-dimensional case, the dispersion of $f(x)$ extends further, ranging from 1 to 31. Despite this, the values of $f(x)$ are still highly concentrated in the range 1 to 13. Outliers such as $f(x) = 25, 26, 27$, and 31 occur with very low frequency. These results suggest that as the dimensionality increases, while more variation in the distance to the next prime emerges, the likelihood of small prime gaps remains dominant [15].

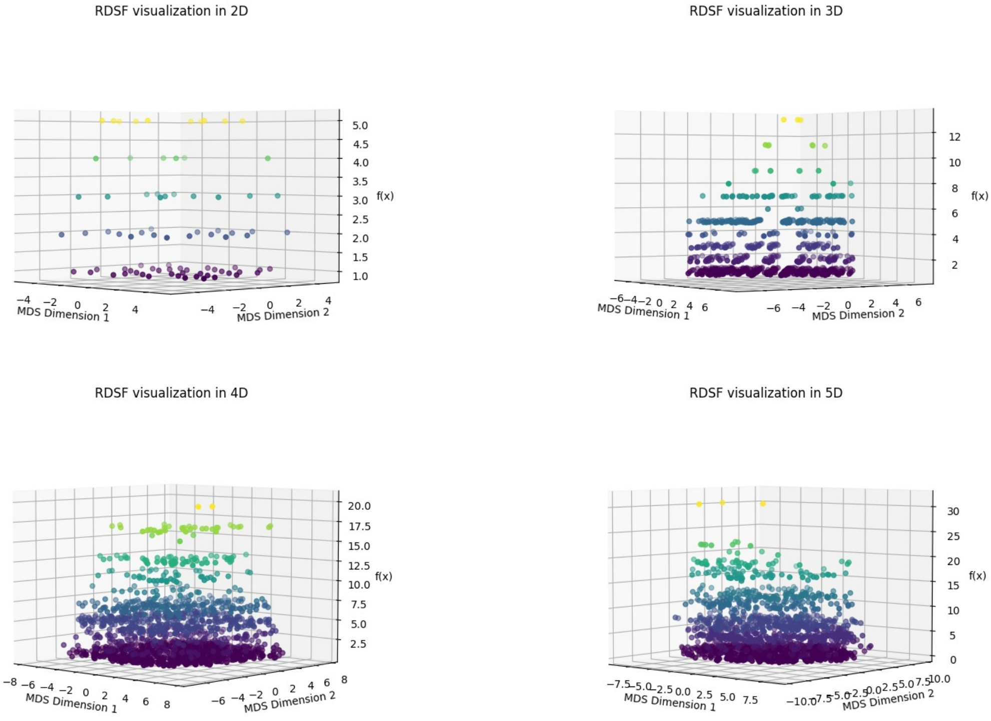

Figure 4. Distribution of $f(x)$ in 2D to 5D spaces: visualizing the distance to the next prime for synthetic integer vectors under the RDSF framework. Each subplot corresponds to a specific dimensionality (2D, 3D, 4D, 5D) and reveals the structured variation in the dispersion of prime distances across dimensions.

Based on the empirical distribution of $f(x)$ observed in multiple dimensions, we propose the following corollary:

Corollary 2. *Let $x \in \{1, \ldots, 9\}^m$ and define $P(x) = \prod_{i=1}^{m} x_i$. Then, the probability that $P(x) + t$ is a prime number decreases as t increases, and is maximized when $t \leq m$.*

This result indicates that within bounded integer grids, products of small integers tend to be closely followed by prime numbers, and this proximity exhibits a structured pattern across dimensions. The RDSF algorithm thus reveals an underlying topological signature in the dispersion of prime numbers.

3.3. Analysis and Investigation of the Behavior of Multivariate Arbitrary Real-Valued Functions

Let $f(x_1, x_2, \ldots, x_n)$ be a real-valued function of n variables. The RDSF algorithm provides a novel geometric and topological perspective for analyzing the behavior of such functions in a neighborhood of interest. Using Python, we randomly generate points $(x_1, x_2, \ldots, x_n)$ from a specified domain to construct the M-space, where each point corresponds to an input vector of the function. The greater the number of points sampled, the more accurate the resulting analysis becomes.

Following the RDSF steps, we evaluate f at each sampled point and then use the MDS algorithm to reduce the n-dimensional input space into two dimensions while preserving pairwise distances. By incorporating the value of the function as the third coordinate, a 3D plot is obtained that reveals how the function behaves near the chosen region of the domain.

Example 3. ***Rastrigin Function.*** *This is a widely-used benchmark in multivariate optimization. Its highly multimodal structure makes it a difficult problem for optimization algorithms [16]:*

$$f(x) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i) \right]$$

where:

- $X = (x_1, x_2, \ldots, x_n)$ *is an n-dimensional input vector,*
- A *is a constant (typically set to* 10*),*
- n *is the number of variables.*

In the two-variable case, the function surface contains numerous local minima (see Fig. 5). Applying the RDSF algorithm reveals this complex landscape.

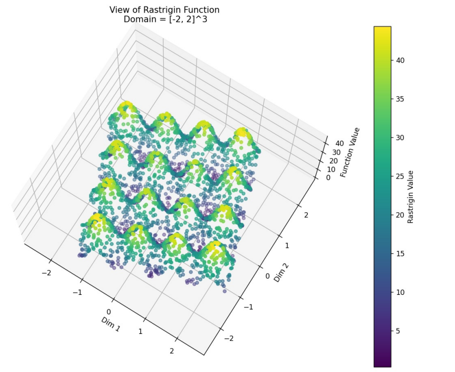

Figure 5. Rastrigin Function (2 variables) in the range $[-2, 2]$

For three variables, where direct 4D visualization is impossible, traditional analysis often requires fixing one variable. In contrast, RDSF embeds the entire 3D space into 2D and then adds the function value to visualize a 3D structure (see Figs. 6 and 7).

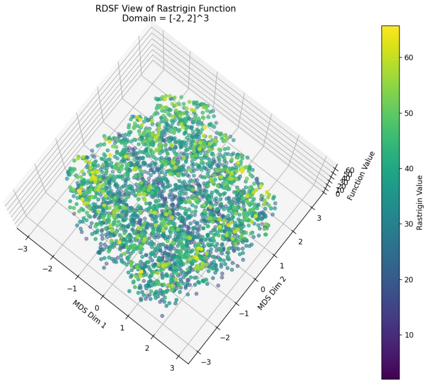

Figure 6. Rastrigin Function (3 variables) in the range $[-2, 2]$

In both ranges, the RDSF algorithm successfully reveals the dense distribution of local minima and the global minimum at the origin, consistent with known analytical results [16].

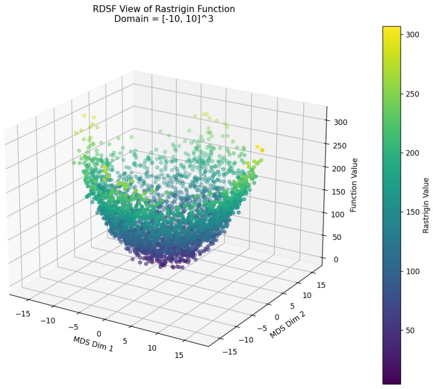

Figure 7. Rastrigin Function (3 variables) in the range $[-10, 10]$

Example 4. ***Ackley Function.*** *Another benchmark in multivariate optimization, characterized by a nearly flat outer region and a central hole with a global minimum [17]. Its complexity, featuring numerous local minima, poses a significant challenge. The n-dimensional Ackley function is defined as:*

$$f(X) = -a \cdot \exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1)$$

where $X = (x_1, x_2, \ldots, x_n)$, and recommended parameters are $a = 20$, $b = 0.2$, $c = 2\pi$. The global minimum occurs at $X = (0, 0, \ldots, 0)$, where $f(X) = 0$.

1. Intuitive Visualization Using RDSF. *We applied the RDSF algorithm to the 3-variable Ackley function without fixing any variable. Using Python, 300 random samples in $[-5, 5]^3$ were generated and evaluated. MDS was applied to embed into 2D, and the function values were used as the third coordinate.*

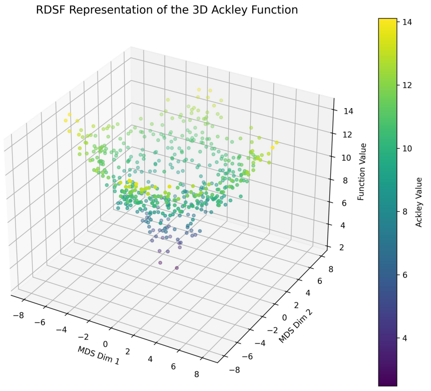

Figure 8. Three-dimensional RDSF representation of the 3-variable Ackley function

As shown, RDSF reveals a central basin (global minimum) surrounded by ridges. This provides intuitive insight into how regions of similar output are arranged in input space.

2. Comparison with Numerical Optimization. *To validate these insights, we used Particle Swarm Optimization (PSO) via the `pyswarms` library [18,19]. With 40 particles and 100 iterations in $[-5, 5]^3$, PSO found a near-global minimum:*

- ***Estimated minimum:*** $f(x^*) = 0.0009409701490210587$
- ***At point:*** $x^* = (-0.00020799, -0.00016255, -0.00030871)$

This aligns well with the RDSF visualization, supporting RDSF's utility as both an exploratory and complementary tool for numerical optimization.

This section highlights the flexibility and insight provided by RDSF in exploring complex function behaviors, whether the functions are highly multimodal (like Rastrigin and Ackley) or smooth with

doi:10.20944/preprints202505.1528.v2

a single minimum. The method enables a unified framework for analyzing arbitrary multivariate real-valued functions without requiring dimension-specific assumptions.

4. Conclusions

In this paper, we introduced the RDSF algorithm—Reducing the Dimension of the Space of Independent and Dependent Variables of Real-Valued Functions—a framework built upon manifold learning principles [20], specifically using MDS for dimensionality reduction [21]. RDSF enables visual exploration of high-dimensional function behavior by projecting independent variables into two dimensions and mapping function values as the third dimension.

We demonstrated its effectiveness across diverse problems: analyzing residuals in the heat equation [22] and revealing structural patterns in prime number distributions [23]. These case studies illustrated how RDSF provides a flexible tool for understanding the behavior of real-valued functions in multidimensional domains.

The visual representations produced by RDSF not only support exploratory data analysis [21] but also offer intuitive insights into mathematical and computational problems. We believe this framework can be extended to other analytical domains, enabling new methods of visualization and interpretation for complex systems.

Future work includes exploring other distance-preserving embedding methods [24,25], applying RDSF to empirical datasets in physics or biology, and formalizing theoretical properties of the algorithm in various functional spaces.

Appendix A. Proof of Theorem 1

We provide a justification for the theorem stating that for a PDE $\Delta(X,U) = 0$ with finite coefficients, there exists an approximate solution of the form

$$U(X) = \varepsilon\Big(f(X) + G^k(X)\Big)$$

where $f(X)$ is a FADF function, $G(X)$ is a known continuous function, k represents the degree of the equation, and $\varepsilon > 0$ is a small constant.

Step 1: Structure of the PDE

Assume the PDE $\Delta(X,U)$ is composed of partial derivatives of U with respect to variables $X = (x_1, \dots, x_n)$ and possibly nonlinear combinations of these derivatives. Let us denote the PDE in general form:

$$\Delta(X,U) = \sum_{|\alpha|\le m} a_\alpha(X) \cdot D^\alpha U(X) + \Phi(X, U(X), \nabla U(X), \dots) = 0$$

where: - α is a multi-index of derivative orders, - D^α denotes the partial derivative of order $|\alpha|$, - $a_\alpha(X)$ are finite-valued coefficient functions, - Φ is a smooth nonlinear expression in U and its derivatives.

Step 2: Inserting the Trial Function

Let $U(X) = \varepsilon\Big(f(X) + G^k(X)\Big)$, with f being FADF and G continuous. Then all required derivatives of U exist and are finite:

$$D^\alpha U(X) = \varepsilon\Big(D^\alpha f(X) + D^\alpha G^k(X)\Big)$$

Step 3: Plug into the PDE

We substitute $U(X)$ into $\Delta(X,U)$:

$$\Delta(X,U) = \sum_\alpha a_\alpha(X) \cdot \varepsilon\Big(D^\alpha f + D^\alpha G^k\Big) + \Phi(X, \varepsilon(f + G^k), \dots)$$

This simplifies to:

$$\Delta(X,U) = \varepsilon \sum_{\alpha} a_{\alpha}(X) \cdot \left(D^{\alpha} f + D^{\alpha} G^{k} \right) + \Phi(X, \varepsilon(f + G^{k}), \dots)$$

Note that: - The first term is $O(\varepsilon)$ because ε is factored out. - The second term, Φ, can be expanded in a Taylor series around $\varepsilon = 0$:

$$\Phi(X, \varepsilon(f + G^{k}), \dots) = \Phi(X, 0, \dots) + \varepsilon \cdot \Psi(X) + O(\varepsilon^{2})$$

for some bounded function Ψ depending on the partials of f and G.

Step 4: Estimating the Residual

Combining both terms:

$$\Delta(X,U) = \varepsilon \cdot T_1(X) + O(\varepsilon^{2})$$

for some bounded function $T_1(X)$. Hence, the total residual is:

$$|\Delta(X,U)| = O(\varepsilon)$$

Step 5: Conclusion

Therefore, by choosing ε small enough, we can ensure that the residual $|\Delta(X,U)|$ becomes arbitrarily small across the domain of interest. This justifies that $U(X) = \varepsilon(f(X) + G^{k}(X))$ is an approximate solution to the PDE:

$$\Delta(X,U) \approx 0 \quad \Rightarrow \quad \text{approximate solution.} \quad \square$$

References

1. Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., & Ziegler, H. (2008). Visual analytics: Scope and challenges. In *Visual Data Mining*, pp. 76–90. Springer.
2. Ma, Y. and Fu, Y. (2011). *Manifold Learning Theory and Applications*. CRC Press.
3. Lee, J. A., and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer.
4. Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics.
5. Borg, I., and Groenen, P. J. F. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer.
6. Buja, A., Cook, D., and Swayne, D. F. (2008). Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 17(4), 880–900.
7. Raissi, M., Perdikaris, P., and 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.
8. Sarnak, P. (1989). Some Applications of Modular Forms. *Cambridge Tracts in Mathematics*, Vol. 99, Cambridge University Press.
9. Kruskal, J. B., and Wish, M. (1978). *Multidimensional Scaling*. Sage Publications.
10. Evans, L. C. (2010). *Partial Differential Equations*. American Mathematical Society.
11. Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer Science & Business Media.
12. G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 6th edition, 2008.
13. P. Ribenboim. *The Little Book of Bigger Primes*. Springer-Verlag, 1996.
14. A. Granville. Harald Cramér and the distribution of prime numbers. *Scandinavian Actuarial Journal*, 1:12–28, 1995.
15. J. Pintz. Cramér vs. Cramér: On Cramér's probabilistic model for primes. *Functiones et Approximatio*, 37(2):361–376, 2007.
16. Rastrigin, L. A. (1974). Systems of extremal control. *Mir*, Moscow.
17. Ackley, D. H. (1987). *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic Publishers.

doi:10.20944/preprints202505.1528.v2

18. Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE.
19. Miranda, L. (2020). pyswarms: a research toolkit for particle swarm optimization in Python. *Journal of Open Source Software*, 5(50), 1996.
20. Ma, Y. and Fu, Y. (2011). *Manifold Learning Theory and Applications*. CRC Press.
21. Borg, I. and Groenen, P. J. F. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media.
22. Strauss, W. A. (1992). *Partial Differential Equations: An Introduction*. John Wiley & Sons.
23. Apostol, T. M. (1976). *Introduction to Analytic Number Theory*. Springer.
24. Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.
25. McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint* arXiv:1802.03426.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.