Report on Solving 1D and 2D Poisson Equations Using Finite Element Method (FEM) and Physics-Informed Neural Networks (PINNs)

Introduction

This project aims to replicate the methods used in the paper to solve 1D and 2D Poisson equations. The Poisson equation is a widely used partial differential equation (PDE) in physics and engineering. I employed both the Finite Element Method (FEM) and Physics-Informed Neural Networks (PINNs) to solve the equations. The project's goal was to compare the performance and accuracy of these two methods.

A partial differential equation (PDE) is a mathematical equation that involves functions of multiple variables and their partial derivatives. PDEs are used to formulate problems involving functions of several variables and are essential in describing various physical phenomena such as heat conduction, wave propagation, fluid dynamics, and electrostatics. In a PDE the usual form of the equation sets the and expression of the partial derivatives equal to a cerain function known as source factor.

Boundary conditions are essential to uniquely determine the solution of a PDE within a specified domain. They specify the behavior of the solution on the boundaries of the domain. There are different types of boundary conditions, including Dirichlet boundary conditions (specifying the value of the function on the boundary) and Neumann boundary conditions (specifying the value of the derivative of the function normal to the boundary).

The Poisson equation is a widely studied PDE given by $-\Delta u=f$ where Δ is the Laplace operator, u is the unknown function, and f is a known source term. We will use boundary conditions to ensure that the problem is well-posed and can be solved uniquely either analytically or numerically using methods like the Finite Element Method (FEM) or Physics-Informed Neural Networks (PINNs).

The Finite Element Method (FEM) is a numerical technique widely used to solve PDEs like the Poisson equation. It works by discretizing the domain into smaller sub-domains called elements, formulating a variational problem, and then solving the resulting system of algebraic equations. FEM is highly effective for problems with complex geometries and boundary conditions. On the other hand, Physics-Informed Neural Networks (PINNs) offer a modern approach by embedding the physical laws described by the PDEs into the neural network training process. PINNs minimize a composite loss function that includes the PDE residuals and boundary condition discrepancies, enabling the network to learn the solution u directly from the data.

1D Poisson Equation

Problem Definition

The 1D Poisson equation considered is:

$$-\frac{d^2}{dx^2}u = f(x)$$

with boundary conditions u(0)=0 and u(1)=exp(−1).

The exact solution was

$$u_e(x) = x * exp(-x^2)$$

And that was known that because the Poisson problem has an analytical solution. Therefore, to control the experiment and test the accuracy of the results, f was purposely selected to be:

$$f(x) = (-6x + 4x^3) * exp(-x^2)$$

Finite Element Method (FEM) Implementation

Methodology

1. **Mesh Generation**: The interval [0,1]] was divided into n elements.
2. **Function Space Definition**: Using Lagrange polynomials of degree 1.
3. **Boundary Conditions**: Dirichlet boundary conditions at x=0 and x=1.
4. **Variational Formulation**: The weak form of the Poisson equation was solved.
5. **Solving the System**: Using the FEniCS library.

Results

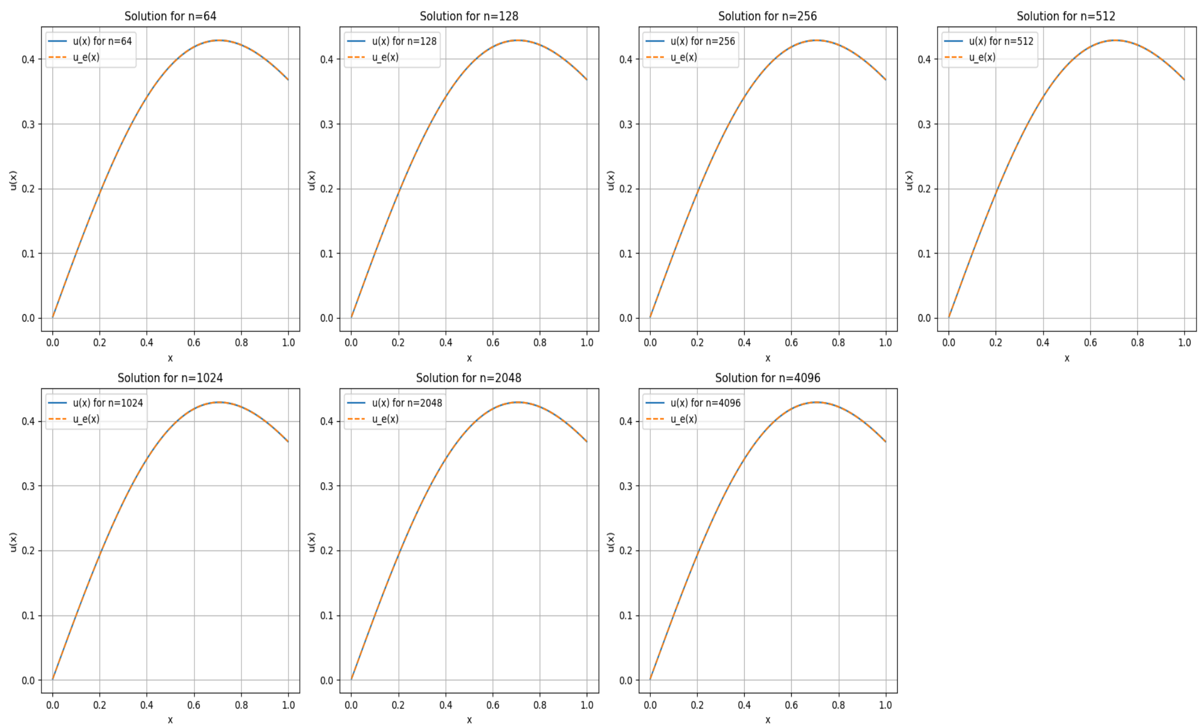
The results obtained for various mesh sizes are as follows:

- **Mean solve times** range from 0.003 seconds for n=64 to 0.012 seconds for n=4096.
- **Mean evaluation times** are approximately 0.002 seconds for all cases.
- **L2 norm error relative** decreases with increasing n, indicating better accuracy with finer meshes.

The results are summarized by the following table

Mesh Size (n)	Mean Solve Time (s)	Mean Eval Time (s)	L2 Norm Error	L2 Norm Error Relative
64	0.0032	0.0022	0.0012169	0.0001581
128	0.0030	0.0021	0.0003038	0.0000395
256	0.0033	0.0022	0.00007596	0.00000987
512	0.0037	0.0021	0.00001901	0.00000247
1024	0.0050	0.0021	0.00000472	0.00000061
2048	0.0074	0.0022	0.00000121	0.00000016
4096	0.0126	0.0023	0.00000029	0.00000004

and we can graphically see how close the approximate solution was to the exact solution in the graphs below



Since the error is fairly small for all n, it is difficult to really see a difference from the graphs but it is a nice way to get a concrete understanding of the effectiveness of this method.

Physics-Informed Neural Networks (PINNs) Implementation

Methodology

1. **Neural Network Architecture**: Various architectures with different layers and neurons.
2. **Loss Function**: Combined loss from the PDE residual and boundary conditions.
3. **Optimization**: Using Adam optimizer for training the network.

Results

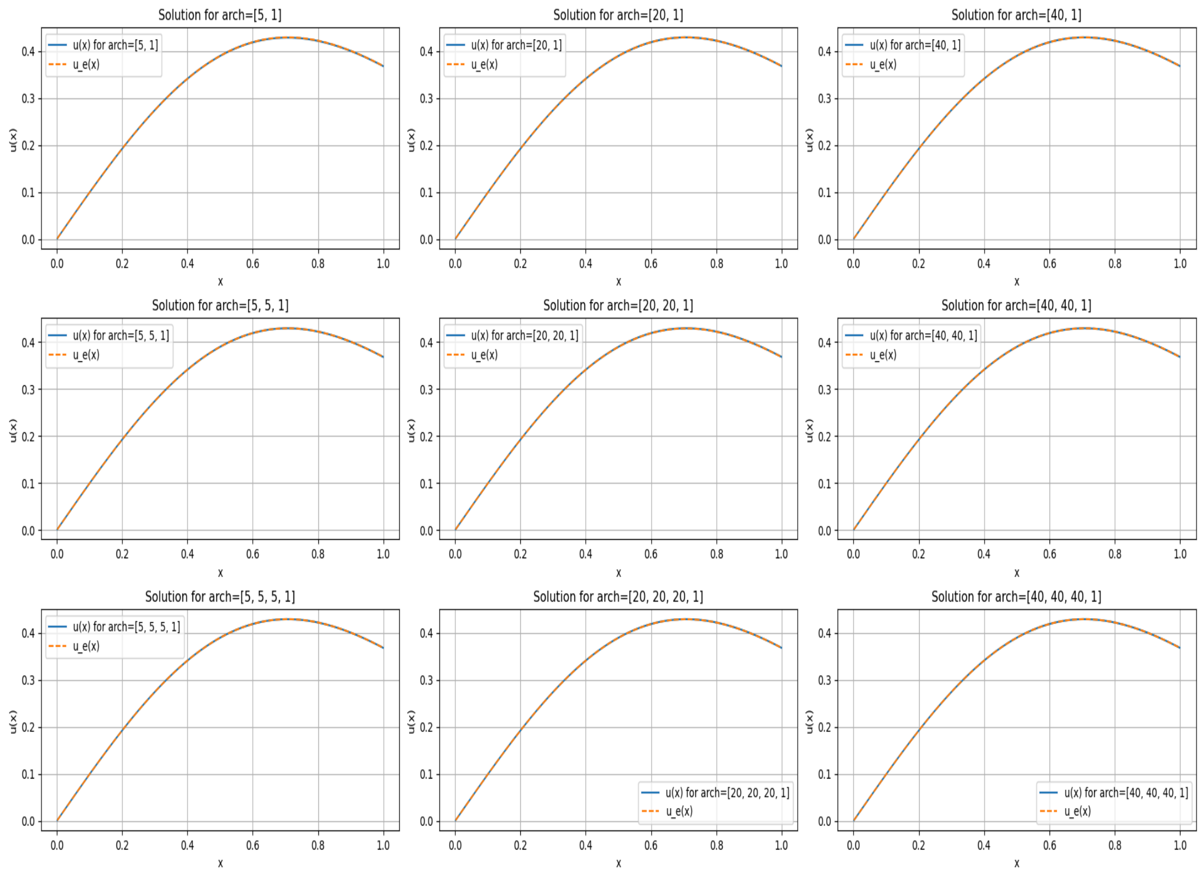
The results obtained for various network architectures are as follows:

- **Mean solve times** range from 0.979 seconds for architecture [5,1] to 8.974 seconds for architecture [40,40,40,1].
- **Mean evaluation times** are approximately 0.01 seconds for all cases.
- **L2 norm error relative** decreases with increasing complexity of the network, indicating better accuracy with deeper and wider networks.

Once again the results are summarized in the table

Architecture	Mean Solve Time (s)	Mean Eval Time (s)	L2 Norm Error	L2 Norm Error Relative
[5, 1]	0.979	0.011	0.006428	0.0008353
[20, 1]	1.677	0.010	0.0004049	0.0000526
[40, 1]	2.273	0.010	0.001384	0.0001799
[5, 5, 1]	1.744	0.0077	0.001105	0.0001437
[20, 20, 1]	3.260	0.0077	0.005051	0.0006563
[40, 40, 1]	5.750	0.0091	0.0003575	0.0000465
[5, 5, 5, 1]	2.677	0.010	0.0008610	0.0001119
[20, 20, 20, 1]	4.998	0.0095	0.001771	0.0002302
[40, 40, 40, 1]	8.974	0.0098	0.0000875	0.0000114

and the solutions are shown in a graph to concretely see the approximation quality



Comparison Between FEM and PINNs

- **FEM** showed a rapid decrease in relative L2 norm error with increasing mesh size. The method was computationally efficient with solve times remaining under 0.01 seconds for mesh sizes up to 4096 elements.
- **PINNs** displayed higher solve times compared to FEM. The accuracy of PINNs improved with more complex architectures, but the relative L2 norm error was generally higher compared to FEM for the same problem.

Finite Element Method (FEM) Implementation

Methodology (continued)

- **Mesh Generation**: The unit square [0,1]×[0,1][0, 1] \times [0, 1][0,1]×[0,1] was divided into n×nn \times nn×n elements.
- **Function Space Definition**: Using Lagrange polynomials of degree 1.
- **Boundary Conditions**: Dirichlet boundary conditions at y=0y = 0y=0 and Neumann boundary conditions elsewhere.
- **Variational Formulation**: The weak form of the Poisson equation was solved.
- **Solving the System**: Using the FEniCS library to solve the linear system derived from the variational formulation.

Results

The results obtained for various mesh sizes are as follows:

- **Mean solve times** range from 0.25 seconds for n=100n = 100n=100 to 15.07 seconds for n=1000n = 1000n=1000.
- **Mean evaluation times** range from 0.039 seconds for n=100n = 100n=100 to 0.619 seconds for n=1000n = 1000n=1000.
- **L2 norm error relative** decreases with increasing nnn, indicating better accuracy with finer meshes.

Δu=∂x2∂2u+∂y2∂2u

Problem Definition

The 2D Poisson equation considered is:

$$-\left(\frac{d^2}{dx^2}u + \frac{d^2}{dy^2}u\right) = f(x)$$

with boundary conditions u(x,0)=0 (Dirichlet condition on the bottom edge), ∂u∂x(0,y)=0 and ∂u∂x(1,y)=0)=0 (Neumann conditions on the left and right edges), and ∂u∂y(x,1)=0 on the top edge

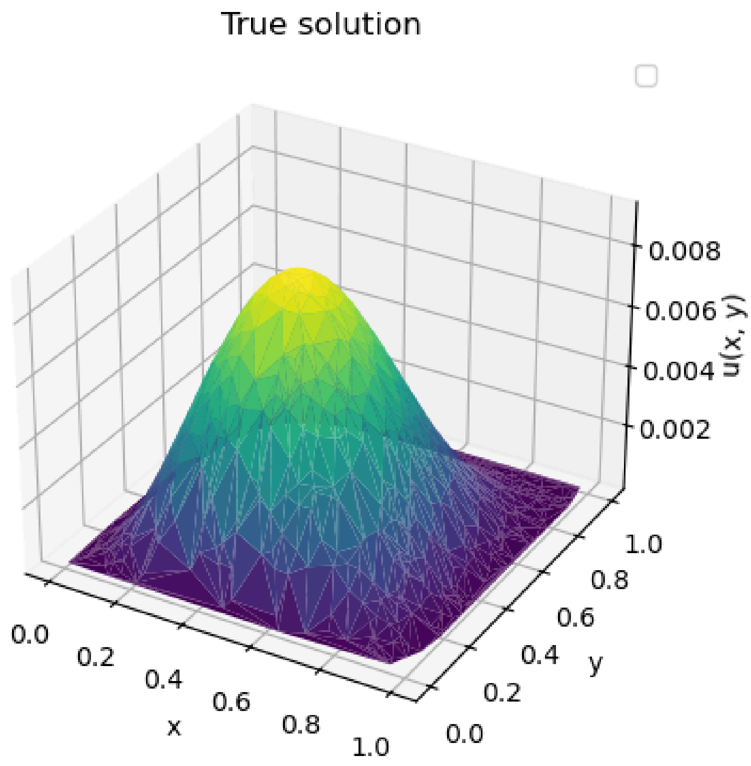
The dominion upon which this problem was solved is the unitary square

The exact solution was

$$u_e(x) = x^2(x - 1)^2 * y(y - 1)^2$$

And that was known that because the Poisson problem has an analytical solution. Therefore, to control the experiment and test the accuracy of the results, f was purposely selected accordingly to be the laplacian operator of that function.

The real solution is plotted below so it can be compared with the results obtained:



Finite Element Method (FEM) Implementation

Methodology (continued)

- **Mesh Generation**: The unit square [0,1]×[0,1][0, 1] \times [0, 1][0,1]×[0,1] was divided into n×nn \times nn×n elements.
- **Function Space Definition**: Using Lagrange polynomials of degree 1.
- **Boundary Conditions**: Dirichlet boundary conditions at y=0y = 0y=0 and Neumann boundary conditions elsewhere.
- **Variational Formulation**: The weak form of the Poisson equation was solved.
- **Solving the System**: Using the FEniCS library to solve the linear system derived from the variational formulation.

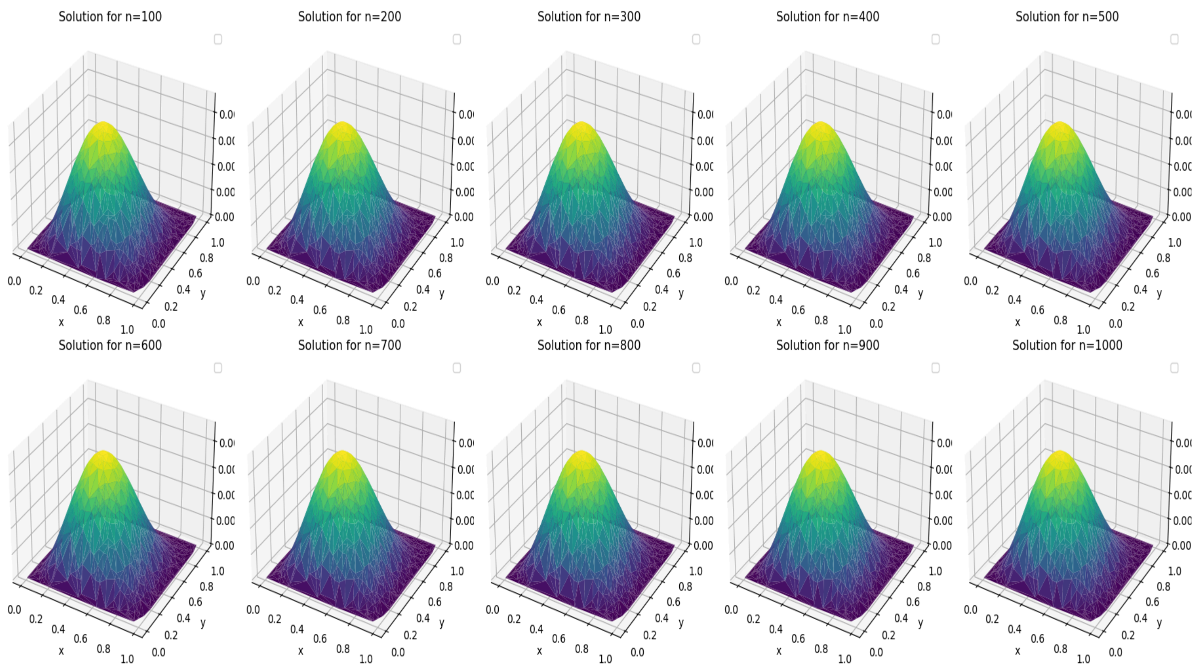
Results

The results obtained for various mesh sizes are as follows:

- **Mean solve times** range from 0.25 seconds for n=100n = 100n=100 to 15.07 seconds for n=1000n = 1000n=1000.
- **Mean evaluation times** range from 0.039 seconds for n=100n = 100n=100 to 0.619 seconds for n=1000n = 1000n=1000.
- **L2 norm error relative** decreases with increasing nnn, indicating better accuracy with finer meshes.

Mesh Size (n)	Mean Solve Time (s)	Mean Eval Time (s)	L2 Norm Error	L2 Norm Error Relative
100	0.2536	0.0388	0.0001909	0.0015510
200	0.8252	0.0780	0.0000477	0.0003876
300	1.9685	0.1303	0.0000212	0.0001721
400	2.8591	0.1603	0.0000119	0.0000969
500	6.0259	0.2090	0.0000076	0.0000621
600	4.5224	0.2052	0.0000053	0.0000432
700	4.8730	0.2242	0.0000039	0.0000317
800	7.8023	0.3462	0.0000030	0.0000242
900	10.2093	0.4903	0.0000024	0.0000191
1000	15.0672	0.6190	0.0000019	0.0000156

Here are the plotted approximations obtained with the FEM method, they all look very similar to the true solution.



Physics-Informed Neural Networks (PINNs) Implementation

Methodology

1. **Neural Network Architecture**: Various architectures with different layers and neurons were tested.
2. **Loss Function**: The loss function was a combination of the PDE residual and boundary conditions.
3. **Optimization**: The Adam optimizer was used for training the network.

Results

1. **Mean Solve Time**: The solve time increases with the complexity of the architecture. Simpler architectures like [5, 1] and [5, 5, 1] have shorter solve times, whereas more complex architectures such as [20, 1] and [20, 20, 20, 1] exhibit significantly longer solve times. This increase in computational cost is expected as the number of parameters and the depth of the network grow.
2. **Mean Evaluation Time**: The evaluation time remains relatively consistent across different architectures, averaging around 0.02 seconds. This indicates that once the network is trained, the time to evaluate the solution is not heavily dependent on the network complexity.
3. **L2 Norm Error**: The L2 norm error indicates the absolute error between the predicted and the exact solution. The results show that simpler networks like [5, 5, 1] achieve lower L2 norm errors, suggesting better accuracy. However, more complex architectures do not always guarantee lower errors, as seen in the case of [20, 20, 20, 1].
4. **L2 Norm Error Relative**: This metric provides a normalized measure of the error. The relative errors follow a similar trend to the absolute errors, with simpler architectures like [5, 5, 1] performing better in terms of relative accuracy.

In summary, the results suggest that while increasing the network complexity generally leads to higher computational costs, it does not always correlate with improved accuracy. Simpler architectures like [5, 5, 1] strike a balance between computational efficiency and solution accuracy, making them suitable for solving the 2D Poisson equation using PINNs. Further optimization of the network architecture and training process could potentially enhance the performance of more complex models.

Architecture	Mean Solve Time (s)	Mean Eval Time (s)	L2 Norm Error	L2 Norm Error Relative
[5, 1]	0.6814	0.0359	0.0932	0.7574
[20, 1]	28.0389	0.0258	0.0170	0.1381
[40, 1]	2.1163	0.0199	0.0200	0.1624
[5, 5, 1]	1.3061	0.0201	0.0107	0.0866
[20, 20, 1]	2.9597	0.0183	0.0135	0.1098
[40, 40, 1]	5.8788	0.0185	0.0196	0.1593
[5, 5, 5, 1]	2.0460	0.0217	0.0190	0.1543
[20, 20, 20, 1]	4.8713	0.0213	0.0237	0.1927

Conclusions

The FEM method confirms to be the most reliable tool in terms of rapidity of execution, and accuracy. The main difference is in the solving time since the networks ar every slow to train compared to the FEM. The evaluation time is actually fast(er) with the PINNs.

The difference between the two methods is particularly evident in the case of the 2D Poisson problem. In this case the PINN has performed particularly poorly and the explanation for this might be that it was trained using a fairly sparse set of points, but for hardware reasons (the computer would crash) i could not use 2000 samples on both the dimensions of the square like it was done in the paper, so i has to stop at 256 which still took hours to train.