Literature Report Presentation

Can deep learning be used to enhance numerical approximation methods like the finite element method?

Titus Ex

February 19, 2021

Overview

1. Problem Definition
2. Optimal Test Functions
3. Deep Learning Approaches
4. Research Questions

The advection-diffusion equation

The PDE that will be used in the problem setup is the advection-diffusion equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - \nabla \cdot (\boldsymbol{v} u) \tag{1}$$

Here, u is some variable of interest, D is the diffusion coefficient, and $\boldsymbol{v}$ is the velocity field of u. This equation describes that transport of physical quantities through a system due to two different effects called *advection* and *diffusion*.

The advection-diffusion equation

Often we are interested in solving (1) together with a set of boundary conditions, so that we may arrive at the following boundary value problem

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u) - \nabla \cdot (\boldsymbol{v} u) \\ u(t,0) = g \\ u(t,1) = h \end{cases} \tag{2}$$

- Typically, one is interested in approximating the solution of (2), especially when the problem does not have a solution in the classical sense.
- One of the most common numerical methods that is used to approximate u in (2) is called the finite element method.
- The finite element method do not only say something about how to find an approximate solution to the linear advection-diffusion equation, but also about the non-linear Navier-Stokes equations, which are much more widely applicable.

The Instability of Numerical Solutions

To see why the the (Galerkin) finite element method can suffer from instability, consider the 1D steady-state advection-diffusion equation

$$\begin{cases} -\epsilon \dfrac{d^2 u}{dx^2} + v \dfrac{du}{dx} = 0 \text{ for } x \in (0,1) \\ u(0) = 0 \\ u(1) = 1 \end{cases} \tag{3}$$

which has solution

$$u(x) = \frac{1 - e^{xv/\epsilon}}{1 - e^{v/\epsilon}} \tag{4}$$

The Instability of Numerical Solutions

To approximate u in this boundary value problem, let's consider the finite difference method which discretises the domain into N points and approximates the first and second derivatives in the following way

$$\begin{aligned} u'(x_i) &= \frac{-u(x_{i-1}) + u(x_{i+1})}{2h} \\ u''(x_i) &= \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2} \end{aligned} \tag{5}$$

Substituting these approximations into (3) gives

$$-\epsilon \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + v \frac{-u_{i-1} + u_{i+1}}{2h} = 0 \tag{6}$$

Instability of Numerical Solutions

If the solution is assumed to be of the form $u_i = r^k = r^{k-1}r$ such that $u_{i-1} = r^{k-1}$ and $u_{k+1} = r^{k-1}r^2$, then it follows from (6) that

$$r = \frac{1 + \frac{vh}{2\epsilon}}{1 - \frac{vh}{2\epsilon}} \tag{7}$$

The ratio between transport through advection and through diffusion is called the Péclet number and is denoted by $Pe = v/\epsilon$. What happens when we examine the grid Péclet number: $vh/2\epsilon$? The approximate solution will oscillate!
Convential Galerkin discretisations lead to the central difference approximations and therefore suffer from the same instability, see [1].

Instability of Numerical Solutions

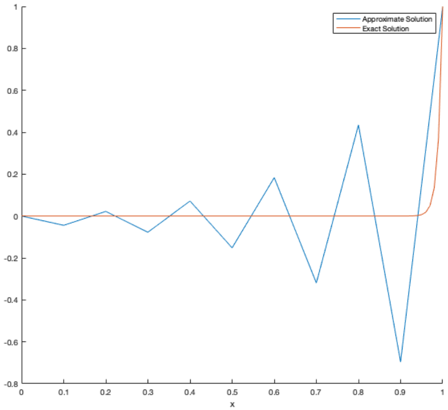

Figure 1: Galerkin FEM approximate solution vs exact solution, given $\epsilon = 0.01$ and $v = 1$.

Optimal Test Functions

Dealing with an unstable solution

- Several approaches have been used to deal with this problem, such as increasing the number of grid points and adding artificial diffusion.
- In this research thesis the problem of instability will be tackled using a Petrov-Galerkin finite element scheme with so called *optimal test functions*.
- The test functions are optimal in the sense that they can guarantee the stability of the approximate solution.

Optimal Test Functions

Consider the following variational boundary value problem

$$\text{Find } u \in U : b(u, v) = l(v) \quad \forall v \in V \tag{8}$$

where U and V are real Hilbert spaces (normed by $\|\cdot\|_U$ and $\|\cdot\|_V$), and l is a continuous linear form defined on the test space V, and $b(\cdot)$ denotes a bilinear form that is defined on $U \times V$ that is continuous

$$|b(u, v)| \leq M\|u\|_U\|v\|_V \tag{9}$$

and that satisfies the inf-sup condition

$$\inf_{\|u\|_U=1} \sup_{\|v\|_V=1} b(u, v) \geq \gamma \tag{10}$$

with $\gamma > 0$ and assume that $\{v \in V : b(u, v) = 0 \quad \forall u \in U\} = \{0\}$.

Optimal Test Functions

If u in (8) is approximated using the following Galerkin method

$$\begin{cases} \text{Find } u_n \in U_n \text{ satisfying} \\ \quad b(u_n, v_n) = l(v_n) \ \forall v \in V_n \end{cases} \tag{11}$$

with $U_n \subseteq U$, $V_n \subseteq V$, $\dim U_n = \dim V_n$, and if (10) holds for U_n and V_n as well, then the following theorem holds

Theorem (Babuška)

Under the above assumptions, the exact and the discrete problems (8) and (11) are uniquely solvable. Furthermore,

$$\|u - u_n\|_U \leq \frac{M}{\gamma_n} \inf_{w_n \in U_n} \|u - w_n\|_U \tag{12}$$

Optimal Test Functions

Let's define a new norm called the *energy norm*

$$\|u\|_E \stackrel{\text{def}}{=} \sup_{\|v\|_V=1} b(u,v) \tag{13}$$

Define the map $T : U \to V$ such that for every $u \in U$, $Tu \in V$ is the unique solution of

$$(Tu, v)_V = b(u,v), \quad \forall v \in V, \tag{14}$$

where $(\cdot,\cdot)_V$ refers to the inner product on V. If we then consider the following finite dimensional trial subspace

$$U_n = \text{span}\{e_j : j = 1, \ldots, n\} \tag{15}$$

for some linearly independent set of functions e_j in U we can formulate the following definition.

Optimal Test Functions

Definition (Optimal Test Space)

Every trial subspace U_n, as in (15), has its corresponding **optimal test space**, defined by

$$V_n = \text{span}\{Te_j : j = 1, \ldots, n\} \tag{16}$$

The "optimal" test spaces defined as above are optimal in that they result in the optimal ratio of continuity constant to stability constant when U is endowed with the energy norm. Specifically, the following theorem holds

Theorem (Best Approximation Error in Energy Norm)

Let V_n be the optimal test space corresponding to a finite dimensional trial space U_n. Then the error in the Petrov-Galerkin scheme (11) using $U_n \times V_n$ equals the best approximation error in the energy norm, i.e.

$$\|u - u_n\|_E = \inf_{w_n \in U_n} \|u - w_n\|_E \tag{17}$$

Deep Learning Approaches

In the past few years, Deep Learning has been used more extensively to solve PDEs and boundary value problems. There are several advantages associated with using deep learning approaches to solve PDEs

- Provided that they converge correctly, neural networks have no issue with instability of the approximate solution.
- Generally, deep learning approaches do not suffer from the curse of dimensionality as no grids are used.
- Once trained, neural networks can produce approximate solutions almost instantaneously.

In this literature presentation two deep learning approaches will be covered: *Variational Physics-Informed Neural Networks* and *Deep Operator Networks*.

Variational Physics-Informed Neural Networks (VPINNs)

Consider the following problem

$$\begin{aligned} \mathcal{L}u(x) &= f(x),\ x \in \Omega \\ &\text{s.t.} \\ u(x) &= h(x),\ x \in \partial\Omega \end{aligned} \tag{18}$$

where $\mathcal{L}$ contains differential operators. The goal is to approximate the solution $u(x)$, using a neural network $u_{NN}(x)$. As the name suggests the VPINN works from the variational form, so the PDE in (18) is multiplied by a test function v and then integrate the result to obtain the variational form

$$\begin{aligned} (\mathcal{L}^q u_{NN}(x), v(x))_\Omega &= (f(x), v(x))_\Omega \\ u(x) &= h(x),\ x \in \partial\Omega \end{aligned} \tag{19}$$

Variational Physics-Informed Neural Networks (VPINNs)

Using the variational form, the variational residual is defined as follows

$$\text{Residual}^v = \mathcal{R} - F - r^b, \tag{20}$$

$$\mathcal{R} = (\mathcal{L}^q u_{\text{NN}}, v)_\Omega, \; F = (f, v)_\Omega,$$

$$r^b(x) = u_{\text{NN}}(x) - h(x), \; \forall x \in \partial\Omega \tag{21}$$

which is enforced for all admissible test functions in the space

$$V_k = \text{span}\{v_k, k = 1, 2, \ldots, K\} \tag{22}$$

Using this residual and r^b the *variational loss function* can be defined

$$L^v = L_R^v + L_u \tag{23}$$

$$L_R^v = \frac{1}{K} \sum_{k=1}^{K} |\mathcal{R}_k - F_k|^2 \,, \; L_u = \tau \frac{1}{N_u} \sum_{i=1}^{N_u} \left| r^b(x_{u_i}) \right|^2$$

Variational Physics-Informed Neural Networks (VPINNs)

There are several advantages that arise from using the variational loss function

- Using integration by parts to reduce the order of the differential operators, reduces the required regularity in the solution space. This means that VPINNs will be less computationally expensive compared to deep learning approaches that work directly from the strong form.
- VPINNs use a relatively small number of quadrature points to evaluate the integrals used in the weak form, when compared to the penalising points required in other deep learning approaches, like the PINNs.

Deep Operator Networks (DeepONets)

Deep Operator Networks (Lu Lu et al., 2020, [3]) use a different approach. Instead of approximating a function, DeepONets can be trained to approximate operators, which are mappings from a space of functions into another space of functions.

- Let G be an operator taking an input function u, let $G(u)$ be the corresponding output function, and let y be a point in the domain of $G(u)$. DeepONets can be used to approximate $G(u)(y)$.
- The universal approximation theorem for continuous functions is very famous in deep learning. The legitimacy of DeepONets comes from a less known result that is due to Chen & Chen [5].

Deep Operator Networks (DeepONets)

Theorem (Universal Approximation Theorem for Operators)

Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X, K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p, m, constants $c_i^k, \xi_{ij}^k, \theta_i^k, \zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$, $j = 1, \ldots, m$, such that

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{branch} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{trunk} \right| < \epsilon \tag{24}$$

holds for all $u \in V$ and $y \in K_2$ and where $G(u)(y)$ denotes the function value of the output function corresponding to the operator G and the input function u at the point y in the domain.

Deep Operator Networks (DeepONets)

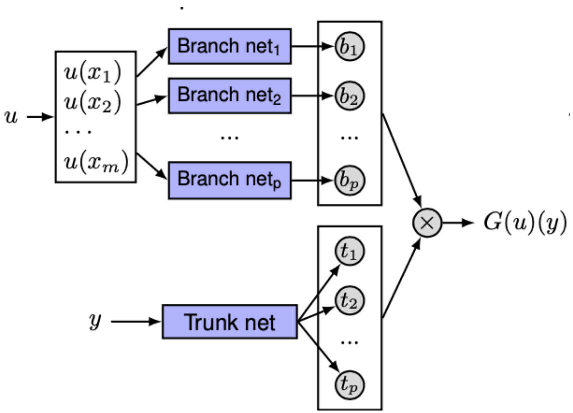

Figure 2: Deep Operator Approach, [3].

Deep Operator Networks (DeepONets)

The DeepONets have several advantages

- They offer a lot more flexibility, since the essentially use the input function u as a variable, whereas the VPINN is trained for a specific value of u (this could be very useful in solving some of the research questions of this thesis).
- They are able to generalise a lot better than regular fully connected neural networks.

Research Questions

Research Question 1

Can deep learning approaches be used to generate optimal test functions corresponding to particular trial functions, that improve the stability/accuracy of finite element methods?

More specifically, the goal is to try to approximate the optimal test function $Tu \in V$, that is, the unique solution of

$$(Tu, v)_V = b(u, v), \quad \forall v \in V, \tag{25}$$

and see whether it can be used to improve the stability/accuracy when compared to classic FEM codes. As Tu is a function, the VPINNs will be used to test out this research first.

Research Question 2

Can deep learning approaches be used to generate optimal test functions, using trial functions as variables, that improve the stability/accuracy of finite element methods?

In this case the goal is to try to approximate the operator T defined as the map from trial to test space, such that

$$(Tu, v)_V = b(u, v), \quad \forall v \in V, \tag{26}$$

This research question differs from the first in that instead of approximating the *function* Tu, the *operator* T will be approximated. For this research question, DeepONets appear to be very suited, as they are designed to approximate operators.

Research Question 3

Can deep learning approaches be used to generate optimal test functions, using problem specific parameters like the diffusion coefficient as variables, to improve the stability/accuracy of finite element methods?

This is essentially a follow up on the previous two research questions. The goal here is to see whether it's possible to use problem specific parameters like the diffusion coefficient as variables for a neural network.

Research Question 4

Can deep learning approaches be used to estimate the finite element integrals?

The previous three research questions all revolved around approximating optimal test functions and using them in the finite element method. However, the finite element method does not use the point-wise values of the test functions, it only uses the integrals in the weak form. Instead of approximating the optimal test functions and then using sampling to approximate their integrals, it could be more efficient and more accurate to approximate the integrals directly.

Research Question 5

Can deep learning methods be used to estimate the perturbation introduced to the finite element integrals, caused by using optimal test functions?

Suppose that v is the usual test function used and w is the optimal test function. The effect of using an optimal test function can be seen as a perturbation to the finite element integrals. If the integral $\int (v - w)u$ is close to zero (with trial function u), then no additional stabilization would be needed. If this integral is not close to zero, it means that using the optimal test function would change the coefficients used in the finite element equations.

Preliminary Results

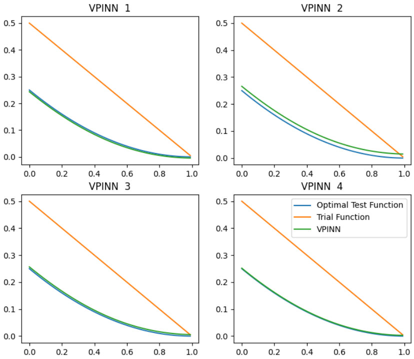

Figure 3: Testing four different VPINNs to approximate the optimal test function corresponding to the 1D pure convection problem.

Preliminary Results

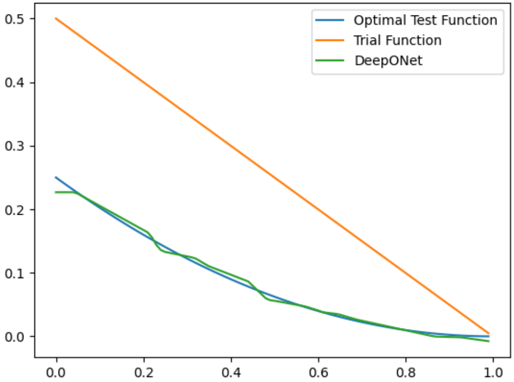

Figure 4: Testing the DeepONet architecture to approximate optimal test functions corresponding to the 1D pure convection problem.

References

[1] Alexander N. Brooks, Thomas J.R. Hughes. *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations.* Computer Methods in Applied Mechanics and Engineering, 1982.

[2] Leszek Demkowicz and Jay Gopalakrishnan. *A Class of Discontinuous Petrov-Galerkin Methods. II. Optimal Test Functions.* Portland State University, PDXScholar, 2010.

[3] Lu Lu, Pengzhan Jin, and George Em Karniadakis. *DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*. Division of Applied Mathematics, Brown University, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 2020.

[4] Ehsan Kharazmi, Zhongqiang Zhang, George EM Karniadakis. *VPINNs: Variational Physics-Informed Neural Networks For Solving Partial Differential Equations*, 2019.

[5] T.Chen and H.Chen. *Universal approximation to non-linear operators by neural networks with arbitrary activation functions and its application to dynamical systems.* IEEE Transactions on Neural Networks, 6(4):911–917, 1995.