Ministry of Higher Education and
Scientific Research
University of Mosul
College of Computer Science and
Mathematics
Department of Mathematics

Solving Volterra Integro-Differential Equations Using the Gauss-Legendre with Physics-Informed Neural Networks

**A Thesis Submitted to the Council of the College of
Computer Science and Mathematics
University of Mosul
as a Partial Fulfillment of Requirements
for the Degree of Doctor of Philosophy in
Mathematics/ Computational Mathematics**

By

Oday Ahmed Jasim Mohammed

Supervised by

Prof. Dr. Abdulghafor Mohammed Ameen Khudhur

2025 A.D. **1446 A.H.**

Abstract

Deep learning (DL) is being widely used and has a key place in many disciplines since it is an effective way to investigate the behavior of many different systems. Deep learning has recently been utilized to solve differential equations using physics-based input. Physics-informed neural networks (PINNs) are a novel deep learning model that excels at solving both inverse and forward non-linear PDE problems. PINNs may be trained as surrogate models for approximation solutions to the partial differential equations (PDEs) without label data by embedding the physical information outlined by PDEs in feedforward neural networks. Owing to neural networks exceptional capacity to depict intricate interactions, several PINN-based techniques have been created to address a range of issues, including fractional partial differential equations, stochastic partial differential equations, and integro-differential equations.

The PINN is used in this thesis to solve Volterra integro-differential equations (VIDEs) of 2^{nd} order. Hence, after being converted to a differential equation (DEs), the series of approximated closed-form iterated solutions is obtained using the generic Gauss legendar quadrate. Furthermore, three examples are provided and resolved to demonstrate the dependability, effectiveness, and suitability of the suggested approach. It is found the $tanh$ and sin is the activation's functions and the Optimization method L-BFGS and L-BFGS-B more appropriate for solving 2^{nd} order VIDEs by PINN. We have found The square of the relative error in all cases ranges between 0.1e-5 and 0.1e-7 and this is an excellent result and too the training duration for all cases was between 160-280 seconds. The results demonstrate the advantages faster convergence and improved accuracy.

This thesis also presents a new neural network implemented using Python and the IDRLnet library. The new neural network of the open-source

deep learning package is designed to solve PDEs using PINNs and is mainly used for physics-informed deep learning. This thesis uses PINN to solve the 2^{nd} order VIDE, in which a series of iterative solutions are established based on the Gauss Legendre quadratic method after converting the 2^{nd} order VIDE into a Differential equation. In addition, three different examples are presented and solved using the new algorithm to highlight the accuracy and efficiency of the pro-posed method. It is found that the L2 relative error in all cases is between 0.1e-3 and 0.1e-5, which is a good result, and the "training" time for all cases are 414.08, 724.2, 1339.31, and 2260.8 seconds. Finally, it is concluded that the interval functions (function domain) in the solution region are inversely proportional to the solution's accuracy. In addition, the number of iterations of training the network directly affects the solution's accuracy. Also, the relatively higher the degree of the Gauss-Legendre method, the more accurate the solution to the Volterra integral equation will be.

After careful evaluation, we have found that the Deepxde library surpasses the IDRLnet library in many aspects: The Deepxde library provides precision absolute errors of 5 to 8 decimal places, while the IDRLnet library yields 3 to 5 decimal places. Consequently, the Deepxde library exhibits higher levels of accuracy. The L2 relative error in the Deepxde library yields an accuracy ranging from 5 to 7 decimal places, but the IDRLnet library only provides an accuracy ranging from 3 to 5 decimal places. The execution time for executing the Python program in the Deepxde library ranges from 160 to 280 seconds, but in the IDRLnet library, it is 414.08, 724.2, 1339.31, and 2260.8 seconds.

Finally, should be note that we used Python (3.9.13) with several libraries such as: Numpy (2.2.0), Deepxde (1.12.2), IDRLnet (2.0.0), PyYAML (6.0.2), paddlepaddle (2.6.2), matplotlib (3.10.0), Torch (2.5.1), mpmath (1.4.0), pandas (2.2.4), packaging (2.24), and many more.

وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم الرياضيات

حل معادلة فولتيرا التكاملية-التفاضلية باستخدام طريقة كاوس-ليجندر مع الشبكات العصبية المستنيرة بالفيزياء

اطروحة مقدمة
الى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة دكتوراه فلسفة في
الرياضيات/ الرياضيات الحاسوبية

من قبل

عـــدي أحمـــــد جاســـم محمـــــد

بإشراف

أ.د. عبد الغفور محمد أمين خضر عبد الكريم

١٤٤٦ هـ ٢٠٢٥ م

المستخلص

يتم استخدام التعلم العميق (DL) على نطاق واسع وله مكانة رئيسية في العديد من التخصصات لأنه وسيلة فعالة للتحقيق في سلوك العديد من الأنظمة المختلفة. تم استخدام التعلم العميق مؤخرًا لحل المعادلات التفاضلية باستخدام مدخلات قائمة على الفيزياء. الشبكات العصبية المستنيرة بالفيزياء (PINNs) هي نموذج جديد للتعلم العميق يتفوق في حل كل من مشاكل PDE غير الخطية العكسية والأمامية. يمكن تدريب الشبكات العصبية المدعومة بالمعلومات الفيزيائية (PINNs) كنماذج بديلة لتقريب الحلول لمعادلات التفاضل الجزئي (PDEs) بدون بيانات تسميات من خلال تضمين المعلومات الفيزيائية المحددة بواسطة PDEs في الشبكات العصبية التلافيفية. بفضل القدرة الاستثنائية للشبكات العصبية على تصوير التفاعلات المعقدة، تم إنشاء العديد من التقنيات المستندة إلى PINN لمعالجة مجموعة من القضايا، بما في ذلك المعادلات التفاضلية الجزئية الكسرية، والمعادلات التفاضلية الجزئية العشوائية، والمعادلات التكاملية التفاضلية.

يتم استخدام PINN في هذه الأطروحة لحل معادلات فولتيرا التكاملية التفاضلية (VIDEs) من الرتبة الثانية. لذا، بعد تحويلها إلى معادلة تفاضلية (DEs)، يتم الحصول على سلسلة من الحلول المغلقة التقريبية المتكررة باستخدام مربع كاوس ليجندر العام. علاوة على ذلك، تم تقديم ثلاثة أمثلة وحلها لإظهار موثوقية وفعالية وملاءمة النهج المقترح. تم العثور على أن دوال التفعيل وطريقة التحسين L-BFGS و L-BFGS-B هي الأكثر ملاءمة لحل معادلات التفاضل الجزئي من الدرجة الثانية باستخدام PINN. لقد وجدنا مربع الخطأ النسبي في جميع الحالات يتراوح بين 0.1e-5 و0.1e-7 وهذا نتيجة ممتازة، وكذلك كانت مدة التدريب لجميع الحالات بين 160-280 ثانية. تظهر النتائج مزايا التقارب الأسرع والدقة المحسنة.

تقدم هذه الأطروحة أيضًا شبكة عصبية جديدة تم تنفيذها باستخدام بايثون ومكتبة IDRLnet. تم تصميم شبكة عصبية جديدة لحزمة التعلم العميق مفتوحة المصدر لحل المعادلات التفاضلية الجزئية باستخدام PINNs وتستخدم بشكل رئيسي في التعلم العميق المستند إلى الفيزياء. تستخدم هذه الأطروحة شبكة PINN لحل معادلة تكاملية تفاضلية من الرتبة الثانية (VIDE)، حيث يتم إنشاء سلسلة من الحلول التكرارية بناءً على طريقة جاوس- ليجندر التربيعية بعد تحويل معادلة VIDE من الرتبة الثانية إلى معادلة تفاضلية. بالإضافة إلى ذلك، تم تقديم ثلاثة أمثلة مختلفة وحلها

باستخدام الخوارزمية الجديدة لتسليط الضوء على دقة وكفاءة الطريقة المقترحة. تبين أن الخطأ النسبي L2 في جميع الحالات يتراوح بين0.1e−3 و0.1e−5، وهو نتيجة جيدة، ووقت "التدريب" لجميع الحالات هو 414.08، 724.2، 1339.31، و2260.8 ثانية. أخيرًا، تم الاستنتاج أن الدوال الفاصلة (مجال الدالة) في منطقة الحل تتناسب عكسيًا مع دقة الحل. بالإضافة إلى ذلك، فإن عدد التكرارات في تدريب الشبكة يؤثر بشكل مباشر على دقة الحل. أيضًا، كلما كان درجة طريقة غاوس–ليجندر أعلى، كانت دقة الحل لمعادلة فولتيرا التكاملية أفضل.

بعد تقييم دقيق، وجدنا أن مكتبة Deepxde تتفوق على مكتبة IDRLnet في العديد من الجوانب: توفر مكتبة Deepxde أخطاء مطلقة بدقة تصل إلى 5 إلى 8 منازل عشرية، بينما تنتج مكتبة IDRLnet أخطاء تصل إلى 3 إلى 5 منازل عشرية. وبالتالي، فإن مكتبة Deepxde تُظهر مستويات أعلى من الدقة. الخطأ النسبي L2 في مكتبة Deepxde ينتج دقة تتراوح بين 5 إلى 7 منازل عشرية، ولكن مكتبة IDRLnet توفر دقة تتراوح بين 3 إلى 5 منازل عشرية فقط. وقت تنفيذ برنامج بايثون في مكتبة Deepxde يتراوح بين 160 إلى 280 ثانية، ولكن في مكتبة IDRLnet، يتراوح بين 414.08، 724.2، 1339.31، و2260.8 ثانية.

أخيرًا، نلاحظ أننا استخدمنا برنامج بايثون (3.9.13) مع العديد من المكتبات مثل: Numpy (2.2.0), Deepxde (1.12.2), IDRLnet (2.0.0), PyYAML (6.0.2), paddlepaddle (2.6.2), matplotlib (3.10.0), Torch (2.5.1), mpmath (1.4.0), packaging (2.2.4), pandas (24.2), والعديد من المكتبات الأخرى.