

PHƯƠNG PHÁP GIẢI SỐ PHƯƠNG TRÌNH VI PHÂN TUYẾN TÍNH BẬC CAO BẰNG MẠNG NƠON

METHODS OF SOLVING HIGHER ORDER LINEAR DIFFERENTIAL EQUATIONS BY NEURAL NETWORKS

Phạm Quý Mười*, Lê Hoàng Nhân, Đỗ Trường Trung

Trường Đại học Sư phạm – Đại học Đà Nẵng, Đà Nẵng, Việt Nam¹

*Tác giả liên hệ / Corresponding author: pqmuoi@ued.udn.vn

(Nhận bài / Received: 24/7/2023; Sửa bài / Revised: 15/8/2023; Chấp nhận đăng / Accepted: 16/8/2023)

Tóm tắt - Bài báo này trình bày hai phương pháp tìm nghiệm xấp xỉ cho bài toán Cauchy trong phương trình vi phân tuyến tính bậc n bằng mạng nơon. Phương pháp thứ nhất là thiết kế mạng nơon sinh ra hàm một biến phụ thuộc vào các tham số của mạng và đề xuất hàm chi phí mà cực tiểu của hàm này ứng mạng nơon xấp xỉ nghiệm của bài toán Cauchy. Phương pháp thứ hai là biến đổi phương trình vi phân tuyến tính bậc n về hệ phương trình vi phân tuyến tính với n ẩn hàm và thiết kế mạng nơon sinh ra hàm vector mà mỗi thành phần ứng với một ẩn hàm cần tìm. Sau đó, đề xuất hàm chi phí để xác định bộ tham số của mạng nơon ứng với hàm vector xấp xỉ nghiệm của hệ. Từ đó nhận được nghiệm xấp xỉ của bài toán Cauchy. Nhóm tác giả áp dụng hai phương pháp vào việc tìm nghiệm số của một số ví dụ cụ thể. Cả hai phương pháp đều hoạt động tốt, có độ chính xác cao.

Từ khóa - Phương trình vi phân tuyến tính bậc cao; bài toán Cauchy; hệ phương trình vi phân tuyến tính bậc nhất; mạng nơon; phương pháp giải phương trình vi phân bằng mạng nơon.

1. Đặt vấn đề

Phương trình vi phân tuyến tính bậc cao có nhiều ứng dụng quan trọng trong nhiều lĩnh vực khoa học và kỹ thuật. Trong vật lý, phương trình vi phân tuyến tính bậc cao thường được sử dụng để mô hình hóa và giải quyết các vấn đề trong cơ học lượng tử, điện tử và cơ học lưu chất. Trong kỹ thuật, phương trình vi phân tuyến tính bậc cao được sử dụng để mô tả và dự đoán các quá trình trong hệ thống điều khiển, xử lý tín hiệu, xử lý ảnh và nhiều lĩnh vực khác. Trong khoa học tự nhiên, các phương trình vi phân tuyến tính bậc cao cũng được sử dụng để mô tả và nghiên cứu các hiện tượng trong hóa học, sinh học, địa chất và nhiều lĩnh vực khoa học tự nhiên khác [1].

Giải các phương trình vi phân tuyến tính bậc cao thường rất phức tạp và trong nhiều trường hợp, người ta không thể giải chính xác bằng các phương pháp giải tích. Vì vậy, người ta cần sử dụng các phương pháp số để tìm nghiệm gần đúng của các phương trình này. Một số phương pháp số phổ biến để giải phương trình vi phân tuyến tính bậc cao bao gồm phương pháp hạ bậc, phương pháp đa bậc, phương pháp phổ. Ý tưởng của phương pháp hạ bậc là chuyển đổi phương trình vi phân tuyến tính bậc cao thành một hệ gồm các phương trình vi phân bậc nhất. Sau đó, hệ này có thể được giải bằng phương pháp Euler hoặc các phương pháp khác cho phương trình bậc nhất [1]. Ý tưởng của phương pháp đa bậc là xấp

Abstract - This article presents two methods to find approximate solutions for the Cauchy problem in n th order linear differential equations by neural networks (NN). The first is designing NN that generates a function of one variable depending on the parameters of the network and proposing a cost function which the minimum of this function corresponds to the NN that approximates the solution of the Cauchy problem. The second is transforming a n th order linear differential equation into a system of linear differential equations with n hidden functions and designing a NN that generates a vector function where each component corresponds to a hidden function to be found. Then, proposing a cost function to determine the set of parameters of the NN corresponding to the vector function approximating the solution of the system and an approximate solution of the Cauchy problem is obtained. The authors apply both methods to find the numerical solutions of some specific examples. Both methods work well, with high accuracy.

Key words - Higher order linear equation; Cauchy problem; systems of first order linear equations; neural networks; methods of solving differential equations by neural networks.

xi nghiệm của phương trình vi phân tuyến tính bậc cao bằng cách sử dụng một số hàm đa bậc. Các hàm này thường được xây dựng từ các hàm cơ bản như hàm lượng giác, hàm mũ, và hàm Bessel. Phương pháp đa bậc được sử dụng rộng rãi trong các ứng dụng về vật lý và kỹ thuật [2, 3]. Ý tưởng của phương pháp là xấp xỉ nghiệm của phương trình vi phân bằng cách sử dụng một hàm xấp xỉ dạng tổ hợp tuyến tính của các hàm cơ sở đặc biệt được gọi là hàm phổ. Các hàm phổ thường được chọn sao cho thỏa mãn điều kiện biên của phương trình. Phương pháp phổ có độ chính xác cao và thích hợp cho các bài toán có dạng đặc biệt [4].

Trong năm năm gần đây, việc giải số phương trình vi phân bằng mạng nơon (neural networks) là một phương pháp mới, rất tiềm năng và được quan tâm bởi nhiều nhà khoa học và ứng dụng khác nhau [5, 6, 7]. Một mạng nơon có thể xem như một hàm số (một biến hoặc nhiều biến, hàm vô hướng hoặc hàm vector tùy thuộc vào kiến trúc của mạng) phụ thuộc tham số. Ý tưởng chính của việc giải phương trình vi phân bằng mạng nơon là đi tìm một mạng nơon sao cho hàm số sinh ra bởi mạng nơon này xấp xỉ nghiệm của phương trình vi phân cần tìm. Thông thường, để xác định tham số trong mạng nơon, thường chọn một hàm chi phí sao cho cực tiểu của hàm chi phí đã chọn là bộ tham số xác định mạng nơon cần tìm. Tùy thuộc vào các bài toán khác nhau, đề xuất các hàm chi phí phù hợp. Sử dụng mạng nơon để

¹ The University of Danang – University of Science and Education, Danang, Vietnam (Pham Quy Muoi, Le Hoang Nhan, Do Truong Trung)

xấp xỉ nghiệm của phương trình vi phân cho phép ta áp dụng sức mạnh tính toán của mạng nơron để tìm nghiệm gần đúng mà không cần phải dựa vào các phép tính phức tạp. Ưu điểm của phương pháp này so với các phương pháp số truyền thống là nó đạt được độ chính xác cao trong việc ước lượng giá trị của hàm số và đạo hàm của hàm số đó tại các điểm không chỉ trong vùng biên mà còn trên toàn miền của bài toán. Hơn nữa, phương pháp này còn giúp chúng ta giảm thiểu được dữ liệu. Bởi lẽ, phương pháp này không cần dữ liệu đầy đủ trong toàn miền của bài toán mà chỉ cần một số lượng nhỏ điểm dữ liệu (dữ liệu điều kiện ban đầu trong bài toán Cauchy) đủ để xấp xỉ hàm số và đạo hàm tại các điểm còn lại trong bài toán.

Trong bài báo này, nhóm tác giả trình bày phương pháp tìm nghiệm số cho bài toán Cauchy trong phương trình vi phân tuyến tính bậc cao bằng mạng nơron. Trước hết, sử dụng ý tưởng của ông M. Raissi và các cộng sự [5] vào giải bài toán được nghiên cứu. Sau đó, kết hợp ý tưởng biến đổi bài toán Cauchy trong phương trình vi phân bậc cao về bài toán Cauchy cho hệ phương trình vi phân tuyến tính bậc nhất và áp dụng ý tưởng của ông M. Raissi và các cộng sự [5] để giải số hệ phương trình này. Cuối cùng sẽ so sánh, phân tích hai phương pháp này thông qua một số ví dụ cụ thể.

2. Phương trình vi phân tuyến tính bậc cao và bài toán Cauchy

Định nghĩa 2.1. Phương trình vi phân tuyến tính bậc n là phương trình có dạng

$$L[y](t) = g(t), \quad (1)$$

Trong đó

$$L[y](t) := \frac{d^n y}{dt^n} + p_1(t) \frac{d^{n-1} y}{dt^{n-1}} + \dots + p_{n-1}(t) \frac{dy}{dt} + p_n(t)y,$$

và $p_i(t)$, ($i = 1, \dots, n$) và $g(t)$ là các hàm liên tục theo biến t và không phụ thuộc vào y .

Định nghĩa 2.2. Bài toán Cauchy cho phương trình vi phân tuyến tính cấp n là bài toán tìm hàm $y(t)$ thỏa mãn phương trình vi phân tuyến tính bậc n :

$$L[y](t) = g(t), \forall t \in I$$

và các điều kiện sau:

$$y(t_0) = y_0, y'(t_0) = y'_0, \dots, y^{(n-1)}(t_0) = y_0^{(n-1)}, \quad (2)$$

trong đó t_0 là điểm bất kỳ trong khoảng I và $y_0, y'_0, \dots, y_0^{(n-1)}$ là các số thực cho trước.

Định lý 2.3.([1]) Nếu các hàm p_1, p_2, \dots, p_n và g là các hàm liên tục trên khoảng mở I , thì tồn tại chính xác một nghiệm $y = \phi(t)$ của phương trình (1) thỏa mãn các điều kiện tại (2).

Việc tìm nghiệm chính xác của phương trình vi phân (1) cũng như Bài toán Cauchy (1)-(2) nhìn chung là rất khó và chỉ có thể trong một số trường hợp đặc biệt. Trong trường hợp phương trình vi phân bậc cao (1) với các hệ số hằng số, chúng ta có thể giải thông qua phương trình đặc trưng như các ví dụ sau. Chúng ta sẽ sử dụng các ví dụ dưới đây để minh họa hai phương pháp số được nghiên cứu trong bài báo này.

Ví dụ 1.1. Giải phương trình:

$$y'''' + y''' - 7y'' - y' + 6y = 0, \quad (i)$$

với điều kiện ban đầu:

$$y(0) = 1, y'(0) = 0, y''(0) = -2, y'''(0) = -1.$$

Giải:

Giả sử rằng $y = e^{rt}$, khi đó phương trình (i) trở thành

$$e^{rt}(r^4 + r^3 - 7r^2 - r + 6) = 0.$$

Vì $e^{rt} > 0, \forall r$ nên ta cần xác định r sao cho

$$r^4 + r^3 - 7r^2 - r + 6 = 0. (*)$$

Phương trình (*) có các nghiệm gồm $r_1 = 1, r_2 = -1, r_3 = 2, r_4 = -3$. Vì vậy nghiệm tổng quát của phương trình (i) là $y = c_1 e^t + c_2 e^{-t} + c_3 e^{2t} + c_4 e^{-3t}$.

Để tìm nghiệm thỏa mãn các điều kiện ban đầu, ta cần xác định c_1, c_2, c_3, c_4 thỏa mãn hệ phương trình

$$\begin{cases} c_1 + c_2 + c_3 + c_4 = 1, \\ c_1 - c_2 + 2c_3 - 3c_4 = 0, \\ c_1 + c_2 + 4c_3 + 9c_4 = -2, \\ c_1 - c_2 + 8c_3 - 27c_4 = -1. \end{cases}$$

Giải hệ phương trình này ta tìm được

$$c_1 = \frac{11}{8}, c_2 = \frac{5}{12}, c_3 = -\frac{2}{3}, c_4 = -\frac{1}{8}.$$

Vậy nghiệm của phương trình là

$$y = \frac{11}{8} e^t + \frac{5}{12} e^{-t} - \frac{2}{3} e^{2t} - \frac{1}{8} e^{-3t}.$$

Ví dụ 1.2. Giải phương trình

$$y'''' - 3y''' + 3y'' - y = 4e^t. \quad (ii)$$

với điều kiện ban đầu:

$$y(1) = \frac{5}{3} e, y'(1) = \frac{14}{3} e, y''(1) = \frac{41}{3} e.$$

Giả sử $y = e^{rt}$, khi đó đa thức đặc trưng cho phương trình thuần nhất tương ứng với phương trình (ii) là

$$r^3 - 3r^2 + 3r - 1 = (r - 1)^3,$$

Vì vậy nghiệm tổng quát của phương trình thuần nhất là $y(t) = c_1 e^t + c_2 t e^t + c_3 t^2 e^t$.

Để tìm nghiệm riêng $Y(t)$ của phương trình (ii), ta bắt đầu bằng việc giả sử rằng $Y(t) = A t^3 e^t$, trong đó A là một hệ số chưa xác định. Ta lấy đạo hàm $Y(t)$ ba lần, thay y bởi Y trong phương trình (ii) ta được

$$6Ae^t = 4e^t.$$

Suy ra, $A = \frac{2}{3}$ và $Y(t) = \frac{2}{3} t^3 e^t$. Do đó, nghiệm tổng quát của phương trình (ii) là

$$y = c_1 e^t + c_2 t e^t + c_3 t^2 e^t + \frac{2}{3} t^3 e^t.$$

Để tìm nghiệm thỏa mãn các điều kiện ban đầu, ta cần xác định c_1, c_2, c_3, c_4 thỏa mãn các hệ phương trình

$$\begin{cases} ec_1 + ec_2 + ec_3 + \frac{2}{3} e = \frac{5}{3} e \\ ec_1 + 2c_2 e + 3c_3 e + \frac{8}{3} e = \frac{14}{3} e \\ ec_1 + 3c_2 e + 7c_3 e + \frac{26}{3} e = \frac{41}{3} e. \end{cases}$$

Giải hệ phương trình này ta tìm được

$$c_1 = 1; c_2 = -1; c_3 = 1.$$

Vậy nghiệm của phương trình là:

$$y = e^t - t e^t + t^2 e^t + \frac{2}{3} t^3 e^t.$$

3. Mạng nơron và một số khái niệm liên quan

Gọi $\sigma: \mathbb{R} \rightarrow \mathbb{R}$ là hàm kích hoạt của mạng nơron. Khi đó, ta định nghĩa $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^n$ được xác định như sau:

$$(\sigma(z))_i = \sigma(z_i), z \in \mathbb{R}^n.$$

Định nghĩa 3.1. Một mạng nơron (Neural Networks) gồm L lớp (01 lớp đầu vào, 01 lớp đầu ra và $L - 2$ lớp ẩn, trong đó lớp thứ l có n_l neuron ($l = 1, 2, \dots, L$), là một hàm số $F(\cdot, \theta): \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_L}$:

$$z^1 = t \in \mathbb{R}^{n_1},$$

$$z^l = \sigma(w^l z^{l-1} + b^l) \in \mathbb{R}^{n_l}, l = 2, 3, \dots, L - 1,$$

$$F(t, \theta) = w^L z^{L-1} + b^L \in \mathbb{R}^{n_L},$$

Trong đó

$$z^l \in \mathbb{R}^{n_l}, b^l \in \mathbb{R}^{n_l}, w^l \in \mathbb{R}^{n_l \times n_{l-1}}, \text{ và}$$

$$\theta = (w^2, w^3, \dots, w^L, b^2, b^3, \dots, b^L).$$

Như vậy, một mạng nơron có thể xem là một hàm số nhiều biến $F(\cdot, \theta)$ phụ thuộc vào tham số θ .

4. Phương pháp thứ nhất giải số phương trình vi phân tuyến tính bậc cao bằng mạng nơron

Ý tưởng của phương pháp này như sau: chúng ta tìm tham số θ mạng nơron sao cho hàm số một biến $F(t, \theta)$ sinh bởi mạng nơron đó thỏa mãn $F(t, \theta) \approx y(t)$, trong đó $y(t)$ là nghiệm của bài toán Cauchy (1)-(2).

Để thực hiện được điều này, thực hiện các bước sau:

1. *Thiết kế mạng nơron: Thiết kế một mạng nơron trong đó lớp đầu vào với 1 nơron, 2 lớp ẩn với 100 nơron mỗi lớp và lớp đầu ra với 1 nơron. Khi đó, mạng nơron xác định một hàm số một biến số $F(t, \theta)$ phụ thuộc vào tham số θ của mạng.*

2. *Rời rạc bài toán: Chọn $T > t_0$, chia đoạn $[t_0, T]$ thành các điểm chia $t_0 < t_1 < t_2 < \dots < t_m = T$. Các điểm chia có thể được sinh ngẫu nhiên hoặc là các điểm chia đều.*

3. *Chọn hàm chi phí:*

$$\phi(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} |L[F(t_i, \theta)] - g(t_i)|^2 + \frac{1}{n} \sum_{i=0}^{n-1} |F^{(i)}(t_0, \theta) - y_0^{(i)}|^2,$$

trong đó $N_r = m + 1$ là số các điểm chia trong đoạn $[t_0, T]$.

4. *Chọn giải thuật tìm cực tiểu hàm chi phí: Tìm nghiệm xấp xỉ cho cực tiểu của hàm $\phi(\theta)$ bằng giải thuật L-BFGS-B [8]. Phương pháp này có tốc độ hội tụ nhanh và cho kết quả rất tốt, thường được dùng trong các công trình nghiên cứu giải phương trình vi phân đạo hàm riêng bằng mạng nơron [5, 6].*

5. Phương pháp thứ hai giải số phương trình vi phân tuyến tính bậc cao bằng mạng nơron

Ý tưởng của phương pháp thứ hai là chuyển bài toán Cauchy (1)-(2) cho phương trình vi phân bậc cao về bài toán Cauchy cho hệ phương trình vi phân tuyến tính bậc nhất. Sau đó, dùng mạng nơron để tìm nghiệm xấp xỉ cho bài toán Cauchy này. Để chuyển đổi phương trình vi phân (1) sang hệ phương trình vi phân tuyến tính, ta định nghĩa các hàm x_1, x_2, \dots, x_n được xác định bởi

$$x_1 = y, x_2 = y', \dots, x_n = y^{(n-1)}.$$

Khi đó, ta có

$$x'_1 = x_2, x'_2 = x_3, \dots, x'_{n-1} = x_n. \quad (3)$$

Do đó, phương trình (1) tương đương với hệ phương trình

$$\begin{cases} x'_1 = x_2 \\ x'_2 = x_3 \\ \vdots \\ x'_{n-1} = x_n \\ x'_n = -p_1(t)x_n - p_2(t)x_{n-1} - \dots - p_n(t)x_1. \end{cases} \quad (4)$$

hay

$$Y' = AY, \quad (5)$$

trong đó

$$Y(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, Y' = \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix},$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -p_n(t) & -p_{n-1}(t) & -p_{n-2}(t) & \dots & -p_1(t) \end{bmatrix}.$$

Điều kiện (2) tương đương với

$$Y(t_0) = Y_0 := \begin{bmatrix} y_0 \\ y'_0 \\ \vdots \\ y_0^{(n-1)} \end{bmatrix}. \quad (6)$$

Như vậy, bài toán Cauchy (1)-(2) của phương trình vi phân cấp cao tương đương với bài toán Cauchy (5)-(6) của hệ phương trình vi phân tuyến tính cấp một.

Để tìm nghiệm xấp xỉ của bài toán Cauchy (1)-(2), chúng ta đi tìm nghiệm xấp xỉ của bài toán Cauchy (5)-(6). Dùng phương pháp mạng nơron, chúng ta tìm một hàm vectơ phụ thuộc tham số

$$F(t, \theta) = (F_1(t, \theta), F_2(t, \theta), \dots, F_n(t, \theta))$$

sao cho $F_1(t, \theta) \approx x_1, F_2(t, \theta) \approx x_2, \dots, F_n(t, \theta) \approx x_n$.

Để thực hiện được điều này, thực hiện các bước sau:

1. *Thiết kế mạng nơron: Thiết kế một mạng nơron với lớp đầu vào có 1 nơron, 2 lớp ẩn với 100 nơron cho mỗi lớp và lớp đầu ra với n nơron. Khi đó, mạng nơron xác định một hàm vectơ $F(t, \theta) = (F_1(t, \theta), F_2(t, \theta), \dots, F_n(t, \theta))$ phụ thuộc vào tham số θ của mạng.*

2. *Rời rạc bài toán (5)-(6): Chọn $T > t_0$, chia đoạn $[t_0, T]$ thành các điểm chia $t_0 < t_1 < t_2 < \dots < t_m = T$.*

3. *Chọn hàm chi phí:*

$$\phi(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \|F'(t_i, \theta) - AF(t_i, \theta)\|^2 + \frac{1}{n} \|F(t_0, \theta) - Y_0\|^2,$$

trong đó $N_r = m + 1$ là số các điểm chia trong đoạn $[t_0, T]$.

4. *Chọn giải thuật tìm cực tiểu hàm chi phí: Tìm nghiệm xấp xỉ cho cực tiểu của hàm $\phi(\theta)$ bằng giải thuật L-BFGS-B.*

Về cơ bản, các bước trong phương pháp thứ hai giống như các bước trong phương pháp thứ nhất. Sự khác biệt cơ bản ở hai phương pháp là ở Bước 1 (thiết kế kiến trúc mạng) và Bước 3 (xác định hàm chi phí). Trong phương pháp thứ nhất, kiến trúc mạng sinh ra hàm một biến phụ thuộc tham số, trong khi trong phương pháp thứ hai, kiến

trúc mạng sinh ra hàm vector một biến phụ thuộc tham số. Trong phương pháp thứ nhất, để tính được hàm chi phí chúng ta cần tính các đạo hàm bậc cao, ngược lại trong phương pháp thứ hai, chúng ta chỉ cần tính các đạo hàm bậc nhất. Chúng ta sẽ xem xét ưu nhược điểm của mỗi phương pháp thông qua các ví dụ số cụ thể ở phần tiếp theo.

6. Một số ví dụ áp dụng

Trong phần này, áp dụng hai phương pháp đã được đề xuất để giải các ví dụ được nêu ở mục 2. Ở mỗi ví dụ, sẽ minh họa nghiệm chính xác và hai nghiệm xấp xỉ nhận được từ hai phương pháp trình bày ở mục 4 và 5. Nhóm tác giả sẽ so sánh sai số giữa nghiệm chính xác và nghiệm xấp xỉ nhận được từ hai phương pháp theo số điểm chia rời rạc trong mỗi bài toán. Từ đó đưa ra nhận định về độ chính xác và tốc độ hội tụ của hai phương pháp.

Ví dụ 6.1. Giải số phương trình: $y'''' + y''' - 7y'' - y' + 6y = 0$, với điều kiện ban đầu

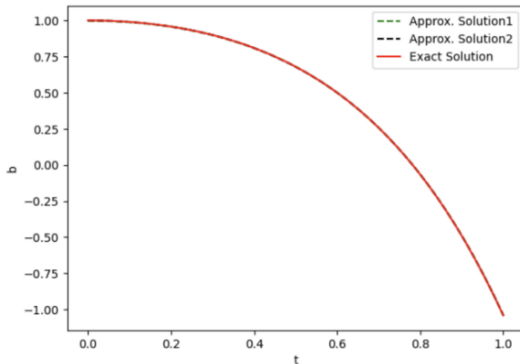
$$y(0) = 1, y'(0) = 0, y''(0) = -2, y'''(0) = -1.$$

Từ Ví dụ 1.1, nghiệm chính xác của bài toán này là

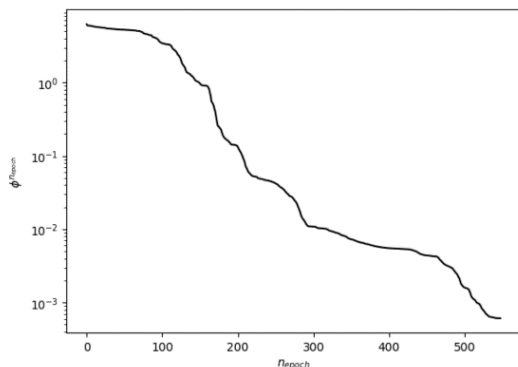
$$y^* = \frac{11}{8}e^t + \frac{5}{12}e^{-t} - \frac{2}{3}e^{2t} - \frac{1}{8}e^{-3t}.$$

Để giải số ví dụ này, cả hai phương pháp được đề xuất nghiệm số trên đoạn $[0,1]$ được chia thành các đoạn con đều nhau bởi $N_r = 5000$ điểm.

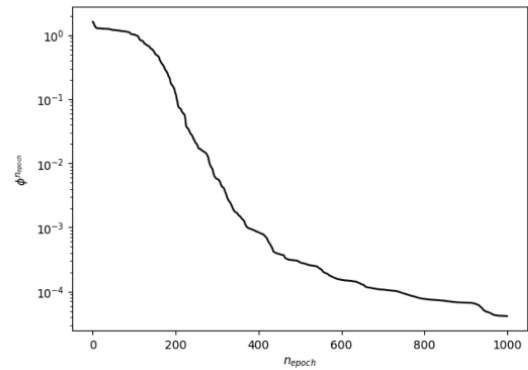
Nghiệm chính xác và nghiệm số nhận được từ hai phương pháp được minh họa ở Hình 1. Chúng ta thấy cả ba nghiệm hoàn toàn trùng khít lên nhau.



Hình 1. Đồ thị của hai nghiệm xấp xỉ theo phương pháp thứ nhất (Approx. Solution1), theo phương pháp thứ hai (Approx. Solution2) và nghiệm chính xác của Ví dụ 6.1



Hình 2. Đồ thị của hàm chi phí $\phi(\theta_n)$ trong phương pháp thứ nhất trong Ví dụ 6.1



Hình 3. Đồ thị của hàm chi phí $\phi(\theta_n)$ trong phương pháp thứ hai trong Ví dụ 6.1

Giá trị của hàm chi phí trong phương pháp thứ nhất và thứ hai lần lượt được minh họa ở Hình 2 và Hình 3. Chúng ta thấy, cả hai hàm chi phí đều giảm nhanh theo vòng lặp của giải thuật L-BFGS-B. Tuy nhiên, với cùng một quy tắc dừng giải thuật, giá trị hàm chi phí trong phương pháp thứ hai nhỏ hơn nhiều so với giá trị của hàm chi phí trong phương pháp thứ nhất. Số vòng lặp của giải thuật trong phương pháp thứ nhất ít hơn số vòng lặp của giải thuật trong phương pháp thứ hai.

Để đánh giá sai số giữa nghiệm chính xác và nghiệm xấp xỉ chúng ta tính trung bình bình phương sai số:

$$E_i = \frac{1}{N_r} \sum_{j=0}^{N_r} |y_i(t_j) - y^*(t_j)|^2,$$

Trong đó, y^* là nghiệm chính xác và y_i là nghiệm xấp xỉ nhận được bởi phương pháp thứ nhất ($i = 1$) và phương pháp thứ hai ($i = 2$).

Sai số giữa nghiệm chính xác và nghiệm số trong mỗi phương pháp theo số điểm chia đoạn $[0,1]$ được cho ở Bảng 1; sai số giữa nghiệm chính xác và nghiệm xấp xỉ trong cả hai phương pháp khá bé. Khi số điểm chia tăng thì sai số giảm. Phương pháp thứ hai cho sai số bé hơn.

Bảng 1. Sai số giữa nghiệm chính xác và nghiệm xấp xỉ theo số điểm chia của đoạn $[0, 1]$

N_r	E_1	E_2
500	1,09e-06	8,24e-08
1000	4,07e-07	6,60e-08
2000	6,09e-08	4,12e-08
5000	4,33e-08	7,51e-09

Ví dụ 6.2. Giải số phương trình:

$$y'''' - 3y'' + 3y' - y = 4e^t$$

với điều kiện ban đầu

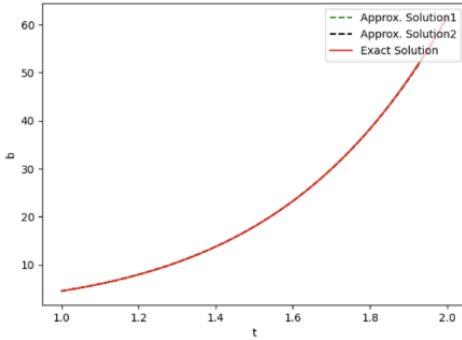
$$y(1) = \frac{5}{3}e, y'(1) = \frac{14}{3}e, y''(1) = \frac{41}{3}e.$$

Nghiệm chính xác của bài toán này là

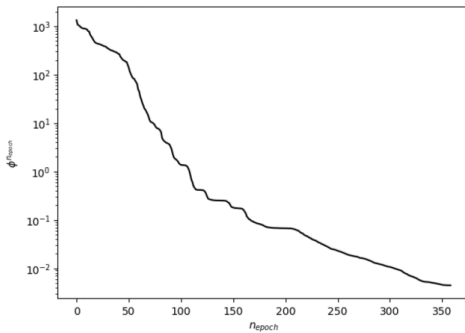
$$y^* = e^t - te^t + t^2e^t + \frac{2}{3}t^3e^t.$$

Để giải số ví dụ này, cả hai phương pháp được đề xuất nghiệm số trên đoạn $[1,2]$ được chia thành các đoạn con đều nhau bởi $N_r = 2000$ điểm.

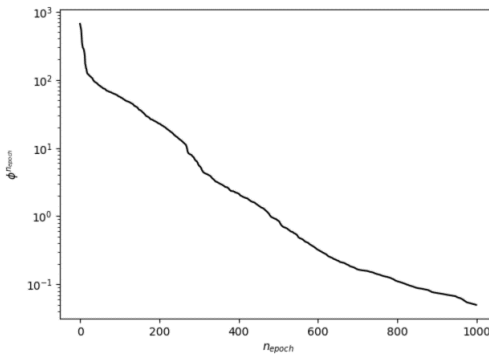
Nghiệm chính xác và nghiệm số nhận được từ hai phương pháp được minh họa ở Hình 4. Tương tự như ở Ví dụ 6.1, thấy rằng cả ba nghiệm hoàn toàn trùng khít lên nhau.



Hình 4. Đồ thị của hai nghiệm xấp xỉ theo phương pháp thứ nhất (Approx. Solution1), theo phương pháp thứ hai (Approx. Solution2) và nghiệm chính xác của Ví dụ 6.2



Hình 5. Đồ thị của hàm chi phí $\phi(\theta_n)$ trong phương pháp thứ nhất trong Ví dụ 6.2



Hình 6. Đồ thị của hàm chi phí $\phi(\theta_n)$ trong phương pháp thứ hai trong Ví dụ 6.2

Giá trị của hàm chi phí trong phương pháp thứ nhất và thứ hai lần lượt được minh họa ở Hình 5 và Hình 6. Chúng ta thấy, cả hai hàm chi phí đều giảm nhanh theo vòng lặp của giải thuật L-BFGS-B. Khác với Ví dụ 6.1, với cùng một quy tắc dừng giải thuật, giá trị hàm chi phí trong phương pháp thứ nhất lại nhỏ hơn giá trị của hàm chi phí trong phương pháp thứ hai. Số vòng lặp của giải thuật trong phương pháp thứ nhất ít hơn số vòng lặp của giải thuật trong phương pháp thứ hai.

Sai số giữa nghiệm chính xác và nghiệm số trong mỗi phương pháp theo số điểm chia đoạn $[1,2]$ được cho ở Bảng 2. Trong cả hai phương pháp, sai số đều lớn hơn so với sai

số ở Ví dụ 6.1 và sai số trong phương pháp thứ nhất thì bé hơn sai số trong phương pháp thứ hai. Các sai số lớn hơn ở Ví dụ 6.1 có thể do: (1) Phương trình vi phân trong Ví dụ 6.2 là phương trình không thuần nhất, trong khi phương trình vi phân ở Ví dụ 6.1 là phương trình thuần nhất; (2) điều kiện ban đầu trong Ví dụ 6.2 được tính xấp xỉ khi rời rạc bài toán và hàm nguồn trong vế phải của phương trình vi phân cũng được tính xấp xỉ khi rời rạc. Tuy nhiên, chúng tôi không thể lí giải được vì sao phương pháp thứ nhất hoạt động tốt hơn phương pháp thứ 2 cho Ví dụ 6.2, nhưng lại kém hơn cho Ví dụ 6.1. Điều này cần phải tiếp tục nghiên cứu để làm rõ nguyên nhân.

Bảng 2. Sai số giữa nghiệm chính xác và nghiệm xấp xỉ theo số điểm chia của đoạn $[1,2]$

N_r	E_1	E_2
500	2,41e-05	5,27e-03
1000	2,25e-06	6,34e-04
2000	1,57e-06	3,01e-04
5000	8,77e-07	1,14e-04

7. Kết luận

Trong bài báo này, đã trình bày hai phương pháp để giải số phương trình vi phân tuyến tính bậc cao bằng mạng nơron. Cả hai phương pháp đều nhận được các nghiệm xấp xỉ tốt. Các ví dụ số đã cho thấy, sai số giữa nghiệm xấp xỉ và nghiệm chính xác khá bé và sẽ giảm đi khi số điểm chia tăng lên. Tùy theo từng ví dụ mà phương pháp này cho kết quả tốt hơn phương pháp kia. Tuy nhiên, việc xác định phương pháp nào là tốt hơn thì chưa có câu trả lời, cần phải tiếp tục nghiên cứu và thực nghiệm cho nhiều tình huống khác.

TÀI LIỆU THAM KHẢO

- [1] W. E. Boyce and R. C. DiPrima, *Elementary differential equations and boundary value problems*, Wiley, 2020.
- [2] D. Funaro, *Polynomial approximation of differential equations* m 8, Springer Science & Business Media, 2008.
- [3] N. Mai-Duy, "An effective spectral collocation method for the direct solution of high-order ODEs", *Communications in numerical methods in engineering*, vol. 22, no. 6, pp. 627-642, 2005.
- [4] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: fundamentals in single domains*, Springer Science & Business Media, 2007.
- [5] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational physics*, vol. 378, pp. 686-707, 2019.
- [6] M. Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations", *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 932-955, 2018.
- [7] J. Han, A. Jentzen, and W. E., "Solving high-dimensional partial differential equations using deep learning", *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505-8510, 2018.
- [8] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization", *ACM Transactions on mathematical software (TOMS)*, vol. 23, no. 4, pp. 550-560, 1997.