Hindawi
Advances in Astronomy
Volume 2021, Article ID 6662217, 18 pages
https://doi.org/10.1155/2021/6662217

Research Article

Conformable Fractional Models of the Stellar Helium Burning via Artificial Neural Networks

Emad A.-B. Abdel-Salam,[1] **Mohamed I. Nouh**,[2] **Yosry A. Azzam**,[2] **and M. S. Jazmati**[3]

[1]*Department of Mathematics, Faculty of Science, New Valley University, El-Kharja 72511, Egypt*
[2]*Astronomy Department, National Research Institute of Astronomy and Geophysics (NRIAG), 11421 Helwan, Cairo, Egypt*
[3]*Department of Mathematics, College of Science, Qassim University, Buraydah 51452, P. O. Box 6644, Saudi Arabia*

Correspondence should be addressed to Yosry A. Azzam; y.azzam@nriag.sci.eg

Received 15 December 2020; Revised 20 January 2021; Accepted 11 February 2021; Published 16 March 2021

Academic Editor: Kwing Lam Chan

The helium burning phase represents the second stage that the star used to consume nuclear fuel in its interior. In this stage, the three elements, carbon, oxygen, and neon, are synthesized. The present paper is twofold: firstly, it develops an analytical solution to the system of the conformable fractional differential equations of the helium burning network, where we used, for this purpose, the series expansion method and obtained recurrence relations for the product abundances, that is, helium, carbon, oxygen, and neon. Using four different initial abundances, we calculated 44 gas models covering the range of the fractional parameter $\alpha = 0.5 - 1$ with step $\Delta\alpha = 0.05$. We found that the effects of the fractional parameter on the product abundances are small which coincides with the results obtained by a previous study. Secondly, we introduced the mathematical model of the neural network (NN) and developed a neural network algorithm to simulate the helium burning network using a feed-forward process. A comparison between the NN and the analytical models revealed very good agreement for all gas models. We found that NN could be considered as a powerful tool to solve and model nuclear burning networks and could be applied to the other nuclear stellar burning networks.

1. Introduction

Nowadays, applications of fractional calculus in physics, astrophysics, and related science are widely used [1, 2]. Examples of the recent applications of the fractional calculus in physics are found in [3] in which the author has introduced a generalized fractional scale factor and a time-dependent Hubble parameter obeying an "Ornstein–Uhlenbeck-like fractional differential equation" which serves to describe the accelerated expansion of a nonsingular universe, in [4] in which the author extended the idea of fractional spin based on two-order fractional derivative operator and in [5] in which the author has generalized the fractional action integral by using the Saigo–Maeda fractional operators defined in terms of the Appell hypergeometric function.

In astrophysics, many problems have been handled using fractional models. Examples of these studies are in [6], where the author introduced an analytical solution to the fractional white dwarf equation, in [7] in which the authors analyzed the fractional incompressible gas spheres and in [8, 9] in which the authors introduced an analytical solution to the first and second types of Lane–Emden equation in the sense of modified Riemann–Liouville fractional derivative. Nouh in [10] solved the fractional helium burning network using a series expansion method. Abdel-Salam and Nouh [11] and Yousif et al. [12] introduced analytical solutions to the conformable polytropic and isothermal gas spheres.

Simulation of ordinary (ODE) and partial differential equations (PDE) using an artificial neural network (ANN) gives very good accuracy when compared with both the numerical and analytical methods. Many authors dealt with this issue and developed many neural algorithms to solve ODE and PDE. Dissanayake and Phan-Thien [13] first introduced the concept of approximating the solutions of differential equations with neural networks, where training was carried out by minimizing losses based on the satisfaction of the network with the boundary conditions and the differential equations themselves. Lagaris et al. [14] demonstrated that the network shape could be selected by construction to satisfy boundary conditions and that automatic differentiation could be used to determine the derivatives that appear in the loss function. This approach has been extended to irregular boundary systems [15, 16], applied to the resolution of PDEs occurring in fluid mechanics [17], and software packages have been developed to facilitate their applications [18–20]. Nouh et al. [21] and Azzam et al. [22] developed a neural network algorithm to solve the first and second types of Lane–Emden equations arising in astrophysics.

The helium burning stage (also known as the triple-alpha process) represents the second stage where the stars undergo the transfer of nuclear energy from the interior to their surface. In this stage, nuclear energy is almost converted to light when passing through the stellar atmosphere. Helium burning (HB) releases energy per unit fuel of about $6\times10^{23}\,\mathrm{MeV/g}\approx10^{18}\,\mathrm{erg/g}$. The reaction equations that govern the HB network may be written as follows [10]:

$$\begin{aligned}3\mathrm{He}^4 &\longrightarrow \mathrm{C}^{12}+\gamma+7.281\,\mathrm{Mev},\\ \mathrm{C}^{12}+\mathrm{He}^4 &\longrightarrow \mathrm{O}^{16}+\gamma+7.150\,\mathrm{Mev},\\ \mathrm{O}^{16}+\mathrm{He}^4 &\longrightarrow \mathrm{Ne}^{20}+\gamma+4.750\,\mathrm{Mev},\end{aligned}\tag{1}$$

where the conversion process from helium to carbon needs 10^8 K.

Clayton [23] set up a model for the helium burning process by taking into account the above reactions. If the number of atoms per unit of stellar material mass for helium, carbon, oxygen, and neon is represented by x, y, z, and r, respectively, then the next four equations (also maybe called the kinetic equations) control the time-dependent change in abundance:

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t}&=-3ax^3-bxy-cxz,\\ \frac{\mathrm{d}y}{\mathrm{d}t}&=ax^3-bxy,\\ \frac{\mathrm{d}z}{\mathrm{d}t}&=bxy-cxz,\\ \frac{\mathrm{d}r}{\mathrm{d}t}&=cxz.\end{aligned}\tag{2}$$

where a, b, and c are the reaction rates.

The system of equation (2) represents the integer version of the helium burning network and solved simultaneously by computational or analytical methods [23–26]. Appendix A includes clarification of the derivation of the set of equation (2). The fractional kinetic equation (like helium burning network) has been solved by many authors. In terms of H-functions, [27] presented a solution to the fractional generalized kinetic equation. The generalized fractional kinetic equations have been solved by [28]. Chaurasia and Pandey [29] solved the fractional kinetic equations in a series form of the Lorenzo–Hartley function.

In the present article, we developed a neural network algorithm to solve the fractional system of differential equations describing the helium burning network. We use the principles of the conformable fractional derivative for the mathematical modeling of the ANN. We used in this research an architecture of ANN which is the feed-forward network having three layers and trained using the algorithm of backpropagation (BP) based on the gradient descent delta rule.

The analytical solution is developed using the series expansion method and a comparison between the ANN and analytical models is performed to declare the efficiency and applicability of the ANN for solving the conformable helium burning network. The paper is organized as follows: Section 2 introduces the details of the analytical solution of the conformable helium burning model using the series expansion method. Section 3 deals with the mathematical modeling of the neural network technique with its gradient computations and back-propagation training algorithm. Section 4 discusses the results obtained and the comparison between the ANN and analytical models. Section 5 gives the details of the conclusion.

2. Analytical Solution to the Conformable Helium Burning Model

By being certainly valid, the techniques of numerical integration may provide very accurate models. However, it is surely worthwhile to obtain modeling with the desired precision if complete analytical formulas are created. Besides, these analytical formulas usually provide much more deep insight into the essence of a model than numerical integration. The power series solution, on the other hand, may serve as the analytical representation of the solution in the absence of a closed analysis solution for a particular differential equation.

The fractional form of equation (2) is given by [10]

$$\begin{aligned}D_t^{\alpha}x&=-3ax^3-bxy-cxz,\\ D_t^{\alpha}y&=ax^3-bxy,\\ D_t^{\alpha}z&=bxy-cxz,\\ D_t^{\alpha}r&=cxz.\end{aligned}\tag{3}$$

If $T=t^{\alpha}$, then x, y, z, r could be represented by

$$\begin{aligned} x &= \sum_{m=0}^{\infty} X_m T^m = X_0 + X_1 T + X_2 T^2 + X_3 T^3 + \cdots \\ &= X_0 + X_1 t^{\alpha} + X_2 t^{2\alpha} + X_3 t^{3\alpha} + \cdots, \\ y &= \sum_{m=0}^{\infty} Y_m T^m = Y_0 + Y_1 T + Y_2 T^2 + Y_3 T^3 + \ldots \\ &= Y_0 + Y_1 t^{\alpha} + Y_2 t^{2\alpha} + Y_3 t^{3\alpha} + \cdots, \\ z &= \sum_{m=0}^{\infty} Z_m T^m = Z_0 + Z_1 T + Z_2 T^2 + Z_3 T^3 + \cdots \\ &= Z_0 + Z_1 t^{\alpha} + Z_2 t^{2\alpha} + Z_3 t^{3\alpha} + \cdots, \\ r &= \sum_{m=0}^{\infty} R_m T^m = R_0 + R_1 T + R_2 T^2 + R_3 T^3 + \cdots \\ &= R_0 + R_1 t^{\alpha} + R_2 t^{2\alpha} + R_3 t^{3\alpha} + \cdots, \end{aligned} \tag{4}$$

where X_m, Y_m, Z_m, R_m are constants to be determined.

In equation (2), the left side of the system depicts the abundances of those elements in which the abundance of helium (x) is raised to power 3. To obtain the fractional derivative of u^n, we apply the fractional derivative of the product of the two functions. Using the series expansion method, we obtained the recurrence relation of the term x^3 by the following.

Let $D_x^{\alpha}u^n = D_x^{\alpha}G$, with $u = \sum_{m=0}^{\infty} A_m x^{\alpha m}, C = \sum_{m=0}^{\infty} Q_m x^{\alpha m}$.

That is,

$$\begin{gathered} nu^{n-1} D_x^{\alpha} u = D_x^{\alpha} G, \\ \text{or } nu^{n} D_x^{\alpha} u = u D_x^{\alpha} G. \end{gathered} \tag{5}$$

Performing the fractional derivative to equation (5) k times, we get

$$\begin{gathered} D^{\overbrace{\alpha,\ldots,\alpha}^{k\text{ times}}}\left[nGD_x^{\alpha}u\right] = D^{\overbrace{\alpha,\ldots,\alpha}^{k\text{ times}}}\left(uD_x^{\alpha}G\right), \\ \text{or } n\sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} u D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} G = \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} G D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} u, \end{gathered} \tag{6}$$

and putting $X = 0$, we get

$$\begin{aligned} & n\sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} u(0) D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} G(0) \\ & \quad = \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} G(0) D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} u(0). \end{aligned} \tag{7}$$

Since

$$\begin{aligned} D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} u(0) &= \alpha^{(j+1)} A_{j+1}, D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} G(0) \\ &= \alpha^{(k-j)} Q_{k-j}, D^{\overbrace{\alpha,\ldots,\alpha}^{j+1\text{ times}}} G(0) = \alpha^{(j+1)} Q_{j+1}, \\ D^{\overbrace{\alpha,\ldots,\alpha}^{k-j\text{ times}}} u(0) &= \alpha^{(k-j)} Q_{j+1}, \end{aligned} \tag{8}$$

we have

$$\begin{aligned} & n\sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} (j+1)!\alpha^{(j+1)} A_{j+1} (k-j)!\alpha^{(k-j)} Q_{k-j} \\ & = \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} (j+1)!\alpha^{(j+1)} Q_{j+1} (k-j)!\alpha^{(k-j)} A_{k-j}, \\ & \Rightarrow n\sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} (j+1)!(k-j)!\alpha^{(k+1)} A_{j+1} Q_{k-j} = \sum_{j=0}^{k}\begin{pmatrix} k \\ j \end{pmatrix} (j+1)!(k-j)!\alpha^{(k+1)} Q_{j+1} A_{k-j}. \end{aligned} \tag{9}$$

After some manipulations, we get

$$k!\,(k+1)A_0Q_{k+1} = n\sum_{j=0}^{k} k!\,(j+1)A_{j+1}Q_{k-j} - \sum_{j=0}^{k-1} k!\,(j+1)A_{k-j}Q_{j+1}, \tag{10}$$

and putting $i = j + 1$ and $i = k - j$ in equation (10), we have

$$k!\,(k+1)A_0Q_{k+1} = n\sum_{i=1}^{k+1} k!\,(i)A_iQ_{k+1-i} - \sum_{i=1}^{k} k!\,(k+1-i)A_iQ_{k+1-i}. \tag{11}$$

If $m = k + 1$, then

$$m!A_0Q_m = n\sum_{i=1}^{m} (m-1)!\,(i)A_iQ_{m-i} - \sum_{i=1}^{m-1} (m-1)!\,(m-i)A_iQ_{m-i}. \tag{12}$$

Adding the zero value $\{-(m-1)!\,(m-m)A_mQ_0\}$ to the second summation of the last equation, we get

$$\begin{aligned} m!A_0Q_m &= n\sum_{i=1}^{m} (m-1)!\,(i)A_iQ_{m-i} - \sum_{i=1}^{m-1} (m-1) \\ &\quad !\,(m-i)A_iQ_{m-i} - (m-1)!\,(m-m)A_mQ_0, \\ m!A_0Q_m &= n\sum_{i=1}^{m} (m-1)!\,(i)A_iQ_{m-i} - \sum_{i=1}^{m} (m-1)!\,(m-i)A_iQ_{m-i}. \end{aligned} \tag{13}$$

From the last equation, we can write the coefficients Q_m as

$$Q_m = \frac{1}{m!A_0}\sum_{i=1}^{m} (m-1)!\,(in - m + i)A_iQ_{m-i}, \quad \forall m \geq 1. \tag{14}$$

Putting $n = 3$ in equation (14), we have

$$Q_m = \frac{1}{m!X_0}\sum_{i=1}^{m} (m-1)!\,(4i - m)X_iQ_{m-i}, \quad \forall m \geq 1, \tag{15}$$

where

$$\begin{aligned} X_0 &= A_0, \\ X_i &= A_i, \\ Q_0 &= X_0^3, \\ Q_1 &= \frac{3X_1Q_0}{X_0}, \end{aligned} \tag{16}$$

taking fractional differentiation α-derivatives to equation (4), we get

$$\begin{aligned} D_t^{\alpha}x &= \sum_{n=1}^{\infty} \alpha nX_nT^{n-1}, \\ D_t^{\alpha}y &= \sum_{n=1}^{\infty} \alpha nY_nT^{n-1}, \\ D_t^{\alpha}z &= \sum_{n=1}^{\infty} \alpha nZ_nT^{n-1}, \\ D_t^{\alpha}r &= \sum_{n=1}^{\infty} \alpha nR_nT^{n-1}, \end{aligned} \tag{17}$$

and inserting equations (4) and (17) into equation (3), the series coefficients $X_{n+1}, Y_{n+1}, Z_{n+1}$, and R_{n+1} could be obtained from

$$\begin{aligned} X_{n+1} &= -\frac{1}{\alpha(n+1)}\left[3aQ_n + b\sum_{j=0}^{n} X_jY_{n-j} + c\sum_{j=0}^{n} X_jZ_{n-j}\right], \\ Y_{n+1} &= \frac{1}{\alpha(n+1)}\left[aQ_n - b\sum_{j=0}^{n} X_jY_{n-j}\right], \\ Z_{n+1} &= \frac{1}{\alpha(n+1)}\left[b\sum_{j=0}^{n} X_jY_{n-j} - c\sum_{j=0}^{n} X_jZ_{n-j}\right], \\ R_{n+1} &= \frac{c}{\alpha(n+1)}\sum_{j=0}^{n} X_jZ_{n-j}. \end{aligned} \tag{18}$$

The recurrence relations corresponding to the integer model could be obtained by putting $\alpha = 1$ in the last four formulas of equation (18) [26].

At $n=0$ and with the initial values of the chemical composition $X_0 = x_0, Y_0 = y_0, Z_0 = z_0, R_0 = r_0$, where x_0, y_0, z_0, r_0 are arbitrary constants, we get

$$
\begin{aligned}
X_1 &= -\frac{1}{\alpha}\left[3aQ_0 + bX_0Y_0 + cX_0Z_0\right] \\
&= -\frac{1}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right], \\
Y_1 &= \frac{1}{\alpha}\left[aQ_0 - bX_0Y_0\right] = \frac{1}{\alpha}\left[ax_0^3 - bx_0y_0\right], \\
Z_1 &= \frac{1}{\alpha}\left[bX_0Y_0 - cX_0Z_0\right] = \frac{1}{\alpha}\left[bx_0y_0 - cx_0z_0\right], \\
R_1 &= \frac{c}{\alpha}X_0Z_0 = \frac{c}{\alpha}x_0z_0, \\
Q_0 &= X_0^3 = x_0^3, Q_1 = \frac{3X_1Q_0}{X_0} = -\frac{3x_0^2}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right],
\end{aligned}
\tag{19}
$$

at $n = 1$, we get

$$
\begin{aligned}
X_2 &= -\frac{1}{2\alpha}\left[3aQ_1 + b\left(X_0Y_1 + X_1Y_0\right) + c\left(X_0Z_1 + X_1Z_0\right)\right] \\
&= -\frac{1}{2\alpha}\left[-\frac{9ax_0^2}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right] + b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right. \\
&\quad \left. + c\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right] \\
Y_2 &= \frac{1}{2\alpha}\left[aQ_1 - b\left(X_0Y_1 + X_1Y_0\right)\right] \\
&= \frac{1}{2\alpha}\left[-\frac{9ax_0^2}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right] + b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right], \\
Z_2 &= \frac{1}{2\alpha}\left[b\left(X_0Y_1 + X_1Y_0\right) - c\left(X_0Z_1 + X_1Z_0\right)\right] \\
&= \frac{1}{2\alpha}\left[b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right) - c\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right], \\
R_2 &= \frac{c}{2\alpha}\left(X_0Z_1 + X_1Z_0\right) = \frac{c}{2\alpha}\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right),
\end{aligned}
\tag{20}
$$

and by applying the same scheme, we can determine the rest of the series terms. So, the product abundance could be represented by the series solution of equation (3) as

$$
\begin{aligned}
x &= \sum_{m=0}^{\infty} X_m t^{\alpha m} = X_0 + X_1 t^{\alpha} + X_2 t^{2\alpha} + X_3 t^{3\alpha} + \ldots \\
&= x_0 - \frac{1}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]t^{\alpha} \\
&\quad - \frac{1}{2\alpha}\left[-\frac{9ax_0^2}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right. \\
&\quad \left. + b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right) + c\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right]t^{2\alpha} + \cdots, \\
y &= \sum_{m=0}^{\infty} Y_m t^{\alpha m} = Y_0 + Y_1 t^{\alpha} + Y_2 t^{2\alpha} + Y_3 t^{3\alpha} + \cdots \\
&= y_0 + \frac{1}{\alpha}\left[ax_0^3 - bx_0y_0\right]t^{\alpha} \\
&\quad + \frac{1}{2\alpha}\left[-\frac{9ax_0^2}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right] + b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right]t^{2\alpha} + \cdots, \\
z &= \sum_{m=0}^{\infty} Z_m t^{\alpha m} = Z_0 + Z_1 t^{\alpha} + Z_2 t^{2\alpha} + Z_3 t^{3\alpha} + \cdots \\
&= z_0 + \frac{1}{\alpha}\left[bx_0y_0 - cx_0z_0\right]t^{\alpha} \\
&\quad + \frac{1}{2\alpha}\left[b\left(\frac{x_0}{\alpha}\left[ax_0^3 - bx_0y_0\right] - \frac{y_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right) - c\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)\right]t^{2\alpha} + \cdots, \\
r &= \sum_{m=0}^{\infty} R_m t^{\alpha m} = R_0 + R_1 t^{\alpha} + R_2 t^{2\alpha} + R_3 t^{3\alpha} + \cdots \\
&= r_0 + \frac{c}{\alpha}x_0z_0t^{\alpha} + \frac{c}{2\alpha}\left(\frac{x_0}{\alpha}\left[bx_0y_0 - cx_0z_0\right] - \frac{z_0}{\alpha}\left[3ax_0^3 + bx_0y_0 + cx_0y_0\right]\right)t^{2\alpha} + \cdots.
\end{aligned}
\tag{21}
$$

It is important to mention that x_0, y_0, z_0, and r_0 are arbitrary initial values that enable us to compute gas models with different chemical compositions, that is, pure helium or rich helium models.

3. Neural Network Algorithm

3.1. Mathematical Modeling of the Problem. To simulate the conformable fractional helium burning network represented by equation (3), we use the neural network architecture shown in Figure 1.

Considering the initial conditions $X_0 = x_0, Y_0 = y_0, Z_0 = z_0, R_0 = r_0$, the neural network could be obtained following the next steps [30].

The form of the neural approximate solution of equation (3) will have two terms: the first represents the initial values and the second represents the feed-forward neural network, where x is the input vector and p is the corresponding vector of adjustable weight parameters. Then, the output of the neural network $N_l(x_l, p)$ is written as

$$
\begin{aligned}
X_t(x, p) &= A_1(x,) + f_1(x, N_1(x, p)), \\
Y_t(y, p) &= A_2(y) + f_2(y, N_2(y, p)), \\
Z_t(z, p) &= A_3(z) + f_3(z, N_3(z, p)), \\
R_t(r, p) &= A_4(r) + f_4(r, N_4(r, p)).
\end{aligned}
\tag{22}
$$

Then, neural network output $N_\ell(x_\ell, p)$ is given by

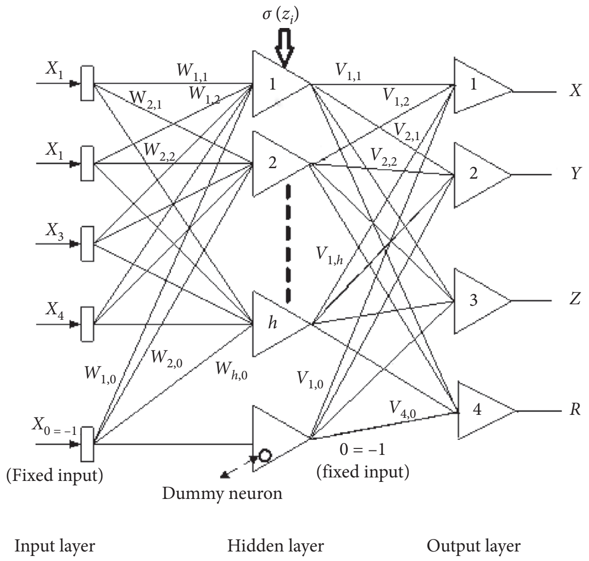


Figure 1: ANN architecture proposed to simulate conformable fractional helium burning network.

$$N_\ell(x_\ell, p) = \sum_{j=1}^{H} v_{ij}\sigma_i(z_j), \quad \ell = 1, 2, 3, 4, \tag{23}$$

where $z_j = \sum_{i=1}^{n} w_{ij}x_j + \beta_i$, w_{ij} is the weight from the input unit j to the hidden unit i, v_i is the weight from the hidden unit i to the output, β_i represents the bias of the i^{th} hidden unit, and $\sigma(z)$ is the sigmoid activation function that has the form $\sigma(x) = (1/(1 + e^{-x}))$, $\sigma(y) = (1/(1 + e^{-y}))$, $\sigma(z) = (1/(1 + e^{-z}))$, and $\sigma(r) = (1/(1 + e^{-r}))$. By differentiating the networks output N with respect to the vector x_j, we get

$$D_{x_j}^{\alpha} N(x, p) = D_{x_j}^{\alpha}\left(\sum_{i=1}^{H} v_i\sigma\left(z_i = \sum_{i=1}^{n} w_{ij}x_j + \beta_i\right)\right) = \sum_{i=1}^{h} v_i w_{ij}\sigma^{(\alpha)}, \quad \sigma^{(\alpha)} = D_x^{\alpha}\sigma(x). \tag{24}$$

Differentiating equation (24) n times gives

$$D_{x_j}^{\overset{n\,\text{times}}{\alpha,\ldots,\alpha}} N(x, p) = \sum_{i=1}^{n} v_i P_i \sigma_i^{(n\alpha)}, \quad P_i = \prod_{k=1}^{n} w_{ik}^{\alpha_k}, \sigma_i = \sigma(z_i). \tag{25}$$

As a result, the solution of the helium burning network is given as

$$\begin{aligned} X_t(x, p) &= x_0 + xN_1(x, p),\\ Y_t(x, p) &= y_0 + yN_2(y, p),\\ Z_t(x, p) &= z_0 + zN_3(z, p),\\ R_t(x, p) &= r_0 + rN_4(r, p), \end{aligned} \tag{26}$$

which fulfills the initial conditions as

$$\begin{aligned} &X_t(0, p) = x_0 + 0.N_1(0, p) = x_0, Y_t(0, p) = y_0 + 0.N_2(0, p) = y_0,\\ &Z_t(0, p) = z_0 + 0.N_3(0, p) = z_0, R_t(0, p) = r_0 + 0.N_4(0, p) = r_0,\\ &D_x^{\alpha}X_t(x, p) = x^{1-\alpha}N_1(x, p) + xD_x^{\alpha}N_1(x, p),\\ &D_x^{\alpha}Y_t(y, p) = y^{1-\alpha}N_2(y, p) + yD_x^{\alpha}N_2(y, p),\\ &D_x^{\alpha}Z_t(z, p) = z^{1-\alpha}N_3(z, p) + zD_z^{\alpha}N_3(z, p),\\ &D_r^{\alpha}R_t(r, p) = r^{1-\alpha}N_4(r, p) + rD_r^{\alpha}N_4(r, p). \end{aligned} \tag{27}$$

3.2. Gradient Computations and Parameter Updating.

Using equation (27) to update the network parameters and computing the gradient, the error quantity needed to be minimized is given by

$$
\begin{aligned}
E(x) = {} & \sum_i \left\{D_t^\alpha X_t(x_i, p) + 3aX_t^3(x_i, p) + bX_t(x_i, p)Y_t(y_i, p) + cX_t(x_i, p)Z_t(z_i, p)\right\}^2 \\
& + \sum_i \left\{D_t^\alpha Y_t(x_i, p) - aX_t^3(x_i, p) + bX_t(x_i, p)Y_t(y_i, p)\right\}^2 + \sum_i \left\{D_t^\alpha R_t(r_i, p) - cX_t(x_i, p)Z_t(z_i, p)\right\}^2 \\
& + \sum_i \left\{D_t^\alpha Z_t(z_i, p) - bX_t(x_i, p)Y_t(y_i, p) + cX_t(x_i, p)Z_t(z_i, p)\right\}^2,
\end{aligned}
\tag{28}
$$

where

$$
\begin{aligned}
D_x^\alpha X_t(x, p) &= x^{1-\alpha} N_1(x, p) + x D_x^\alpha N_1(x, p), \\
D_x^\alpha Y_t(y, p) &= y^{1-\alpha} N_2(y, p) + y D_x^\alpha N_2(y, p), \\
D_x^\alpha Z_t(z, p) &= z^{1-\alpha} N_3(z, p) + z D_z^\alpha N_3(z, p), \\
D_r^\alpha R_t(r, p) &= r^{1-\alpha} N_4(r, p) + r D_r^\alpha N_4(r, p),
\end{aligned}
\tag{29}
$$

where $D_x^\alpha N(x, p)$ is given by equation (25). So, the problem is converted into an unconstrained optimization problem.

To update the network parameters, we train the neural network for the optimized parameter values. After the training process, we obtained the network parameters and computed the following:

$$
\begin{aligned}
X_t(x, p) &= x_0 + xN_1(x, p), \\
Y_t(x, p) &= y_0 + yN_2(y, p), \\
Z_t(x, p) &= z_0 + zN_3(z, p), \\
R_t(x, p) &= r_0 + rN_4(r, p).
\end{aligned}
\tag{30}
$$

Now, N with one hidden layer is analogous to the conformable fractional derivative. By replacing the hidden unit transfer function with the n$^{\text{th}}$ order fractional derivative, the fractional N gradient differentiating with respect to v_i, β_i, and w_{ij} could be written as

$$
\begin{aligned}
D_{v_i}^\alpha N &= P_i \sigma_i^{(n\alpha)}, \\
D_{\beta_i}^\alpha N &= v_i P_i \sigma_i^{((n+1)\alpha)}, \\
D_{w_{ij}}^\alpha N &= x_i v_i P_i \sigma_i^{((n+1)\alpha)} + v_i \alpha_j w_{ij}^{1-\alpha_j} \left(\prod_{k=1, k \neq j} w_{ik}^{\alpha_k} \right) \sigma_i^{(n\alpha)}.
\end{aligned}
\tag{31}
$$

The network parameters updating rule can be given as

$$
\begin{aligned}
v_i(x+1) &= v_i(x) + aD_{v_i}^\alpha N, \\
\beta_i(x+1) &= \beta_i(x) + bD_{\beta_i}^\alpha N, \\
w_{ij}(x+1) &= w_{ij}(x) + cD_{w_{ij}}^\alpha N,
\end{aligned}
\tag{32}
$$

where a, b, c are learning rates, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, h$.

In the stellar helium burning model based on ANN, the neuron is the fundamental processing unit that can process a local memory and carry out localised information. At each neuron, the net input (z) is calculated by supplementing the received weights to obtain an aggregate weight of those inputs and add it to a bias (β). The net input (z) is then passed by a nonlinear activation function, which results in the neuron output u_j (as seen in Figure 1) [31].

3.3. Training of BP Algorithm. The backpropagation (BP) training algorithm is a gradient algorithm aimed to minimize the average square error between the desired output and the actual output of a feed-forward network.

It requires continuously differentiable nonlinearity. Figure 2 displays a flow chart of a backpropagation offline learning algorithm [32].

The algorithm is a recursive algorithm that starts at the output units, working back to the first hidden layer. A comparison of the output X_j, Y_j, Z_j, R_j at the output layer with the desired outputs tx, ty, tz, tr is performed using an error function which has the following form:

$$
\delta_j = X_j\left(tx_j - X_j\right)\left(1 - X_j\right) + Y_j\left(ty_j - Y_j\right)\left(1 - Y_j\right) + Z_j\left(tz_j - Z_j\right)\left(1 - Z_j\right) + R_j\left(tr_j - R_j\right)\left(1 - R_j\right).
\tag{33}
$$

For the hidden layer, the error function takes the form:

$$
\delta_j = \left\{X_j\left(1 - X_j\right) + Y_j\left(1 - Y_j\right) + Z_j\left(1 - Z_j\right) + R_j\left(1 - R_j\right)\right\} \sum_k \delta_k w_k,
\tag{34}
$$

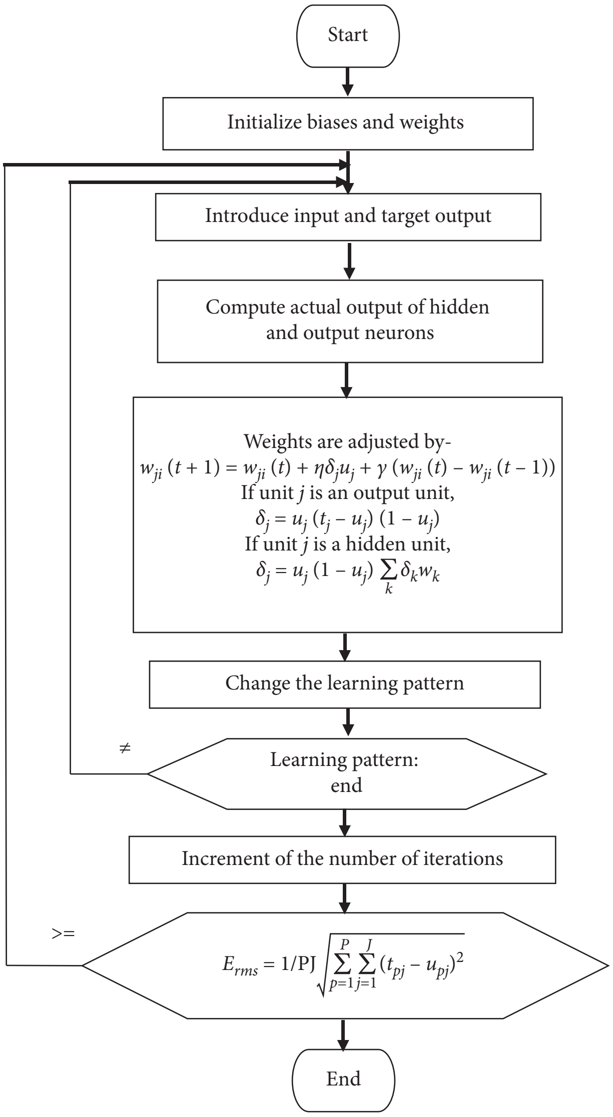

Start
Initialize biases and weights
Introduce input and target output
Compute actual output of hidden and output neurons
Weights are adjusted by-
$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j u_j + \gamma(w_{ji}(t) - w_{ji}(t-1))$
If unit j is an output unit,
$\delta_j = u_j(t_j - u_j)(1 - u_j)$
If unit j is a hidden unit,
$\delta_j = u_j(1 - u_j)\sum_k \delta_k w_k$
Change the learning pattern
≠
Learning pattern: end
Increment of the number of iterations
>=
$E_{rms} = 1/\mathrm{PJ}\sqrt{\sum_{p=1}^{P}\sum_{j=1}^{J}(t_{pj} - u_{pj})^2}$
End

Figure 2: Flowchart of an offline backpropagation training algorithm.

where δ_j is the error term of the output layer, and w_k is the weight between the output and hidden layers. The update of the weight of each connection is implemented by replicating the error in a backward direction from the output layer to the input layer as follows:

$$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j u_j + \gamma\big(w_{ji}(t) - w_{ji}(t-1)\big). \tag{35}$$

The value of learning rate η is chosen such that it is neither too large leading to overshooting nor very small leading to a slow convergence rate. The value of the momentum term found in the last part in equation (36) which is affixed with a constant γ (momentum) is used to accelerate the error convergence of the backpropagation learning algorithm and also to assist in pushing the changes of the energy function over local increases and boosting the weights in the direction of the overall downhill [33]. This term adds a fraction of the most recent weight values to the current weight values. Both η and γ terms are set at the start of the training phase and determine the network speed and stability [31, 34].

The process is repeated for each input pattern until the output error of the network is decreased to a prespecified threshold value. The final weight values are frozen and utilized to get the precise product abundances during the test session. The quality and success of training of ANN are assessed by calculating the error for the whole batch of training patterns using the normalized RMS error that is defined as

$$E_{\mathrm{rms}} = \frac{1}{PJ}\sqrt{\sum_{p=1}^{P}\sum_{j=1}^{J}\left\{\left(t_{pj} - X_{pj}\right)^{2} + \left(t_{pj} - Y_{pj}\right)^{2} + \left(t_{pj} - Z_{pj}\right)^{2} + \left(t_{pj} - R_{pj}\right)^{2}\right\}}, \tag{36}$$

where J is the number of output units; P is the number of training patterns; tx_{pj}, ty_{pj}, tz_{pj}, and tr_{pj} are the desired outputs at unit j, whereas X_{pj}, Y_{pj}, Z_{pj}, and R_{pj} are the actual outputs at the same unit j. A zero error denotes that all the output patterns computed by the stellar helium burning model match the expected values perfectly and that the stellar helium burning model is fully trained. Similarly, internal unit thresholds are adjusted by supposing they are connection weights on links from the input with an auxiliary constant value. The previous algorithm has been programmed using C++ programming language running on Windows 7 of a CORE i7 PC.

4. Results and Discussion

4.1. Data Preparation. Based on the recurrence relations (equation (18)), we computed one pure helium gas model, $X_0 = 1$, $Y_0 = 0$, $Z_0 = 0$, $R_0 = 0$, and three rich helium gas models, $X_0 = 0.95$, $Y_0 = 0.05$, $Z_0 = 0$, $R_0 = 0$; $X_0 = 0.9$, $Y_0 = 0.1$, $Z_0 = 0$, $R_0 = 0$; and $X_0 = 0.85$, $Y_0 = 0.15$, $Z_0 = 0$, $R_0 = 0$. The fractional parameter covers the range $\alpha = 0.5 - 1$ with a step of 0.05. The calculations are performed for a time $T = 2100$ s. Consequently, we have a total sum of 44 fractional helium burning models.

Figure 3 plots the two product abundances from gas models calculated at $\alpha = 0.95$, where the solid lines are for the pure helium model with initial abundance $X_0 = 1$, $Y_0 = 0$, $Z_0 = 0$, $R_0 = 0$; and the dashed lines are for the rich helium model with initial abundances, $X_0 = 0.95$, $Y_0 = 0.05$, $Z_0 = 0$, $R_0 = 0$. The effects of changing the composition of the gas are remarkable, especially for the carbon C^{12}.

In Figure 4, we illustrated the effects of changing the fractional parameters on the product abundances calculated for a gas model with initial abundance $X_0 = 0.85$, $Y_0 = 0.15$, $Z_0 = 0$, $R_0 = 0$. It is clear that the effects of the change of the fractional parameter on the behavior of the product abundances are small. This result is similar to the results obtained by [10] for the models computed in the sense of the modified Riemann–Liouville fractional derivative.

4.2. ANN Training. For the training of ANN used to simulate the helium burning network, we used part of the data calculated in the previous subsection. The data used for training of the ANN are as shown in the second column of Table 1.

The neural network (NN) architecture used in this paper for the helium burning network has three layers as shown in Figure 1. These layers are the input layer, hidden layer, and output layer. Different configurations of hidden neurons of 10, 20, and 40 have been tested, where we concluded that 20 neurons in a single hidden layer are giving the best model of the network to simulate the helium burning network. This number of neurons in the hidden layer was found to give the minimum value of RMS error of 0.000005 in an almost similar number of training iterations. As a result, the configuration of the NN we used was 4-20-4, where the input layer has four inputs which are the fractional parameter α, the time t (t takes values from 3 to 2100 in steps of 3 seconds), two of the initial abundances which are the helium (X_0), and carbon (Y_0). We excluded the other two initial abundances (Z_0 and R_0) because their values are always zero as indicated in Table 1. The output layer has 4 nodes which are the time-dependent product abundances for helium (X), carbon (Y), oxygen (Z), and neon (R).

During the training of the NN, we used a value for the learning rate ($\eta = 0.035$) and for the momentum ($\gamma = 0.5$). Those values for η and γ were proved to quicken the convergence of the backpropagation training algorithm without exceeding the solution. For the demonstration of the convergence and stability of the values computed for weight parameters of network layers, the behaviors of the convergence of the input layer weights, bias, and output layer weights (w_{i}, $\beta_{\mathrm{i},}$ and ν_{i}) for the helium burning network are displayed in Figure 5. As these figures show, the weight values are initialized to random values and after somewhat considerable iterations they converged to stable values.

4.3. Comparison between the NN Model and Analytical Model. After the end of the training phase of NN, we used the final frozen weight values in the test phase to predict the time-dependent product abundances for helium (X), carbon (Y), oxygen (Z), and neon (R). In this test phase, we used values for a fractional parameter α not being used in the training phase to predict the helium burning network model. These values are shown in the third column of Table 1. The results of the predicted values show very good agreement with the analytical values for different helium modes. A comparison between the predicted NN model values and analytical model for two values of the fractional parameters ($\alpha = 0.55$ and $\alpha = 0.95$) along with different helium modes shown in Table 1 are displayed in the range of figures , that is, Figures 6–9 for one pure helium gas model, $X_0 = 1$, $Y_0 = 0$, $Z_0 = 0$, $R_0 = 0$, and three rich helium gas models, $X_0 = 0.95$, $Y_0 = 0.05$, $Z_0 = 0$, $R_0 = 0$; $X_0 = 0.9$, $Y_0 = 0.1$, $Z_0 = 0$, $R_0 = 0$; and $X_0 = 0.85$, $Y_0 = 0.15$, $Z_0 = 0$, $R_0 = 0$. In all these figures, the very good agreement between both the NN model and analytical model is clear, which elects the NN to be considered as a powerful tool to solve and model nuclear burning networks and could be applied to the other nuclear stellar burning networks.

From the performed calculations, one can examine the effect of changing the fraction parameter over four elements. Figures 6–9 illustrate the fractional product abundances of He^4, C^{12}, O^{16}, and Ne^{20} as a function of time, where some features could be obtained. For all gas models, the difference between the abundances of He^4 computed for the two values of the fractional parameters ($\alpha = 0.55$, 0.95) is very small when the time $t \leq 200$ seconds, after that the difference

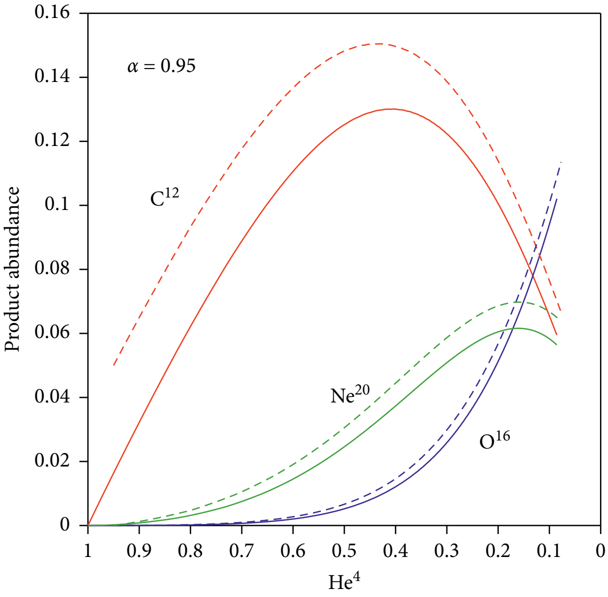


Figure 3: The product abundance (C^{12}, O^{16}, and Ne^{20}) computed analytically at $\alpha = 0.95$ for the two different abundances. The solid lines represent the helium network with $X_0 = 1$, $Y_0 = 0$, $Z_0 = 0$, and $R_0 = 0$, and the dashed lines represent the helium network $X_0 = 0.95$, $Y_0 = 0.05$, $Z_0 = 0$, and $R_0 = 0$.

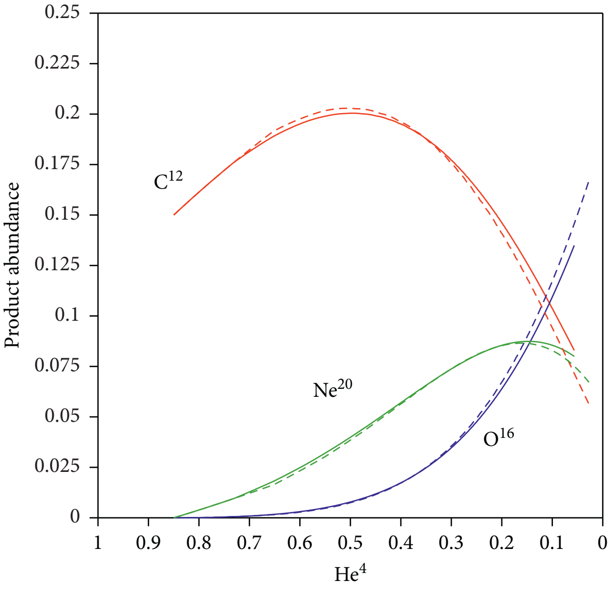
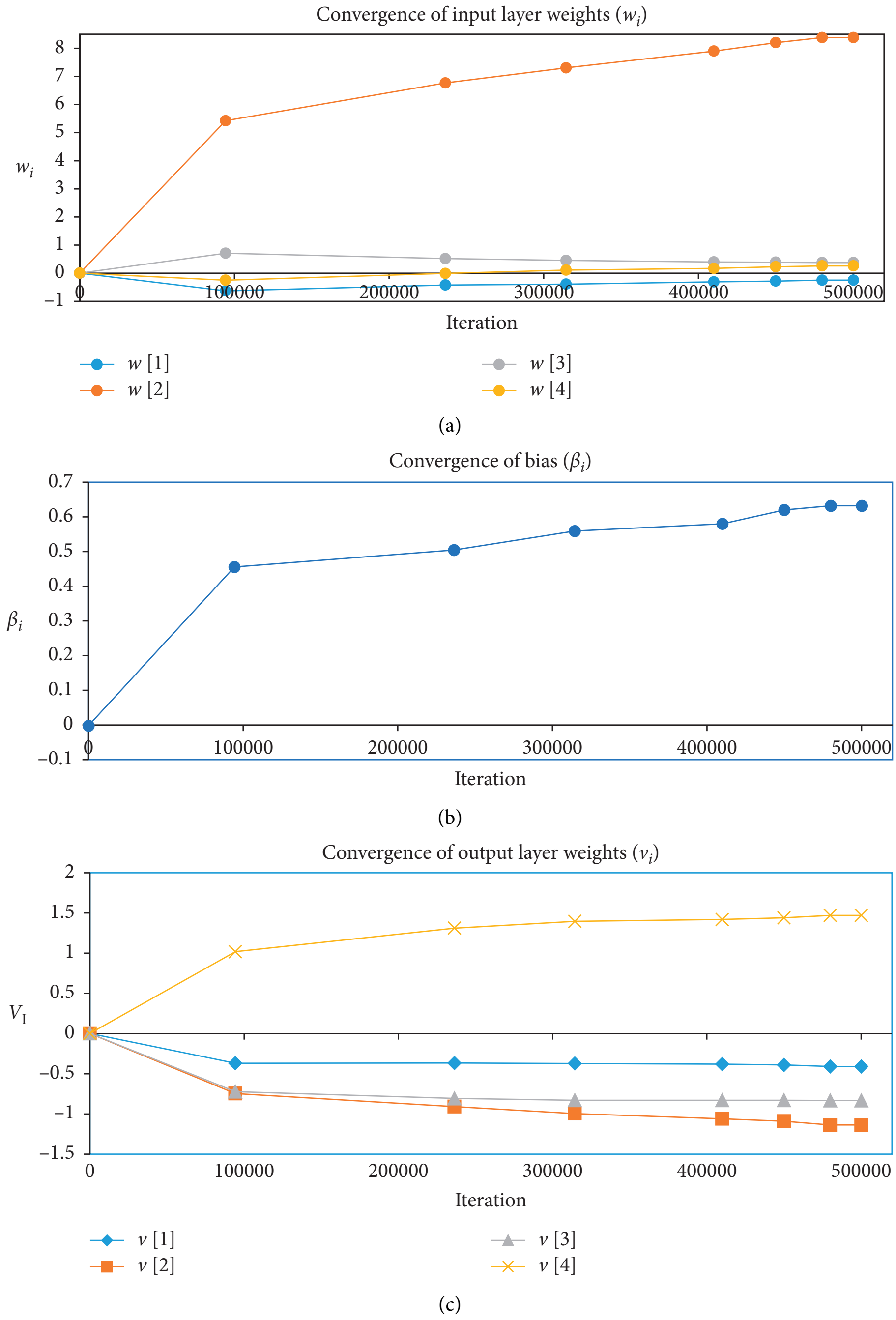


Figure 4: The product abundance (C^{12}, O^{16}, and Ne^{20}) computed analytically at $X_0 = 0.85$, $Y_0 = 0.15$, $Z_0 = 0$, and $R_0 = 0$ and for two different fractional parameters. The solid lines represent the helium network with $\alpha = 0.9$ and the dashed lines represent the helium network $\alpha = 0.5$.

Table 1: Training and testing data for the helium burning network.

	Training phase	Testing phase
α	0.5, 0.6, 0.7, 0.8, 0.9, 1	0.55, 0.65, 0.75, 0.85, 0.95
Time	0–2100 sec ($\Delta t = 3$)	0–2100 sec ($\Delta t = 3$)
Initial abundances of the HB	$X_0 = 0.85,\ Y_0 = 0.15,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 0.90,\ Y_0 = 0.1,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 0.95,\ Y_0 = 0.05,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 1,\ Y_0 = 0,\ Z_0 = 0,\ R_0 = 0$	$X_0 = 0.85,\ Y_0 = 0.15,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 0.90,\ Y_0 = 0.1,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 0.95,\ Y_0 = 0.05,\ Z_0 = 0,\ R_0 = 0$ $X_0 = 1,\ Y_0 = 0,\ Z_0 = 0,\ R_0 = 0$



(a)

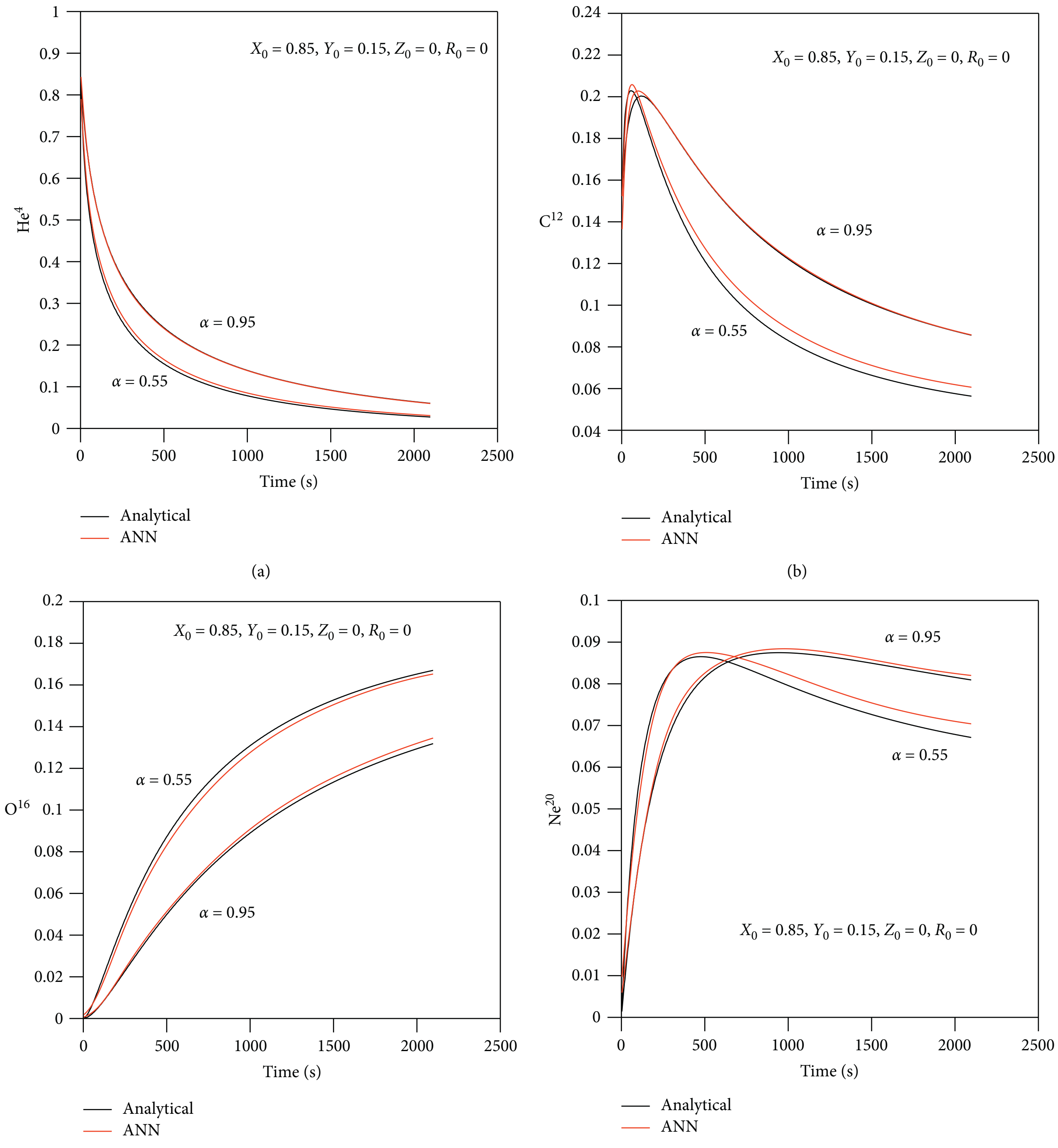
(b)

(c)

Figure 5: Convergence of the weights of input, bias, and output layers for the training of the NN used to simulate helium burning network. (a) The convergence of input layer weights (w_i). (b) The convergence of bias (β_i). (c) Convergence of output layer weights (v_i).

becomes larger. Also, it is noticed clearly that the abundance of C^{12} has the same behavior.

The behaviors of the fractional product abundances of O^{16} and Ne^{20} are different from those of He^4 and C^{12}. The differences between the fractional product abundances of O^{16} are large after just the beginning of the ignition, whereas the differences between the fractional product abundances of Ne^{20} are very small for $t \leq 100$ seconds and increase after that time.



(a)

(b)

(c)

(d)

Figure 6: The distribution of the product abundance with time for the rich helium burning network, $X_0 = 0.85$, $Y_0 = 0.15$, $Z_0 = 0$, and $R_0 = 0$.

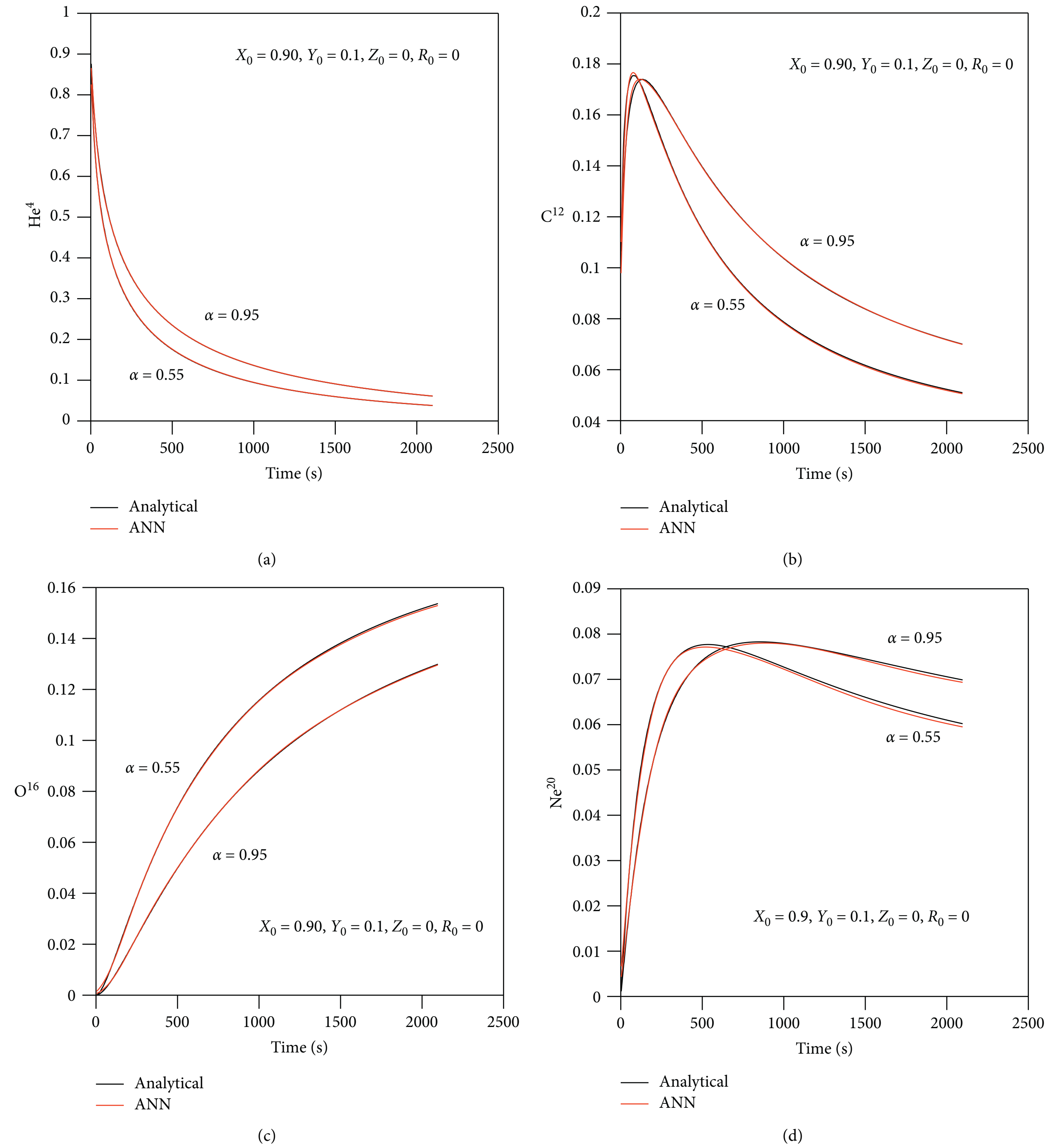
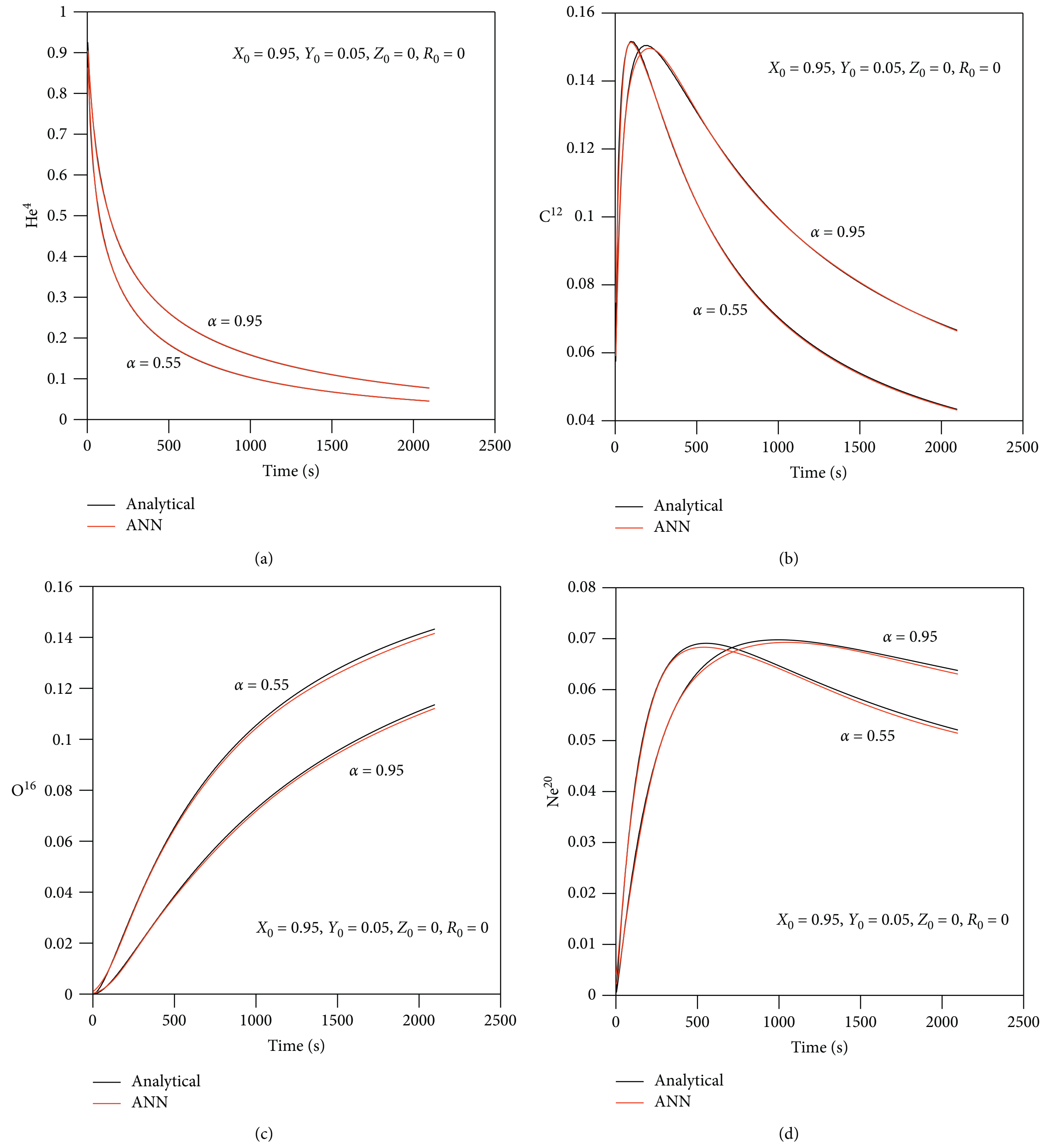


FIGURE 7: The distribution of the product abundance with time for the rich helium burning network, $X_0 = 0.9$, $Y_0 = 0.1$, $Z_0 = 0$, and $R_0 = 0$.



(a)

(b)

(c)

(d)

FIGURE 8: The distribution of the product abundance with time for the rich helium burning network, $X_0 = 0.95$, $Y_0 = 0.05$, $Z_0 = 0$, and $R_0 = 0$.

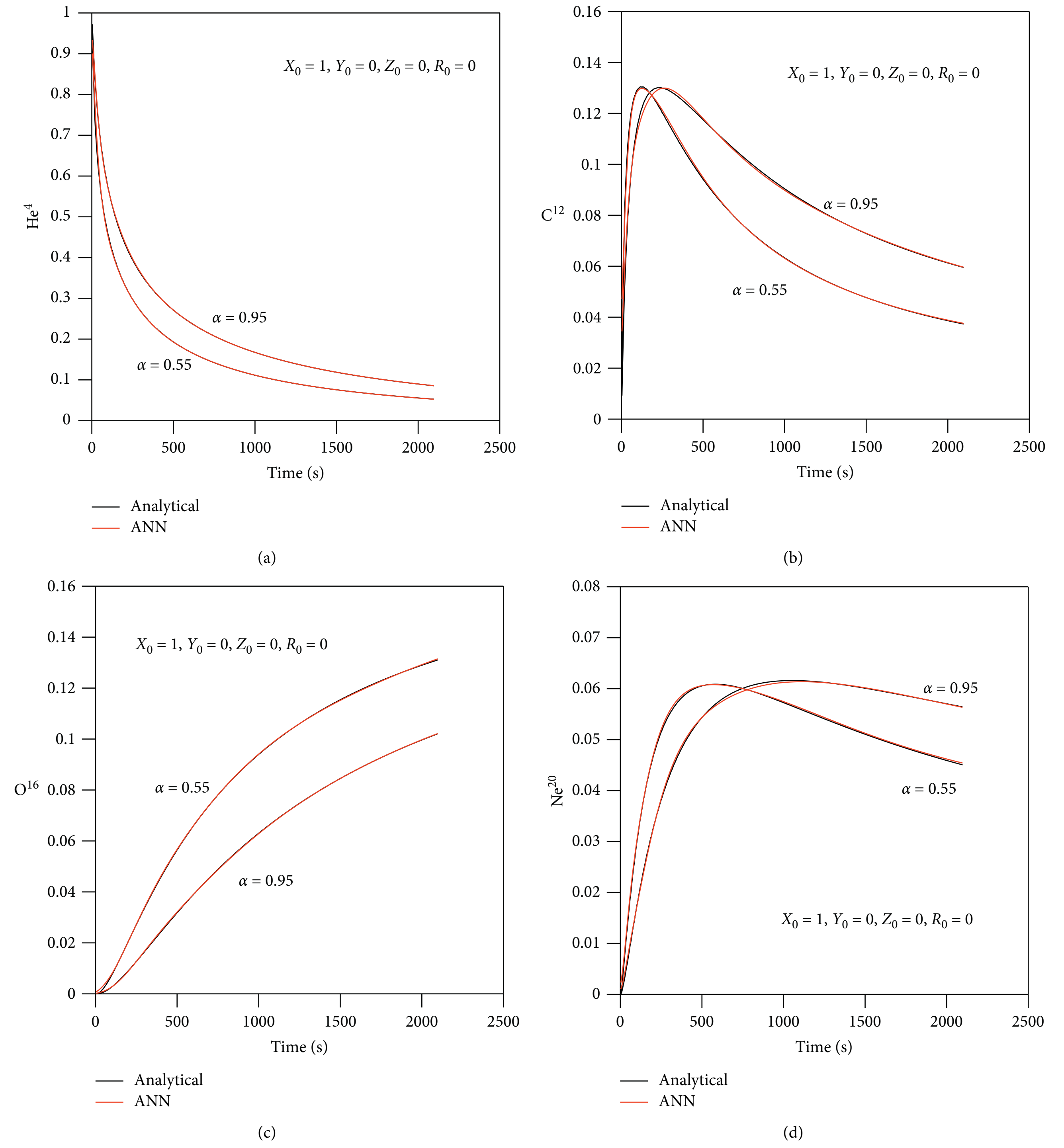


FIGURE 9: The distribution of the product abundance with time for the pure helium burning network, $X_0 = 1$, $Y_0 = 0$, $Z_0 = 0$, and $R_0 = 0$.

5. Conclusion

In the current research, we introduced an analytical solution to the conformable fractional helium burning network via a series expansion method where we obtained the product abundances of the syntheses elements as a function of time. The calculations are performed for the four different initial abundances: $(X_0 = 1,\ Y_0 = 0,\ Z_0 = 0,\ R_0 = 0)$, $(X_0 = 0.95,\ Y_0 = 0.05,\ Z_0 = 0,\ R_0 = 0)$; $(X_0 = 0.9,\ Y_0 = 0.1,\ Z_0 = 0,\ R_0 = 0)$ and $(X_0 = 0.85,\ Y_0 = 0.15,\ Z_0 = 0,\ R_0 = 0)$. The results of the analytical solution revealed that the conformable models have the same behaviors as the fractional models computed using the modified Riemann–Liouville fractional derivative. Second, we used the NN in its feed-forward type to simulate the system of the differential equations of the HB. To do that, we performed the mathematical modeling of a NN to simulate the conformable helium burning network. We trained the NN using the backpropagation delta rule algorithm and used training data for models with the fractional parameter range $\alpha = 0.5 - 1$ with step $\Delta\alpha = 0.1$. We predicted the fractional models for the range $\alpha = 0.55 - 0.95$ with step $\Delta\alpha = 0.1$. The comparison with the analytical solutions gives a very good agreement for most cases, a small difference obtained for the model with fractional parameters $\alpha = 0.55$. The results obtained in this research prove that modeling of nuclear burning networks using NN gives very good results and validates the NN to be an accurate, robust, and trustworthy method to solve and model similar networks and could be applied to other nuclear stellar burning networks comprised of conformable fractional differential equations.

Appendix

A. The Helium Burning Network

Kinetic equations governing the change in the number density N_i of species i over time are describing the nucleosynthesis of the elements in stars [35]:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = -\sum_j N_i N_j \langle\sigma v\rangle_{ij} + \sum_{k,l\neq i} N_l N_k \langle\sigma v\rangle_{kl}, \tag{A.1}$$

where $\langle\sigma v\rangle_{mn}$ for the interaction involving species m and n constitutes the reaction cross section and all the reactions producing or destroying species i shall be summed up. The number density N_i of the species i is expressed by its abundance X_i by the relation

$$N_i = \frac{\rho N_A X_i}{A_i}, \tag{A.2}$$

where N_A is Avogadro's number and A_i is the mass of i in mass units.

The reaction rate is given by

$$r_{ij} = \rho^2 N_A^2 \frac{X_i X_j}{A_i A_j} \langle\sigma v\rangle_{ij}. \tag{A.3}$$

For the helium burning reactions, the rates of the three reactions in the units of s^{-1} could be written as [23]

$$\begin{aligned} r_{3\alpha} &= \langle\sigma v\rangle_{3\alpha}\left(\mathrm{He}^4\right)^3, \\ r_{\alpha 12} &= \langle\sigma v\rangle_{\alpha 12}\mathrm{He}^4\mathrm{C}^{12}, \\ r_{\alpha 16} &= \langle\sigma v\rangle_{\alpha 16}\mathrm{He}^4\mathrm{O}^{16}. \end{aligned} \tag{A.4}$$

Now, by putting $x = \mathrm{He}^4, y = \mathrm{C}^{12}, z = \mathrm{O}^{16}, r = \mathrm{Ne}^{20}$ for the helium, carbon, oxygen, and neon abundances in number density (in units of cm^{-3}), respectively, and implementing equations (A.2)–(A.4), the abundance differential equation (Equation (A.1)) could be written as

$$\begin{aligned} \frac{dN_1}{dt} &= \frac{\rho N_A}{A_x}\frac{dx}{dt} = -\langle\sigma v\rangle_{3\alpha} x^3 - \langle\sigma v\rangle_{\alpha 12} xy - \langle\sigma v\rangle_{\alpha 16} xz, \\ \frac{\mathrm{d}N_2}{\mathrm{d}t} &= \frac{\rho N_A}{A_y}\frac{\mathrm{d}y}{\mathrm{d}t} = \langle\sigma v\rangle_{3\alpha} x^3 - \langle\sigma v\rangle_{\alpha 12} xy, \\ \frac{\mathrm{d}N_3}{\mathrm{d}t} &= \frac{\rho N_A}{A_z}\frac{\mathrm{d}z}{\mathrm{d}t} = \langle\sigma v\rangle_{\alpha 12} xy - \langle\sigma v\rangle_{\alpha 16} xz, \\ \frac{\mathrm{d}N_4}{\mathrm{d}t} &= \frac{\rho N_A}{A_r}\frac{\mathrm{d}r}{\mathrm{d}t} = \langle\sigma v\rangle_{\alpha 16} xz. \end{aligned} \tag{A.5}$$

Using equations (A.4) and (A.5) could be written as

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= -r_{3\alpha}x^3 - r_{\alpha 12}xy - r_{\alpha 16}xz, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= r_{3\alpha}x^3 - r_{\alpha 12}xy, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= r_{\alpha 12}xy - r_{\alpha 16}xz, \\ \frac{\mathrm{d}r}{\mathrm{d}t} &= r_{\alpha 16}xz, \end{aligned} \tag{A.6}$$

where the abundances x, y, z, and r are expressed in number instead of number density. By replacing the reaction rates $r_{3\alpha}, r_{\alpha 12}$, and $r_{\alpha 16}$ in equation (A.6) by a, b, and c, respectively, we obtained equation (2).

Data Availability

The Excel data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the Academy of Scientific Research and Technology (ASRT), Egypt (Grant no. 6413), under the project Science Up. ASRT is the second affiliation of this research.

References

[1] T. M. Michelitsch, B. A. Collet, A. P. Riascos, A. F. Nowakowski, and F. C. G. A. Nicolleau, "A fractional generalization of the classical lattice dynamics approach," *Chaos, Solitons and Fractals*, vol. 92, pp. 1339–1351, 2016, ISSN 0960-0779.

[2] R. Hilfer, (ed), *Applications of Fractional Calculus in Physics*, World Scientific, Singapore, 2000.

[3] R. A. El-Nabulsi, "Implications of the Ornstein-Uhlenbeck-Like fractional differential equation in cosmology," *Revista Mexicana de Fisica*, vol. 62, pp. 240–250, 2016.

[4] R. A. El-Nabulsi, “On generalized fractional spin, fractional angular momentum, fractional momentum operators in quantum mechanics,” *Few-Body System*, vol. 61, p. 25, 2020.

[5] R. A. El-Nabulsi, “Saigo-maeda operators involving the Appell function, real spectra from symmetric quantum Hamiltonians and violation of the second law of thermodynamics for quantum damped oscillators,” *International Journal of Theoretical Physics*, vol. 59, no. 12, pp. 3721–3736, 2020.

[6] A. R. El-Nabulsi, “The fractional white dwarf hydrodynamical nonlinear differential equation and emergence of quark stars,” *Applied Mathematics and Computation*, vol. 218, no. 6, p. 2837, 2011.

[7] S. S. Bayin and J. P. Krisch, “Fractional incompressible stars,” *Astrophysics and Space Science*, vol. 359, no. 2, 2015.

[8] E. A.-B. Abdel-Salam and M. I. Nouh, “Approximate solution to the fractional second-type lane-emden equation,” *Astrophysics*, vol. 59, no. 3, p. 398, 2016.

[9] M. I. Nouh and E. A.-B. Abdel-Salam, “Analytical solution to the fractional polytropic gas spheres,” *European Physical Journal Plus*, vol. 133, p. 149, 2018.

[10] M. I. Nouh, “Computational method for a fractional model of the helium burning network,” *New Astronomy*, vol. 66, p. 40, 2019.

[11] E. A.-B. Abdel-Salam and M. I. Nouh, “Conformable fractional polytropic gas spheres,” *New Astronomy*, vol. 76, p. 101322, 2020.

[12] E. A. Yousif, A. M. A. Adam, A. A. Hassaballa, and M. I. Nouh, “Conformable fractional isothermal gas spheres,” *New Astronomy*, vol. 84, p. 101511, 2021.

[13] M. W. M. G. Dissanayake and N. Phan-Thien, “Neural-network-based approximations for solving partial differential equations,” *Communications in Numerical Methods in Engineering*, vol. 10, no. 3, pp. 195–201, 1994.

[14] I. E. Lagaris, A. Likas, and D. I. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations,” *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[15] I. E. Lagaris, A. C. Likas, and D. G. Papageorgiou, “Neural-network methods for boundary value problems with irregular boundaries,” *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1041–1049, 2000.

[16] K. S. McFall and J. R. Mahan, “Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions,” *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1221–1233, 2009.

[17] M. Baymani, A. Kerayechian, and S. Effati, “Artificial neural networks approach for solving Stokes problem,” *Applied Mathematics*, vol. 01, no. 4, pp. 288–292, 2010.

[18] L. Lu, X. Meng, Z. Mao, E. George, and K. DeepXDE, “A deep learning library for solving differential equations,” 2020, http://arxiv.org/abs/1907.04502.

[19] K. Alexander, R. Khudorozkov, and S. Tsimfer, “PYDens: a python framework for solving differential equations with neural networks,” 2019, http://arxiv.org/abs/1909.11544.

[20] F. Chen, D. Sondak, P. Protopapas et al., “NeuroDiffEq: a Python package for solving differential equations with neural networks,” *Journal of Open Source Software*, vol. 5, no. 46, p. 1931, 2020.

[21] M. I. Nouh, Y. A. Azzam, and E. A.-B. Abdel-Salam, “Modeling fractional polytropic gas spheres using artificial neural network,” *Neural Computation and Application*, 2020.

[22] Y. A. Azzam, E. A.-B. Abdel-Salam, and M. I. Nouh, “Artificial neural network modeling of the conformable fractional isothermal gas spheres,” *RMxAA*, vol. 57, no. 1, 2021.

[23] D. D. Clayton, *Principles of Stellar Evolution and Nucleosynthesis*, University of Chicago Press, Chicago, IL, USA, 1983.

[24] H. L. Duorah and R. S. Kushwaha, “Helium-Burning reaction products and the rate of energy generation,” *The Astrophysical Journal*, vol. 137, p. 566, 1963.

[25] W. R. Hix and F.-K. Thielemann, “Computational methods for nucleosynthesis and nuclear energy generation,” *Journal of Computational and Applied Mathematics*, vol. 109, no. 1-2, p. 321, 1999.

[26] M. I. Nouh, M. A. Sharaf, and A. S. Saad, “Symbolic analytical solutions for the abundances differential equations of the Helium burning phase,” *Astronomische Nachrichten*, vol. 324, no. 5, p. 432, 2003.

[27] H. J. Haubold and A. M. Mathai, “The fractional kinetic equation and thermonuclear functions,” *Astrophysics and Space Science*, vol. 273, no. 1/4, pp. 53–63, 2000.

[28] R. K. Saxena, A. M. Mathai, and H. J. Haubold, “On fractional kinetic equations,” *Astrophysics and Space Science*, vol. 282, no. 1, pp. 281–287, 2002.

[29] V. B. L. Chaurasia and S. C. Pandey, “Computable extensions of generalized fractional kinetic equations in astrophysics,” *Research in Astronomy and Astrophysics*, vol. 10, no. 1, p. 22, 2010.

[30] N. Yadav, A. Yadav, and M. Kumar, *An Introduction to Neural Network Methods for Differential Equations*, Springer, Berlin, Germany, 2015.

[31] H. K. Elminir, Y. A. Azzam, and F. I. Younes, “Prediction of hourly and daily diffuse fraction using neural network, as compared to linear regression models,” *Energy*, vol. 32, no. 8, pp. 1513–1523, 2007.

[32] T. Fukuda, Y. Hasegawa, K. Sekiyama, and A. Tadayoshi, *Multi-Locomotion Robotic Systems: New Concepts of Bio-Inspired Robotics*, Springer, Berlin, Germany, 2012.

[33] C. Denz, *Optical Neural Networks*, Springer, Berlin, Germany, 1998.

[34] I. A. Basheer and M. Hajmeer, “Artificial neural networks: fundamentals, computing, design, and application,” *Journal of Microbiological Methods*, vol. 43, no. 1, pp. 3–31, 2000.

[35] V. Kourganoff, *Introduction to the Physics of Stellar Interiors*, D. Reidel Publishing Company, Dordrecht, The Netherlands, 1973.