foundations

Article

Iterated Crank–Nicolson Runge–Kutta Methods and Their Application to Wilson–Cowan Equations and Electroencephalography Simulations

Jinjie Liu *, Qi Lu, Hacene Boukari and Fatima Boukari

Division of Physics, Engineering, Mathematics, and Computer Science, Delaware State University, Dover, DE 19901, USA; qilu@desu.edu (Q.L.); hboukari@desu.edu (H.B.); fboukari@desu.edu (F.B.)
* Correspondence: jliu@desu.edu

Abstract: The Wilson–Cowan model has been widely applied for the simulation of electroencephalography (EEG) waves associated with neural activities in the brain. The Runge–Kutta (RK) method is commonly used to numerically solve the Wilson–Cowan equations. In this paper, we focus on enhancing the accuracy of the numerical method by proposing a strategy to construct a class of fourth-order RK methods using a generalized iterated Crank–Nicolson procedure, where the RK coefficients depend on a free parameter c_2. When c_2 is set to 0.5, our method becomes a special case of the classical fourth-order RK method. We apply the proposed methods to solve the Wilson–Cowan equations with two and three neuron populations, modeling EEG epileptic dynamics. Our simulations demonstrate that when c_2 is set to 0.4, the proposed RK4-04 method yields smaller errors compared to those obtained using the classical fourth-order RK method. This is particularly visible when the spectral radius of the connection matrix or the excitation-inhibition coupling coefficient is relatively large.

Keywords: Runge–Kutta method; iterated Crank–Nicolson method; Wilson–Cowan equations; EEG simulation

Citation: Liu, J.; Lu, Q.; Boukari, H.; Boukari, F. Iterated Crank–Nicolson Runge–Kutta Methods and Their Application to Wilson–Cowan Equations and Electroencephalography Simulations. *Foundations* **2024**, *4*, 673–689. https://doi.org/10.3390/foundations4040042

Academic Editor: Jay Jahangiri

Received: 1 July 2024
Revised: 8 October 2024
Accepted: 30 October 2024
Published: 13 November 2024

1. Introduction

In computational neuroscience, the Wilson–Cowan model is an important tool for studying neural activities in the brain [1–3]. It describes the interactions between excitatory and inhibitory neuron populations, and is widely used to simulate electroencephalography (EEG) waves [4–8]. For example, in [7], the Wilson–Cowan model is extended to a system of three equations (one excitatory and two inhibitory neuron populations), and is employed to simulate EEG waves in the context of epileptic dynamics. In [9], a four-population network is introduced to study sleep regulation, consisting of excitatory, inhibitory, sleep-promoting, and wake-promoting neurons.

The Wilson–Cowan model is a system of nonlinear ordinary differential equations (ODEs). Due to its nonlinearity and the use of a sigmoid function, insights from the Wilson–Cowan model rely mainly on numerical solutions. For instance, when the Wilson–Cowan model is applied to modeling EEG signals, the solutions typically exhibit highly oscillatory behavior. The robustness and efficiency of the used numerical methods then become important considerations and integral parts for the validity of the solutions.

Here, the Crank–Nicolson (CN) method [10] is a widely used numerical algorithm for solving differential equations. This algorithm produces an implicit system, which is typically solved using an iterative solver, leading to the development of the iterated Crank–Nicolson (ICN) method. The ICN algorithm has been applied to numerically solve differential equations associated with diverse physical phenomena, including relativity [11–13], peridynamics [14], beam propagation [15,16], and electromagnetism (Maxwell's equations) [17,18]. The original ICN algorithm is a second-order accurate method [11], and it has recently been extended to third-order accuracy [19,20]. To our knowledge, no work has been performed to generalize this method to fourth-order accuracy. It is worth to noting that the ICN method can be viewed as a type of Runge–Kutta (RK) method, which is commonly used to solve differential equations [21–23]. A more popular explicit RK method is the classical fourth-order RK method, known as the RK4 method [23].

In this paper, we propose a strategy to extend the ICN algorithm to fourth-order accuracy. Since the proposed methods can also be interpreted as RK methods, we refer to them as Iterated Crank–Nicolson Runge–Kutta (ICN-RK) methods. Specifically, we develop a class of four-stage, fourth-order algorithms where the coefficients depend on a free parameter. When this parameter is set to 0.5, the corresponding method is the classical RK4 method. The proposed fourth-order RK methods are employed to solve the Wilson–Cowan equations for two and three neuron populations. The simulated EEG signals include single-spike and poly-spike waves, as well as the transition from single to poly-spike waves. The use of the free parameter enables us to investigate the relationship between this parameter and the accuracy of solutions. Through a series of numerical simulations, we examine the proposed RK methods as the free parameter varies from 0.1 to 0.9. First, we verify the convergence rates of these methods. Second, we confirm that the EEG waves simulated using our methods are comparable to the simulation results and the clinical data reported in [7]. Finally, we compare our methods to the classical fourth-order RK method by evaluating their performance with different connection matrices, including those with varying spectral radii and excitation-inhibition coupling coefficients.

The paper is organized as follows: In Section 2, we briefly review the Wilson–Cowan equations. In Section 3, we present the derivation of a new class of fourth-order RK methods based on a generalized ICN procedure. Section 4 provides numerical examples of Wilson–Cowan equations for EEG simulations, followed by a discussion and conclusion.

2. The Wilson–Cowan Model

The original Wilson–Cowan model can be written as a system of two differential equations [1,2],

$$\tau_e \frac{dE(t)}{dt} = -E(t) + (1 - rE(t))S(C_{11}E(t) + C_{12}I(t) + P), \tag{1}$$

$$\tau_i \frac{dI(t)}{dt} = -I(t) + (1 - rI(t))S(C_{21}E(t) + C_{22}I(t) + Q), \tag{2}$$

where $E(t)$ and $I(t)$ represent the excitatory and inhibitory neuron populations, respectively. τ_e and τ_i are time constants, and r is the refractory period. The coefficients C_{ij} form a connection matrix C, where C_{11} and C_{22} represent the feedback strength from the excitatory and inhibitory neurons to themselves, respectively. C_{12} and C_{21} are the excitation-inhibition coupling coefficients, where C_{12} represents the inhibition strength from the inhibitory neurons to the excitatory neurons, and C_{21} represents the excitation strength from the excitatory neurons to the inhibitory neurons. P and Q are the external inputs. S is a sigmoid (logistic) function [1],

$$S(x) = \frac{1}{1 + e^{-a(x-b)}}, \tag{3}$$

where a indicates the steepness of the sigmoid function and b is the threshold. Because of the sigmoid function S and nonlinearity, the Wilson–Cowan system generally does not have analytical solutions. Traditionally, this system is solved using numerical methods. We can write this system (1)–(2) in matrix form,

$$\frac{dU}{dt} = T^{-1}(-U + AS(CU + B)), \tag{4}$$

where $U = \begin{pmatrix} E \\ I \end{pmatrix}$, $T = \begin{pmatrix} \tau_e & 0 \\ 0 & \tau_i \end{pmatrix}$, $A = \begin{pmatrix} 1 - rE & 0 \\ 0 & 1 - rI \end{pmatrix}$, $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$, and $B = \begin{pmatrix} P \\ Q \end{pmatrix}$.

Using the matrix Equation (4), the Wilson–Cowan system can be generalized to higher dimensions to model multiple excitatory and inhibitory neuron populations, where U and B are n dimensional vectors and T, A, and C are $n \times n$ matrices. In [7], the general Wilson–Cowan system with three neuron populations is used to simulate EEG for epileptic dynamics. This three-population system includes one excitatory and two inhibitory neuron populations, and the equations are given by [7]:

$$\tau_e \frac{dE(t)}{dt} = -E(t) + S(C_{11}E(t) + C_{12}I(t) + C_{13}J(t) + P), \tag{5}$$

$$\tau_i \frac{dI(t)}{dt} = -I(t) + S(C_{21}E(t) + C_{22}I(t) + C_{23}J(t) + Q), \tag{6}$$

$$\tau_j \frac{dJ(t)}{dt} = -J(t) + S(C_{31}E(t) + C_{32}I(t) + C_{33}J(t) + R), \tag{7}$$

where $E(t)$ is the excitatory neuron population, $I(t)$ and $J(t)$ represent two inhibitory neuron populations. τ_e, τ_i, and τ_j are the three time constants. The refractory period r in the original Wilson–Cowan system is chosen to be zero. The C_{ij} terms form a 3×3 connectivity matrix, and P, Q, and R are the three external inputs.

Similarly, system (5)–(7) can also be written in matrix form,

$$\frac{du}{dt} = T^{-1}(-u + S(Cu + B)), \tag{8}$$

where $u = (E, I, J)^T$, $T = \text{diag}(\tau_e, \tau_i, \tau_j)$, $C = (C_{ij})$, and $B = (P, Q, R)^T$. Here, v^T represents the transpose of vector v. In fact, τ, C, and B can be time-dependent. For example, in [7], P gradually increases from 3 to 5 in order to simulate the transition from single-spike wave to poly-spike wave.

If we let $f(t,u) = T^{-1}(-u + S(Cu + B))$ in Equation (8), then the Wilson–Cowan system can be written in the form of a standard differential equation,

$$\frac{du}{dt} = f(t,u). \tag{9}$$

To find a particular solution to this equation, we need an initial condition $u(0) = u_0$, where u_0 is a constant.

3. Iterated Crank–Nicolson Runge–Kutta Methods

Since the Wilson–Cowan system is a nonlinear system, commonly used numerical methods include the Runge–Kutta algorithm, Crank–Nicolson method, and others. In this section, we start with a general iterated Crank–Nicolson procedure, and construct a class of fourth-order explicit Runge–Kutta methods for the nonlinear ODE system (9).

The Crank–Nicolson algorithm is based on the following implicit update equation [10]:

$$u^{n+1} = u^n + h\left(\frac{1}{2}f(t_n, u^n) + \frac{1}{2}f(t_{n+1}, u^{n+1})\right). \tag{10}$$

where h is the time step, and u^n and u^{n+1} represent the solutions at two consecutive time levels, $t_n = nh$ and $t_{n+1} = t_n + h$, respectively.

The CN Equation (10) can be solved explicitly using iterations, leading to the following iterated Crank–Nicolson (ICN) algorithm [11]:

$$u_1 = u^n + hf(t_n, u^n), \tag{11}$$

$$u_j = u^n + h\left(\frac{1}{2}f(t_n, u^n) + \frac{1}{2}f(t_n + h, u_{j-1})\right), \quad j = 2, 3, ..., s, \tag{12}$$

$$u^{n+1} = u_s, \tag{13}$$

where s represents the number of iterations. The original ICN method is second-order accurate. Increasing the number of iterations does not increases the order of accuracy [11], but it does reduce the numerical dissipation [20].

To extend the ICN method to higher orders, we modify the weight $\frac{1}{2}$ to θ_j in the j-th iteration, and use a parameter-dependent time step $c_j h$. This yields the following generalized ICN algorithm [12,19]:

$$
\begin{aligned}
u_1 &= u^n + c_1 h f(t_n, u^n), && (14)\\
u_j &= u^n + c_j h\big((1-\theta_j) f(t_n, u^n) + \theta_j f(t_n + c_{j-1}h, u_{j-1})\big), \quad j = 2, 3, ..., s, && (15)\\
u^{n+1} &= u_s, && (16)
\end{aligned}
$$

which can also be written as

$$
\begin{aligned}
k_1 &= f(t_n, u^n), && (17)\\
k_{j+1} &= f(t_n + c_j h, u^n + c_j h((1-\theta_j)k_1 + \theta_j k_j)), \quad j = 1, 2, ..., s-1, && (18)\\
u^{n+1} &= u^n + c_s h((1-\theta_s)k_1 + \theta_s k_s). && (19)
\end{aligned}
$$

For instance, when $s = 3$, the generalized ICN method can reach third-order accuracy if the coefficients c_j and θ_j satisfy:

$$
c_1 = 1, c_2 = 2/3, c_3 = 1, \theta_1 = 1/2, \theta_2 = 1/3, \theta_3 = 3/4. \tag{20}
$$

When the coefficients satisfy Condition (20), the method can be written as

$$
\begin{aligned}
k_1 &= f(t_n, u^n), && (21)\\
k_2 &= f(t_n + h, u^n + hk_1), && (22)\\
k_3 &= f\left(t_n + \frac{2}{3}h, u^n + \frac{4}{9}hk_1 + \frac{2}{9}hk_2\right), && (23)\\
u^{n+1} &= u^n + \frac{1}{4}hk_1 + \frac{3}{4}hk_3. && (24)
\end{aligned}
$$

This RK method, (21)–(24), is third-order accurate, as it satisfies the criteria given in [24]. Furthermore, if $f = f(u)$ is linear, then this method is a strong stability preserving method [24,25]. More detailed derivations are provided in [20].

When $s = 4$, Equations (17)–(19) become

$$
\begin{aligned}
k_1 &= f(t_n, u^n), && (25)\\
k_2 &= f(t_n + c_1 h, u^n + c_1 hk_1), && (26)\\
k_3 &= f(t_n + c_2 h, u^n + c_2 h[(1-\theta_2)k_1 + \theta_2 k_2]), && (27)\\
k_4 &= f(t_n + c_3 h, u^n + c_3 h[(1-\theta_3)k_1 + \theta_3 k_3]), && (28)\\
u^{n+1} &= u^n + c_4 h((1-\theta_4)k_1 + \theta_4 k_4). && (29)
\end{aligned}
$$

For the nonlinear case, to achieve fourth-order accuracy, the coefficients must satisfy the following conditions [22]:

$$
\begin{aligned}
c_4\theta_4 c_3 &= \frac{1}{2}, && (30)\\
c_4\theta_4 c_3^2 &= \frac{1}{3}, && (31)\\
c_4\theta_4 c_3^3 &= \frac{1}{4}. && (32)
\end{aligned}
$$

However, this system has no solution. Therefore, the algorithm given by Equations (17)–(19) is not fourth order when $s = 4$, regardless of the choice of θ_j and c_j.

To achieve fourth-order accuracy, we modify Equation (19) by including all k_j in the calculation of u^{n+1}. The resulting algorithm becomes

$$k_1 = f(t_n, u^n), \tag{33}$$
$$k_2 = f(t_n + c_1 h, u^n + c_1 h k_1), \tag{34}$$
$$k_3 = f(t_n + c_2 h, u^n + c_2 h[(1-\theta_2)k_1 + \theta_2 k_2]), \tag{35}$$
$$k_4 = f(t_n + c_3 h, u^n + c_3 h[(1-\theta_3)k_1 + \theta_3 k_3]), \tag{36}$$
$$u^{n+1} = u^n + w_1 h k_1 + w_2 h k_2 + w_3 h k_3 + w_4 h k_4, \tag{37}$$

and the coefficients must satisfy the following eight equations [22]:

$$w_1 + w_2 + w_3 + w_4 = 1, \tag{38}$$
$$w_2 c_1 + w_3 c_2 + w_4 c_3 = \frac{1}{2}, \tag{39}$$
$$w_2 c_1^2 + w_3 c_2^2 + w_4 c_3^2 = \frac{1}{3}, \tag{40}$$
$$w_2 c_1^3 + w_3 c_2^3 + w_4 c_3^3 = \frac{1}{4}, \tag{41}$$
$$w_3 \theta_2 c_2 c_1 + w_4 \theta_3 c_3 c_2 = \frac{1}{6}, \tag{42}$$
$$w_3 \theta_2 c_2 c_1^2 + w_4 \theta_3 c_3 c_2^2 = \frac{1}{12}, \tag{43}$$
$$w_3 \theta_2 c_2^2 c_1 + w_4 \theta_3 c_3^2 c_2 = \frac{1}{8}, \tag{44}$$
$$w_4 \theta_3 \theta_2 c_3 c_2 c_1 = \frac{1}{24}. \tag{45}$$

One solution to the system of Equations (38)–(45) is

$$c_1 = c_2 = \frac{1}{2}, c_3 = 1, \theta_2 = \theta_3 = 1, w_1 = w_4 = \frac{1}{6}, w_2 = w_3 = \frac{1}{3}, \tag{46}$$

where the RK algorithm (33)–(37) is exactly the classical fourth-order RK (RK4) method,

$$k_1 = f(t_n, u^n), \tag{47}$$
$$k_2 = f(t_n + \frac{h}{2}, u^n + \frac{1}{2} h k_1), \tag{48}$$
$$k_3 = f(t_n + \frac{h}{2}, u^n + \frac{1}{2} h k_2), \tag{49}$$
$$k_4 = f(t_n + h, u^n + h k_3), \tag{50}$$
$$u^{n+1} = u^n + \frac{1}{6} h (k_1 + 2k_2 + 2k_3 + k_4). \tag{51}$$

Now, we work on the general solution to the system of Equations (38)–(45), as each solution set leads to a fourth-order RK method. We observe that Equations (42)–(44) lead to the following equation:

$$4c_2^2 - 5c_2 + 3c_1 - 4c_1 c_3 + 2c_3 = 0, \tag{52}$$

so, given two of the three unknowns, we can calculate the other one using this equation.

In this work, we focus on deriving a family of fourth-order RK algorithms when $c_3 = 1$, because the same c_3 is used in the classical RK4 method. When $c_3 = 1, 0 < c_2 < 1$, $c_2 \neq \frac{3}{4}$, and $c_2 \neq \frac{1}{4}$, we solve the system of Equations (38)–(45) and obtain their explicit solutions. Each solution set leads to a fourth-order RK method. We summarize this result in the following theorem.

Theorem 1. *The Runge–Kutta algorithm (33)–(37) is fourth-order if* $c_3 = 1$, $0 < c_2 < 1$, $c_2 \neq \frac{1}{4}$, $c_2 \neq \frac{3}{4}$, *and the other parameters are calculated using the following sequence of equations explicitly:*

$$c_1 = 4c_2^2 - 5c_2 + 2, \tag{53}$$

$$\theta_2 = \frac{c_2 - 1}{c_1(4c_2 - 3)}, \tag{54}$$

$$w_3 = -\frac{4c_2 - 3}{24c_2(1 - c_2)^2}, \tag{55}$$

$$w_2 = \frac{\frac{1}{6} - w_3c_2 + w_3c_2^2}{c_1(1 - c_1)}, \tag{56}$$

$$w_4 = \frac{-c_1^2(\frac{1}{2} - w_3c_2) + c_1(\frac{1}{3} - w_3c_2^2)}{c_1(1 - c_1)}, \tag{57}$$

$$\theta_3 = \frac{1}{24c_1c_2\theta_2w_4}, \tag{58}$$

$$w_1 = 1 - w_2 - w_3 - w_4. \tag{59}$$

Proof. Equations (38)–(45) are the order conditions for the fourth-order RK method, as they coincide with the fourth-order conditions given in Equations (235a)–(235h) in Butcher's book [22]. Next, we need to show that the coefficients computed using Equations (53)–(59) satisfy the order conditions (38)–(45), ensuring that the proposed RK algorithm (33)–(37) is fourth-order.

Note that when $c_2 = \frac{3}{4}$ or $c_2 = \frac{1}{4}$, the system of Equations (38)–(45) has no solution. If $c_2 = \frac{1}{4}$, then $c_1 = 1$ by Equation (53), and the system of Equations (39)–(41) becomes inconsistent. If $c_2 = \frac{3}{4}$, then $c_1 = \frac{1}{2}$, and the system of Equations (42)–(44) is inconsistent. Therefore, we assume $c_2 \neq \frac{1}{4}$ and $\frac{3}{4}$.

First, by substituting $c_3 = 1$ into Equation (52), we obtain Equation (53) for calculating c_1. Next, we solve the system of Equations (42)–(44) for $w_3\theta_2$ and $w_4\theta_3$ in terms of c_1 and c_2,

$$w_3\theta_2 = \frac{1}{24c_1c_2(1 - c_2)}, \tag{60}$$

$$w_4\theta_3 = \frac{3 - 4c_2}{24c_2(1 - c_2)}. \tag{61}$$

By solving Equation (45) for θ_2, and then Equation (60) for w_3, we obtain Equations (54) and (55) for calculating θ_2 and w_3, respectively. Next, solving the system of Equations (39)–(41) for w_2 and w_4, which lead to Equations (56) and (57), respectively. θ_3 can then be calculated using Equation (45), yielding Equation (58). Note that θ_3 can also be computed using Equation (61), resulting in the same solution. Finally, by solving Equation (38) for w_1, we obtain Equation (59). □

In particular, when c_2 is chosen to be a rational number, all other coefficients are rational. Therefore, we obtain a class of fourth-order RK methods where the corresponding coefficients are determined by a free parameter c_2 ($c_2 \neq \frac{3}{4}$ and $c_2 \neq \frac{1}{4}$). For instance, if we let c_2 vary from 0.1 to 0.9, and c_3 is fixed at 1, the corresponding coefficients of fourth-order RK methods are listed in Table 1. We use RK4-0x to denote the fourth-order RK method with coefficients determined by the parameter 0.x. For example, RK4-05 is the classical fourth-order RK method. When $c_2 = 0.4$, the corresponding RK method is RK4-04, and it can be written as

$$k_1 = f(t_n, u^n), \tag{62}$$

$$k_2 = f(t_n + \frac{16}{25}h, u^n + \frac{16}{25}hk_1), \tag{63}$$

$$k_3 = f(t_n + \frac{2}{5}h, u^n + \frac{37}{280}hk_1 + \frac{15}{56}hk_2), \tag{64}$$

$$k_4 = f(t_n + h, u^n - \frac{127}{188}hk_1 + \frac{315}{188}hk_3), \tag{65}$$

$$u^{n+1} = u^n + \frac{h}{10368}(1539k_1 + 3125k_2 + 4200k_3 + 1504k_4). \tag{66}$$

Table 1. Coefficients for fourth-order RK methods when $c_3 = 1$ and c_2 varies from 0.1 to 0.9.

Method	c_1	c_2	c_3	θ_2	θ_3	w_1	w_2	w_3	w_4
RK4-01	$\frac{77}{50}$	0.1	1	$\frac{225}{1001}$	$\frac{1755}{659}$	$-\frac{113}{154}$	$-\frac{3125}{56133}$	$\frac{325}{243}$	$\frac{659}{1458}$
RK4-02	$\frac{29}{25}$	0.2	1	$\frac{100}{319}$	$\frac{110}{131}$	$-\frac{41}{348}$	$-\frac{3125}{11136}$	$\frac{275}{384}$	$\frac{131}{192}$
RK4-03	$\frac{43}{50}$	0.3	1	$\frac{175}{387}$	$-\frac{105}{23}$	$\frac{19}{258}$	$\frac{3125}{6321}$	$\frac{25}{49}$	$-\frac{23}{294}$
RK4-04	$\frac{16}{25}$	0.4	1	$\frac{75}{112}$	$\frac{315}{188}$	$\frac{19}{128}$	$\frac{3125}{10368}$	$\frac{175}{432}$	$\frac{47}{324}$
RK4-05	$\frac{1}{2}$	0.5	1	1	1	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$
RK4-06	$\frac{11}{25}$	0.6	1	$\frac{50}{33}$	$\frac{35}{53}$	$\frac{7}{44}$	$\frac{3125}{7392}$	$\frac{25}{96}$	$\frac{53}{336}$
RK4-07	$\frac{23}{50}$	0.7	1	$\frac{75}{23}$	$\frac{135}{511}$	$\frac{51}{322}$	$\frac{3125}{5589}$	$\frac{25}{189}$	$\frac{73}{486}$
RK4-08	$\frac{14}{25}$	0.8	1	$-\frac{25}{14}$	$-\frac{55}{248}$	$\frac{121}{672}$	$\frac{3125}{3696}$	$-\frac{25}{96}$	$\frac{31}{132}$
RK4-09	$\frac{37}{50}$	0.9	1	$-\frac{25}{111}$	$-\frac{65}{327}$	$\frac{143}{666}$	$\frac{3125}{1443}$	$-\frac{25}{9}$	$\frac{109}{78}$

4. Numerical Simulations of EEG

In this section, we apply the proposed fourth-order RK methods as shown in Table 1 (RK4-01 to RK4-09) to the Wilson–Cowan equations, and compare their performance.

For the sigmoid function $S(x)$ in Equation (3), we set $a = 1$ and $b = 4$, following [7]. The simulation runs from $t = 0$ to $t = T$, with the computational domain having N grid points, so the step size $h = T/N$.

4.1. Wilson–Cowan Equation of Two Neuron Populations

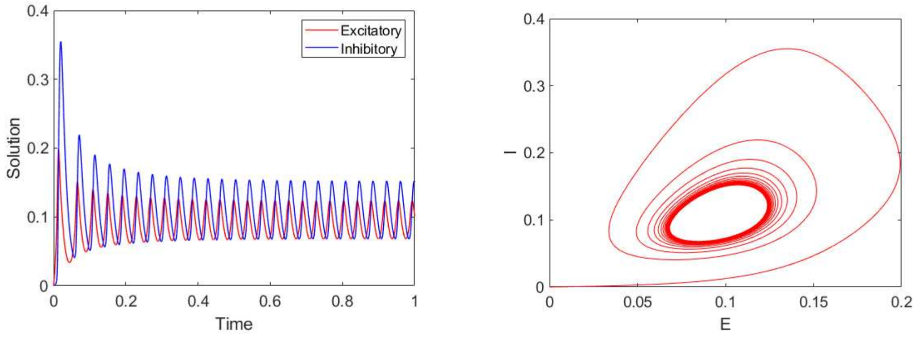
In the first test, we solve the Wilson–Cowan system (1)–(2). We set $\tau_e = \tau_i = 0.013$, $P = 1.5$, $Q = -2$, and

$$C = \begin{pmatrix} 24 & -20 \\ 40 & 0 \end{pmatrix}. \tag{67}$$

These parameters were taken from [7], where the Wilson–Cowan model was studied for epileptic dynamics and compared with clinical data. In [7], the model includes three neuron populations (one excitatory and two inhibitory populations). Ignoring the effect of the second inhibitory neuron population reduces it to a system with two neuron populations. The initial conditions are $E(0) = I(0) = 0$, and the simulation runs until time $T = 1$.

Figure 1a shows the numerical solutions where the excitatory (E) and inhibitory (I) neuron populations are functions of time. Figure 1b illustrates the E-I phase plane where the E-I curve converges to a limit cycle. A comparison of four different fourth-order RK methods is illustrated in Table 2. All methods achieve fourth-order convergence rates, measured by L_1, L_2, and L_∞ error norms. Solutions from a very fine mesh N = 32,000 are used as the exact solution. Among the methods tested in this example, RK4-03 produces the largest errors, RK4-05 and RK4-06 yield similar error magnitudes, and RK4-04 produces the smallest error.

To provide a comprehensive comparison of all methods, we list the errors when $N = 8000$ in Table 3. Among these nine methods, the RK4-04 method exhibits the best accuracy. The RK4-05 and RK4-06 methods produce error magnitudes approximately 3 to 4 times larger than that of the RK4-04 method.



(**a**) (**b**)

Figure 1. Numerical solution of the Wilson–Cowan Equations (1) and (2). (**a**) Time course of the excitatory (red curve) and the inhibitory (blue curve) neuron populations. (**b**) The excitatory-inhibitory (E-I) phase plane plot. Parameters: $\tau_e = \tau_i = 0.013$, $P = 1.5$, $Q = -2$, and C is given in Equation (67).

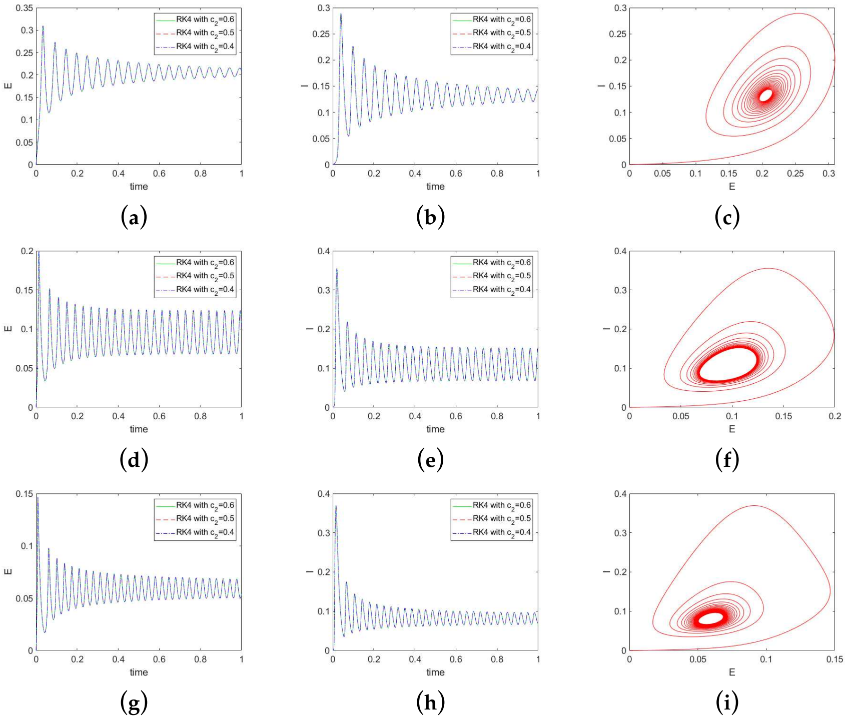
Table 2. Comparison of fourth-order RK methods when c_2 varies from 0.3 to 0.6. ϵ_1, ϵ_2 and ϵ_∞ represent the numerical errors measured in the L_1, L_2, and L_∞ norms, respectively.

Method	N	ϵ_1	Rate	ϵ_2	Rate	ϵ_∞	Rate
RK4-06 $c_2 = 0.6$	1000	7.96×10^{-5}		2.98×10^{-6}		2.12×10^{-4}	
	2000	4.60×10^{-6}	4.11	1.21×10^{-7}	4.62	1.17×10^{-5}	4.18
	4000	2.61×10^{-7}	4.14	4.87×10^{-9}	4.64	6.50×10^{-7}	4.16
	8000	1.53×10^{-8}	4.10	2.01×10^{-10}	4.60	3.76×10^{-8}	4.11
RK4-05 $c_2 = 0.5$	1000	7.90×10^{-5}		3.01×10^{-6}		2.29×10^{-4}	
	2000	4.57×10^{-6}	4.11	1.22×10^{-7}	4.63	1.26×10^{-5}	4.18
	4000	2.63×10^{-7}	4.12	4.94×10^{-9}	4.62	7.10×10^{-7}	4.15
	8000	1.55×10^{-8}	4.08	2.06×10^{-10}	4.59	4.14×10^{-8}	4.10
RK4-04 $c_2 = 0.4$	1000	2.29×10^{-5}		9.33×10^{-7}		8.38×10^{-5}	
	2000	1.32×10^{-6}	4.11	3.67×10^{-8}	4.67	4.31×10^{-6}	4.28
	4000	7.10×10^{-8}	4.22	1.38×10^{-9}	4.74	2.20×10^{-7}	4.29
	8000	3.94×10^{-9}	4.17	5.37×10^{-11}	4.68	1.19×10^{-8}	4.21
RK4-03 $c_2 = 0.3$	1000	3.79×10^{-4}		1.55×10^{-5}		1.36×10^{-3}	
	2000	2.68×10^{-5}	3.82	7.49×10^{-7}	4.38	8.70×10^{-5}	3.97
	4000	1.70×10^{-6}	3.98	3.31×10^{-8}	4.50	5.34×10^{-6}	4.03
	8000	1.05×10^{-7}	4.02	1.44×10^{-9}	4.52	3.27×10^{-7}	4.03

Next, we modify the connection matrix C by multiplying it by a factor, σ. As a result, the connection matrix σC has a larger spectral radius. Figure 2 shows the solutions (the excitatory and inhibitory neuron populations), and the phase plane plots for $\sigma = 0.5$, 1, and 1.5. The spectral radius of the original connection matrix C is $\rho = 28$, so the corresponding spectral radii of the connection matrices used in each row of this figure are 14, 28, and 42, respectively. Table 4 illustrates the results of five RK methods with c_2 varying from 0.3 to 0.7. For the two rows corresponding to $\rho = 0.1$ and $\rho = 0.5$, the smallest numerical error occurs in the fifth column where $c = 0.5$ (the RK4-05 method). On the other hand, for the other four rows where $\rho \geq 1$, the smallest errors are observed in the fourth column, which corresponds to the RK4-04 method ($c_2 = 0.4$). In all cases, the smallest numerical errors are obtained with either the RK4-05 or RK4-04 method.

Table 3. Comparison of nine fourth-order RK methods when c_2 varies from 0.1 to 0.9. $N = 8000$. ϵ_1, ϵ_2 and ϵ_∞ represent the numerical errors measured in the L_1, L_2, and L_∞ norms, respectively.

Method	c_2	ϵ_1	ϵ_2	ϵ_∞
RK4-01	0.1	1.11×10^{-8}	1.86×10^{-10}	8.91×10^{-8}
RK4-02	0.2	1.47×10^{-8}	2.18×10^{-10}	5.91×10^{-8}
RK4-03	0.3	1.05×10^{-7}	1.44×10^{-9}	3.27×10^{-7}
RK4-04	0.4	3.94×10^{-9}	5.37×10^{-11}	1.19×10^{-8}
RK4-05	0.5	1.55×10^{-8}	2.06×10^{-10}	4.14×10^{-8}
RK4-06	0.6	1.53×10^{-8}	2.01×10^{-10}	3.76×10^{-8}
RK4-07	0.7	2.70×10^{-8}	3.86×10^{-10}	9.30×10^{-8}
RK4-08	0.8	1.25×10^{-7}	1.70×10^{-9}	3.71×10^{-7}
RK4-09	0.9	5.37×10^{-8}	7.19×10^{-10}	1.50×10^{-7}



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 2. Solutions for different matrix C using three fourth-order RK methods with $c_2 = 0.6, 0.5,$ and 0.4. First row (**a**–**c**): $\sigma = 0.5$; second row (**d**–**f**): $\sigma = 1$; third row (**g**–**i**): $\sigma = 1.5$. Left column: the excitatory neural population; middle column: the inhibitory neural population; right column: the phase plane plot.

Table 4. Comparison of L_2 errors for fourth-order RK methods with different c_2 values when σ varies from 0.1 to 5. The corresponding spectral radius ρ varies from 3 to 146. $N = 8000$.

σ	ρ	$c_2 = 0.3$	$c_2 = 0.4$	$c_2 = 0.5$	$c_2 = 0.6$	$c_2 = 0.7$
0.1	3	2.62×10^{-15}	2.03×10^{-15}	2.06×10^{-15}	2.03×10^{-15}	1.78×10^{-15}
0.5	15	2.36×10^{-11}	6.25×10^{-12}	3.82×10^{-12}	3.94×10^{-12}	1.52×10^{-11}
1	29	3.09×10^{-9}	1.41×10^{-10}	4.18×10^{-10}	3.85×10^{-10}	8.40×10^{-10}
1.5	44	3.36×10^{-9}	2.81×10^{-10}	8.98×10^{-10}	1.04×10^{-9}	6.42×10^{-10}
2	58	1.72×10^{-9}	4.97×10^{-10}	1.39×10^{-9}	1.90×10^{-9}	2.31×10^{-9}
5	146	2.70×10^{-8}	1.90×10^{-9}	4.58×10^{-9}	7.05×10^{-9}	1.41×10^{-8}

Furthermore, through a series of tests, we examine the effect when one entry of the connection matrix C is amplified. To accomplish this, we modify the last entry C_{22} to a nonzero value: $C_{22} = 2$, so the connection matrix C becomes:

$$C = \begin{pmatrix} 24 & -20 \\ 40 & 2 \end{pmatrix}. \tag{68}$$

We multiply each entry C_{ij} by a factor, with values ranging from 0.1 to 5. The results shown in Tables 5–8 correspond to the case where C_{11}, C_{12}, C_{21}, and C_{22} are multiplied by factors α, β, γ, and δ, respectively. We primarily compare RK4-04 and RK4-05, as the other methods produce larger errors than these two methods for most of the test cases. From Tables 5 and 7, we observe that the RK4-05 ($c_2 = 0.5$) method produces smaller error than the RK4-04 method, except the case when $\alpha = 1$, $\gamma = 0.5$, and $\gamma = 1$. Table 6 shows a different trend, where RK4-04 achieves a smaller error than the RK4-05 method when $\beta \geq 0.5$. Table 8 illustrates that smallest errors are obtained by the RK4-04 method.

In summary, we observe that the RK4-04 method achieves smaller errors compared to the other methods when the coupling coefficient C_{12} in the excitatory equation is relatively large. Conversely, in most other cases, the classical RK4-05 method produces the smallest error among the methods tested.

Table 5. Comparison of L_2 errors for fourth-order RK methods with different c_2 values, as α varies from 0.1 to 5. $N = 8000$.

α	ρ	$c_2 = \mathbf{0.3}$	$c_2 = \mathbf{0.4}$	$c_2 = \mathbf{0.5}$	$c_2 = \mathbf{0.6}$	$c_2 = \mathbf{0.7}$
0.1	28	2.59×10^{-14}	4.40×10^{-15}	3.98×10^{-15}	4.18×10^{-15}	1.19×10^{-14}
0.5	29	4.88×10^{-13}	8.17×10^{-14}	8.48×10^{-14}	7.43×10^{-14}	1.61×10^{-13}
1	29	3.09×10^{-9}	1.41×10^{-10}	4.18×10^{-10}	3.85×10^{-10}	8.40×10^{-10}
1.5	30	4.08×10^{-11}	5.34×10^{-11}	2.45×10^{-11}	1.27×10^{-11}	3.35×10^{-11}
2	30	1.95×10^{-10}	1.61×10^{-10}	9.07×10^{-11}	5.58×10^{-11}	6.21×10^{-11}
5	113	7.08×10^{-10}	2.86×10^{-9}	1.83×10^{-9}	1.45×10^{-9}	2.33×10^{-9}

Table 6. Comparison of L_2 errors for for fourth-order RK methods with different c_2 values, as β varies from 0.1 to 5. $N = 8000$.

β	ρ	$c_2 = \mathbf{0.3}$	$c_2 = \mathbf{0.4}$	$c_2 = \mathbf{0.5}$	$c_2 = \mathbf{0.6}$	$c_2 = \mathbf{0.7}$
0.1	19	1.90×10^{-11}	1.04×10^{-11}	5.83×10^{-12}	3.28×10^{-12}	3.33×10^{-12}
0.5	21	4.26×10^{-11}	3.94×10^{-13}	1.76×10^{-12}	4.79×10^{-12}	2.04×10^{-11}
1	29	3.09×10^{-9}	1.41×10^{-10}	4.18×10^{-10}	3.85×10^{-10}	8.40×10^{-10}
1.5	35	3.45×10^{-10}	1.02×10^{-11}	3.71×10^{-11}	3.11×10^{-11}	1.07×10^{-10}
2	41	1.23×10^{-10}	5.39×10^{-12}	1.23×10^{-11}	8.81×10^{-12}	4.12×10^{-11}
5	64	7.78×10^{-12}	4.82×10^{-13}	6.23×10^{-13}	3.54×10^{-13}	2.73×10^{-12}

Table 7. Comparison of L_2 errors for fourth-order RK methods with different c_2 values, as γ varies from 0.1 to 5. $N = 8000$.

γ	ρ	$c_2 = \mathbf{0.3}$	$c_2 = \mathbf{0.4}$	$c_2 = \mathbf{0.5}$	$c_2 = \mathbf{0.6}$	$c_2 = \mathbf{0.7}$
0.1	19	2.17×10^{-11}	1.48×10^{-11}	8.61×10^{-12}	5.35×10^{-12}	5.68×10^{-12}
0.5	21	5.98×10^{-9}	7.38×10^{-11}	1.59×10^{-10}	1.21×10^{-10}	2.41×10^{-9}
1	29	3.09×10^{-9}	1.41×10^{-10}	4.18×10^{-10}	3.85×10^{-10}	8.40×10^{-10}
1.5	35	6.25×10^{-11}	3.08×10^{-12}	2.54×10^{-12}	1.39×10^{-12}	2.61×10^{-11}
2	41	1.91×10^{-11}	1.98×10^{-12}	1.35×10^{-12}	1.23×10^{-12}	8.85×10^{-12}
5	64	1.26×10^{-11}	2.70×10^{-12}	1.87×10^{-12}	1.87×10^{-12}	7.33×10^{-12}

Table 8. Comparison of L_2 errors for fourth-order RK methods with different c_2 values, as δ varies from 0.1 to 5. $N = 8000$.

δ	ρ	$c_2 = 0.3$	$c_2 = 0.4$	$c_2 = 0.5$	$c_2 = 0.6$	$c_2 = 0.7$
0.1	28.4	1.57×10^{-9}	6.10×10^{-11}	2.23×10^{-10}	2.17×10^{-10}	4.20×10^{-10}
0.5	28.7	2.17×10^{-9}	9.43×10^{-11}	3.03×10^{-10}	2.86×10^{-10}	5.84×10^{-10}
1	29.1	3.09×10^{-9}	1.41×10^{-10}	4.18×10^{-10}	3.85×10^{-10}	8.40×10^{-10}
1.5	29.5	4.18×10^{-9}	1.88×10^{-10}	5.47×10^{-10}	4.96×10^{-10}	1.16×10^{-9}
2	29.9	5.50×10^{-9}	2.34×10^{-10}	6.93×10^{-10}	6.18×10^{-10}	1.55×10^{-9}
5	32.2	2.60×10^{-11}	1.92×10^{-12}	2.06×10^{-11}	2.89×10^{-11}	1.92×10^{-11}

4.2. Wilson–Cowan Equation of Three Neuron Populations

In this section, we apply our methods to the Wilson–Cowan system (5)–(7) for epileptic dynamics. We conduct a series of simulations with parameters obtained from [7], as listed in Table 9. In [7], these simulations are shown to exhibit good agreement with the clinical EEG recordings. Unless otherwise stated, the initial condition is $E(0) = I(0) = J(0) = 0$, and the simulations run until $T = 3$. High-resolution solutions with $N = 128,000$ are used as exact solutions when computing numerical errors.

Table 9. Parameters for the Wilson–Cowan model for epileptic dynamic simulations [7].

Test	Figure in [7]	(τ_e, τ_i, τ_j)	(P, Q, R)	C
T01	3c	$(0.013, 0.013, 0.267)$	$(3, -2, 0)$	$\begin{pmatrix} 24 & -20 & -15 \\ 40 & 0 & 0 \\ 7 & 0 & 0 \end{pmatrix}$
T02	3f	$(0.015, 0.013, 0.267)$	$(0.5, -5, -5)$	$\begin{pmatrix} 23 & -15 & -10 \\ 35 & 0 & 0 \\ 10 & 0 & 0 \end{pmatrix}$
T03	3i	$(0.0225, 0.03, 0.12)$	$(4, -5, -3)$	$\begin{pmatrix} 25 & -15 & -10 \\ 35 & 0 & 0 \\ 10 & 0 & 0 \end{pmatrix}$
T04	3l	$(0.015, 0.013, 0.267)$	$(3, -5, -5)$	$\begin{pmatrix} 23 & -15 & -10 \\ 35 & 0 & 0 \\ 10 & 0 & 0 \end{pmatrix}$
T05	4	$(0.013, 0.013, 0.267)$	$(5, -2, 0)$	$\begin{pmatrix} 38 & -29 & -10 \\ 40 & 0 & 0 \\ 20 & 0 & 0 \end{pmatrix}$
T06	5d	$(0.017, 0.017, 0.25)$	$(5, -2, 0)$	$\begin{pmatrix} 38 & -29 & -10 \\ 40 & 0 & 0 \\ 6 & 0 & 0 \end{pmatrix}$
T07	5e	$(0.017, 0.017, 0.25)$	$(5, -2, 0)$	$\begin{pmatrix} 38 & -29 & -10 \\ 40 & 0 & 0 \\ 15 & 0 & 0 \end{pmatrix}$
T08	7	$(0.013, 0.013, 0.267)$	$(-0.5,-5,0)$	$\begin{pmatrix} 35 & -30 & -10 \\ 40 & 0 & 0 \\ 15 & 0 & 0 \end{pmatrix}$

First, single-spike waves (test cases: T01 through T03) and poly-spike waves (test cases: T04 through T07) are plotted in Figures 3 and 4, respectively. These results align with the simulations and the clinical data [7]. We then compare four methods, RK4-03 through RK4-06, by examining their L_2 error norms as c_2 varies from 0.3 to 0.6, with $N = 32,000$. In Table 10, we list the L_2 error norms for the seven tests. The RK4-05 method ($c_2 = 0.5$) achieves the smallest errors for most of the cases (T02, T04-07). In contrast, the RK4-04 method ($c_2 = 0.4$) achieves the smallest errors for the first test case T01. We also see that the RK4-06 method has the smallest error for the case T03.

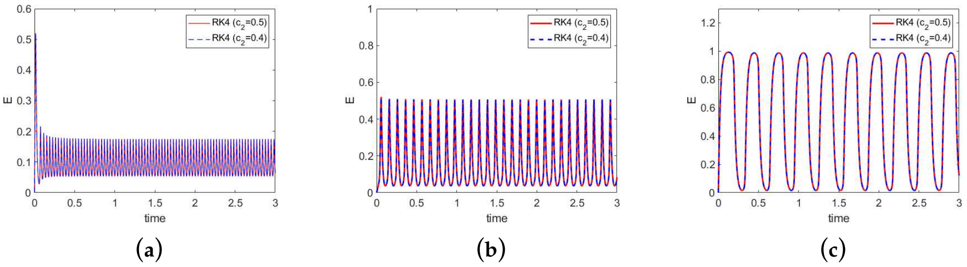


Figure 3. Excitatory populations for single-spike waves: (**a**) T01, (**b**) T02, and (**c**) T03. Results of RK4-04 ($c_2 = 0.4$) and RK4-05 ($c_2 = 0.5$) are plotted in blue and red, respectively.

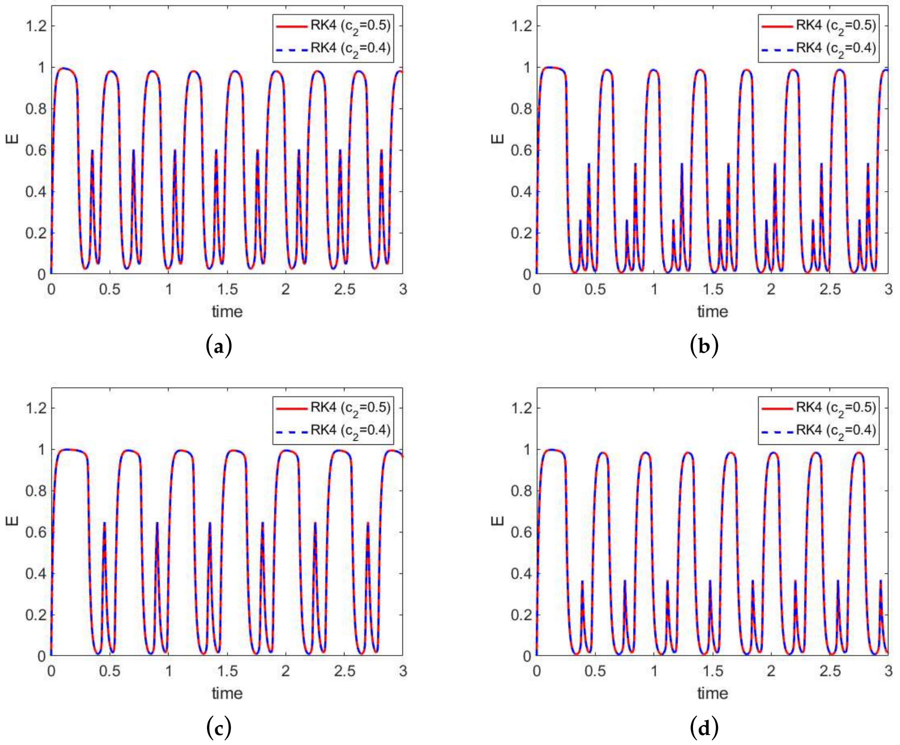


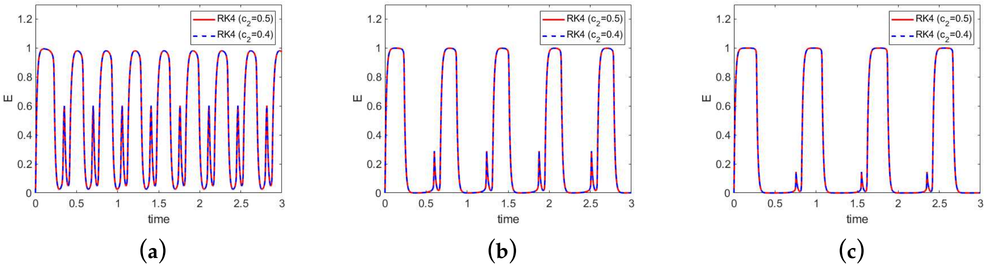
Figure 4. Excitatory populations for poly-spike waves: (**a**) T04, (**b**) T05, (**c**) T06, and (**b**) T07. Results of RK4-04 ($c_2 = 0.4$) and RK4-05 ($c_2 = 0.5$) are plotted in blue and red, respectively.

Table 10. L_2 errors of fourth-order RK methods with different c_2 values for tests T01 through T07.

Test	T01	T02	T03	T04	T05	T06	T07
$c_2 = 0.3$	1.17×10^{-8}	5.78×10^{-9}	2.23×10^{-11}	1.61×10^{-9}	4.56×10^{-8}	4.36×10^{-8}	6.62×10^{-9}
$c_2 = 0.4$	5.81×10^{-10}	7.87×10^{-10}	1.52×10^{-11}	1.84×10^{-10}	4.89×10^{-9}	5.58×10^{-9}	6.04×10^{-10}
$c_2 = 0.5$	1.80×10^{-9}	4.22×10^{-10}	8.10×10^{-12}	8.97×10^{-11}	4.05×10^{-9}	4.13×10^{-9}	5.26×10^{-10}
$c_2 = 0.6$	1.77×10^{-9}	6.14×10^{-10}	4.48×10^{-12}	1.44×10^{-10}	6.40×10^{-9}	6.14×10^{-9}	8.83×10^{-10}

Next, focusing on test case T04, we multiply either the connection matrix by a factor σ or the coupling coefficient C_{12} by a factor β, with both σ and β varying from 1 to 3. The results of σ-dependent solutions are shown in Figure 5 and Table 11. Table 12 shows the β-dependent results. For relatively small σ or β, the RK4-05 method achieves smaller error than the other methods. However, for relatively large σ or β, the RK4-04 method produces the smallest errors among the four methods.

A similar observation can be made for a series of tests based on T07 (Figure 6), where the connection matrix is multiplied by σ, with σ taking values 1, 1.5, and 2. The results are illustrated in Table 13. When $\sigma = 1$, the error of the RK4-05 method is about 15% smaller than that of the RK4-04 method. However, when $\sigma = 1.5$ and 2, the errors of RK4-04 method are 20% and 40% smaller than the error of the RK4-05 method, respectively.



(a) (b) (c)

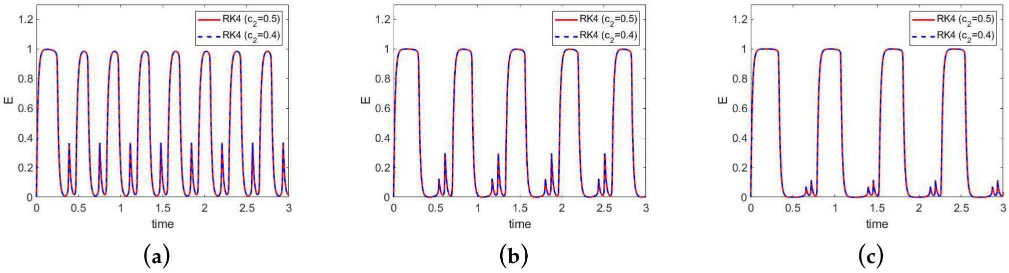
Figure 5. Excitatory populations for test T04 with connection matrix σC, where (**a**) $\sigma = 1$, (**b**) $\sigma = 2$, and (**c**) $\sigma = 3$. Results of RK4-04 ($c_2 = 0.4$) and RK4-05 ($c_2 = 0.5$) are plotted in blue and red, respectively.

Table 11. L_2 errors of fourth-order RK methods for test T04 with connection matrix σC, where σ varies from 1 to 3.

Method	$\boldsymbol{\sigma = 1}$	$\boldsymbol{\sigma = 2}$	$\boldsymbol{\sigma = 3}$
RK4-03 ($c_2 = 0.3$)	1.61×10^{-9}	1.99×10^{-8}	3.15×10^{-8}
RK4-04 ($c_2 = 0.4$)	1.84×10^{-10}	2.46×10^{-9}	3.20×10^{-9}
RK4-05 ($c_2 = 0.5$)	8.97×10^{-11}	2.13×10^{-9}	3.99×10^{-9}
RK4-06 ($c_2 = 0.6$)	1.44×10^{-10}	3.19×10^{-9}	6.21×10^{-9}

Table 12. L_2 errors of fourth-order RK methods for test T04 with coupling coefficient βC_{12}, where β varies from 1 to 3.

Method	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$
RK4-03 ($c_2 = 0.3$)	1.61×10^{-9}	2.30×10^{-8}	6.84×10^{-8}
RK4-04 ($c_2 = 0.4$)	1.84×10^{-10}	5.45×10^{-11}	3.61×10^{-9}
RK4-05 ($c_2 = 0.5$)	8.97×10^{-11}	4.55×10^{-10}	5.63×10^{-9}
RK4-06 ($c_2 = 0.6$)	1.44×10^{-10}	1.91×10^{-9}	1.06×10^{-8}



(a) (b) (c)

Figure 6. Excitatory populations of test T07 with connection matrix σC, where (**a**) $\sigma = 1$, (**b**) $\sigma = 1.5$, and (**c**) $\sigma = 2$. Results of RK4-04 ($c_2 = 0.4$) and RK4-05 ($c_2 = 0.5$) are plotted in blue and red, respectively.

Table 13. L_2 errors of fourth-order RK methods for test T07 with connection matrix σC, where σ varies from 1 to 2.

Method	$\boldsymbol{\sigma = 1}$	$\boldsymbol{\sigma = 1.5}$	$\boldsymbol{\sigma = 2}$
RK4-03 ($c_2 = 0.3$)	6.62×10^{-9}	2.26×10^{-8}	2.19×10^{-9}
RK4-04 ($c_2 = 0.4$)	6.04×10^{-10}	2.57×10^{-9}	5.07×10^{-10}
RK4-05 ($c_2 = 0.5$)	5.26×10^{-10}	3.31×10^{-9}	9.02×10^{-10}
RK4-06 ($c_2 = 0.6$)	8.83×10^{-10}	5.02×10^{-9}	1.52×10^{-9}

The last test is based on the parameters of T05, where the connection matrix is σC, and P gradually increases from 0 to 5. The simulations run until $T = 5$. The P value increases to its maximum value when time reaches $T/2$, making P time-dependent and computed using the following formula:

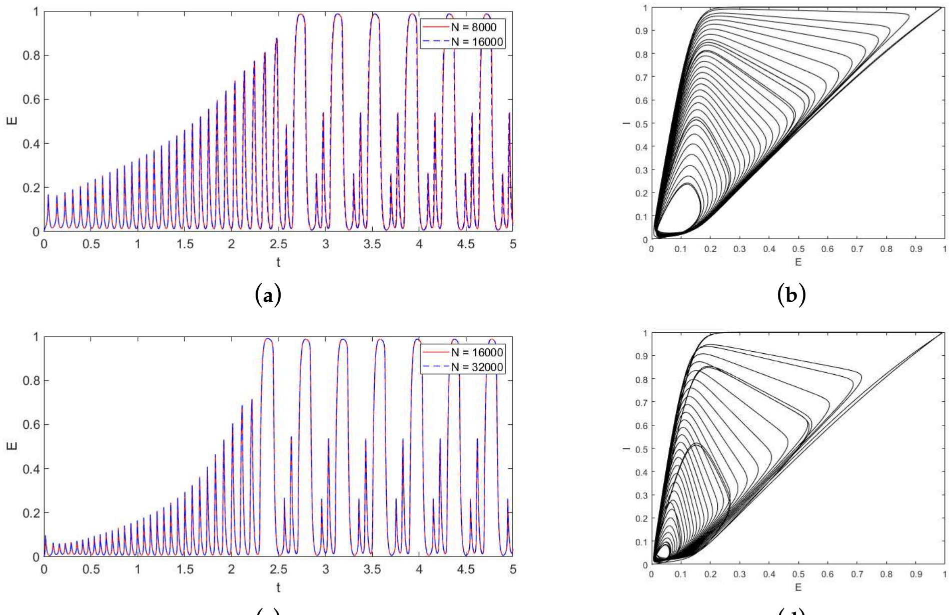
$$P(t) = \begin{cases} P_1 + 2(P_2 - P_1)t/T, & t < T/2 \\ P_2, & t \geq T/2, \end{cases} \tag{69}$$

where $P_1 = 0$ and $P_2 = 5$. We try two cases of σ, one where it is fixed at 1, and another where it decreases from 2 to 1 using the following formula:

$$\sigma(t) = \begin{cases} \sigma_1 + 2(\sigma_2 - \sigma_1)t/T, & t < T/2 \\ \sigma_2, & t \geq T/2, \end{cases} \tag{70}$$

where $\sigma_1 = 2$ and $\sigma_2 = 1$.

The plots of excitatory neuron populations and the phase plane plot are illustrated in Figure 7. Results at two resolutions, $N = 16{,}000$ and $N = 32{,}000$, are plotted, and show no visible difference, indicating that the solution has achieved reasonable accuracy. As P increases from 0 to 5, the wave transitions from single-spike to poly-spike wave, regardless of whether σ is fixed at 1 or gradually decreases from 2 to 1. For a quantitative error analysis, Table 14 presents the L_2 error norms of four methods (RK4-03 to RK4-06) when $N = 32{,}000$. Similarly to the previous tests, the RK4-04 and RK4-05 methods produce smaller errors than the other methods. The error of the RK4-05 method is about 25% smaller than that of the RK4-04 method when σ is fixed. In contrast, when σ is time-varying, the error of the RK4-04 method is about 30% smaller than that of the RK4-05 method.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. Excitatory populations and the phase plane plots for test T05 with P gradually increases from 0 to 5. The connection matrix is σC. Upper row (**a**,**b**): σ is fixed at 1; lower row (**c**,**d**): σ gradually decreases from 2 to 1. Left column (**a**,**c**): excitatory neuron populations. Results of two resolutions N = 16,000 and N = 32,000 are plotted in red and blue, respectively. Right column (**b**,**d**): phase plane plots.

Table 14. L_2 errors of fourth-order RK methods for test T05, as P gradually increases from 0 to 5. σ is fixed at 1 or gradually decreases from 2 to 1.

Method	$\sigma = 1$	σ Decreases from 2 to 1
RK4-03 ($c_2 = 0.3$)	2.19×10^{-6}	1.82×10^{-6}
RK4-04 ($c_2 = 0.4$)	2.70×10^{-7}	1.89×10^{-7}
RK4-05 ($c_2 = 0.5$)	2.18×10^{-7}	3.06×10^{-7}
RK4-06 ($c_2 = 0.6$)	3.26×10^{-7}	4.65×10^{-7}

5. Discussion

The Wilson–Cowan model is a system of nonlinear differential equations for which an analytical solution is not available, so it is traditionally solved using numerical methods. In particular, the classical explicit fourth-order Runge–Kutta (RK4) method is one of the most popular approaches. The four-stage, fourth-order RK methods we developed in this work can be considered a generalization of the classical RK4 method by introducing a free parameter c_2. The classical fourth-order RK method corresponds to the case where $c_2 = 0.5$.

Our work was motivated by recent developments in generalized iterated Crank–Nicolson (ICN) algorithms, which achieve up to third-order accuracy [20]. We modified the final update equation of the four-stage ICN method by introducing an additional coefficient, enabling fourth-order accuracy. The order conditions (38)–(45) result in one free variable, c_2, from which the other coefficients can be calculated explicitly. In Table 1, we list nine sets of RK coefficients where c_2 varies from 0.1 to 0.9 in increments of 0.1. To our knowledge, this is the first time these fourth-order RK methods (with $c_2 \neq 0.5$) have been proposed.

The introduction of a free parameter allows for the selection of the most efficient numerical solver by comparing the performance of the methods with different parameter values. One notable observation is that the accuracy of the methods is sensitive to the connection matrix C, particularly to its spectral radius and the magnitude of the excitation-inhibition coupling coefficient C_{12}. In most simulations, the classical fourth-order RK method ($c_2 = 0.5$) produced smaller errors than the other methods tested, demonstrating its robustness. However, in a few test cases where the spectral radii or coupling coefficients were relatively large, the RK4-04 method ($c_2 = 0.4$) demonstrated superior accuracy compared to the other methods tested, including the classical fourth-order RK method. For instance, in one of the test cases, as shown in Table 4, the numerical error of the RK4-04 method was less than half that of the classical method when the spectral radius exceeded 100.

6. Conclusions

In this paper, we have developed a class of four-stage, fourth-order Runge–Kutta (RK) methods based on a generalized iterated Crank–Nicolson procedure. The coefficients of the proposed RK methods are derived as functions of a free parameter c_2, with the classical fourth-order RK (RK4) algorithm being a special case when $c_2 = 0.5$. We use the notation RK4-0x to denote the fourth-order RK method when $c_2 = 0.x$. Specifically, the RK4-05 method corresponds to the classical RK4 method. This free parameter provides a convenient way to compare the performance of different RK methods within this framework.

Focusing on the Wilson–Cowan systems with two and three neuron populations modeling EEG epileptic dynamics, we conducted a series of numerical simulations to evaluate the performance of the proposed methods in comparison with the classical RK4 (RK4-05) method. Our simulations included both single-spike and poly-spike waveforms, as well as the transitions from single-spike to poly-spike waves. The results confirmed that the proposed methods achieve fourth-order accuracy, and the simulated EEG waves align with those reported by other research groups and the clinical data. In particular, the RK4-05 ($c_2 = 0.5$) and RK4-04 ($c_2 = 0.4$) methods produced smaller errors than the other methods tested. The comparison of the RK4-04 and the RK4-05 methods illustrated that their numerical errors are sensitive to the spectral radius of the connection matrix and the excitation-inhibition coupling coefficient.

Our current analysis is based on numerical simulations focused on a specific type of EEG wave associated with epileptic dynamics. In the future, we plan to extend our work to other EEG applications, including insomnia and sleep regulation.

Author Contributions: Methodology, J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L., Q.L., F.B. and H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by USAF/UARC/RITA Grant number FA955023D0001, NSF Grant 1955664, NSF Grant 2219731, and NIH Grant P20GM103446.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wilson, H.R.; Cowan, J.D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* **1972**, *12*, 1–24. [CrossRef] [PubMed]
2. Wilson, H.R.; Cowan, J.D. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* **1973**, *13*, 55–80. [CrossRef] [PubMed]
3. Wilson, H.R.; Cowan, J.D. Evolution of the Wilson–Cowan equations. *Biol. Cybern.* **2021**, *115*, 643–653. [CrossRef] [PubMed]
4. Sakaguchi, H. Oscillatory and excitable behaviors in a population of model neurons. *Prog. Theor. Phys.* **1988**, *79*, 1061–1068. [CrossRef]
5. Borisyuk, R.M.; Kirillov, A.B. Bifurcation analysis of a neural network model. *Biol. Cybern.* **1992**, *66*, 319–325. [CrossRef]
6. Monteiro, L.; Bussab, M.; Berlinck, J.C. Analytical results on a Wilson-Cowan neuronal network modified model. *J. Theor. Biol.* **2002**, *219*, 83–91. [CrossRef]
7. Wang, Y.; Goodfellow, M.; Taylor, P.N.; Baier, G. Phase space approach for modeling of epileptic dynamics. *Phys. Rev. E* **2012**, *85*, 061918. [CrossRef]
8. Cowan, J.D.; Neuman, J.; van Drongelen, W. Wilson–Cowan equations for neocortical dynamics. *J. Math. Neurosci.* **2016**, *6*, 1. [CrossRef]
9. Xie, R. Mathematically Modeling the Neuron Network Involved in Sleep Regulation. Ph.D. Thesis, Brandeis University: Waltham, MA, USA, 2021.
10. Crank, J.; Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*; Cambridge Univ Press: Cambridge, UK, 1947; Volume 43, pp. 50–67.
11. Teukolsky, S.A. Stability of the iterated Crank-Nicholson method in numerical relativity. *Phys. Rev. D* **2000**, *61*, 087501. [CrossRef]
12. Duez, M.D.; Marronetti, P.; Shapiro, S.L.; Baumgarte, T.W. Hydrodynamic simulations in 3+ 1 general relativity. *Phys. Rev. D* **2003**, *67*, 024004. [CrossRef]
13. Duez, M.D.; Liu, Y.T.; Shapiro, S.L.; Stephens, B.C. General relativistic hydrodynamics with viscosity: Contraction, catastrophic collapse, and disk formation in hypermassive neutron stars. *Phys. Rev. D* **2004**, *69*, 104030. [CrossRef]
14. Xu, P.; Liu, J. Iteration-Based Temporal Subgridding Method for the Finite-Difference Time-Domain Algorithm. *Mathematics* **2024**, *12*, 302. [CrossRef]
15. Yioultsis, T.V.; Ziogos, G.D.; Kriezis, E.E. Explicit finite-difference vector beam propagation method based on the iterated Crank-Nicolson scheme. *JOSA A* **2009**, *26*, 2183–2191. [CrossRef] [PubMed]
16. Ketzaki, D.A.; Rekanos, I.T.; Kosmanis, T.I.; Yioultsis, T.V. Beam Propagation Method Based on the Iterated Crank–Nicolson Scheme for Solving Large-Scale Wave Propagation Problems. *IEEE Trans. Magn.* **2015**, *51*, 7204404. [CrossRef]
17. Shibayama, J.; Nishio, T.; Yamauchi, J.; Nakano, H. Explicit FDTD method based on iterated Crank–Nicolson scheme. *Electron. Lett.* **2022**, *58*, 16–18. [CrossRef]
18. Wu, P.; Wang, X.; Xie, Y.; Jiang, H.; Natsuki, T. Iterated Crank-Nicolson Procedure with Enhanced Absorption for Nonuniform Domains. *IEEE J. Multiscale Multiphysics Comput. Tech.* **2022**, *7*, 61–68. [CrossRef]
19. Tran, Q.; Liu, J. Modified iterated Crank-Nicolson method with improved accuracy for advection equations. *Numer. Algorithms* **2024**, *95*, 1539–1560. [CrossRef]
20. Liu, J.; Appiah-Adjei, S.; Brio, M. Iterated Crank–Nicolson Method for Peridynamic Models. *Dynamics* **2024**, *4*, 192–207. [CrossRef]
21. Butcher, J. Runge-kutta methods. *Scholarpedia* **2007**, *2*, 3147. [CrossRef]
22. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
23. Burden, R.L. *Numerical Analysis*; Brooks/Cole Cengage Learning: Boston, MA, USA, 2011.

24. Gottlieb, S.; Shu, C.W. Total variation diminishing Runge-Kutta schemes. *Math. Comput.* **1998**, *67*, 73–85. [CrossRef]
25. Gottlieb, S.; Shu, C.W.; Tadmor, E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **2001**, *43*, 89–112. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.