


Article

Rosenbrock Type Methods for Solving Non-Linear Second-Order in Time Problems

Maria Jesus Moreta 

IMUVA. Departamento de Análisis Económico y Economía Cuantitativa, Campus de Somosaguas, Universidad Complutense de Madrid, Pozuelo de Alarcón, 28223 Madrid, Spain; mjesusmoreta@ccee.ucm.es

Abstract: In this work, we develop a new class of methods which have been created in order to numerically solve non-linear second-order in time problems in an efficient way. These methods are of the Rosenbrock type, and they can be seen as a generalization of these methods when they are applied to second-order in time problems which have been previously transformed into first-order in time problems. As they also follow the ideas of Runge–Kutta–Nyström methods when solving second-order in time problems, we have called them Rosenbrock–Nyström methods. When solving non-linear problems, Rosenbrock–Nyström methods present less computational cost than implicit Runge–Kutta–Nyström ones, as the non-linear systems which arise at every intermediate stage when Runge–Kutta–Nyström methods are used are replaced with sequences of linear ones.

Keywords: second-order in time problems; non-linear problems; Rosenbrock type methods



Citation: Moreta, M.J. Rosenbrock Type Methods for Solving Non-Linear Second-Order in Time Problems. *Mathematics* **2021**, *9*, 2225. <https://doi.org/10.3390/math9182225>

Academic Editor: Stefano De Marchi

Received: 30 July 2021

Accepted: 6 September 2021

Published: 10 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the literature, there are many non-linear second-order in time problems which must be numerically solved by means of appropriate methods. It is usual to use numerical methods that have been designed to numerically solve first-order in time problems by transforming the initial problem into a first-order one. In this way, we have several options to choose from, such as Runge–Kutta methods (RK methods), fractional step Runge–Kutta methods (FSRK methods) or exponential integrators. These methods are a good option if the problem we have is a linear one but they present a high computational cost when the problem we are solving is a non-linear one, as we have to solve non-linear problems at every intermediate stage (see [1–4]). In this case, a good option is to use Rosenbrock methods or exponential Rosenbrock type methods [5–7], for example. However, in this case, when converting the original problem to a first-order one, the dimension of the problem doubles, so the computational cost increases.

Another possibility is to use numerical integrators specially derived to numerically solve second-order in time problems. In this way, we can use, for example, Runge–Kutta–Nyström methods (RKN methods) or fractional step Runge–Kutta–Nyström methods (FSRKN methods) [2,8,9]. When we use this type of method, we have to choose whether to use explicit or implicit methods. When we use explicit methods, there may be stability problems when the problem we are solving is a stiff one. On the other hand, if we choose an implicit method, we can select a method with an infinite stability interval, but with a high computational cost [8] when the problem is non-linear or/and multidimensional in space. In order to avoid the high computational cost that implicit RKN methods present when multidimensional problems in space are solved, FSRKN methods were developed and studied in [9]. The idea of these methods is to split the spatial operator in a suitable way so that at every intermediate stage the problem to be solved is simpler in a certain way than the original one.

In order to avoid all the previous drawbacks when solving a non-linear second-order in time problem, in this paper, we present a new class of methods, which we call Rosenbrock–Nyström methods. These methods avoid the non-linear systems which arise when RKN

methods are used by replacing them with sequences of linear ones. Rosenbrock–Nyström methods arise in a natural way as a generalization of Rosenbrock ones when they are applied to second-order in time equations that have been previously transformed into first-order in time problems.

In the literature, the construction of methods of the Rosenbrock type to numerically solve second-order non-linear systems of ordinary differential equations has already been studied [10]. However, the methods presented in that paper differ from the ones presented here. We remark here on three of the most important differences between the methods in [10] and our methods when solving a problem such as $y''(t) = f(t, y)$, $y(0) = y_0$, $y'(0) = v_0$, with $0 \leq t \leq T < \infty$. The first one is that to use the methods presented in [10], we have to define $u = (y, v, t)^T$, with $v = y'(t)$ and then we have to convert the problem to a first-order one in the following way: $u' = g(y) = (v, f(t, y), 1)^T$, $u(0) = (y_0, v_0, t_0)^T$, while with our methods we do not need to convert the problem to a first-order one. The second difference is that in [10], we have to solve two linear systems at each intermediate step, instead of the one linear system we have to solve at each intermediate step when using the methods presented here. The last one is that in the mentioned paper, operator $g(y)$ and its Jacobian have to be evaluated at every intermediate stage. When using our method, we evaluate operator $f(t, y)$ at every intermediate stage, but we use fixed values of $f_t(t, y)$ and $f_y(t, y)$ at every time step, so the number of function evaluations per intermediate stage is smaller with our methods.

In this article, we show the development of Rosenbrock–Nyström methods, as well as the conditions that must be satisfied to obtain the desired classical order (up to order four) and the main ideas in order to have stability when linear problems are solved. In addition, we show some numerical experiments that prove the good behavior of these new methods.

This paper is structured as follows: in the next section, we give a brief description of Rosenbrock–Nyström methods, together with their development. In Section 3, we describe the stability requirements that these methods should satisfy when integrating linear problems. In Section 4, we deal with the conditions that the coefficients of the method should satisfy in order to have up to classical order four. The construction of such methods is studied in [11]. Finally, in Section 5, we present some numerical experiments in order to test such methods.

2. Development of Rosenbrock–Nyström Methods

Non-linear second-order in time problems can be written in an abstract form as follows: “Find $y : [0, T] \rightarrow \mathcal{H}$ solution of

$$\begin{cases} y''(t) &= f(t, y(t)), & 0 \leq t \leq T < \infty, \\ y(0) &= y_0, \\ y'(0) &= v_0, \end{cases} \quad (1)$$

where, typically, \mathcal{H} is a Hilbert space of functions defined in a certain bounded domain $\Omega \subseteq \mathbb{R}^M$, integer $M \geq 1$ with smooth boundary Γ . This formulation involves lots of different problems: partial differential equations, ordinary differential equations, etc.

Example 1. Let us show here two problems that will be solved in the numerical experiments (Section 5) and can be solved by using Rosenbrock–Nyström methods.

The first problem is a modification of the non-linear wave propagation suggested in [12]

$$\begin{aligned} u_j''(t) &= F(u_{j+1} - u_j) - F(u_j - u_{j-1}) + g_j(t), & j = 1, \dots, N, \\ u_j(0) &= \sin\left(\frac{2\pi j}{N+1}\right), & j = 1, \dots, N, \\ u_j'(t) &= 0 & j = 1, \dots, N, \end{aligned} \quad (2)$$

with

$$\begin{aligned} F(u) &= \lambda u + \alpha u^p, \\ u_0(t) &= u_{N+1}(t) = 0, \end{aligned}$$

$g_j(t)$ is such that the exact solution is given by $u_j(t) = \sin\left(\frac{2\pi j}{N+1}\right)\cos(t)$. Parameter λ controls the stiffness of the problem.

Example 2. The second example is the following non-linear Euler–Bernoulli equation:

$$\begin{cases} u_{tt} = -Au^2 + h(t, x), & x \in [-1, 1] & 0 \leq t \leq T = 1, \\ u(0, x) = (x^2 - 1)^3 \cos(x), & x \in [-1, 1], \\ u_t(0, x) = -(x^2 - 1)^3 \sin(x), & x \in [-1, 1], \\ u(t, -1) = u(t, 1) = 0, & 0 \leq t \leq T = 1 \\ u_x(t, -1) = u_x(t, 1) = 0, & 0 \leq t \leq T = 1 \end{cases} \quad (3)$$

with $h(t, x)$ such that the exact solution is $u(t, x) = (x^2 - 1)^3 \cos(t + x)$. Operator A is such that $Au = u_{xxxx}$. This problem can be discretized in space by taking, for example, a pseudo-spectral discretization.

When solving this problem with an explicit Runge–Kutta–Nyström method, we have to impose very restrictive hypotheses over the time step size in order to guarantee stability.

When we solve a problem such as (1) with a Rosenbrock–Nyström method, the numerical approximation to the exact solution and its derivative, $(y(t_n), v(t_n))$ is given by (y_n, v_n) , where $t_n = t_0 + n\tau$, with τ as the time step size. The numerical approximation (y_n, v_n) proposed by us is calculated as

$$\begin{aligned} v_{n+1} &= v_n + \tau \sum_{i=1}^s b_i f(t_n + \alpha_i \tau, y_n + \sum_{j=1}^{i-1} \alpha_{ij} K_{n,j}) \\ &\quad + \tau^2 (\beta^T \cdot e) f_t(t_n, y_n) + \tau f_y(t_n, y_n) \sum_{i=1}^s \beta_i K_{n,i}, \\ y_{n+1} &= y_n + \sum_{i=1}^s b_i K_{n,i}, \end{aligned}$$

where $K_{n,i}, i = 1, \dots, s$ are the intermediate stages and $e = (1, \dots, 1)^T$. These intermediate stages are given by

$$\begin{aligned} K_{n,i} &= \tau v_n + \tau^2 \sum_{j=1}^i \delta_{ij} f(t_n + \alpha_j \tau, y_n + \sum_{l=1}^{j-1} \alpha_{jl} K_{n,l}) \\ &\quad + \tau^2 \sum_{j=1}^i \gamma_{ij} (\tau f_t(t_n, y_n) + f_y(t_n, y_n) K_{n,j}) \end{aligned} \quad (4)$$

We use notation $f_t(t, y)$ and $f_y(t, y)$ to refer to $\frac{\partial f}{\partial t}(t, y)$ and $\frac{\partial f}{\partial y}(t, y)$, respectively. When the problem we are solving is autonomous, that is, of the form

$$\begin{cases} y''(t) &= f(y), \\ y(0) &= y_0, \\ y'(0) &= v_0, \end{cases}$$

methods applied to problem (1) when it is transformed to a first-order in time problem. Let us remember that the coefficients of Rosenbrock methods are given by an array of the form

$$\begin{array}{c|c|c|c} \tilde{\alpha} & \tilde{A}_\alpha & \tilde{\gamma} & \tilde{A}_\gamma \\ \hline & \tilde{b}^T & & \end{array} = \begin{array}{c|c|c|c|c} \tilde{\alpha}_1 & 0 & & & \tilde{\gamma}_1 & \tilde{\gamma}_{11} \\ \tilde{\alpha}_2 & \tilde{\alpha}_{21} & \ddots & & \tilde{\gamma}_2 & \tilde{\gamma}_{21} & \ddots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\ \tilde{\alpha}_s & \tilde{\alpha}_{s1} & \dots & \tilde{\alpha}_{ss-1} & 0 & \tilde{\gamma}_s & \tilde{\gamma}_{s1} & \dots & \tilde{\gamma}_{ss-1} & \tilde{\gamma}_{ss} \\ \hline & \tilde{b}_1 & \dots & & \tilde{b}_s & & & & & \end{array},$$

and the equations that these methods give when solving a problem such as

$$\begin{cases} u'(t) = F(t, u), \\ u(0) = u_0, \end{cases} \tag{7}$$

are

$$Q_{n,i} = \tau F(t_n + \tilde{\alpha}_i \tau, u_n + \sum_{j=1}^{i-1} \tilde{\alpha}_{ij} Q_{n,j}) + \tau^2 \tilde{\gamma}_i F_t(t_n, u_n) + \tau F_y(t_n, u_n) \sum_{j=1}^i \tilde{\gamma}_{ij} Q_{n,j},$$

$$u_{n+1} = u_n + \sum_{j=1}^s \tilde{b}_j Q_{n,j},$$

where $t_n = t_0 + n\tau$, with τ as the time step size and u_n as the numerical approximation to $u(t_n)$. $Q_{n,i}$, $i = 1, \dots, s$, are the intermediate stages.

To solve problem (1) with a Rosenbrock method, we first write it as a first-order one, by defining $v(t) = y'(t)$. In this way, we define $u(t) = (y(t), v(t))^T$ and, therefore, we have a problem like (7), being $F(t, u) = (v(t), f(t, y))^T$. The initial condition is $u(0) = u_0$ with $u_0 = (y_0, v_0)^T$. We apply a Rosenbrock method to this problem, by using notation $Q_{n,i} = (Q_{n,i}^y, Q_{n,i}^v)$.

We operate in the equations that give the intermediate stages, replacing $Q_{n,j}^v$ with its expression in the equations for $Q_{n,j}^y$. From this, we obtain

$$Q_{n,i}^y = \tau v_n + \tau^2 \sum_{j=1}^i (\tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}) f(t_n + \tilde{\alpha}_j \tau, y_n + \sum_{l=1}^{j-1} \tilde{\alpha}_{jl} Q_{n,l}^y) + \tau^3 \sum_{j=1}^i (\tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}) \tilde{\gamma}_j f_t(t_n, y_n)$$

$$+ \tau^2 \sum_{j=1}^i (\tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}) \sum_{l=1}^j \tilde{\gamma}_{lj} Q_{n,l}^y,$$

$$v_{n+1} = v_n + \tau \sum_{j=1}^s \tilde{b}_j f(t_n + \tilde{\alpha}_j \tau, y_n + \sum_{l=1}^{j-1} \tilde{\alpha}_{jl} Q_{n,l}^y)$$

$$+ \tau^2 \sum_{j=1}^s \tilde{b}_j \tilde{\gamma}_j f_t(t_n, y_n) + \tau f_y(t_n, y_n) \sum_{j=1}^s \tilde{b}_j \sum_{l=1}^j \tilde{\gamma}_{jl},$$

$$y_{n+1} = y_n + \sum_{j=1}^s \tilde{b}_j Q_{n,j}^y$$

Now, let us assume that the Rosenbrock method satisfies

$$\tilde{\alpha}_i = \sum_{j=1}^{i-1} \tilde{\alpha}_{ij}, \quad \tilde{\gamma}_i = \sum_{j=1}^i \tilde{\gamma}_{ij}, \quad \sum_{j=1}^s \tilde{b}_j = 1.$$

The first two conditions are usual restrictions satisfied for many Rosenbrock methods and the third one is the condition to have classical order 1. Then, if we denote $K_{n,i} = Q_{n,i}^y$, $i = 1, \dots, s$ and if we define

$$\begin{aligned}
 \alpha &= \tilde{\alpha}, \\
 \mathcal{A}_\alpha &= \tilde{\mathcal{A}}_\alpha \\
 \mathcal{A}_\gamma &= (\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)\tilde{\mathcal{A}}_\gamma, \\
 \mathcal{A}_\delta &= \tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma, \\
 b^T &= \tilde{b}^T, \\
 \beta^T &= \tilde{b}^T \tilde{\mathcal{A}}_\gamma,
 \end{aligned}
 \tag{8}$$

what we obtain is precisely Equation (4). Furthermore, we can construct Rosenbrock–Nyström methods that do not come from Rosenbrock ones. This fact gives much more freedom to obtain the coefficients of the desired methods. The proposed methods with classical order 3 and 4 that are used in the numerical experiments have been obtained without using existing Rosenbrock ones.

3. Stability When Solving Linear Ordinary Differential Equations

Following the ideas given for Rosenbrock methods in [6], in this part we deal with the stability of Rosenbrock–Nyström methods when they are applied to a simplified problem such as

$$\begin{aligned}
 U''(t) &= -B^2U(t), \\
 U(0) &= U_0, \\
 U'(0) &= V_0,
 \end{aligned}
 \tag{9}$$

where B is a given symmetric positive defined matrix of order $m \geq 1$ and $U(t)$, U_0 and $V_0 \in \mathbb{R}^m$.

Here, we study the stability in the energy norm, which is the natural norm for the study of the well-posedness of problem (9). This norm is given by

$$\|(U(t), U'(t))^T\|_B^2 = \|BU(t)\|_2^2 + \|U'(t)\|_2^2,$$

with $\|\cdot\|$ being the Euclidean norm in \mathbb{R}^m .

When solving problem (9) with a Rosenbrock–Nyström method, we obtain

$$\begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix} = \begin{bmatrix} r_{11}(\tau B) & B^{-1}r_{12}(\tau B) \\ Br_{21}(\tau B) & r_{22}(\tau B) \end{bmatrix} \begin{bmatrix} U_0 \\ V_0 \end{bmatrix},
 \tag{10}$$

where terms $r_{ij}(kB)$, $1 \leq i, j \leq 2$ are given, in tensorial form, by

$$\begin{aligned}
 r_{11}(\tau B) &= I - (b^T \otimes \tau^2 B^2)M(\mathcal{A}_\delta, \mathcal{A}_\alpha, \mathcal{A}_\gamma, B)^{-1}(\mathcal{A}_\delta e \otimes I), \\
 r_{12}(\tau B) &= (b^T \otimes \tau B)M(\mathcal{A}_\delta, \mathcal{A}_\alpha, \mathcal{A}_\gamma, B)^{-1}(e \otimes I) \\
 r_{21}(\tau B) &= -\tau B[(b^T e \otimes I) - ((b^T \mathcal{A}_\alpha + \beta^T) \otimes \tau^2 B^2)M(\mathcal{A}_\delta, \mathcal{A}_\alpha, \mathcal{A}_\gamma, B)^{-1}(\mathcal{A}_\delta e \otimes I)], \\
 r_{22}(\tau B) &= I - ((b^T \mathcal{A}_\alpha + \beta^T) \otimes \tau^2 B^2)M(\mathcal{A}_\delta, \mathcal{A}_\alpha, \mathcal{A}_\gamma, B)^{-1}(e \otimes I).
 \end{aligned}$$

with $M(\mathcal{A}_\delta, \mathcal{A}_\alpha, \mathcal{A}_\gamma, B) = \mathcal{I} \otimes I + (\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma) \otimes \tau^2 B^2$. These elements form matrix $R(\tau B)$,

$$R(\tau B) = \begin{bmatrix} r_{11}(\tau B) & r_{12}(\tau B) \\ r_{21}(\tau B) & r_{22}(\tau B) \end{bmatrix}.$$

By bounding (10) in the energy norm, we obtain that the proof of stability is related to the boundedness of the powers of matrix $R(\tau B)$. As matrix B is assumed to be symmetric and positive definite, then B is normal and we can use the following spectral result:

$$\|R(\tau B)^n\|_2 \leq \sup_{\theta \in \sigma(\tau B)} \|R(\theta)^n\|_2, \tag{11}$$

with $\sigma(\tau B)$ being the spectrum of τB . Then, the boundedness of the powers of matrix $R(\tau B)$ is reduced to the study of the boundedness of matrix $R(\theta)$. (Note: if we assume that B is not normal, we can use a similar result to (11), but considering the numerical range instead of the spectrum [14].)

Following the results in [15], the following definitions and theorem can be stated.

Definition 1. The interval $C = [0, \beta_{stab})$ is the interval of stability of the Rosenbrock–Nyström method if $\beta_{stab} \in \mathbb{R}^+$ is the highest value, such that

$$C \subset \{\theta \in \mathbb{R}^+ \cup \{0\} / \rho(R(\theta)) \leq 1 \text{ and } R(\theta) \text{ is simple when } \rho(R(\theta)) = 1\}.$$

The Rosenbrock–Nyström method is said to be R -stable if $C = \mathbb{R}^+ \cup \{0\}$.

Definition 2. The interval $C^* = [0, \beta_{per})$ is the interval of periodicity of the Rosenbrock–Nyström method if $\beta_{per} \in \mathbb{R}^+$ is the highest value, such that

$$C^* \subset \{\theta \in \mathbb{R}^+ \cup \{0\} / R(\theta) \text{ is simple and for all } \lambda \in \sigma(R(\theta)), |\lambda| = 1\}.$$

We can say that the Rosenbrock–Nyström method is P -stable if $C^* = \mathbb{R}^+ \cup \{0\}$.

Theorem 1. Under assumptions that

- (i) The method is R -stable.
- (ii) $\sigma(\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma) \cap (-\infty, 0] = \emptyset$.
- (iii) There exists a value $\bar{\theta} \in \mathbb{R}$ such that $R(\bar{\theta})$ does not have double eigenvalues.
- (iv) $(b^T \mathcal{A}_\alpha + \beta^T)(\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma)^{-1}(\mathcal{A}_\delta e) = 1$.

Then,

$$\|R(kB)^n\|_2 \leq C, \quad n \in \mathbb{N}.$$

where C is independent of the size of $\sigma(kB)$.

This result cannot be obtained if assumption (iv) is not satisfied.

Proof. The proof of this theorem is straightforward by using the results of Theorems 5 and 7 in [15]. \square

Corollary 1. When the Rosenbrock–Nyström method comes from a Rosenbrock one with classical order greater than or equal to one, the condition

$$(b^T \mathcal{A}_\alpha + \beta^T)(\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma)^{-1}(\mathcal{A}_\delta e) = 1$$

is always satisfied.

Proof. Let us assume that the Rosenbrock–Nyström method comes from a Rosenbrock one with Butcher array

$$\begin{array}{c|c|c|c} \tilde{\alpha} & \tilde{\mathcal{A}}_\alpha & \tilde{\gamma} & \tilde{\mathcal{A}}_\gamma \\ \hline & \tilde{b}^T & & \end{array}$$

and that the coefficients of the Rosenbrock–Nyström method satisfy relations (8). Then,

$$\begin{aligned}
 & (b^T \mathcal{A}_\alpha + \beta^T)(\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma)^{-1}(\mathcal{A}_\delta e) \\
 &= (\tilde{b}^T \tilde{\mathcal{A}}_\alpha + \tilde{\beta}^T \tilde{\mathcal{A}}_\gamma)((\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)\tilde{\mathcal{A}}_\alpha + (\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)\tilde{\mathcal{A}}_\gamma)^{-1}(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)e \\
 &= \tilde{b}^T(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)((\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma))^{-1}(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)e \\
 &= \tilde{b}^T(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)((\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)^2)^{-1}(\tilde{\mathcal{A}}_\alpha + \tilde{\mathcal{A}}_\gamma)e \\
 &= \tilde{b}^T e = 1
 \end{aligned}$$

□

4. Order Conditions for Rosenbrock–Nyström Methods

Let us see the conditions that Rosenbrock–Nyström methods should satisfy to obtain the highest possible order when integrating a problem such as (1). In a similar way as with Runge–Kutta–Nyström methods, a Rosenbrock–Nyström method is said to have classical order p if

$$\begin{aligned}
 \rho_{n+1} &= y(t_{n+1}) - \hat{y}_{n+1} = O(\tau^{p+1}), \\
 \zeta_{n+1} &= y'(t_{n+1}) - \hat{v}_{n+1} = O(\tau^{p+1}),
 \end{aligned}$$

where $(\hat{y}_{n+1}, \hat{v}_{n+1})^T$ is the numerical solution obtained from the exact solution $(y(t_n), y'(t_n))^T = (\tilde{y}_{n+1}, \tilde{v}_{n+1})^T$ with time step size τ .

In order to study the order conditions, it is useful to write the equations as in the autonomous case, given by (5), in the following way,

$$\begin{aligned}
 g_{n,i}^J &= y_n^J + \sum_{j=1}^{i-1} \alpha_{ij} K_{n,j}^J, \\
 K_{n,i}^J &= \tau v_n^J + \tau^2 \sum_{j=1}^i \delta_{ij} f^J(g_j) + \tau^2 \sum_K f_K^J(y_n) \sum_{j=1}^i \gamma_{ij} K_{n,j}^K, \\
 v_{n+1}^J &= v_n^J + \tau \sum_{i=1}^s b_i f^J(g_i) + \tau \sum_K f_K^J(y_n) \sum_{i=1}^s \beta_i K_{n,i}^K, \\
 y_{n+1}^J &= y_n^J + \sum_{i=1}^s b_i K_{n,i}^J,
 \end{aligned}$$

where the superscript indices in capital letters indicate the component of the vector we are using (in this part, the notation is similar to that used in [6]). In the following, $\partial f^J / \partial y^K$ will be denoted by f_K^J , $\partial^2 f^J / \partial y^K \partial y^L$ by f_{KL}^J , etc.

To obtain the order conditions, we compare the Taylor series of \hat{y}_{n+1} and \hat{v}_{n+1} obtained from the exact solution $(\tilde{y}_n, \tilde{v}_n)^T$ with the Taylor series of the exact solution.

In this part, the following formulae are used:

$$\begin{aligned}
 (\tau \varphi(\tau))^{(q)}|_{\tau=0} &= q \varphi^{(q-1)}(\tau)|_{\tau=0}, \quad q \geq 1, \\
 (\tau^2 \varphi(\tau))^{(q)}|_{\tau=0} &= q(q-1) \varphi^{(q-2)}(\tau)|_{\tau=0}, \quad q \geq 2.
 \end{aligned}$$

These formulae, which can be proved in a recursive manner, are obtained by using that

$$\begin{aligned}
 (\tau \varphi(\tau))^{(q)} &= q \varphi^{(q-1)}(\tau) + \tau \varphi^{(q)}(\tau), \quad q \geq 1, \\
 (\tau^2 \varphi(\tau))^{(q)} &= q(q-1) \varphi^{(q-2)}(\tau) + 2q\tau \varphi^{(q-1)}(\tau) + \tau^2 \varphi^{(q)}(\tau), \quad q \geq 2.
 \end{aligned}$$

We differentiate by using the notation $\varphi(\tau) = \sum_{j=1}^i (\delta_{ij} f^J(g_j) + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) K_{n,j}^K)$ together with the previous formulae. Then, we obtain

$$\begin{aligned}
 (K_{n,i}^J)^{(0)}|_{\tau=0} &= 0, \\
 (K_{n,i}^J)^{(1)}|_{\tau=0} &= \tilde{v}_n^J|_{\tau=0} + \left(2\tau \sum_{j=1}^i (\delta_{ij} f^J(g_j) + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) K_{n,j}^K) \right)|_{\tau=0} \\
 &\quad + \left(\tau^2 \sum_{j=1}^i (\delta_{ij} f^J(g_j) + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) K_{n,j}^K)^{(1)} \right)|_{\tau=0} \\
 &= \tilde{v}_n^J, \\
 (K_{n,i}^J)^{(2)}|_{\tau=0} &= 2 \sum_{j=1}^i (\delta_{ij} f^J(g_j) + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) K_{n,j}^K)|_{\tau=0} \\
 &= 2 \sum_{j=1}^i \delta_{ij} f^J(\tilde{y}_n) \\
 (K_{n,i}^J)^{(3)}|_{\tau=0} &= 6 \sum_{j=1}^i (\delta_{ij} f^J(g_j)^{(1)} + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) (K_{n,j}^K)^{(1)})|_{\tau=0} \\
 &= 6 \sum_{j=1}^i (\delta_{ij} \alpha_j \sum_K f_K^J(\tilde{y}_n) \tilde{v}_n^K + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) \tilde{v}_n^K) \\
 &= 6 \sum_{j=1}^i (\delta_{ij} \alpha_j + \gamma_{ij}) \sum_K f_K^J(\tilde{y}_n) \tilde{v}_n^K, \\
 (K_{n,i}^J)^{(4)}|_{\tau=0} &= 12 \sum_{j=1}^i (\delta_{ij} (f^J(g_j))^{(2)} + \gamma_{ij} \sum_K f_K^J(\tilde{y}_n) (K_{n,j}^K)^{(2)})|_{\tau=0} \\
 &= 12 \sum_{j=1}^i \delta_{ij} \alpha_j^2 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{v}_n^K \tilde{v}_n^L + 24 \sum_{j=1}^i \sum_{l=1}^{i-1} \sum_{m=1}^l \delta_{ij} \alpha_{jl} \delta_{lm} \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \\
 &\quad + 24 \sum_{j=1}^i \sum_{l=1}^j \gamma_{ij} \delta_{jl} \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n),
 \end{aligned}$$

where we have used that

$$\begin{aligned}
 f^J(g_i)|_{\tau=0} &= f^J(\tilde{y}_n), \\
 (f^J(g_i))^{(1)}|_{\tau=0} &= \sum_K \alpha_i f_K^J(\tilde{y}_n) \tilde{v}_n^K, \\
 (f^J(g_i))^{(2)}|_{\tau=0} &= \sum_{K,L} \alpha_i^2 f_{K,L}^J(\tilde{y}_n) \tilde{v}_n^K \tilde{v}_n^L + 2 \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} \delta_{jl},
 \end{aligned}$$

together with

$$\begin{aligned}
 (g_i^K)^{(0)}|_{\tau=0} &= \tilde{y}_n^K, \\
 (g_i^K)^{(1)}|_{\tau=0} &= \sum_{j=1}^{i-1} \alpha_{ij} (K_{n,j}^J)^{(1)}|_{\tau=0} \\
 &= \sum_{j=1}^{i-1} \alpha_{ij} \tilde{v}_n^J \\
 &= \alpha_i \tilde{v}_n^J, \\
 (g_i^K)^{(2)}|_{\tau=0} &= \sum_{j=1}^{i-1} \alpha_{ij} (K_{n,j}^J)^{(2)}|_{\tau=0} \\
 &= 2 \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} \delta_{jl} f^J(\tilde{y}_n), \\
 (g_i^K)^{(3)}|_{\tau=0} &= \sum_{j=1}^{i-1} \alpha_{ij} (K_{n,j}^J)^{(3)}|_{\tau=0} \\
 &= 6 \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} (\delta_{jl} \alpha_l + \gamma_{jl}) \sum_K f_K^J(\tilde{y}_n) \tilde{v}_n^K.
 \end{aligned}$$

Then, by using the expressions obtained for $(K_{n,i}^J)^{(l)}, l = 1, \dots, 4$, we have

$$\begin{aligned}
 (\vartheta_{n+1}^J)^{(1)}|_{\tau=0} &= \sum_{i=1}^s b_i f^J(g_i)|_{\tau=0} + \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \beta_i (K_{n,i}^K)^{(0)}|_{\tau=0} \\
 &= f^J(\tilde{y}_n) \sum_{i=1}^s b_i, \\
 (\vartheta_{n+1}^J)^{(2)}|_{\tau=0} &= 2 \sum_{i=1}^s b_i (f^J(g_i))^{(1)}|_{\tau=0} + 2 \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \beta_i (K_{n,i}^K)^{(1)}|_{\tau=0} \\
 &= 2 \sum_{i=1}^s b_i \sum_K f_K^J(\tilde{y}_n) \alpha_i \tilde{\vartheta}_n^K + 2 \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \beta_i \tilde{\vartheta}_n^K \\
 &= 2 \sum_{i=1}^s (b_i \alpha_i + \beta_i) \sum_K f_K^J(\tilde{y}_n) \tilde{\vartheta}_n^K, \\
 (\vartheta_{n+1}^J)^{(3)}|_{\tau=0} &= 3 \sum_{i=1}^s b_i (f^J(g_i))^{(2)}|_{\tau=0} + 3 \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \beta_i (K_{n,i}^K)^{(2)}|_{\tau=0} \\
 &= 3 \sum_{i=1}^s b_i \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \alpha_i^2 \tilde{\vartheta}_n^K \tilde{\vartheta}_n^L \\
 &\quad + 6 \sum_{i=1}^s b_i \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} \delta_{jl} \\
 &\quad + 6 \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \sum_{j=1}^i \beta_i \delta_{ij} f^K(\tilde{y}_n) \\
 &= 3 \sum_{i=1}^s b_i \alpha_i^2 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{\vartheta}_n^K \tilde{\vartheta}_n^L \\
 &\quad + 6 \sum_{i=1}^s \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \left(\sum_{j=1}^{i-1} \sum_{l=1}^j b_i \alpha_{ij} \delta_{jl} + \sum_{j=1}^i \beta_i \delta_{ij} \right), \\
 (\vartheta_{n+1}^J)^{(4)}|_{\tau=0} &= 4 \sum_{i=1}^s b_i (f^J(g_i))^{(3)}|_{\tau=0} + 4 \sum_K f_K^J(\tilde{y}_n) \sum_{i=1}^s \beta_i (K_{n,i}^K)^{(3)}|_{\tau=0} \\
 &= 4 \sum_{i=1}^s b_i \alpha_i^3 \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\vartheta}_n^K \tilde{\vartheta}_n^L \tilde{\vartheta}_n^M \\
 &\quad + 16 \sum_{i=1}^s b_i \alpha_i \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} \delta_{jl} \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^K(\tilde{y}_n) \tilde{\vartheta}_n^L \\
 &\quad + 8 \sum_{i=1}^s b_i \alpha_i \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} \delta_{jl} \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^L(\tilde{y}_n) \tilde{\vartheta}_n^K \\
 &\quad + 24 \sum_{i=1}^s b_i \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} (\delta_{jl} \alpha_l + \gamma_{jl}) \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) \tilde{\vartheta}_n^L \\
 &\quad + 24 \sum_{i=1}^s \sum_{j=1}^i \beta_i (\delta_{ij} \alpha_j + \gamma_{ij}) \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) \tilde{\vartheta}_n^L \\
 &= \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\vartheta}_n^K \tilde{\vartheta}_n^L \tilde{\vartheta}_n^M \sum_{i=1}^s 4b_i \alpha_i^3 \\
 &\quad + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^K(\tilde{y}_n) \tilde{\vartheta}_n^L \sum_{i=1}^s \sum_{j=1}^{i-1} \sum_{l=1}^j 24b_i \alpha_i \alpha_{ij} \delta_{jl} \\
 &\quad + 24 \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) \tilde{\vartheta}_n^L \sum_{i=1}^s \left(\sum_{j=1}^{i-1} \sum_{l=1}^j b_i \alpha_{ij} (\delta_{jl} \alpha_l + \gamma_{jl}) + \sum_{j=1}^i \beta_i (\delta_{ij} \alpha_j + \gamma_{ij}) \right)
 \end{aligned} \tag{12}$$

where we have used that

$$(f^J(g_i))^{(3)} = \sum_{K,L,M} f_{K,L,M}^J(g_i)(g_i^K)^{(1)}(g_i^L)^{(1)}(g_i^M)^{(1)} + 2 \sum_{K,L} f_{K,L}^J(g_i)(g_i^K)^{(2)}(g_i^L)^{(1)} + \sum_{K,L} f_{K,L}^J(g_i)(g_i^K)^{(1)}(g_i^L)^{(2)} + \sum_K f_K^J(g_i)(g_i^K)^{(3)}$$

and therefore

$$(f^J(g_i))^{(3)}|_{\tau=0} = \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L \tilde{\sigma}_n^L \alpha_i^3 + 4 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_i \alpha_{ij} \delta_{jl} f^K(\tilde{y}_n) \tilde{\sigma}_n^L + 2 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_i \alpha_{ij} \delta_{jl} f^L(\tilde{y}_n) \tilde{\sigma}_n^K + 6 \sum_{K,L} f_K^J(\tilde{y}_n) \sum_{j=1}^{i-1} \sum_{l=1}^j \alpha_{ij} (\delta_{jl} \alpha_l + \gamma_{jl}) f_L^K(\tilde{y}_n) \tilde{\sigma}_n^L.$$

For $(\hat{y}_{n+1}^J)^{(l)}, l = 1, \dots, 4$, we have

$$\begin{aligned} (\hat{y}_{n+1}^J)^{(0)}|_{\tau=0} &= \hat{y}_n^J, \\ (\hat{y}_{n+1}^J)^{(1)}|_{\tau=0} &= \sum_{i=1}^s b_i (K_{n,i}^J)^{(1)}|_{\tau=0} \\ &= \tilde{\sigma}_n^J \sum_{i=1}^s b_i, \\ (\hat{y}_{n+1}^J)^{(2)}|_{\tau=0} &= \sum_{i=1}^s b_i (K_{n,i}^J)^{(2)}|_{\tau=0} \\ &= 2f^J(\tilde{y}_n) \sum_{i=1}^s \sum_{j=1}^i b_i \delta_{ij}, \\ (\hat{y}_{n+1}^J)^{(3)}|_{\tau=0} &= \sum_{i=1}^s b_i (K_{n,i}^J)^{(3)}|_{\tau=0} \\ &= 6 \sum_{i=1}^s b_i \sum_{j=1}^i (\delta_{ij} \alpha_j + \gamma_{ij}) \sum_K f_K^J(\tilde{y}_n) \tilde{\sigma}_n^K \\ &= 6 \sum_K f_K^J(\tilde{y}_n) \tilde{\sigma}_n^K \sum_{i=1}^s \sum_{j=1}^i b_i (\delta_{ij} \alpha_j + \gamma_{ij}), \tag{13} \\ (\hat{y}_{n+1}^J)^{(4)}|_{\tau=0} &= \sum_{i=1}^s b_i (K_{n,i}^J)^{(4)}|_{\tau=0} \\ &= 12 \sum_{i=1}^s b_i \sum_{j=1}^i \delta_{ij} \alpha_j^2 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L \\ &\quad + 24 \sum_{i=1}^s b_i \sum_{j=1}^i \sum_{l=1}^{j-1} \sum_{m=1}^l \delta_{ij} \alpha_{il} \delta_{lm} \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \\ &\quad + 24 \sum_{i=1}^s b_i \sum_{j=1}^i \sum_{l=1}^j \gamma_{ij} \delta_{jl} \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \\ &= 12 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L \sum_{i=1}^s \sum_{j=1}^i b_i \delta_{ij} \alpha_j^2 \\ &\quad + 24 \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n) \sum_{i=1}^s \sum_{j=1}^i \sum_{l=1}^j b_i (\gamma_{ij} \delta_{jl} + \delta_{ij} \alpha_{jl} \sum_{m=1}^l \delta_{lm}). \end{aligned}$$

Now, we calculate the derivatives of the exact solution, by taking into account that $y'' = f(y)$,

$$\begin{aligned}
 (\tilde{y}_n^J)^{(1)} &= \tilde{\sigma}_n^J, \\
 (\tilde{y}_n^J)^{(2)} &= (\tilde{\sigma}_n^J)^{(1)} = f^J(\tilde{y}_n), \\
 (\tilde{y}_n^J)^{(3)} &= (\tilde{\sigma}_n^J)^{(2)} = (f^J(\tilde{y}_n))^{(1)} = \sum_K f_K^J(\tilde{y}_n) \tilde{\sigma}_n^K, \\
 (\tilde{y}_n^J)^{(4)} &= (\tilde{\sigma}_n^J)^{(3)} = (f^J(\tilde{y}_n))^{(2)} \\
 &= \sum_{K,L} f_{KL}^J(\tilde{y}_n) \tilde{\sigma}_n^K (\tilde{y}_n^L)^{(1)} + \sum_K f_K^J(\tilde{y}_n) (\tilde{\sigma}_n^K)^{(1)} \\
 &= \sum_{K,L} f_{KL}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L + \sum_K f_K^J(\tilde{y}_n) f^K(\tilde{y}_n), \\
 (\tilde{y}_n^J)^{(5)} &= (\tilde{\sigma}_n^J)^{(4)} = (f^J(\tilde{y}_n))^{(3)} \\
 &= \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L (\tilde{y}_n^M)^{(1)} + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) (\tilde{\sigma}_n^K)^{(1)} \tilde{\sigma}_n^L \\
 &\quad + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{\sigma}_n^K (\tilde{\sigma}_n^L)^{(1)} + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) (\tilde{y}_n^L)^{(1)} f^K(\tilde{y}_n) \\
 &\quad + \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) (\tilde{y}_n^L)^{(1)} \\
 &= \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L \tilde{\sigma}_n^M + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^K(\tilde{y}_n) \tilde{\sigma}_n^L \\
 &\quad + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) \tilde{\sigma}_n^K f^L(\tilde{y}_n) + \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^K(\tilde{y}_n) \tilde{\sigma}_n^L \\
 &\quad + \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) \tilde{\sigma}_n^L \\
 &= \sum_{K,L,M} f_{K,L,M}^J(\tilde{y}_n) \tilde{\sigma}_n^K \tilde{\sigma}_n^L \tilde{\sigma}_n^M + 3 \sum_{K,L} f_{K,L}^J(\tilde{y}_n) f^K(\tilde{y}_n) \tilde{\sigma}_n^L \\
 &\quad + \sum_{K,L} f_K^J(\tilde{y}_n) f_L^K(\tilde{y}_n) \tilde{\sigma}_n^L.
 \end{aligned}
 \tag{14}$$

To obtain the order conditions up to order four, we compare the results in (12) and (13) with the results in (14). Then, these order conditions are:

Order 1: We compare $(\hat{\sigma}_n^J)^{(1)}$ with $(\tilde{\sigma}_n^J)^{(1)}$ and $(\hat{y}_n^J)^{(1)}$ with $(\tilde{y}_n^J)^{(1)}$

$$\sum_{i=1}^s b_i = 1.$$

Order 2: We compare $(\hat{\sigma}_n^J)^{(2)}$ with $(\tilde{\sigma}_n^J)^{(2)}$ and $(\hat{y}_n^J)^{(2)}$ with $(\tilde{y}_n^J)^{(2)}$

$$\sum_{i=1}^s (b_i \alpha_i + \beta_i) = \frac{1}{2},$$

$$\sum_{i=1}^s \sum_{j=1}^i b_i \delta_{ij} = \frac{1}{2}.$$

Order 3: We compare $(\hat{\vartheta}_n^J)^{(3)}$ with $(\tilde{\vartheta}_n^J)^{(3)}$ and $(\hat{y}_n^J)^{(3)}$ with $(\tilde{y}_n^J)^{(3)}$

$$\begin{aligned} \sum_{i=1}^s b_i \alpha_i^2 &= \frac{1}{3}, \\ \sum_{i=1}^s \sum_{j=1}^{i-1} \sum_{l=1}^j b_i \alpha_{ij} \delta_{jl} + \sum_{i=1}^s \sum_{j=1}^i \beta_i \delta_{ij} &= \frac{1}{6}, \\ \sum_{i=1}^s \sum_{j=1}^i b_i (\delta_{ij} \alpha_j + \gamma_{ij}) &= \frac{1}{6}. \end{aligned}$$

Order 4: We compare $(\hat{\vartheta}_n^J)^{(4)}$ with $(\tilde{\vartheta}_n^J)^{(4)}$ and $(\hat{y}_n^J)^{(4)}$ with $(\tilde{y}_n^J)^{(4)}$

$$\begin{aligned} \sum_{i=1}^s b_i \alpha_i^3 &= \frac{1}{4}, \\ \sum_{i=1}^s \sum_{j=1}^{i-1} \sum_{l=1}^j b_i \alpha_i \alpha_{ij} \delta_{jl} &= \frac{1}{8}, \\ \sum_{i=1}^s \sum_{j=1}^{i-1} \sum_{l=1}^j b_i \alpha_{ij} (\delta_{jl} \alpha_l + \gamma_{jl}) + \sum_{i=1}^s \sum_{j=1}^i \beta_i (\delta_{ij} \alpha_j + \gamma_{ij}) &= \frac{1}{24}, \\ \sum_{i=1}^s \sum_{j=1}^i b_i \delta_{ij} \alpha_j^2 &= \frac{1}{12}, \\ \sum_{i=1}^s \sum_{j=1}^i \sum_{l=1}^{j-1} \sum_{m=1}^l b_i \delta_{ij} \alpha_{jl} \delta_{lm} + \sum_{i=1}^s \sum_{j=1}^i \sum_{l=1}^j b_i \gamma_{ij} \delta_{jl} &= \frac{1}{24}. \end{aligned}$$

By using notation $\alpha^j = (\alpha_1^j, \dots, \alpha_s^j)^T$ and $(b\alpha)^T = (b_1\alpha_1, \dots, b_s\alpha_s)$, these order conditions can be written as follows:

Order 1:

$$b^T e = 1.$$

Order 2:

$$b^T \alpha + \beta^T e = \frac{1}{2}, \quad b^T \mathcal{A}_\delta e = \frac{1}{2}.$$

Order 3:

$$b^T \alpha^2 = \frac{1}{3}, \quad (b^T \mathcal{A}_\alpha + \beta^T) \mathcal{A}_\delta e = \frac{1}{6}, \quad b^T (\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma) e = \frac{1}{6}.$$

Order 4:

$$\begin{aligned} b^T \alpha^3 &= \frac{1}{4}, & (b\alpha)^T \mathcal{A}_\alpha \mathcal{A}_\delta e &= \frac{1}{8}, & (b^T \mathcal{A}_\alpha + \beta^T) (\mathcal{A}_\delta \alpha + \mathcal{A}_\gamma) e &= \frac{1}{24}, \\ b^T \mathcal{A}_\delta \alpha^2 &= \frac{1}{12}, & b^T (\mathcal{A}_\delta \mathcal{A}_\alpha + \mathcal{A}_\gamma) \mathcal{A}_\delta e &= \frac{1}{24}. \end{aligned}$$

5. Numerical Experiments

This section is devoted to the numerical experiments we have carried out in order to prove the advantages of these methods when solving a non-linear equation.

The Rosenbrock–Nyström methods we have chosen are the ones that are developed and studied in [11]. The first one is the one-stage method given by:

$$\begin{array}{c|c|c|c} 0 & 0 & \frac{1}{2} & \frac{1}{4} \\ \hline & \frac{1}{2} & & \\ \hline & 1 & & \end{array}$$

This method can be obtained from the well-known Rosenbrock method with

$$\tilde{\alpha} = 0, \quad \tilde{A}_\alpha = 0, \quad \tilde{A}_\gamma = \frac{1}{2}, \quad \tilde{b}^T = 1,$$

Therefore, the equations that this Rosenbrock–Nyström method provides are the same that we have with the Rosenbrock method when converting the second order problem in a first order one.

This Rosenbrock method is A-stable, but not L-stable [6]. The method presented here is the only Rosenbrock–Nyström method with one stage and classical order 2 that is stable. In fact, it can be proved that this method is P-stable as the eigenvalues of its stability matrix are $\frac{4-\theta^2}{4+\theta^2} \pm \frac{4\theta}{4+\theta^2}i$, for $\theta \in \mathbb{R}$. They are complex conjugate with modulus 1 except for $\theta = 0$ and $\theta \rightarrow \infty$, where they are double and equal to 1. There are no one-stage Rosenbrock–Nyström methods with classical order 2 that are just R-stable but not P-stable, so in order to obtain methods of this type, we should use at least two stages. Another possibility is to construct one-stage Rosenbrock–Nyström methods with complex coefficients, following the ideas given in [16] for Rosenbrock methods. This will be the idea of future works.

The second one is an R-stable method with two stages and classical order 3, which we will call RN3. The coefficients of this method are given by the following array:

$$\begin{array}{c|c|c|c|c} 0 & 0 & \frac{2}{3} & & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & 0 & -\frac{2}{9} & \frac{2}{3} & -\frac{10}{9} & \frac{2}{3} \\ \hline & -\frac{3}{4} & \frac{3}{4} & & & & \\ \hline & \frac{1}{4} & \frac{3}{4} & & & & \end{array}$$

The third one is a method with three stages and classical order 4. This method is called RN4. The coefficients of this method are given by the following array:

$$\begin{array}{c|c|c|c|c|c|c|c|c} 0 & 0 & & & \frac{3}{2} & & & & \frac{3}{2} \\ -\frac{111}{20} & -\frac{111}{20} & 0 & & -\frac{2331}{290} & \frac{3}{2} & & & -\frac{333}{40} & \frac{3}{2} \\ -\frac{27}{31} & -\frac{10,877,913}{13,938,344} & -\frac{1,261,935}{13,938,344} & 0 & -\frac{4,064,531,049}{2,365,464,982} & -\frac{6,527,250}{285,487,153} & \frac{3}{2} & & -\frac{24,378,057}{23,356,144} & -\frac{3,785,805}{27,876,688} & \frac{3}{2} \\ \hline & \frac{160,081,141}{288,754,956} & \frac{3,659,778,205}{93,075,347,484} & \frac{94,178}{234,981} & & & & & & & \\ \hline & \frac{411,283}{998,001} & -\frac{134,000}{35,743,221} & \frac{417,074}{704,943} & & & & & & & \end{array}$$

In the tables, the local and global orders are given. The global error has been calculated as the difference between the exact solution at $T = 1$ and the numerical one obtained with our method.

Example 3. The first problem we have solved is the first problem presented in Example 1, that is, problem (2). Here, the parameters we have selected are

$$N = 20, \quad \lambda = 1000, \quad \alpha = 2, \quad p = 3.$$

We present the results obtained for the method with classical order 2 in Table 1, where we can see that the expected orders are obtained for the solution as well as for the derivative. For the local error in the solution, we obtain one order more than the one expected since the odd derivatives vanish.

The results for the method with classical order 3 are given in Table 2 where we can see that the expected orders are obtained for the solution as well as for the derivative. For the local error in the derivative, we obtain one order more than the one expected since the odd derivatives vanish.

The results for the method with classical order 4 are presented in Table 3, where again we can see that the expected results are obtained. The local order in the solution is one unit higher than the expected one as the odd derivatives vanish.

Table 1. Local and global errors and orders for problem (2) with RN2.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	8.7268×10^{-7}	5.4684×10^{-8}	3.4200×10^{-9}	2.1379×10^{-10}	1.3362×10^{-11}	8.3524×10^{-13}
Order		3.9962	3.9991	3.9998	3.9999	3.9998
Loc. err. $u'(t)$	1.3910×10^{-4}	1.7433×10^{-5}	2.1806×10^{-6}	2.7262×10^{-7}	3.4079×10^{-8}	4.2599×10^{-9}
Order		2.9962	2.9991	2.9998	2.9999	3.0000
Glob. err. $u(t)$	1.9668×10^{-4}	4.9142×10^{-5}	1.2282×10^{-5}	3.0700×10^{-6}	7.6745×10^{-7}	1.9186×10^{-7}
Order		2.0008	2.0004	2.0002	2.0001	2.0001
Glob. err. $u'(t)$	1.3275×10^{-4}	3.5692×10^{-5}	9.0829×10^{-6}	2.2811×10^{-6}	5.7098×10^{-7}	1.4279×10^{-7}
Order		1.8951	1.9744	1.9934	1.9982	1.9994

Table 2. Local and global errors and orders for problem (2) with RN3.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	7.4745×10^{-7}	4.8042×10^{-8}	3.0236×10^{-9}	1.8931×10^{-10}	1.1837×10^{-11}	7.3971×10^{-13}
Order		3.9596	3.9899	3.9975	3.9994	4.0002
Loc. err. $u'(t)$	2.5478×10^{-6}	8.0729×10^{-8}	2.5316×10^{-9}	7.9180×10^{-11}	2.4749×10^{-12}	7.7275×10^{-14}
Order		4.9800	4.9950	4.9988	4.9997	5.0012
Glob. err. $u(t)$	4.6594×10^{-6}	4.0170×10^{-7}	3.8542×10^{-8}	4.0814×10^{-9}	4.6400×10^{-10}	5.5107×10^{-11}
Order		3.5360	3.3816	3.2393	3.1369	3.0738
Glob. err. $u'(t)$	9.4688×10^{-5}	1.2141×10^{-5}	1.5315×10^{-6}	1.9215×10^{-7}	2.4058×10^{-8}	3.0095×10^{-9}
Order		2.9633	2.9869	2.9947	2.9977	2.9989

Table 3. Local and global errors and orders for problem (2) with RN4.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	6.1363×10^{-7}	9.9822×10^{-9}	1.5757×10^{-10}	2.4682×10^{-12}	3.8384×10^{-14}	4.5776×10^{-16}
Order		5.9419	5.9853	5.9964	6.0068	6.3898
Loc. err. $u'(t)$	1.8146×10^{-6}	6.9583×10^{-8}	2.2799×10^{-9}	7.2081×10^{-11}	2.2592×10^{-12}	7.0702×10^{-14}
Order		4.7048	4.9317	4.9832	4.9958	4.9979
Glob. err. $u(t)$	2.6326×10^{-6}	1.9208×10^{-7}	1.2682×10^{-8}	8.1048×10^{-10}	5.1163×10^{-11}	3.2167×10^{-12}
Order		3.7767	3.9209	3.9678	3.9856	3.9914
Glob. err. $u'(t)$	7.9785×10^{-5}	2.7301×10^{-6}	9.5300×10^{-8}	3.5643×10^{-9}	1.4775×10^{-10}	6.9183×10^{-12}
Order		4.8691	4.8403	4.7408	4.5924	4.4166

Example 4. The second problem is the equation of motion of a soliton in an exponential lattice. This problem was first proposed in [17] and it is a highly non-linear system.

$$\begin{aligned}
 u_j''(t) &= 2e^{-u_j} - e^{-u_{j-1}} - e^{-u_{j+1}}, \quad j = 1, \dots, N, \\
 u_j(0) &= -\ln(1 + \beta^2 \operatorname{sech}^2(\alpha j)), \\
 u_j'(0) &= \frac{2\beta^3 \operatorname{sech}^2(\alpha j) \tanh(\alpha j)}{1 + \beta^2 \operatorname{sech}^2(\alpha j)},
 \end{aligned}
 \tag{15}$$

with $\alpha = 2$, $\beta = \sinh(\alpha)$ and $N = 20$. The solution of this problem is $u_j(t) = -\ln(1 + \beta^2 \operatorname{sech}^2(\alpha j + \beta t))$.

The results for method RN2 can be seen in Table 4. There, we can observe that we also obtain the expected local and global order for the solution and the derivative.

The results for the second method, RN3, can be seen in Table 5, for which the expected results have been obtained.

Finally, the results for method RN4 are presented in Table 6.

Table 4. Local and global errors and orders for problem (15) with RN2.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	2.6003×10^{-6}	3.0800×10^{-7}	3.7439×10^{-8}	4.6137×10^{-9}	5.7258×10^{-10}	7.1314×10^{-11}
Order		3.0777	3.0403	3.0206	3.0104	3.0052
Loc. err. $u'(t)$	1.1920×10^{-4}	1.4952×10^{-5}	1.8721×10^{-6}	2.3420×10^{-7}	2.9286×10^{-8}	3.6615×10^{-9}
Order		2.9949	2.9976	2.9989	2.9994	2.9997
Glob. err. $u(t)$	1.1100×10^{-4}	2.5216×10^{-5}	5.9894×10^{-6}	1.4582×10^{-6}	3.5965×10^{-7}	8.9303×10^{-8}
Order		2.1381	2.0739	2.0383	2.0195	2.0098
Glob. err. $u'(t)$	2.6457×10^{-4}	6.6306×10^{-5}	1.6598×10^{-5}	4.1520×10^{-6}	1.0383×10^{-6}	2.5963×10^{-7}
Order		1.9964	1.9982	1.9991	1.9995	1.9998

Table 5. Local and global errors and orders for problem (15) with RN3.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	6.0678×10^{-7}	3.8272×10^{-8}	2.4025×10^{-9}	1.5048×10^{-10}	9.4150×10^{-12}	5.8841×10^{-13}
Order		3.9868	3.9937	3.9969	3.9985	4.0001
Loc. err. $u'(t)$	7.2800×10^{-7}	4.5052×10^{-8}	2.8007×10^{-9}	1.7456×10^{-10}	1.0895×10^{-11}	6.7900×10^{-13}
Order		4.0143	4.0078	4.0040	4.0020	4.0041
Glob. err. $u(t)$	3.6685×10^{-6}	4.6022×10^{-7}	5.7584×10^{-8}	7.2002×10^{-9}	9.0011×10^{-10}	1.1252×10^{-10}
Order		2.9948	2.9986	2.9996	2.9999	2.9999
Glob. err. $u'(t)$	3.2844×10^{-6}	4.1504×10^{-7}	5.2155×10^{-8}	6.5363×10^{-9}	8.1810×10^{-10}	1.0233×10^{-10}
Order		2.9843	2.9924	2.9962	2.9981	2.9991

Table 6. Local and global errors and orders for problem (15) with RN4.

τ	1/80	1/160	1/320	1/640	1/1280	1/2560
Loc. err. $u(t)$	1.3797×10^{-7}	2.4952×10^{-9}	4.8387×10^{-11}	1.0514×10^{-12}	2.5920×10^{-14}	4.6849×10^{-16}
Order		5.7891	5.6884	5.5242	5.3421	5.7899
Loc. err. $u'(t)$	1.6797×10^{-6}	5.3774×10^{-8}	1.6885×10^{-9}	5.2794×10^{-11}	1.6491×10^{-12}	5.2897×10^{-14}
Order		4.9652	4.9931	4.9992	5.0006	4.9624
Glob. err. $u(t)$	1.4563×10^{-6}	5.7019×10^{-8}	3.4328×10^{-9}	2.3119×10^{-10}	1.5242×10^{-11}	9.8139×10^{-13}
Order		4.6748	4.0540	3.8922	3.9229	3.9571
Glob. err. $u'(t)$	4.0638×10^{-6}	2.5938×10^{-7}	1.6313×10^{-8}	1.0216×10^{-9}	6.3895×10^{-11}	3.9944×10^{-12}
Order		3.9697	3.9909	3.9971	3.9990	3.9997

Example 5. The last problem we have solved is the non-linear Euler–Bernoulli Equation (3) presented in Example 2.

We first discretized in space, by using a spectral method suggested in [18] and deeply studied in [8]. This spectral method has been derived to discretize in space a problem such as

$$\begin{cases} -Au = F, & x \in \Omega, \\ \partial u = G, & x \in \Gamma \end{cases}$$

with $Au = u_{xxxx}$, $F \in X$, $G \in Y$, and $u \in D(A)$, with X being a Hilbert space of functions and Y being a suitable space of functions defined on the boundary. In particular, for the problem we

have, we take $\Omega = (-1, 1)$ with boundary $\Gamma = \{-1, 1\}$ and boundary conditions $u(-1) = g_{-1}$, $u(1) = g_1$, $u_x(-1) = h_{-1}$, $u_x(1) = h_1$. For the spatial discretization of this problem, we consider a grid $\Omega_J, \Omega \cup \Gamma$ associated with a natural parameter J (related to the number of nodes on it). Then, we take X_J as the space of polynomials of a degree less than or equal to J , integer $J \geq 4$, and $X_{J,0}$ as the subspace of X_J such that they and their derivatives vanish on the boundary. We will consider in X_J an approximation of the L^2 -norm in $X = L^2(\Omega)$, which will be denoted by $\|\cdot\|_J$. Then, the spatial discretization is given by

$$-A_{J,0}U_J + C_J\partial u = P_JF$$

with $A_{J,0}$ being the matrix that discretizes A_0 and C_J and P_J being the operators associated with the discretization of A which take into account the information on the boundary of u and F , respectively. The interior nodes are denoted by $\mu_j, j = 1, \dots, J$. They are the zeros of the second derivative of the Legendre polynomial of degree $J + 2$.

In this way, after the spatial discretization, our problem reads

$$\begin{cases} U_J'' = -A_{J,0}U_J + C_J\partial u + U_J^2 + P_JH(t), & 0 \leq t \leq T = 1, \\ U_J(0) = U_{J,0} \\ U_J'(0) = V_{J,0} \end{cases}$$

with $P_JH(t)_j = h(t, \mu_j)$, $(U_{J,0})_j = u(0, \mu_j)$ and $(V_{J,0})_j = u_t(0, \mu_j)$.

In the numerical experiments, we have taken $J = 40$.

In this part, we present the results that have been obtained with method RN2 in Table 7, the results obtained with method RN3 in Table 8, and the results that have been obtained with method RN4 in Table 9. In this case, with RN4, we do not obtain order 4 in the global error in the derivative because of the order reduction phenomenon, which has been deeply studied for other numerical integrators such as Runge–Kutta methods, Rosenbrock methods, or RKN methods [19–21]. The way to avoid this order reduction is the objective of a forthcoming paper.

Finally, we present the comparison in terms of computational cost between RN2, RN3, RN4, and the RKN method with two stages and classical order 3 which is given in [22]:

$\frac{9+\sqrt{33}}{12}$	$\frac{19+3\sqrt{33}}{48}$	
$\frac{15-\sqrt{33}}{24}$	$-\frac{33+17\sqrt{33}}{192}$	$\frac{19+3\sqrt{33}}{48}$
	$\frac{5-\sqrt{33}}{24}$	$\frac{7+\sqrt{33}}{24}$
	$\frac{9-\sqrt{33}}{24}$	$\frac{15+\sqrt{33}}{24}$

We will denote this method by the abbreviation RKN3.

The results can be seen in Figure 1 and Table 10, where we have written the values of τ and the value of the computational cost to obtain global errors in the solution in the range $[10^{-12}, 10^{-6}]$. In the graph, the error has been plotted as a function of τ in double logarithmic scale. In this graph and this table, we can see that the method which generates more computational cost to achieve the desired error is RN2. Given a fixed value of τ , this method presents a lower computational cost than the other three, but as its classical order is only two, it needs a lower value of τ to obtain the same error than the other three methods, and in this way, it needs more time steps. For example, if we take $\tau = \frac{1}{320}$, we can see that RN2 takes 1.8751×10^{-2} seconds to obtain the result, but RN4 needs 3.7180×10^{-2} seconds, nearly double. However, for this value, the error that is found with RN2 is 7.2957×10^{-7} , but with RN4 it is 6.6974×10^{-9} . To obtain this error with the RN2 method, we need a value of τ between $\frac{1}{2560}$ and $\frac{1}{5120}$, so the number of steps is much greater and so is the computational cost.

As we can also see, both RN3 and RN4 are better in terms of the computational cost than RKN3. As RN4 has more stages than RN3, for bigger values of τ , RN3 needs fewer steps to obtain the desired error, but when τ is smaller, because of the classical order of the RN4 method, it requires fewer time steps and the computational cost of this method is lower.

Table 7. Local and global errors and orders for problem (3) with RN2.

τ	1/80	1/160	1/320	1/640	1/1280
Loc. err. $u(t)$	3.5093×10^{-7}	2.3204×10^{-8}	1.5422×10^{-9}	1.1241×10^{-10}	1.0014×10^{-11}
Order		3.9187	3.9114	3.7781	3.4887
Loc. err. $u'(t)$	5.5860×10^{-5}	7.2984×10^{-6}	9.2722×10^{-7}	1.1657×10^{-7}	1.4601×10^{-8}
Order		2.9361	2.9766	2.9917	2.9972
Glob. err. $u(t)$	1.1780×10^{-5}	2.9189×10^{-6}	7.2957×10^{-7}	1.8249×10^{-7}	4.5639×10^{-8}
Order		2.0129	2.0003	1.9993	1.9994
Glob. err. $u'(t)$	1.9962×10^{-4}	4.8801×10^{-5}	1.2060×10^{-5}	2.9991×10^{-6}	7.4888×10^{-7}
Order		2.0322	2.0167	2.0076	2.0017

Table 8. Local and global errors and orders for problem (3) with RN3.

τ	1/80	1/160	1/320	1/640	1/1280
Loc. err. $u(t)$	2.1291×10^{-7}	1.7820×10^{-8}	1.2296×10^{-9}	7.9634×10^{-11}	5.0414×10^{-12}
Order		3.5786	3.8572	3.9487	3.9815
Loc. err. $u'(t)$	1.3417×10^{-5}	9.5553×10^{-7}	6.8098×10^{-8}	4.8287×10^{-9}	3.3715×10^{-10}
Order		3.8117	3.8106	3.8179	3.8402
Glob. err. $u(t)$	1.2456×10^{-6}	1.6021×10^{-7}	2.0110×10^{-8}	2.5093×10^{-9}	3.1001×10^{-10}
Order		2.9587	2.9940	3.0026	3.0169
Glob. err. $u'(t)$	1.7959×10^{-5}	2.4474×10^{-6}	3.5574×10^{-7}	5.0192×10^{-8}	6.8569×10^{-9}
Order		2.8754	2.7824	2.8253	2.8718

Table 9. Local and global errors and orders for problem (3) with RN4.

τ	1/80	1/160	1/320	1/640	1/1280
Loc. err. $u(t)$	1.7070×10^{-6}	6.1236×10^{-8}	2.1793×10^{-9}	7.7642×10^{-11}	2.7480×10^{-12}
Order		4.8010	4.8125	4.8109	4.8204
Loc. err. $u'(t)$	4.8165×10^{-5}	3.4176×10^{-6}	2.4071×10^{-7}	1.6643×10^{-8}	1.1149×10^{-9}
Order		3.8170	3.8276	3.8543	3.8999
Glob. err. $u(t)$	3.0516×10^{-6}	1.6639×10^{-7}	6.6974×10^{-9}	2.8418×10^{-10}	1.4797×10^{-11}
Order		4.1969	4.6348	4.5587	4.2634
Glob. err. $u'(t)$	2.1788×10^{-4}	2.5549×10^{-5}	2.8936×10^{-6}	3.1875×10^{-7}	3.4444×10^{-8}
Order		3.0922	3.1423	3.1824	3.2101

Table 10. Computational cost for problem (3) with RNRN2, RN3, RN4, and RKN3 4.

τ	CPU RN2	Error RN2	CPU RN3	Error RN3	CPU RN4	Error RN4	CPU RKN3	Error RKN3
1/80			7.32×10^{-3}	1.24×10^{-6}	9.66×10^{-3}	3.05×10^{-6}	1.72×10^{-2}	6.34×10^{-7}
1/160	1.04×10^{-2}	2.92×10^{-6}	1.60×10^{-2}	1.60×10^{-7}	2.05×10^{-2}	1.66×10^{-7}	3.13×10^{-2}	8.04×10^{-8}
1/320	1.88×10^{-2}	7.30×10^{-7}	2.70×10^{-2}	2.01×10^{-8}	3.72×10^{-2}	6.70×10^{-9}	5.83×10^{-2}	1.01×10^{-8}
1/640	3.70×10^{-2}	1.82×10^{-7}	5.49×10^{-2}	2.51×10^{-9}	7.07×10^{-2}	2.84×10^{-10}	1.12×10^{-1}	1.26×10^{-9}
1/1280	7.23×10^{-2}	4.56×10^{-8}	1.02×10^{-1}	3.10×10^{-10}	1.36×10^{-1}	1.48×10^{-11}	2.13×10^{-1}	1.53×10^{-10}
1/2560	1.34×10^{-1}	1.14×10^{-8}						
1/5120	2.61×10^{-1}	2.86×10^{-9}						
1/10240	5.12×10^{-1}	7.17×10^{-10}						
1/20480	1.03×10^0	1.82×10^{-10}						
1/40960	2.04×10^0	4.86×10^{-11}						

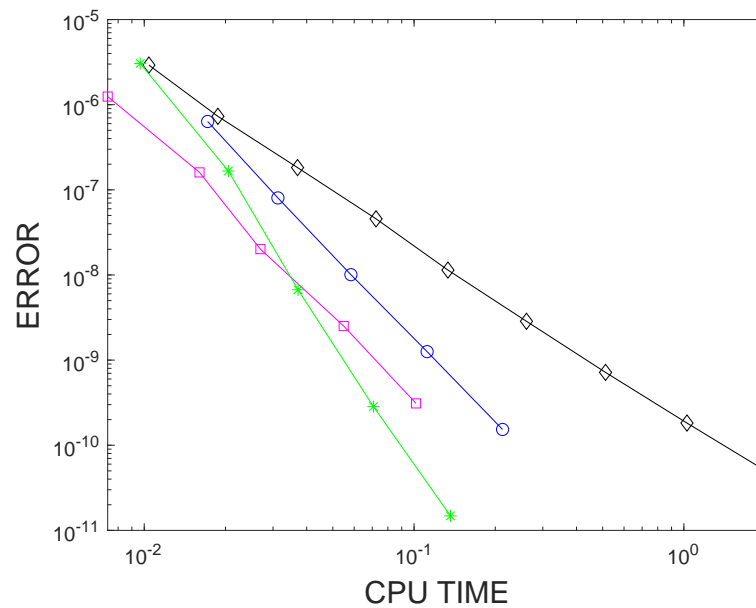


Figure 1. Graph for the global error in the solution with RK2 (\diamond , black), RK3 (\square , magenta), RK4 ($*$, green), and RKN3 (o , blue).

6. Conclusions and Future Work

As we have seen in this work, these new methods are a good choice when we have a second-order in time problem to solve. By using these methods, we avoid the drawback that other integrators like Runge–Kutta–Nyström ones present when the problem we are solving is a non-linear one.

This work is an introductory work about these methods and there is more work to do in the near future. For example, we have the problem of the order reduction that this type of method presents when we solve a second-order in time PDE with non-vanishing order conditions. This problem has already been studied for other existing methods such as Runge–Kutta, Rosenbrock, and Runge–Kutta–Nyström.

Another piece of work to be done is the construction and study of Rosenbrock–Nyström methods with complex coefficients. As we have seen, the only stable method that exists with only one stage is a P -stable one. To obtain R -stable methods (not P -stable) with classical order two, we should have two intermediate stages. Rosenbrock methods with complex coefficients have been proven to be very efficient when solving stiff problems [16,23]. In this way, one piece of future work is to use this idea of complex coefficients to obtain efficient Rosenbrock–Nyström methods to solve second-order in time problems.

To conclude the work to be done, we are also interested in the comparison between different Rosenbrock–Nyström methods, in order to study the best method to be used in practice. One high-stage scheme could be worse in practice than one method with fewer stages because of machine round-off errors.

Funding: This research was funded by the Ministerio de Ciencia e Innovación through project PGC2018-101443-B-I00.

Acknowledgments: The author acknowledges the useful comments given by the referees.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bujanda, B.; Jorge, J.C. Fractional Step Runge-Kutta methods for time dependent coefficient parabolic problems. *Appl. Numer. Math.* **2003**, *45*, 99–122. [[CrossRef](#)]
2. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I, Nonstiff Problems*, 2nd ed.; Springer: Berlin, Germany, 2000.
3. Hochbruck, M.; Ostermann, A. Exponential integrators. *Acta Numer.* **2010**, *19*, 209–286. [[CrossRef](#)]
4. Peaceman, D.W.; Rachford, H.H. The numerical solution of parabolic and elliptic differential equations. *J. SIAM* **1955**, *3*, 28–42. [[CrossRef](#)]
5. Rosenbrock, H.H. Some general implicit processes for the numerical solution of differential equations. *Comput. J.* **1963**, *5*, 329–330. [[CrossRef](#)]
6. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, 2nd ed.; Springer: Berlin, Germany, 1996.
7. Hochbruck, M.; Ostermann, A.; Schweitzer, J. Exponential Rosenbrock-type methods. *SIAM J. Numer. Anal.* **2008**, *47*, 786–803. [[CrossRef](#)]
8. Moreta, M.J. Discretization of Second-Order in Time Partial Differential Equations by Means of Runge-Kutta-Nyström Methods. Ph.D. Thesis, Department of Applied Mathematics, University of Valladolid, Valladolid, Spain, 2005.
9. Moreta, M.J.; Bujanda, B.; Jorge, J.C. Numerical resolution of linear evolution multidimensional problems of second order in time. *Numer. Methods Partial. Differ. Equ.* **2012**, *28*, 597–620.
10. Goyal, S.; Serbin, S.M. A class of Rosenbrock-type schemes for second-order nonlinear systems of ordinary differential equations. *Comput. Math. Appl.* **1987**, *13*, 351–362. [[CrossRef](#)]
11. Moreta, M.J. Construction of Rosenbrock-Nyström methods up to order four. 2021, in preparation.
12. Fermi, E.; Pasta, J.; Ulam, S. Studies of nonlinear problems I. *Lect. Appl. Math.* **1974**, *15*, 143–156.
13. Goldstein, J.A. *Semigroups of Linear Operators and Applications*; Oxford University Press: New York, NY, USA, 1985.
14. Crouzeix, M. *Numerical Range and Hilbertian Functional Calculus*; Institute of Mathematical Research of Rennes, Université de Rennes: Rennes, France, 2003.
15. Alonso-Mallo, I.; Cano, B.; Moreta, M.J. Stability of Runge-Kutta-Nyström methods. *J. Comp. Appl. Math.* **2006**, *189*, 120–131. [[CrossRef](#)]
16. Al'shin, A.B.; Al'shina, E.A.; Kalitkin, N.N.; Koryagina, A.B. Rosenbrock schemes with complex coefficients for stiff and differential-algebraic systems. *Comput. Math. Math. Phys.* **2006**, *46*, 1320–1340. [[CrossRef](#)]
17. Toda, M. Waves in nonlinear lattice. *Suppl. Prog. Theor. Phys.* **1970**, *45*, 174–200. [[CrossRef](#)]
18. Bernardi, C.; Maday, Y. *Approximations Spectrales de Problèmes aux Limites Elliptiques*; Springer: Berlin, Germany, 1992.
19. Alonso-Mallo, I.; Cano, B. Spectral/Rosenbrock discretizations without order reduction for linear parabolic problems. *Appl. Numer. Math.* **2002**, *47*, 247–268. [[CrossRef](#)]
20. Alonso-Mallo, I.; Cano, B. Avoiding order reduction of Runge-Kutta discretizations for linear time-dependent parabolic problems. *BIT* **2004**, *44*, 1–20. [[CrossRef](#)]
21. Alonso-Mallo, I.; Cano, B.; Moreta, M.J. Optimal time order when implicit Runge-Kutta-Nyström methods solve linear partial differential equations. *Appl. Numer. Math.* **2008**, *58*, 539–562. [[CrossRef](#)]
22. Alonso-Mallo, I.; Cano, B.; Moreta, M.J. Stable Runge-Kutta-Nyström methods for dissipative problems. *Numer. Algorithms* **2006**, *42*, 193–203. [[CrossRef](#)]
23. Al'shin, A.B.; Al'shina, E.A.; Limonov, A.G. Two-stage complex Rosenbrock schemes for stiff systems. *Comput. Math. Math. Phys.* **2009**, *49*, 261–278 [[CrossRef](#)]