A numerical method for the solution of an autonomous initial value problem

Flavius Pătrulescu[1],

[1] *Tiberiu Popoviciu* Institute of Numerical Analysis
P.O. Box 68-1, 400110 Cluj-Napoca, Romania

Abstract. Using a known interpolation formula we introduce a class of numerical methods for approximating the solutions of scalar initial value problems for first order differential equations, which can be identified as explicit Runge-Kutta methods. We determine bounds for the local truncation error and we also compare the convergence order and the stability region with those for explicit Runge-Kutta methods, which have convergence order equal with number of stages (i.e. with 2, 3 and 4 stages). The convergence order is only two, but our methods have a larger absolute stability region than the above mentioned methods. In the last section a numerical example is provided, and the obtained numerical approximation is compared with the corresponding exact solution.

2010 Mathematics Subject Classification: 65L05, 65L06.

Keywords: initial value problem, stability region, convergence order, local truncation error

1 Introduction

Consider a scalar initial value problem (IVP):

$$\begin{cases} y' = f(x,y), \quad x \in I \\ y(x_0) = y_0, \end{cases} \tag{1.1}$$

where: $I \subseteq \mathbb{R}$, $y_0 \in \mathbb{R}$, $f : I \times \mathbb{R} \to \mathbb{R}$ and $x_0 \in I$.

We assume that the function f satisfies all requirements necessary to insure the existence of a unique solution y on the finite interval $I = [x_0, x_0 + T]$, $0 < T < \infty$, see [1] for details.

In this paper we give a method to approximate the solution of the above initial value problem using an approximation formula for functions given in [3] and [5].

One denotes by I_T the closed interval determined by the two distinct points x_0, $x_0 + T \in \mathbb{R}$, where $T \in \mathbb{R}$. For a 3-times derivable function $h : I_T \to \mathbb{R}$ consider the function $g : I_T \to \mathbb{R}$ given by:

$$g(x) = h(x_0) + (x - x_0)h'(x_0 + \tfrac{1}{2}(x - x_0)), \quad x \in I_T. \tag{1.2}$$

In [3] is showed that function g verifies:

$$h^{(i)}(x_0) = g^{(i)}(x_0), \quad i = \overline{0,2} \text{ and } |h(x) - g(x)| \leq \tfrac{7}{24} M_3 |x - x_0|^3, \quad x \in I_T, \tag{1.3}$$

where $M_3 = \sup_{x \in I_T} |h^{(3)}(x)|$.

In the next sections, for simplicity, we consider only the autonomous case, i.e. $f = f(y)$, the general case can be treated in the same manner.

The paper is structured as follows. In Section 2 we describe the numerical method together with its versions corresponding to the particular cases of the parameter. In Section 3 we study the convergence of the method and in Section 4 we provide its stability analysis. Two particular cases are investigated in Section 5 and, finally, a numerical example is presented in Section 6.

2 The numerical method

We suppose that the exact solution y of initial value problem (1.1) is 3-times differentiable (conditions for regularity of exact solution of an initial value problem can be found in [1]). Using the results described above we deduce that there exists an approximation $\tilde{y}$ given by

$$\begin{aligned} \tilde{y}(x) &= y(x_0) + (x - x_0)y'(x_0 + \tfrac{1}{2}(x - x_0)) = \\ &= y(x_0) + (x - x_0)f(y(x_0 + \tfrac{1}{2}(x - x_0))), \end{aligned} \tag{2.1}$$

for all $x \in [x_0, x_0 + T]$. From (1.3) this approximation verifies

$$\tilde{y}^{(i)}(x_0) = y^{(i)}(x_0), \quad i = \overline{0,2}, \text{ and } |\tilde{y}(x) - y(x)| < \tfrac{7}{24} M_3 |x - x_0|^3, \quad x \in I,$$

where $M_3 = \sup_{x \in I} |y^{(3)}(x)|$.

Repeating (1.2), the unknown quantity $y(x_0 + \frac{1}{2}(x - x_0))$ in (2.1) can be approximated in the same manner and we obtain

$$\begin{aligned} y(x_0 + \tfrac{1}{2}(x - x_0)) &= y(x_0) + \tfrac{1}{2}(x - x_0)y'(x_0 + \tfrac{1}{2^2}(x - x_0)) = \\ &= y(x_0) + \tfrac{1}{2}(x - x_0)f(y(x_0 + \tfrac{1}{2^2}(x - x_0))). \end{aligned}$$

Continuing this procedure for $y(x_0+\frac{1}{2^2}(x-x_0))$ and for the next unknown values of y, after p steps, we obtain:

$$\begin{aligned}\tilde{y}(x) &= y(x_0)+(x-x_0)f(y(x_0)+\tfrac{1}{2}(x-x_0)f(y(x_0)+\tfrac{1}{2^2}(x-x_0)f(y(x_0)+\\ &+ \ldots+\tfrac{1}{2^{p-1}}(x-x_0)f(y(x_0+\tfrac{1}{2^p}(x-x_0))\ldots). \end{aligned} \tag{2.2}$$

We can write (2.2) in a more suitable form using the recurrence:

$$\begin{aligned} u_0(x) &= y(x_0+\tfrac{1}{2^p}(x-x_0)) \\ u_1(x) &= y(x_0)+\tfrac{1}{2^{p-1}}(x-x_0)f(u_0(x)) \\ \ldots &\quad \ldots \quad \ldots \quad \ldots \\ u_p(x) &= y(x_0)+(x-x_0)f(u_{p-1}(x)), \end{aligned} \tag{2.3}$$

and we can define $\tilde{y}(x)=u_p(x)$, $x\in[x_0,x_0+T]$.

Taking into account that $\frac{1}{2^p}(x-x_0)\to 0,$ as $p\to\infty$, we can choose $p=p_0$ such that $y(x_0)$ approximates $y(x_0+\frac{1}{2^{p_0}}(x-x_0))$ with any given accuracy since y is a continuous function.

For $p=p_0$ we obtain

$$\tilde{u}_0(x)=y(x_0),\quad \tilde{u}_i(x)=y(x_0)+\tfrac{1}{2^{p_0-i}}(x-x_0)f(\tilde{u}_{i-1}(x)),\quad i=1,\ldots,p_0, \tag{2.4}$$

and we use this recurrence relation to construct a numerical method for a scalar initial value problem. The continuous interval $[x_0,x_0+T]$ is partitioned by the point set

$$x_0<x_1<\ldots<x_N=x_0+T. \tag{2.5}$$

Replacing x by x_1 in (2.4) we obtain an approximation $\tilde{u}_{p_0}(x_1)$ for the exact value $y(x_1)$. We denote this approximation by y_1 and we shall apply the algorithm (2.4) for the point x_2 but instead of $y(x_0)=y_0$ we consider the value y_1 previously computed. By repeating this procedure we obtain for each x_s, $s=\overline{1,N}$ an algorithm that can be described in the following way:

$$y_{s+1}=u^s_{p_0},\quad s=\overline{1,N-1} \tag{2.6}$$

where $u^s_{p_0}$ is obtained by the following method

$$u^s_0=y_s,\quad u^s_i=y_s+\tfrac{1}{2^{p_0-i}}(x_{s+1}-x_s)f(u^s_{i-1}),\quad i=\overline{1,p_0} \tag{2.7}$$

and y_s represents the exact value or an approximation of y at $x=x_s$.

Thus, we obtain a numerical method given by

$$y_{s+1}=y_s+h_sf(y_s+\tfrac{1}{2}h_sf(y_s+\tfrac{1}{2^2}h_sf(y_s+\ldots+\tfrac{1}{2^{p_0-1}}h_sf(y_s))\ldots), \tag{2.8}$$

where $h_s=x_{s+1}-x_s$, $s=0,\ldots,N-1$ represents the length of the step.

We have the following equivalence result.

Theorem 2.1 *The method (2.8) is equivalent with a p_0 stages explicit Runge-Kutta method with the Butcher array given by:*

$$\begin{array}{c|ccccccc}
 & 0 & & & & & & \\
\frac{1}{2^{p_0-1}} & \frac{1}{2^{p_0-1}} & 0 & & & & & \\
\frac{1}{2^{p_0-2}} & 0 & \frac{1}{2^{p_0-2}} & 0 & & & & \\
\vdots & & \ldots & & & & & \\
\frac{1}{2^2} & 0 & 0 & 0 & \ldots & \frac{1}{2^2} & 0 & \\
\frac{1}{2} & 0 & 0 & 0 & \ldots & 0 & \frac{1}{2} & 0 \\
\hline
 & 0 & 0 & 0 & \ldots & 0 & 0 & 1
\end{array}$$

Proof. Following [2], for a p_0 stages explicit Runge-Kutta method the Butcher array can be represented as follows:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Here $A = (a_{ij})$, $i, j = \overline{1, p_0}$ is strictly a inferior lower matrix (i.e. $a_{ij} = 0$ for $j \geq i$), $b^T = [b_1, \ldots, b_{p_0}]$ and $c^T = [c_1, \ldots, c_{p_0}]$, $0 \leq c_i < 1$. The values k_i, $i = \overline{1, p_0}$, for the intermediate points $x_s + c_i h_s$ between x_s and x_{s+1}, $s = \overline{0, N-1}$ are defined as follows:

$$k_1 = f(x_s, y_s), \quad k_i = f(x_s + c_i h_s, y_s + h_s \sum_{j=1}^{i-1} a_{ij} k_j), i = \overline{2, p_0}.$$

The approximation for the new point $x_{s+1} = x_s + h_s$ is given by

$$y_{s+1} = y_s + h_s \sum_{i=1}^{p_0} b_i k_i.$$

In the autonomous case, for the above Butcher array we obtain

$$k_1 = f(y_s), \quad k_i = f(y_s + h_s \tfrac{1}{2^{p_0-i+1}} k_{i-1}), \quad i = \overline{2, p_0},$$

and, therefore,

$$y_{s+1} = y_s + h_s k_{p_0}. \tag{2.9}$$

It is easy to see that (2.9) provides a rule of the form (2.8), which concludes the proof. □

Next, we present some particular cases, obtained for different values of the parameter p_0. For $p_0 = 1$ we obtain the *Euler forward method* given by

$$y_{s+1} = y_s + h_s f(y_s), \quad s = \overline{1, N-1}.$$

For $p_0 = 2$ we obtain the *Midpoint rule* given by

$$y_{s+1} = y_s + h_s f(y_s + \tfrac{1}{2} h_s f(y_s)), \quad s = \overline{1, N-1}.$$

For $p_0 = 3$ we obtain the method

$$y_{s+1} = y_s + h_s f(y_s + \tfrac{1}{2} h_s f(y_s + \tfrac{1}{2^2} h_s f(y_s))), \quad s = \overline{1, N-1} \tag{2.10}$$

with the Butcher array

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

For $p_0 = 4$ we obtain the method

$$y_{s+1} = y_s + h_s f(y_s + \tfrac{1}{2} h_s f(y_s + \tfrac{1}{2^2} h_s f(y_s + \tfrac{1}{2^3} h_s f(y_s)))), \quad s = \overline{1, N-1} \tag{2.11}$$

with the Butcher array

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2^3} & \frac{1}{2^3} & 0 & & \\ \frac{1}{2^2} & 0 & \frac{1}{2^2} & 0 & \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ \hline & 0 & 0 & 0 & 1 \end{array}$$

In the next sections, for the methods (2.10) and (2.11), we study the local truncation error, stability, consistency, convergence order and we compare them with others known numerical methods.

3 Local Truncation error

In this section we study the local truncation error and convergence order for the method (2.8) when $p_0 \geq 3$, and we determine the bounds for the coefficient of principal local truncation error, as well.

As in [4], we suppose that

$$\|f\| < M \text{ and } \|f^{(j)}\| < \frac{L^j}{M^{j-1}} \text{ on } [x_0, x_0 + T], \tag{3.1}$$

where $\|f\| = \sup\{|\mathrm{f}(t)| : t \in I\}$ and M, L are positive real numbers.

The convergence order of the method is provided in the following result.

Theorem 3.1 *The method (2.8) has convergence order* 2 *and the coefficient of principal local truncation error C_3 has the following bound*

$$\|C_3\| \leq \tfrac{1}{12} M L^2. \tag{3.2}$$

Proof. In order to obtain the local truncation error of the method (2.8), following [2] we consider the operator

$$\mathcal{L}[z(x), h] = z(x+h) - z(x) - hf(z(x) + \tfrac{1}{2} hf(z(x) + \ldots + \tfrac{1}{2^{p_0-1}} hf(z(x)) \ldots), \tag{3.3}$$

where z is an arbitrary function defined on $[x_0, x_0+T]$, 3-times differentiable at least and $z'(x) = f(z(x))$, $x \in [x_0, x_0+T]$. For $p_0 \geq 3$, using the Taylor series with respect x we obtain

$$\mathcal{L}[z(x), h] = \tfrac{1}{24}h^3[f(z(x))(f'(z(x)))^2 + f^2(z(x))f''(z(x))] + O(h^4).$$

Using the definition for the convergence order given in [2] we deduce that the method has second-order of accuracy. Also, substituting z by the exact solution y, x by x_s, and supposing the *localizing assumption* $y_i = y(x_i)$, $i = \overline{1,s}$ then the *local truncation error* of the method (see [2]) can be written as

$$T_{s+1} = \frac{1}{24}h^3[f(y(x_s))(f'(y(x_s)))^2 + f^2(y(x_s))f''(y(x_s))] + O(h^4), \tag{3.4}$$

and the coefficient of *principal local truncation error* is given by

$$C_3 = \frac{1}{24}[f(y(x_s))(f'(y(x_s)))^2 + (f(y(x_s)))^2 f''(y(x_s))]. \tag{3.5}$$

Then, using (3.1) we have for C_3 the bound

$$\|C_3\| = \tfrac{1}{24}\|f''f^2 + (f')^2 f\| \leq \tfrac{1}{24}[\tfrac{L^2}{M}M^2 + L^2 M] = \tfrac{1}{12}ML^2,$$

which concludes the proof. □

Note that the method defined above is a *zero-stable* method because it verifies *root-condition*. Also, since the convergence order is 2 we conclude that it satisfies the *consistency condition*. It follows that our method represents a convergent method, see [2] for details.

4 Stability analysis

In this section we define the stability function and the absolute-stability region for the method (2.8), when $p_0 \geq 3$. Our main result is the following.

Theorem 4.1 *The method (2.8) has the stability function given by:*

$$R(q) = 1 + \sum_{k=1}^{p_0} \frac{1}{2^{\frac{k(k-1)}{2}}} q^k, \quad q \in \mathbb{C}. \tag{4.1}$$

Proof. We apply (2.8) to scalar test equation (see [2])

$$y' = \lambda y, \quad \lambda \in \mathbb{C}, \quad Re\lambda < 0,$$

and we obtain the difference equation:

$$y_{s+1} = [1 + h\lambda + \tfrac{1}{2}(h\lambda)^2 + \tfrac{1}{2^3}(h\lambda)^3 + \ldots + \frac{1}{2^{\frac{p_0(p_0-1)}{2}}}(h\lambda)^{p_0}]y_s.$$

Denoting $q = h\lambda$ the stability function is obtained as: $R(q) = 1 + \sum_{k=1}^{p_0} \frac{1}{2^{\frac{k(k-1)}{2}}} q^k$. $\square$

The *absolute-stability region* (see [2]) is given by

$$\mathcal{R} = \{q \in \mathbb{C} : |R(q)| < 1\} = \{q \in \mathbb{C} : |1 + \sum_{k=1}^{p_0} \frac{1}{2^{\frac{k(k-1)}{2}}} q^k| < 1\}.$$

We study this region in the next section for the methods (2.10) and (2.11).

5 Method (2.8) when $p_0 = 3$ and $p_0 = 4$

We restrict ourselves to methods (2.10) and (2.11), which will be studied in the following section. For $p_0 \geq 5$ methods obtained from (2.8) have a form much more complicated with a high cost of calculus and the results concerning the accuracy are not so outstanding, because from (3.4) we deduce that the convergence order is only 2.

The method (2.10), obtained for $p_0 = 3$ and defined by

$$y_{s+1} = y_s + h_s f(y_s + \tfrac{1}{2} h_s f(y_s + \tfrac{1}{2^2} h_s f(y_s))) \tag{5.1}$$

requires three evaluations of the function f for each mesh point.

From (3.4) we deduce that the convergence order for this method is 2, unlike the 3-stages explicit Runge-Kutta methods of order 3 which also require three function evaluations.

But this method has an advantage concerning absolute stability region. From (4.1) we deduce that the absolute stability function for (2.10) is defined by

$$R(q) = 1 + q + \tfrac{q^2}{2} + \tfrac{q^3}{8}. \tag{5.2}$$

Using the *scanning technique* (see [2]) the absolute stability region is plotted, using Matlab software, in Figure 1 (a) with continuous line. We know that the absolute stability function for 3 stages explicit Runge-Kutta methods of order 3 (see [2]) is defined by

$$R(q) = 1 + q + \tfrac{q^2}{2} + \tfrac{q^3}{6} \tag{5.3}$$

and the absolute stability region is also plotted, using the same technique as above, in Figure 1 (a) with dashed line. We observe that the absolute stability region for method (2.10) is larger than absolute stability region for 3 stages explicit Runge-Kutta methods of order 3. The method (2.11), obtained for $p_0 = 4$ and defined by

$$y_{s+1} = y_s + h_s f(y_s + \tfrac{1}{2} h_s f(y_s + \tfrac{1}{2^2} h_s f(y_s + \tfrac{1}{2^3} h_s f(y_s)))) \tag{5.4}$$

requires four evaluations of the function f for each mesh point.

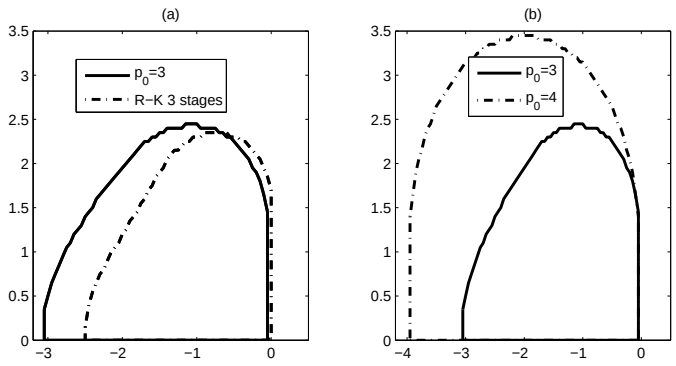

Figure 1: Absolute stability regions

From (3.4) we deduce that the convergence order for this method is 2, and concerning the accuracy it does not compare with 4 stages explicit Runge-Kutta methods of order 4 which also require four function evaluations.

But this method has an advantage concerning absolute stability region. From (4.1) we deduce that the absolute stability function for (2.11) is defined by

$$R(q) = 1 + q + \frac{q^2}{2} + \frac{q^3}{2^3} + \frac{q^4}{2^6} \tag{5.5}$$

and the absolute stability region is plotted in Figure 1 (b) with dashed line. In Figure 1 (b) is also plotted with continuous line the absolute stability region for method (2.10) and we observe that the absolute stability region for method (2.11) is larger than absolute stability region for method (2.10).

We know that the absolute stability function for 4 stages explicit Runge-Kutta methods of order 4 (see [2]) is defined by

$$R(q) = 1 + q + \frac{q^2}{2} + \frac{q^3}{6} + \frac{q^4}{24} \tag{5.6}$$

and the absolute stability region is plotted in Figure 2 (a) with continuous line. In Figure 2 (a) is also plotted the absolute stability region for the method (2.11) with dashed line and we observe that the absolute stability region for method (2.11) is larger than absolute stability region for 4 stages explicit Runge-Kutta methods of order 4.

6 Numerical example

To test the performance of the proposed methods, (2.10) and (2.11), we consider the autonomous initial value problem:

$$\begin{cases} y'(x) = \cos^2(y(x)), \quad x \in [0, 20], \\ y(0) = 0. \end{cases} \tag{6.1}$$

The exact solution of the problem is $y : [0, 20] \to \mathbb{R}$, $y(x) = \arctan(x)$ and is plotted, for the interval $[0, 5]$, in Figure 2 (b) with continuous line. The numerical solutions,

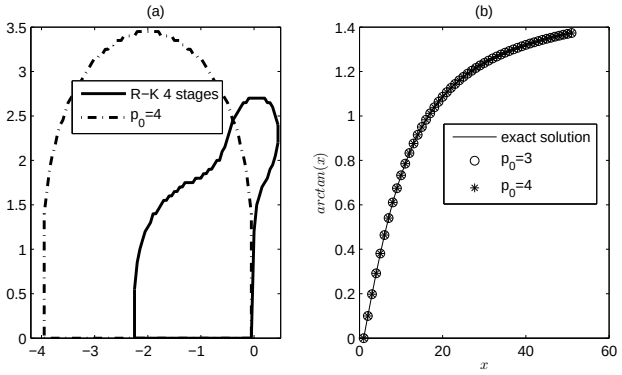

Figure 2: (a) Absolute stability regions (b) Exact solution and numerical solutions for $h = 0.1$

obtained with the methods (2.10) and (2.11), for the fixed step size $h = 0.1$, are also plotted, using Matlab software, in Figure 2 (b) with circle marker and star marker respectively. We observe a very good agreement between exact solution and numerical solutions.

In Table 1 are presented the results for the method (2.10) and (2.11) and Midpoint rule for different values of h. The errors have been obtained as the maximum of the absolute errors on the mesh points $x_s = sh$, $s = \overline{1, N}$:

$$E_{\max} = \max\{|y(x_s) - y_s| : s = 0, 1, \ldots, N\}.$$

h	*Midpoint*	(2.10)	(2.11)
0.1	$4.527354e-4$	$2.289041e-4$	$2.279995e-4$
0.01	$4.255123e-6$	$2.261048e-6$	$2.260270e-6$
10^{-3}	$4.228619e-8$	$2.257633e-8$	$2.257555e-8$
10^{-4}	$4.225559e-10$	$2.257583e-10$	$2.257574e-10$
10^{-5}	$4.998224e-12$	$2.207456e-12$	$2.207456e-12$
10^{-6}	$2.032729e-11$	$2.032796e-11$	$2.032796e-11$

We note that when length of the step decreases 10 times then the error magnitude decreases 100 times. This result represents a validation of the fact that the convergence order for methods (2.10) and (2.11) is 2.

References

[1] Crouzeix, M., Mignot, A.L., *Analyse numérique des équations différentielles*, Masson, Paris, 1989.

[2] Lambert, J. D., *Numerical Methods for Ordinary Differential Systems-The Initial Value Problem*, John Wiley&Sons, 1990.

[3] Păvăloiu, I., *On an approximation formula*, Rev. Anal. Numér. Théor. Approx., **26** (1997), ns. 1-2, pp. 179-183.

[4] Ralston, A., *Runge-Kutta methods with minimum error bounds*, Math. Comp.,**16** (1962), no. 80, pp. 431–437.

[5] Traub, J.F., *Iterative Methods for the Solution of Equations*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1964.