



An adaptive stepsize algorithm for the numerical solving of initial-value problems

Romulus MILITARU

Abstract

The present paper focuses on the efficient numerical solving of initial-value problems (IVPs) using digital computers and one-step numerical methods. We start from considering that the integration stepsize is the crucial factor in determining the number of calculations required and the amount of work involved to obtain the approximate values of the exact solution of a certain problem for a given set of points, within a prescribed computational accuracy, is proportional to the number of accomplished iterations. We perform an analysis of the local truncation error and we derive an adaptive stepsize algorithm which coupled with a certain one-step numerical method makes the use of this structure more computationally effective and insures that the estimated values of the exact solution are in agreement with an imposed accuracy. We conclude with numerical computations proving the efficiency of the proposed step selection algorithm.

1 Introduction

A lot of problems in engineering and science can be formulated in terms of differential equations. Differential equations occur in the description of many physical phenomena, for example: a condenser discharging, a gas escaping under pressure from a container, heat flow involved by the high temperatures

Key Words: initial value problems, adaptive numerical methods, local truncation error approach, one-step numerical methods, numerical computation.

2010 Mathematics Subject Classification: Primary 65L05, 65L70, 65L12.

Received: 2 May, 2014

Revised: 25 June, 2014

Accepted: 28 June, 2014

that act on a porous material, uninhibited bacteriological growth, and recently in antropology, [1],[9], [12],[13]. Most of these problems require the solution to an initial-value problem IVP, that is, the solution to a differential equation that satisfies a given initial condition. The majority real-world initial-value problems are so complicated that they cannot be solved using analytical methods. Therefore, we must resort to numerical methods, which are designed to calculate approximate values of the solution function by some process requiring a finite number of steps, using a computer code. The numerical treatment of the initial value problems is an intensively researched field, the ability to numerically approximate the exact solution being the only way to obtain information about the trajectory. The high-efficiency computers (speed and memory), their wide distribution and the computer software allow us to create efficient adaptive numerical algorithms for solving IVPs, thus giving the possibility to study and predict a large range of realistic problems. In addition, the numerical processing allows the treatment of particular conditions which are many times impossible to simulate in experimental tests.

2 Problem statement

Let the first order initial-value problem:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases} \quad (1)$$

where $f : [a, b] \times \mathbb{R} \rightarrow \mathbb{R}$, $a < b$, is a real valued function of two variables, continuous and satisfying a Lipschitz condition with respect to variable y , $x_0 \in [a, b]$ and $y = y(x)$ a real valued function defined on $[a, b]$. The imposed regularity conditions for the function f guarantee that the initial-value problem (1) possesses an unique solution, [4],[8],[18].

The numerical solving of an IVP may be accomplished by the mean of two main techniques, [7]:

* construction of a function $\bar{y}(x)$ belonging to a certain class of standard functions and which approximates "sufficiently" smooth the exact solution $y(x)$ over some range of values x , that is $|y(x) - \bar{y}(x)| \leq \varepsilon$, ε being the accuracy imposed by the user;

* approximation of the values of the exact solution $y(x)$ for a certain discrete abscissae x_i , $i = 1, \dots, n$, $n \in \mathbb{N}^*$. Usually the chosen points form an arithmetic series with stepsize h .

We will focus on the second type of numerical techniques. Particularly we consider the case of one-step methods, whose main feature is that they involve for the approximation of the exact value of the solution in a certain point

only information from the previous point. Such a method has the following algorithm, [2],[18],[17]:

using the initial values (x_0, y_0) of the (IVP) and a stepsize $h \in \mathbb{R}^*$ we get the approximate values ω_i of the exact solution for the given discrete points $x_i, i = 1, 2, \dots, n$, as follows:

for $i = 0, 1, \dots, n - 1$ do
 $x_{i+1} = x_i + h;$
 $\omega_{i+1} = \omega_i + h\Phi(x_i, \omega_i; h);$

where $\omega_j := \omega(x_j, h)$ denotes the approximate values of $y(x_j)$ and the analytical expression of the function $\Phi = \Phi(x, y; h) : [a, b] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ depends on the chosen numerical method and we suppose it being continue.

Examples: Euler methods, Trapezoidal method, Runge-Kutta methods, [10].

In practice it is often needed to find the approximation values of the exact solution within a given precision and that leads to another problem: choosing an adequate value for the stepsize. In consequence an important problem for the numerical solving of an IVP is to control the global error. Unfortunately this is usually not possible. Instead, we are looking to estimate and control the local truncation error, which is done by adjusting the stepsize h so that the local truncation error remains below a given precision.

Generally, an adaptive stepsize searching algorithm should put more grid points in the regions where the solution undertakes a great variability.

3 Stepsize selection and practical implementation

In the sequel we assume that the considered one-step method is of order p , meaning that the local truncation error:

$$\tau_{i+1}(h) = \frac{1}{h}(y(x_i + h) - y(x_i)) - \Phi(x_i, y(x_i); h)$$

verifies $\tau_{i+1}(h) = O(h^p), (\forall)i = 0, 1, \dots, n - 1, [2],[9],[11]$

According to these formulations, Euler method is a first order method, Trapezoidal method is a second order method. There are Runge-Kutta methods of second, third, fourth order, [3],[19].

Remark 1. *Considering that $y(x_i) \approx \omega_i$, for a certain $i = 1, 2, \dots$, and using the expression of the local truncation error, it follows for a chosen stepsize h :*

$$\begin{aligned} \tau_{i+1}(h) &= \frac{1}{h}(y(x_{i+1}) - \omega_i) - \Phi(x_i, \omega_i; h) = \\ &= \frac{1}{h}(y(x_{i+1}) - (\omega_i + h\Phi(x_i, \omega_i; h))) = \frac{1}{h}(y(x_{i+1}) - \omega_{i+1}) \end{aligned} \quad (2)$$

Choosing stepsize h , function of the precision desired in the approximation of the values of the exact solution, represents the main problem.

Generally speaking there are two ways to improve the accuracy of the approximations, [5]:

- reduce the stepsize h so as to lower the local truncation error. Then the amount of steps increases, leading to increased round-off error and also the amount of work and in consequence to a computational inefficiency;
- use a method of higher convergency order. Increase of the convergency order p is reasonable only up to some limit; for example in the case of Runge-Kutta methods, for $s > 5$ there are no explicit methods with the convergency order $p = s$, s being the number of stages (number of function evaluations per step) of the considered method, [3].

In consequence when we use the numerical computation to estimate the exact values of the solution of a first order IVP, for particular values of the independent variable x , we are interested to determine a stepsize h , as large as possible, in order that the local truncation error remains below an accuracy ε imposed by user.

Let $F_q(a, b)$ be the set of functions for which all partial derivatives up to including order q exist and are continuous and bounded on $[a, b]$. There is the following general result, [15]:

Theorem 1. (Gragg, 1963) *Let $f \in F_{N+2}(a, b)$ and let $\omega(x, h)$ be the approximate solution obtained by a one-step method of order p , $p \leq N$, to the solution $y(x)$ of the initial value problem (1). Then $\omega(x, h)$ has an asymptotic expansion of the form:*

$$\omega(x, h) = y(x) + h^p e_p(x) + h^{p+1} e_{p+1}(x) + \dots + h^N e_N(x) + h^{N+1} \tilde{e}_{N+1}(x, h) \quad (3)$$

where $e_k(x_0) = 0$, $k = p, p+1, \dots$, which is valid for all $x \in [a, b]$ and all $h = h_N = (x - x_0)/N$, $N = 1, 2, \dots$. The functions $e_i(x)$ therein are independent of h and the remainder term $\tilde{e}_{N+1}(x, h)$ is bounded for fixed x and all $h = h_N = (x - x_0)/N$, $N = 1, 2, \dots$

Thus for a sufficiently small h and $e_p(x) \neq 0$, based on (3) we get as a first approximation:

$$\omega(x, h) - y(x) = h^p e_p(x) + O(h^{p+1}) \quad (4)$$

From (2) and (4) it results that:

$$|\tau_{i+1}(h)| \approx |h^{p-1} e_p(x_{i+1})| \quad (5)$$

Assuming that x_{i+1} is a certain point denoted by x and the point x_i where the method is supposed to be exact ($y(x_i) \approx \omega_i$) is considered to be x_0 , then

using a Taylor development about the point x_0 , since $e_p(x_0) = 0$, we obtain from (5):

$$|\tau_{i+1}(h)| \approx |(x - x_0)h^{p-1}e'_p(x_0)| = |h^p e'_p(x_0)| \quad (6)$$

Proposition 1. For a fixed order p and $(\forall)m \geq 1$ there is the following estimation:

$$(\omega(x, h/m) - \omega(x, h/(m+1))) \leq e_p(x) \left(\frac{h}{m+1}\right)^p \cdot \frac{1}{m^2} (p(m-1) + 2^p - 1) \quad (7)$$

Proof: Using (4) we get:

$$\begin{aligned} \omega(x, h/m) - y(x) &= e_p(x) \left(\frac{h}{m}\right)^p + O(h^{p+1}) \\ \omega(x, h/(m+1)) - y(x) &= e_p(x) \left(\frac{h}{m+1}\right)^p + O(h^{p+1}) \end{aligned}$$

Thus, it follows:

$$\begin{aligned} \omega(x, h/m) - \omega(x, h/(m+1)) &= e_p(x) \left(\frac{h}{m+1}\right)^p \left(\left(1 + \frac{1}{m}\right)^p - 1 \right) = \\ &= e_p(x) \left(\frac{h}{m+1}\right)^p \frac{1}{m} \left(1 + \sum_{k=1}^{p-1} \left(1 + \frac{1}{m}\right)^k \right) \end{aligned}$$

Based on the inequality

$$(1+a)^t \leq 1 + (2^t - 1)a, \quad (\forall)t \geq 1, a \in [0, 1]$$

we obtain:

$$(\omega(x, h/m) - \omega(x, h/(m+1))) \leq e_p(x) \left(\frac{h}{m+1}\right)^p \cdot \frac{p(m-1) + 2^p - 1}{m^2}$$

Taking into consideration that we are looking for an approximate value "sufficiently" close to the corresponding exact one, we will consider in the sequel:

$$e_p(x) = \left(\frac{m+1}{h}\right)^p (\omega(x, h/m) - \omega(x, h/(m+1))) \cdot \frac{m^2}{p(m-1) + 2^p - 1} \quad (8)$$

We will estimate $e'_p(x_0)$ by a second order accuracy forward finite difference formula, [6]:

$$e'_p(x_0) = \frac{-3e_p(x_0) + 4e_p(x_0 + h) - e_p(x_0 + 2h)}{2h} + O(h^2)$$

Thus, using (8) we get:

$$e'_p(x_0) \approx \frac{1}{2h^{p+1}} \frac{m^2(m+1)^p}{p(m-1) + 2^p - 1} \cdot (4(\omega(x_0 + h, h/m) - \omega(x_0 + h, h/(m+1))) - \frac{\omega(x_0 + 2h, 2h/m) - \omega(x_0 + 2h, 2h/(m+1))}{2^p}) \quad (9)$$

Let ε denote the accuracy of the approximation. Then from (6) and (9) the following statement holds:

Proposition 2. *The appropriate stepsize h which insures that the local truncation error after one step using this one remains bounded by ε , is characterized by:*

$$h \approx \frac{1}{2\varepsilon} \frac{m^2(m+1)^p}{p(m-1) + 2^p - 1} \cdot (|4(\omega(x_0 + h, h/m) - \omega(x_0 + h, h/(m+1))) - \frac{\omega(x_0 + 2h, 2h/m) - \omega(x_0 + 2h, 2h/(m+1))}{2^p}|) \quad (10)$$

Proof: Imposing $|\tau_{i+1}(h)| = \varepsilon$ and using (6) it follows that $\varepsilon \approx |h^p e'_p(x_0)|$. Thus (9) leads to:

$$\varepsilon \approx \left| \frac{1}{2h} \frac{m^2(m+1)^p}{p(m-1) + 2^p - 1} \cdot (4(\omega(x_0 + h, h/m) - \omega(x_0 + h, h/(m+1))) - \frac{\omega(x_0 + 2h, 2h/m) - \omega(x_0 + 2h, 2h/(m+1))}{2^p}) \right|$$

and eventually to the conclusion.

Concerning the practical implementation, the above adaptive stepsize strategy allows the construction of an algorithm which determines the stepsize h so that the local truncation error after one step using h remains bounded by the given precision ε :

1. Initialize x_0 , ω_0 and the starting stepsize h ; $m := 1$;

2. Evaluate:

2.1 the approximate values $\omega(x_0 + h, h/m)$, $\omega(x_0 + h, h/(m + 1))$,
 $\omega(x_0 + 2h, 2h/m)$, $\omega(x_0 + 2h, 2h/(m + 1))$, using the selected numerical
method of order p ;

2.2 the expression:

$$E = \left| \frac{1}{2h} \frac{m^2(m+1)^p}{p(m-1) + 2^p - 1} \cdot (4(\omega(x_0 + h, h/m) - \omega(x_0 + h, h/(m+1))) - \frac{\omega(x_0 + 2h, 2h/m) - \omega(x_0 + 2h, 2h/(m+1))}{2^p}) \right|$$

3. Test:

If $E \leq \varepsilon$ then replace x_0, ω_0 by $x_0 + h, \omega(x_0 + h, h)$ and return to Step 1;

Else replace h by the value given by (10):

If $x_0 + h > x$ then $h := |x - x_0|$ and return to Step 1;

Else return to Step 1.

Remark 2. (a). Initially we can consider $h := |x - x_0|$, x being the point where we are looking to estimate the value of the exact solution of the given IVP, within a certain accuracy ε ;

(b). In order to determine the approximate value of the exact solution for a certain point $x \neq x_0$, we apply the sequence Step 1 - Step 3 until $x_0 + h = x$;

(c). All the numerical methods for a single first order differential equation may be extended for the case of a system of first order differential equations, the dependent variable being replaced by a vector. Also, taking into consideration the fact that any higher differential equation can be written as a system of first order differential equations it follows that the methods used for the numerical processing of higher differential equations are identical to those for first order equations, [5],[2],[15].

4 Numerical examples

We will exemplify the functionality and the efficiency of the presented stepsize selection algorithm for the case of numerical processing of some practical IVPs.

1. As a consequence to the Kirchoff law, the intensity of the current I respects the following differential equation in the case of certain electrical circuits:

$$\frac{dI}{dt} = -\frac{R}{L}I + \frac{U_0}{L} \sin(\omega t)$$

where L, U_0, R, ω are constants Assuming $L = 1, R = 50, U_0 = 1, \omega = \pi$, respectively the initial condition $I(0) = 0$, we obtain the following IVP:

$$\begin{cases} \frac{dI}{dt} = -50I + \sin(\pi t) \\ I(0) = 0 \end{cases}$$

We are looking to estimate the values of the exact solution $I = I(t)$ at the moments $t_i = 0.1 \cdot i, i = \overline{1, 15}$, with three exact digits, by the Euler method.

Remark 3. *The exact solution is*

$$I(t) = \frac{1}{2500 + \pi^2} (50 \sin(\pi t) - \pi \cos(\pi t) + \pi e^{-50t})$$

Selecting Euler method and using the presented stepsize selection procedure, imposing the accuracy $\varepsilon = 10^{-4}$, we get the following approximations, by means of C++ code:

t	approximate value of $I(t)$	exact value of $I(t)$
0.1	0.004972	0.004974
0.2	0.010702	0.010696
0.3	0.015417	0.015380
0.4	0.018604	0.018559
0.5	0.019970	0.019921
0.6	0.019381	0.019333
0.7	0.016895	0.016852
0.8	0.012756	0.012722
0.9	0.007367	0.007346
1.0	0.001258	0.001251
1.1	-0.004992	-0.004965
1.2	-0.010698	-0.010696
1.3	-0.015383	-0.015380
1.4	-0.018564	-0.018559
1.5	-0.019979	-0.019921

2. Given the initial value problem

$$\begin{cases} y' = \frac{5}{3}y^{\frac{2}{3}} \\ y(1) = 1 \end{cases}$$

we are looking to estimate the values of the exact solution $y = y(x)$ in the data points $x_i = 1 + 0.3 \cdot i, i = \overline{1, 10}$ with three exact digits, using Euler method.

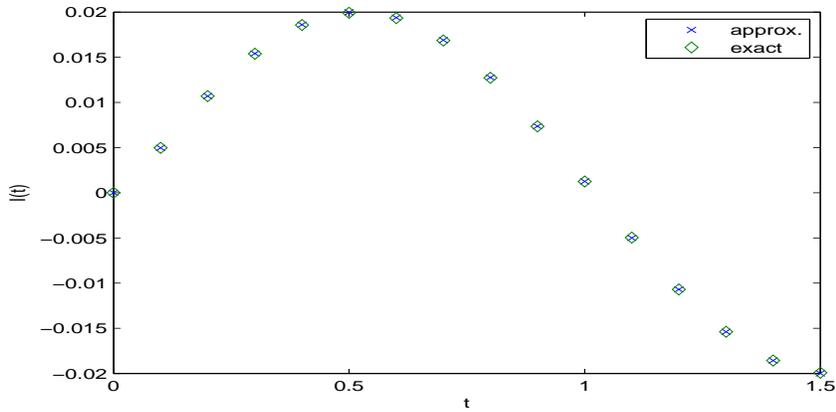


Figure 1: The graphical profile of the exact solution and the numerical one

Remark 4. *The exact solution is $y(x) = x^{5/3}$.*

The approximated values of the exact solution $y = y(x)$ for the given data points, obtained by Euler method and the above adaptive stepsize algorithm, within the given accuracy $\varepsilon = 10^{-4}$ are presented below:

x	approximate value of $y(x)$	exact value of $y(x)$
1.3	1.54846	1.54847
1.6	2.18874	2.18876
1.9	2.91462	2.91466
2.2	3.72131	3.72138
2.5	4.60495	4.60503
2.8	5.56230	5.56241
3.1	6.59062	6.59077
3.4	7.68756	7.68773
3.7	8.85102	8.85121
4.0	10.07914	10.07936

3. Given the initial value problem

$$\begin{cases} y' = -1000y + \sin(x) \\ y(0) = -0.000001 \end{cases}$$

we are looking to estimate the values of the exact solution $y = y(x)$ in the data points $x_i = 0.05 \cdot i, i = \overline{1, 150}$ with four exact digits, using a fourth order Runge-Kutta method.

Remark 5. The exact solution is $y(x) = \frac{1000 \sin(x) - \cos(x)}{1000001}$.

We choose the following fourth order Runge-Kutta method:

$$\left\{ \begin{array}{l} \text{for } i = 0, 1, \dots \\ x_{i+1} = x_i + h \\ \omega_{i+1} = \omega_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = h \cdot f(x_i, \omega_i) \\ k_2 = h \cdot f(x_i + \frac{h}{2}, \omega_i + \frac{k_1}{2}) \\ k_3 = h \cdot f(x_i + \frac{h}{2}, \omega_i + \frac{k_2}{2}) \\ k_4 = h \cdot f(x_i + h, \omega_i + k_3) \end{array} \right. \quad (11)$$

A selection of the approximated values of the exact solution $y = y(x)$ for the given data points, within the given accuracy $\varepsilon = 10^{-5}$ and obtained by using the presented stepsize strategy are presented below:

x	approximate value of $y(x)$	exact value of $y(x)$
0.05	4.8980363e-5	4.8980370e-5
0.3	2.9456453e-4	2.9456457e-4
0.55	5.2183411e-4	5.2183418e-4
0.8	7.1665857e-4	7.1665866e-4
2.5	5.99272685e-4	5.99272688e-4
3	1.42109855e-4	1.42109858e-4
4.7	-9.99909863e-4	-9.99909888e-4
5.35	-8.04114624e-4	-8.04114629e-4

4. In many practical applications, circuits consists of numerous energy-storing elements. Differential equations describing such circuits are, generally, of a high order. Consider a RLC series circuit where $L = 1H$, $R = 10\Omega$, $C = 0.25F$, and $U(t) = \sin(t)$, $t > 0$. The current $I = I(t)$ verifies the following second order differential equation:

$$I'' = -\frac{R}{L}I' - \frac{1}{LC}I + \frac{1}{L}U'$$

Considering that $I(0) = 0$, $I'(0) = 0$ we are looking to estimate the exact values of the current at the moments $t_i = 0.03 \cdot i$, $i = \overline{1, 214}$, using a Runge-Kutta IV-th order method, within the accuracy $\varepsilon = 10^{-6}$.

We can rewrite the given IVP in the general form (1) by setting:

$$\mathbf{y} = (I, v)^t$$

$$\mathbf{f} = \mathbf{f}(t, I, v) = (v, -\frac{R}{L}v - \frac{1}{LC}I + \frac{1}{L} \cos(t))^t$$

$$\mathbf{y}_0 = (I(0), v(0))^t = (0, 0)^t$$

We present a selection of approximated values of the solution $I = I(t)$, obtained choosing the same Runge-Kutta IV-th order as above, coupled with our stepsize selection procedure, imposing the accuracy $\varepsilon = 10^{-6}$.

We make a comparison with the corresponding values resulted by using Matlab.

t	approximate value of $I(t)$ obtained by presented algorithm	value of $I(t)$ predicted by Matlab
0.09	0.003055547	0.003055647
0.57	0.042154236	0.042153271
0.72	0.052480411	0.052478963
0.81	0.057772281	0.057770503
1.17	0.071412234	0.071410577
1.5	0.072724035	0.072723090
1.95	0.057852034	0.057852937
2.28	0.036724438	0.036718395
3.0	-0.025388922	-0.025387030
4.95	-0.087601896	-0.087599945
5.01	-0.084432621	-0.084431538
5.28	-0.066842206	-0.066844414
5.49	-0.049988838	-0.049988295
6.0	-0.002378324	-0.002375236
6.24	0.020669166	0.020671030
6.42	0.037120647	0.037122718

5. Given the initial value problem

$$\begin{cases} y''' = -2y'' - 5y' - y + 4 \\ y(0) = 0 \\ y'(0) = 0 \\ y''(0) = 0 \end{cases}$$

we are looking to tabulate the values of the exact solution $y = y(x)$ in the data points $x_i = 0.02 \cdot i, i = \overline{1, 50}$ within the accuracy $\varepsilon = 6 \cdot 10^{-5}$, using a fourth order Runge-Kutta method.

The above IVP can be rewritten in the general form (1) by setting:

$$\mathbf{y} = (y, v, w)^t$$

$$\mathbf{f} = \mathbf{f}(x, y, v, w) = (v, w, -2w - 5v - y + 4)^t$$

$$\mathbf{y}_0 = (y(0), v(0), w(0))^t = (0, 0, 0)^t$$

A selection of the numerical values of the exact solution $y = y(x)$, within the imposed accuracy, obtained by using the fourth order Runge-Kutta method (11) and the presented adaptive stepsize algorithm, by mean of C++ program, are given in the following table. We performed a comparison with the exact values given by Matlab.

x	approximate value of $y(x)$	exact value of $y(x)$
0.12	0.001072	0.001082
0.28	0.012511	0.012556
0.46	0.049709	0.049802
0.5	0.062209	0.062313
0.6	0.100679	0.100697
0.78	0.194788	0.194827
0.82	0.219788	0.219827
0.96	0.317416	0.317473
1	0.347879	0.347934

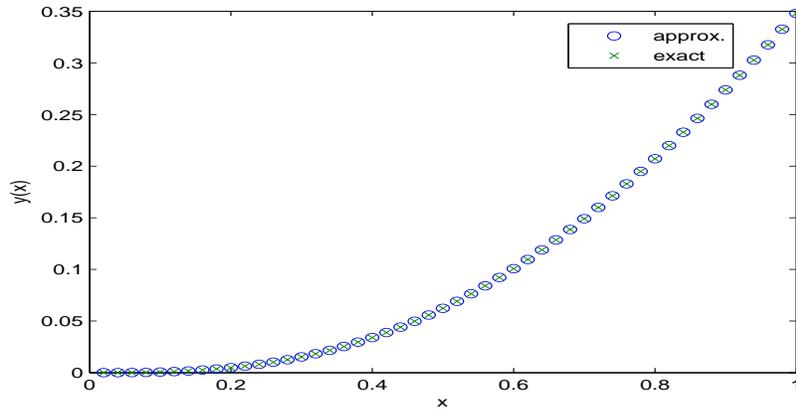


Figure 2: The graphical profile of the exact solution and the numerical one

5 Conclusions

In this paper we expose a stepsize searching algorithm based on an analysis of the local truncation error for the case of a one-step numerical method of order p , for the numerical resolution of the initial-value problems. Starting from the set of possible choices for the stepsize and taking into consideration that the amount of work involved to obtain the final results is proportional to the number of accomplished iterations, the algorithm allows to adjust this one, in such way that the estimate of the local truncation error remains smaller than a given accuracy. The numerical experiments performed by using several one-step methods coupled with the presented algorithm, for first order and higher order IVPs, prove its efficiency concerning the achievement of results which are in agreement with a precision imposed by the user. We emphasize that this algorithm can improve the accuracy of the estimated values predicted by any one-step numerical method.

References

- [1] Abraham, O., Bolarin, G., *On error estimation on Runge-Kutta methods*, Leonardo Journal of Sciences, vol. **18** (2011), pg. 1-10.
- [2] Burden, R.L., Faires, J.D., *Numerical Analysis*, Pws Publishing Company, Boston, USA, 2001.
- [3] Butcher, J.C., *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*, Wiley, 1987.
- [4] Conte, S. D., deBoor, C., *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd Ed., McGraw-Hill, New York, 1980.
- [5] Cheney, W., Kincaid, D., *Numerical Mathematics and Computing*, Brooks/Cole Publishing Company, Fourth Edition, 1995.
- [6] Chung, T.J., *Computational Fluid Dynamics*, 2-nd Edition, Cambridge University Press, 2010.
- [7] Guiplin, C., *Manuel de Calcul Numérique Appliqué*, EDP Science, 1999.
- [8] Henrici, P., *Elements of Numerical Analysis*, Wiley, 1964.
- [9] Holmes, M.H., *Introduction to Numerical Methods in Differential Equations*, Springer, 2007.
- [10] Jedrzejewski, F., *Introduction aux méthodes numériques*, Springer-Verlag, 2005.

- [11] Lambert, J.D., *Computational Methods in Ordinary Differential Equations*, Wiley, 1973.
- [12] Marin, M., Florea O., *On temporal behavior of solutions in Thermoelasticity of porous micropolar bodies*, An. St. Univ. Oidius Constanta vol. **22**, issue 1, (2014), 169-188.
- [13] Marin, M., Sharma, K., *Reflection and transmission of waves from imperfect boundary between two heat conducting micropolar thermoelastic solids*, An. St. Univ. Oidius Constanta vol. **22**, issue 2, (2014), 151-175.
- [14] *MATLAB The Language of Technical Computing*, MathWorks, 2004.
- [15] Philips, G.M., Taylor, P.J., *Theory and Applications of Numerical Analysis*, 2-nd Ed., Academic Press, 1996.
- [16] Press, H.W., Teukolsky, S.A., Vetterling, T.W., Flannery, B., *Numerical Recipes*, 3-rd Ed., Cambridge University Press, 2007.
- [17] Ricardo, H., *A modern introduction to differential equations*, Elsevier Academic Press, 2009.
- [18] Quarteroni, A., Sacco, R., Saleri, F., *Méthodes Numériques - Algorithmes, analyse et applications*, Springer, 2007.
- [19] Wu, X., *A class of Runge-Kutta formulae of order three and four with reduced evaluations of function*, Applied Mathematics and Computation **146** (2003), 417-432.

Romulus MILITARU,
Department of Applied Mathematics,
University of Craiova,
13 A.I. Cuza, 200589 Craiova, Romania.
Email: militaruromulus@yahoo.com