Emirates Journal for Engineering Research

Volume 25 | Issue 3 Article 3

6-26-2020

Numerical Solution of Ordinary Differential Equations Using Continuous Runge-Kutta Methods (Feldberg of Order Four and Five)

Madeha yousif
University of Technology, Iraq, 100049@utechnology.edu.iq

NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS USING CONTINUOUS RUNGE-KUTTA METHODS (FELDBERG OF ORDER FOUR AND FIVE)

Madeha Sh.yousif

University of Technology-Applied Science Department

100049@utechnology.edu.iq

(Received 7th April 2020 and Accepted 26nd June 2020)

طرائق رانج– كوتا المستمرة للإيجاد الحل العددي للمعادلات التفاضلية الاعتيادية (فليبرج من الرتبة الرابعة والخامسة)

الخلاصة

في هذه الورقة نستخدم طريقة رانج–كوتا المستمرة (طريقة رانج–كوتا فليبرج من الرتبة الرابعة والخامسة) للإيجاد الحل العددي للمعادلة التفاضلية الاعتيادية ليس فقط في نقاط محددة وايضا جميع النقاط بينها. وقد أُستخدم برنامج الماتلاب لحساب النتائج.

الكلمات المفتاحية: معادلات تفاضلية اعتيادية, طرائق رانج كوتا المستمرة

Abstract

In this paper the continuous Runge-Kutta method (Runge-Kutta Feldberg method of order four and five) have been used to find the numerical solution of ordinary differential equation not only at the mesh points but also the all points between them. the results are computed using matlab program..

Key words: Ordinary Differential Equations, Continuous Runge-Kutta Methods

1-Introduction

The initial value problem can be defined as

$$y' = f(x, y(x)) \qquad x \ge x_0 \qquad \ldots(1)$$
$$y(x_0) = y_0$$

Which can be solved by the following s-stage Runge-Kutta method,and s denotes the number of stages of methods [1].

the coefficients $\{a_{ij}\}$, $\{c_i\}$ and $\{b_j\}$ fully characterize Runge-Kutta method and we shall assume that the following condition holds.

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \qquad \ldots(2)$$

where

$$k_i = f(x_n + c_i h\,, y_n + h\sum_{j=1}^{s} a_{ij} k_j) \qquad ,i = 1,2,\ldots,s$$

$$c_i = \sum_{j=1}^{s} a_{ij} \quad , \qquad i=1 ,\ldots s$$

But this requires a significant amount of computation for the smaller step size and must be repeated if it is determined that the agreement is not good enough[5].

The Runge-Kutta Feldberg of order four and five method is one way to try to resolve this problem .It has a procedure to determine if the proper step size h is being used .At each step , two different approximations for the solution are made and compared .If the two results are in close agreement , the approximations is accepted. If the two results do not agree to a specified accuracy, the step size is reduced If the results agree to more significant digits than required, the step size is increased[6]

In(2000); Shampine [9] he gave Matlab program to CRK method. In (2002) Josef

conjunction with this defining algorithm, the analysis for these defining RK algorithms is local in nature, requiring no assumptions about previous approximate solution values.
Using equation (2) to find the solution of (1) $x_1=x_0+h$:

$$y(x_0+h)=y_1=y_0+h\sum_{i=0}^{s}b_i k_i \qquad \ldots(3)$$

where

$$k_0 = f(x_0, y_{01}) \qquad \ldots(4)$$

and

$$k_i = f(x_0+c_0 h, y_0+h\sum_{j=0}^{i-1}a_{ij}k_j) \qquad \ldots(5)$$

For i=1, 2,…s, where s is defined as the number of stages of the algorithm and where the a ,b and c coefficients have been selected so that the algorithmic solution, y_1 is equivalent to a Taylor sum of order p.
Specifically, subtracting a Taylor series expansion in h of y_1 from an expansion of $y(x_0+h)$ results in certain truncation error coefficients TEC i,j, as coefficients of the local truncation error, where TEC i,j term are expressions in c, a ,b and for the order m, with $i=1,2,\ldots,\lambda_m$ where λ_m increases with increasing order. For an algorithm to be of order p, all TEC i,j terms must vanish for m=1,…,p , $i=1,\ldots,\lambda_m$. The a_{ij} may be determined from the relations:

$$\sum_{j=0}^{i-1}a_{ij} = c_i \qquad \ldots(6)$$

for i=1,2,…,s
The form of the continuous algorithm parallels that of the defining algorithm since the efficiency of the new methods requires the use of derivative evaluations (4,5) as the core of the new system. One assumes that the solution has been advanced from x_0 to $x_1=x_0+h$, i.e., that $k_1,k_2,\ldots k_s$ have been determined and that the solution is required at some intermediate point, $x^*=x_0+h^*$ with

$$h^*=\delta h \qquad \ldots(7)$$

where generally the scaling factor $\delta \in (0, 1)$.
The continuous algorithm assumes the basic form of the defining algorithm

$$y(x_0+h^*)=y_1^*=y_0+h^*\sum_{i=0}^{s^*}b_i^* k_i^* \qquad \ldots(8)$$

where

$$k_0^* = f(x_0, y_0) \qquad \ldots(9)$$

and

$$k_i^* = f(x_0+c_i^* h^*, y_0+h^*\sum_{j=0}^{i-1}a_{ij}^* k_j^*) \qquad \ldots(10)$$

Diblik [7] gave some functional differential equations to solve them by CRK method.
In (2004) Mathews, and et al[6] Using Matlab for Numerical Methods In 2012, Tay and et al. [8] used A Spreadsheet to Solution System of Ordinary Differential Equations using Fourth-Order Runge-Kutta Method

$$k_1=hf(x_n, y_{n1})$$

$$k_2=hf(x_n+\frac{1}{4}h, y_n+\frac{1}{4}k_1)$$

$$k_3=hf(x_n+\frac{3}{8}h, y_n+\frac{3}{32}k_1+\frac{9}{32}k_2)$$

$$k_4=hf(x_n+\frac{12}{13}h, y_n+\frac{1932}{2197}k_1-\frac{7200}{2197}k_2+\frac{7296}{2197}k_3)$$

$$k_5=hf(x_n+h, y_n+\frac{439}{216}k_1-8k_2+\frac{3680}{513}k_3-\frac{845}{410}k_4)$$

$$k_6=hf(x_n+\frac{1}{2}h, y_n-\frac{8}{27}k_1+2k_2-\frac{3544}{2565}k_3+\frac{1859}{4104}k_4-\frac{11}{40}k_5)$$

Then an approximation to the solution of the initial value problem using a Runge-Kutta method of order four:

$$y_{n+1}=y_n+\frac{25}{216}k_1+\frac{1408}{2565}k_3\frac{2197}{4101}k_4-\frac{1}{5}k_5$$

where the four function values k_1 , k_3, k_4 and k_5 are used k_2
better value for the solution is determined using a Runge-Kutta method of order five:-

$$Z_{n+1}=y_n+\frac{16}{135}k_1+\frac{6656}{12825}k_3+\frac{28561}{56430}k_4-\frac{9}{50}k_5+\frac{2}{55}k_6$$

The optimal step size (sh) can be determined by multiplying the scalar s times the current step size h, the scalar s is

$$s=\left(\frac{tol.h}{2\left|z_{n+1}-y_{n+1}\right|}\right)^{\frac{1}{4}}\approx 0.84\left(\frac{tol.h}{\left|z_{n+1}-y_{n+1}\right|}\right)^{\frac{1}{4}}$$

where tol is the specified error control tolerance[5].

2-Construction of Continuous Runge-Kutta Methods[1,2,3,4].

We can extend the Explicit Runge-Kutta methods into Continuous Runge-Kutta method which determines the solution of the IVP at any point $x^*=x_0+\delta h$, once the solution has been evaluated at $x_1=x_0+h$, where the scaling factor, δ, generally ranges between 0 and 1.

By using the derivative evaluations generated in obtaining $y(x_1)$, this new continuous solution may be determined at a cost of only a few additional derivative evaluations. Thus, for a given Runge-Kutta algorithm, a family of continuous algorithms may be developed, which, when used in method is available with the advantage that once additional derivative evaluation has been made, the solution may be determined at any number of points within the given step for a few arithmetic operations, while still maintaining the accuracy of the defining algorithm.

When the RKF(4)5 coefficients RKF(4)5 are used and $c_6=1$ and $a_{61}=a_{62}=a_{63}=0$, are chosen with $b_1^*=0$, the continuous coefficients become:-

$b_0^* = 1-\delta(301/120+\delta(-269/108+\delta(311/360)))$

$b_2^* = \delta(7168/1425+\delta(-4096/513+\delta(14848/4275)))$

$b_3^* = \delta(-2561/8360+\delta(199927/22572+\delta(-371293/75240)))$

$b_4^* = \delta(57/50+\delta(-3+\delta(42/25)))$ … (14)

$b_5^* = \delta(-96/55+\delta(40/11+\delta(-102/55)))$

$b_6^* = \delta(3/2+\delta(-4+\delta(5/2)))$

Remark[2][3]

We can construct a CRK method of high order by following the same approach

4-Numerical Examples

Example(1):

Consider the following differential equation of the first order:

$$y'=y^2\sin x \qquad y(0)=1 \qquad 0\le x\le 1$$

The exact solution has the form

$$y(x)=\sec x$$

Table (1) presents the numerical results for the global error, RKF(4^{th} and 5^{th}) and CRK are used with differential value of δ.

For $i=1,2,\ldots,s^*$, where $s^*\ge s$ .If c^* and a^* are chosen so that

$$k_i^* = k_i \qquad \ldots(11)$$

for $i=1,\ldots,s$ then the major expense of applying the algorithm has been avoided since $k_1,k_2,\ldots,k_s$ have already been determined during the evaluation of $y(x_1)$.Thus, only $k_{s+1},\ldots,k_{s^*}$ need to be evaluated.

The proposed strategy requires that the c_i^* and a_{ij}^* parameters of the continuous algorithm be related to their respect c_i and a_{ij} coefficients in the defining algorithm, for $i=1,\ldots,s$. The additional parameters c_i^* and a_{ij}^*, $i=s+1,\ldots,s^*$are then used to generate a continuous algorithm at accuracy comparable to that of the defining algorithm for a particular value of δ.

Comparing (4,5) with (9,10)one finds that if:-

$$c_i^* = \frac{c_i h}{h^*} = \frac{c_i}{\delta} \qquad \ldots(12)$$

and

$$a_{ij}^* = \frac{a_{ij} h}{h^*} = \frac{a_{ij}}{\delta} \qquad \ldots(13)$$

for $i=1,2,\ldots,s$, $j=0,1,\ldots i-1$, then $k_i^*=k_i$ for $i=1,\ldots,s$.

Thus for $i\le s$, the c_i^* and a_{ij}^* parameters are already determined, being related to their corresponding coefficients in the defining algorithm by a factor of $\frac{1}{\delta}$. These k_i evaluations will have already been computed during the application of the defining algorithm

3-A Continuous Fourth-order Runge-Kutta Method[5,7]

Using the coefficients of the defining algorithm of RKF(4)5 continuous fourth-order

Table(1)

h	x	RKF(4)	RKF(5)	CRK		
				δ =1	**δ=0.25**	**δ=0. 5**
0.1	0.1	4.7404e-003	4.7404e-003	4.7404e-003	1.0696e-003	4.3555e-003
	0.5	1.0404e-001	1.0404e-001	1.0404e-001	-3.7055e-001	6.3020e-002
	1.0	5.4800e-001	5.4800e-001	5.4800e-001	8.7594e-002	1.9501e-001
0.05	0.1	4.7404e-003	-6.6912e-011	4.7404e-003	-8.7819e-003	-6.3256e-003
	0.5	1.0404e-001	-1.0160e-009	1.0404e-001	-6.3252e-001	-1.9327e-001
	1	5.4800e-001	-3.2211e-008	5.4800e-001	-1.8626e+000	-8.4477e+000

numerical results RKF(4^{th} and 5^{th}) and CRK**)**

$y'=y+3x-x^2 \qquad y(0)=1 \qquad 0 \le x \le 1$

The exact solution has the form

$y(x)=2e^x+x^2-x-1$

Table (2) presents the numerical results for the global error, RKF(4^{th} and 5^{th}) and CRK are used with differential value of δ

Example(2):

Consider the following differential equation of the first order:

Table(2)

h	x	RKF(4)	RKF(5)	CRK		
				δ =1	δ=0.25	δ=0.5
0.1	0.1	-9.3655e-002	-9.3654e-002	-9.3655e-002	-3.5137e-001	-3.2182e-001
	0.5	-1.3830e-001	-1.3830e-001	-1.3830e-001	-6.5630e+000	-8.3591e-001
	1.0	1.4373e+000	1.4373e+000	1.4373e+000	-5.9260e+001	1.1769e-000
0.05	0.1	-3.0567e-009	1.0674e-010	1.0674e-010	-3.9671e-001	-1.3513e-001
	0.5	-2.0911e-008	7.3629e-010	7.3629e-010	-7.4570e+000	-1.8175e+000
	1	-6.3094e-008	2.2421e-009	2.2421e-009	-7.0439e+001	-8.3140e+000

numerical results RKF(4^{th} and 5^{th}) and CRK**)**

$$y(x)=\left(\frac{x^2+3}{3}\right)^{\frac{1}{2}}$$

Table (3) presents the numerical results for the global error, RKF(4^{th} and 5^{th}) and CRK are used with differential value of δ.

Example(3):

Consider the following differential equation of the first order:

$y'=xy^{1/3} \qquad y(0)=1 \qquad 0 \le x \le 1$

The exact solution has the form

Table(3)

h	x	RKF(4)	RKF(5)	CRK		
				δ =1	δ=0.25	δ=0.5
0.1	0.1	-3.3389e-003	-3.3389e-003	-3.3389e-003	-6.2446e-003	-1.0843e-002
	0.5	-8.6736e-002	-8.6736e-002	-8.6736e-002	-4.2287e-001	-2.3240e-001
	1	-3.490e-001	-3.490e-001	-3.490e-001	-2.1890e+000	-1.0268e+000
0.01	0.1	-3.3389e-003	-3.3389e-003	-3.3389e-003	-1.7264e-002	-8.6019e-003
	0.5	-8.6736e-002	-8.6736e-002	-8.6736e-002	-4.9189e-001	-2.2063e-001
	`1	-3.490e-001	-3.490e-001	-3.490e-001	-2.3897e+000	-9.9998e-001

numerical results RKF(4^{th} and 5^{th}) , CRK**)**

3- If $0 \le \delta < 1$ takes different values, we obtain the solution between the discrete points. The error may be increased because we need rounding error.

6-Refrences

5-Conclusions

1- The obtained results show that the CRK method is better than RKF method of the same order.

2- If δ=1 in CRK method the results are approximately equal to the results of RKF(4^{th})5^{th} method.

[1] Burden R.I., and Faires J.D." Numerical Analysis", Fourth Edition, PWS-Kent Publishing, (1989).

[2] Butcher, J.C.,"The Numerical Analysis of Ordinary Differential Equations", Johan Wiley & Sons (1987).

[3] Enright. W.H., "Continuous Numerical Method for ODE's with Defect Control", University of Toronto, Canada, M 533, G4, December (1999).

[4] Eman.A [1]&, Madeha Sh.f [2], Niran [3] ,(GALERKIN'S METHOD TO SOLVE THE LINEAR SECOND ORDER DELAY MULTI-VALUE PROBLEMS), Sci.Int.(Lahore),31(1),1-4, 2019 ISSN 1013-5316;CODEN: SINTE 8 1

[5] Horn, M.K, "Fourth and Fifth Order, Scaled Runge-Kutta Algorithms for Treating Dense Output", SIAM J. Num. Anal. Vol.20, no 3, June 1983.

[6] Mathews, J.H. , Kurtis, D.F.; "Numerical Methods Using Matlab" Pearson Prentice Hall (2004).

[7] Josef Diblik. "Special Aspects to Functional Differential Equations", (2002).

[8] Tay Kim , S L , R A,"A Spreadsheet Solution of a System of Ordinary Differential Equations Using the Fourth-Order Runge-Kutta Method, Spreadsheets in Education (eJSiE),Vol. 5, Iss. 2 [2012], Art.5.

[9] Shampine L.F, and Thomposn S., "Solving Delay Differential Equations in Matlab", Journal of App.Number. Math., Vol.4, No.37.