P- ISSN 1991-8941 E-ISSN 2706-6703
2020,14 (2):77- 82

Journal of University of Anbar for Pure Science (JUAPS)

Open Access

Predictor Corrector Parallel Based on the Geometric Mean Runge-Kutta Formula for Solving Initial Value Problems

Mahmood D. Jasim
Department of Mathematics, College of Basic Education, Mosul University – Mosul - Iraq

ARTICLE INFO

Received: 10 / 6 /2020
Accepted: 10 / 7 / 2020
Available online: 1 / 12 / 2020
DOI: 10.37652/juaps.2022.172392

Keywords:
Initial value problems,
Predictor corrector method,
Geometric Runge-Kutta method,

ABSTRACT

The purpose of this study is to present a new connection of the famous Runge-Kutta methods by using more than one old technique and obtain a new method, which has acceptable results for solving Initial Value problems.

1. INTRODUCTION

One of the best methods for solving ordinary differential equations (ODEs) numerically are the Runge-Kutta methods. Many methods were discovered from 1895 tell now. "The search for better methods is always up to time" [1]. Here we made a connection between the predictor corrector (PC) methods see[2,3], the Geometric mean (GM) formula and expanded "the front of computation" to have a new parallel method. "Evans, Introduced a new Runge-Kutta method used the Geometric mean (GM) formula [4]". We gathered those ideas by using the Implicit Runge-Kutta methods,which represented the backward form, from the Explicit Runge-Kutta methods which represented the forward form [5]. At last, we introduce the new parallel method namely (PPCGM1 method).

1.1. Definition of the Computation Front

"The computation front is the imaginary straight line that separates the values which are next to be computed (be numerical algorithms) from all previously computing value problems [6,7]". It is shown in figure1 bellow.

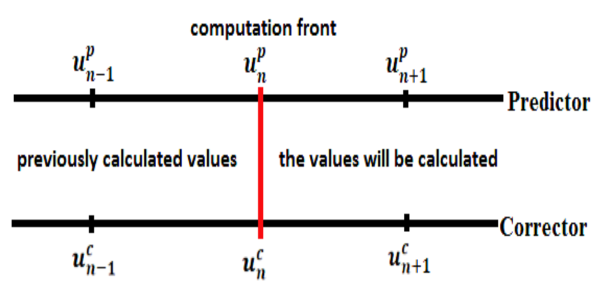

Fig.1: illustrates the definition above.

2. METHODOLGY

2.1. The Gm Method

$u_{n+1} - u_n = \frac{h}{n}\left(\sum_{i=1}^{n}\sqrt{w_i w_{i+1}}\right) \ldots \quad (1)$

Where,

$w_i = \phi\left(t_n + r_i h_i \, , u_n + h\sum_{j=1}^{i-1} s_{ij} w_j\right) \;\; i = 1,2,3,\ldots,n \;\; \ldots (2)$

Where $r_i, s_{ij} \geq 0$, h_i is the length of step, $u_n = u_n(t_n)$.

If we regard ϕ as a function of only u. "this will considerably reduce the lengthy Taylor series expansions of w_i , $i = 1,2,,$

So (2) becomes,

$w_i = \phi\left(u_n + h\sum_{j=1}^{i-1} s_{ij} w_j \right) \;\; i = 1,2,3,\ldots,n$ [8]". .... (3)

*Corresponding author at: Department of Mathematics, College of Basic Education, Mosul University, Iraq. Tel.: +964 7703053530 ORCID:0000-0003-1778-6137
. E-Mail: mahoodalani@yahoo.com

2.2. The Ppcgm1 Method

PPCGM1 calculates u_{n+1}^{p} depended on u_{n-1}^{c} and calculates u_n^c depended on u_{n-1}^{c} and u_n^p , which has the form,

$$u_{n+1}^{p} - u_{n-1}^{c} = 2h(\sqrt{w_1 w_2}) \ \dots (4)$$

$$w_1 = \phi(t_{n-1}, u_{n-1}^{c}) \ , \ w_2 = \phi(t_{n-1} + 2h, u_{n-1}^{c} + 2hw_1)$$

And,

$$u_n^c - u_{n-1}^{c} = h\left(\sqrt{L_1 L_2}\right) \dots (5)$$

$$L_1 = \phi(t_n, u_n^p) \ , \ L_2 = \phi(t_n - h, u_n^p - hL_1)$$

Process of PPCGM1 mode is shown in fig.2 where u_n^p is the predictor values, u_n^c is the corrector values and the red line represents "the computation front".

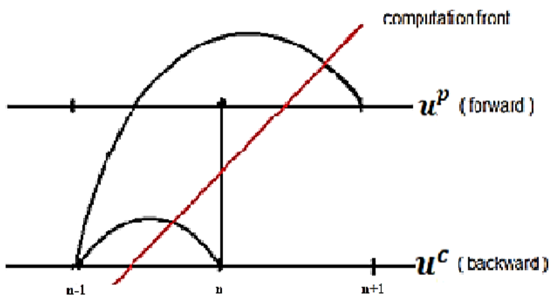

Fig. 2: PPCGM1 mode.

Derivation of PPCGM1 Method

The predictor part of Geometric mean second order, which has the form:

$$u_{n+1}^{p} - u_{n-1}^{p} = 2h\left(\sqrt{aw_1 bw_2}\right) \dots (6)$$

$w_1 = \phi(u_{n-1}^{c})$, $w_2 = \phi(u_{n-1}^{c} + \beta hw_1)$, where $a, b, \beta > 0$

To derive (predictor part) of PPCGM1, expansion of w_1 and w_2 gives

$$w_1 = \phi(u_{n-1}^{c}) = \phi$$

$$w_2 = \phi(u_{n-1}^{c} + \beta hw_1) = \phi + \beta hw_1 \phi_{u_{n-1}^{c}} + o(h^2)$$

Now,

$$abw_1 w_2 = ab\phi^2 + ab\beta h\phi^2 \phi_{u_{n-1}^{c}} + o(h^2)$$

So,

$$(abw_1 w_2)^{1/2} = \left(ab\phi^2 + ab\beta h\phi^2 \phi_{u_{n-1}^{c}} + o(h^2)\right)^{1/2}$$

$$= (ab)^{1/2}\phi\left(1 + \beta h\phi_{u_{n-1}^{c}} + o(h^2)\right)^{1/2}$$

$$(abw_1 w_2)^{1/2} = (ab)^{1/2}\phi\left(1 + \frac{\beta h}{2}\ \phi_{u_{n-1}^{c}} + o(h^2)\right) \dots (7)$$

When substituting eq. (7) in (6), we obtain

$$u_{n+1}^{p} - u_{n-1}^{p} = 2(ab)^{1/2}\phi\left(h + \frac{\beta h^2}{2}\ \phi_{u_{n-1}^{c}} + o(h^3)\right). (8)$$

Compare the last eq. (8) with Taylor eq.[9] of the form,

$$u_{n+1} - u_{n-1} = 2h\phi + h^2\phi\phi_{u_{n-1}} + o(h^3) \dots (9)$$

We get,

$$2(ab)^{1/2} = 2 \ , \beta = 1$$

Which are 2 equations and 3 parameters, that is mean 1 freedom degree, by choosing $a = 1$ (since $a \in (0, \infty)$) then $b = 1$.

Getting the system,

$$u_{n+1}^{p} - u_{n-1}^{c} = 2h\left(\sqrt{w_1 w_2}\right) \dots (10)$$

$$w_1 = \phi(t_{n-1}, u_{n-1}^{c}) \ , \ w_2 = \phi(t_{n-1} + 2h, u_{n-1}^{c} + 2hw_1)$$

Now, the corrector part of PPCGM1 is from the backward technique,

$$u_{n-1}^{c} - u_n^c = -h\left(\sqrt{aL_1 bL_2}\right) \dots (11)$$

and,

$$L_1 = \phi(u_n^p) \quad , \quad L_2 = \phi(u_n^p - \beta hL_1)$$

Now, by expanding L_1 and L_2 , we get

$$L_1 = \phi(u_n^p) = \phi$$

$$L_2 = \phi(u_n^p - \beta hL_1) = \phi - \beta h\phi\phi_{u_n^p} + o(h^2)$$

$$aL_1 = a\phi$$

$$bL_2 = b\phi - b\beta h\phi\phi_{u_n^p} + o(h^2)$$

$$abL_1 L_2 = ab\phi^2 - ab\beta h\phi^2 \phi_{u_n^p} + o(h^2)$$

$$(abL_1 L_2)^{1/2} = \left(ab\phi^2 - ab\beta h\phi^2\phi_{u_n^p} + o(h^2)\right)^{1/2}$$

$$= (ab)^{1/2}\phi\left(1 - \frac{\beta h}{2}\ \phi_{u_n^p} + o(h^2)\right) \dots (12)$$

Substitute (12) in (11),

$$u_{n-1}^{c} - u_n^c = -h(ab)^{1/2}\phi + \frac{\beta h^2}{2}\ \phi_{u_n^p} + o(h^3) \quad \dots (13)$$

Compare eq. (13) with Taylor eq. of the form [9],

$$u_{n-1} - u_n = -h\phi + \frac{h^2}{2}\phi\phi_{u_n} + o(h^3) \dots\dots\dots\dots (14)$$

We get,

$$2(ab)^{1/2} = 2 \ , \beta = 1$$

Which is 2 equations with 3 parameters, so having 1 degree of freedom.

Choosing $b = 1$ then $a = 1$.

The corrector part of PPCGM1 has the form:

$$u_n^c - u_{n-1}^c = h\left(\sqrt{L_1 L_2}\right) \dots (15)$$

$L_1 = \phi(t_n , u_n^p)$ and $L_2 = \phi(t_n - h , u_n^p - hL_1)$

Equations (10) and (15) represent our PPCGM1 method, u_{n+1}^p represent predictor part and u_n^c is the corrector part.

2.3 Analysis of Ppcgm1 Stability

Runge-Kutta methods have an important feature, that they are stable," if we take a suitable small step size h [10]".

"We test the stability of Runge-Kutta methods by using the known test equation $\acute{u} = \lambda u$ [11,12] where $\lambda = \partial\phi/\partial u$ is constant ".

To test the stability of (the predictor part) for PPCGM1 method which had the form,

$$u_{n+1}^p - u_{n-1}^c = 2h(\sqrt{w_1 w_2})$$

$$w_1 = \phi(t_{n-1} , u_{n-1}^c) , \ w_2 = \phi(t_{n-1} + 2h , u_{n-1}^c + 2hw_1)$$

And to find the interval of absolute stability, we used "the test equation $\acute{u} = \lambda u$ ", So,

$$w_1 = \phi(t_{n-1} , u_{n-1}^c) = \lambda u_{n-1}^c \quad \dots (16)$$

$$w_2 = \phi(t_{n-1} + 2h , u_{n-1}^c + 2hw_1) = \lambda(u_{n-1}^c + 2h\lambda u_{n-1}^c) \quad \dots (17)$$

Equations (16) and (17) substituting in (10) we get,

$$u_{n+1}^p - u_{n-1}^c = 2h\left((\lambda u_{n-1}^c) \times (\lambda u_{n-1}^c + 2h\lambda^2 u_{n-1}^c)\right)^{1/2}$$

$$u_{n+1}^p = u_{n-1}^c + 2h\lambda u_{n-1}^c(1 + 2h\lambda)^{1/2} \dots (18)$$

Dividing equation (18) by u_{n-1}^c and putting $z = h\lambda$ we get,

$$\upsilon = \frac{u_{n+1}^p}{u_{n-1}^c} = 1 + 2z(1 + 2z)^{1/2} \dots (19)$$

Equation (19) satisfies the condition of absolute stability if $|\upsilon| < 1$ where z satisfies the condition when $z \in$ (-2,0) this interval is considered as the stability region of PPCGM1.

3. Test PPCGM1 in examples

3.1 Example1

We use the IVP example $\acute{u} = -tu^2$, u (0) = 2, and h= 0.005, to test our PPCGM1 it is shown in table 1 below.

Table 1: Results of PPCGM1 Method applied to example1

Val. of t	Exact solution $u(t)$	u^c	$\|u - u^c\|$	u^p	$\|u - u^p\|$	The classical RK method $u^*(t)$	$\|u - u^*\|$
0.005	1.9999500	2.0001414	4.9998e-05	2	0.0001999	1.99980000	4.9998e-05
0.010	1.9998000	2.0003865	0.00034141	2.000346	0.0007962	1.99940013	1.9998e-08
0.015	1.9995501	2.0007332	0.00083640	2.00070708	0.0015067	1.99880065	0.00014996
0.020	1.9992003	2.0011810	0.00153291	2.00116116	0.0024103	1.99800191	0.00039966
0.025	1.9987507	2.0017299	0.00243029	2.00171335	0.0035117	1.99700439	0.00074886
0.030	1.9982016	2.0023797	0.00352828	2.00236509	0.0048120	1.99580868	0.00119722
0.035	1.9975529	2.0031307	0.00482673	2.00311710	0.0063119	1.99441549	0.00174431
0.040	1.9968051	2.0039829	0.00632559	2.00396985	0.0080116	1.99282566	0.00238961
0.045	1.9959581	2.0049368	0.00802480	2.00492372	0.0099112	1.99104014	0.00313251
0.050	1.9950124	2.0059924	0.00992435	2.00597909	0.0120108	1.98905997	0.00397232

0.055	1.9939682	2.0071503	0.01202425	2.00713633	0.0143105	1.98688633	0.00490827
0.060	1.9928258	2.0084106	0.01432449	2.00839584	0.0168102	1.98452050	0.00593949
0.065	1.9915855	2.0097738	0.01682509	2.00975804	0.0195102	1.98196388	0.00706504
0.070	1.9902477	2.0112404	0.01952609	2.01122336	0.0224104	1.97921794	0.00828390
0.075	1.9888129	2.0128107	0.02242750	2.01279227	0.0255108	1.97628430	0.00959498
0.080	1.9872813	2.0144853	0.02552936	2.01446530	0.0288116	1.97316466	0.01099709
0.085	1.9856536	2.0162648	0.02883173	2.01624296	0.0323128	1.96986082	0.01248898
0.090	1.9839301	2.0181496	0.03233464	2.01812583	0.0360143	1.96637467	0.01406934
0.095	1.9821114	2.0201403	0.03603817	2.02011452	0.0399165	1.96270820	0.01573677
0.100	1.9801980	2.0222377	0.03994237	2.022209	0.0440192	1.95886352	0.01748981

3.2. Example2

We use the IVP example $\acute{u} = (2t-1)/u^2$, u (0) = 1, and h= 0.005, to test our PPCGM1 it is shown in table 2 below.

Table 2: Results of PPCGM1 Method applied to example2

Val. of t	Exact solution $u(t)$	u^c	$\lvert u - u^c \rvert$	u^p	$\lvert u - u^p \rvert$	The classical RK method $u^*(t)$	$\lvert u - u^* \rvert$
0.005	0.9950496	1.00495043	0.0099008	1	0.00495037	0.99000051	0.0049503
0.010	0.9901970	1.00975548	0.0195584	1.00999948	0.01980244	0.97989899	0.0001965
0.015	0.9854401	1.01446786	0.0290277	1.01474919	0.02930907	0.96969117	0.0055411
0.020	0.9807767	1.01909031	0.0383135	1.01935987	0.03858309	0.95937245	0.0110856
0.025	0.9762050	1.02362502	0.0474200	1.02388296	0.04767795	0.94893793	0.0168325
0.030	0.9717228	1.02807405	0.0563511	1.02832094	0.05659808	0.93838232	0.0227849
0.035	0.9673284	1.03243938	0.0651109	1.03267578	0.06534730	0.92769996	0.0289461
0.040	0.9630200	1.03672292	0.0737028	1.03694933	0.07392930	0.91688474	0.0353200
0.045	0.9587957	1.04092647	0.0821307	1.04114338	0.08234763	0.90593003	0.0419110
0.050	0.9546539	1.04505177	0.0903978	1.04525962	0.09060569	0.89482870	0.0487239

0.105	0.100	0.095	0.090	0.085	0.080	0.075	0.070	0.065	0.060	0.055
0.9140976	0.9174311	0.9208315	0.9242998	0.9278374	0.9314456	0.9351256	0.9388789	0.9427069	0.9466111	0.9505929
1.08569715	1.08233615	1.07891198	1.07542351	1.07186959	1.06824900	1.06456050	1.06080278	1.05697449	1.05307422	1.04910049
0.1715994	0.1649049	0.1580804	0.1511236	0.1440321	0.1368034	0.1294348	0.1219238	0.1142675	0.1064630	0.0985075
1.08582860	1.08247313	1.07905472	1.07557226	1.07202462	1.06841060	1.06472895	1.06097841	1.05715763	1.05326521	1.04929971
0.17173093	0.16504193	0.15822320	0.15127242	0.14418718	0.13696499	0.12960330	0.12209943	0.11445064	0.10665408	0.09870678
0.76019613	0.77362698	0.78676863	0.79964179	0.81226494	0.82465458	0.83682556	0.84879130	0.86056392	0.87215447	0.88357299
0.1404706	0.1306625	0.1211897	0.1120349	0.1031828	0.0946200	0.0863343	0.0783150	0.0705525	0.0630381	0.0557642

0.150	0.145	0.140	0.135	0.130	0.125	0.120	0.115	0.110
0.8869179	0.8896995	0.8925383	0.8954355	0.8983918	0.9014084	0.9044862	0.9076263	0.9108297
1.11326853	1.11043206	1.10754083	1.10459402	1.10159080	1.09853031	1.09541164	1.09223387	1.08899604
0.2263505	0.2207325	0.2150024	0.2091585	0.2031989	0.1971218	0.1909253	0.1846075	0.1781662
1.11335923	1.11052660	1.10763936	1.10469671	1.10169781	1.09864182	1.09552784	1.09235495	1.08912220
0.22644127	0.22082710	0.21510099	0.20926119	0.20330594	0.19723337	0.19104159	0.18472862	0.17829243
0.62104301	0.63868389	0.65563360	0.67197124	0.68776250	0.70306260	0.71791845	0.73237024	0.74645272
0.2482340	0.2340658	0.2205671	0.2076730	0.1953292	0.1834899	0.1721160	0.1611736	0.1506336

4. CONCLUSIONS

We note that the results of our PPCGM1 method are acceptable when it comparing with the exact solution. With corresponding to the classical Arithmetical Runge-Kutta (ARK) method. It has better stability by observing the results**.**

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author is very grateful to the University of Mosul College of Basic Education for their provided facilities, which helped to improve the quality of this work.

REFERENCES

[1] Hippolyte S., and Assui R. K., 2019, "A New Seventh Order Runge Kutta Family: Comparison with the Method of Butcher and Presentation of a Calculation Software." Mathematics and Computer Science. 4(3), 68-75.

[2] Khalaf, B. M. S. and Abdulhabib A. A. M., 2004, " Parallel Implicit Runge Kutta Methods for Stiff ODEs ", Raf. J. of Comp. & Math's, 1(2).

[3] Singh, N., 2014, " Predictor Corrector Method of Numerical Analysis-New Approach ", International Journal of Advanced Research in Computer Science, 5(3).

[4] Evans, D. J. and Sanugi B. B. ., 1989, "A Parallel Runge Kutta Integration Method", parallel computing 11, 245-251.

[5] Cash, J. R.,1975, "A Class of Implicit Runge Kutta Methods for the Numerical Solution of Stiff Ordinary Differential Equations "; Journal of the Association for Computing Machinery, 22(4), 504-511.

[6] Khalaf, B. M. S.,1990, "Parallel Numerical Algorithms for Solving Ordinary Differential Equations ", Ph. D. thesis, University of Leeds, School of Computer Studies, U. K.

[7] Murshed, A. A., 2000, "New Parallel numerical algorithms for solving stiff ordinary diff. eq.s adapted for MIMD Computers, Ph.D. thesis, Univ. of Mosul.

[8] Sharmila R. G., 2017, " Fourth Order Runge Kutta-Method Based on Geometric Mean for Hybrid Fuzzy Initial Value Problems ", IOSR Journal of Mathematics, 13 (2), 43-51.

[9] Mahmood, D. A., 2000, "parallel Runge Kutta methods for MIMD computers ", M.Sc. Thesis, Department of Mathematics, University of Mosul, Mosul, Iraq.

[10] Lambert J. D., 2000, "Numerical Methods for Ordinary Differential Systems, the initial value Problem", John Wiley & Sons Ltd.

[11] Agbeboh G.U., 2013, " On the Stability Analysis of a Geometric Mean 4th Order Runge Kutta Formula", Mathematical Theory and Modeling, 3(4).

[12] Bazuaye, F. E., 2018, " A New 4th Order Hybrid Runge Kutta Methods for Solving Initial Value Problems (IVPs).", Pure and Applied Mathematics Journal, 7(6), 78-87.

طريقة المخمن المصحح متوازية باستخدام صيغة رانج كوتا ذات الوسط الهندسي لحل مسائل القيم الابتدائية

محمود ضياء جاسم العاني

كلية التربية الاساسية – قسم الرياضيات – جامعة الموصل – العراق

الخلاصة:

نقدم في هذا البحث طريقة من الرتبة الثانية من طرائق رانج كوتا المعروفة لحل مسائل القيم الابتدائية، وذلك بتطوير هذه الطريقة باستخدام صيغة المخمن والمصحح واستخدام تقنية تعريض مقدمة الحساب للحصول على طريقة جديدة متوازية.