International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 07 - Issue 05 || May 2022 || PP. 43-45

Coding Iteration Approach in a Predictor – Corrector for Linear Implicit One-step Methods

Pham Thi Minh Hanh[1]
[1](*Thai Nguyen University of Technology, TNU, Vietnam*)

Abstract: The paper aims to presenta new approach to implement the implicit one-step methods. The implicit one-step methodsdevelopped in the study was constructed on the basis of Gauss-Legendre polynomials, the implicit Runge-Kutta methods. However, the approach we use could be applied for other implicit one-step methods for solving ordinary differential equations with initial value in general. The improvement produced by this approach is verified in the numerical experiment. This is because the approach takes both advantages from an implicit one-step method of only three stages to approximate the stiff problems with fewer number of calculations and the predictor-corrector technique which reduces the number of functional evaluations compared to different techniques solving a non-linear equation.

Keywords: Predcitor-Corrector, implementation, Matlab code, linear implicit one-step method, ordinary differential equation.

I. INTRODUCTION

Consider the initial value problem

$$y' = f(t,y), a \le t \le b, y(a) = \alpha. \qquad (1)$$

A Runge-Kutta method of s-stages and of order p is generally presented by

$$w_{n+1} = w_n + \sum_{i=1}^{s} b_i k_i$$
$$(\forall n, 0 \le n \le N) \qquad (2)$$
$$k_r = hf\left(t_n + c_j h, w_n + \sum_{r=1}^{s} a_{jr} k_r\right), \forall r = 1,2,\dots,s,$$

where the step size $h = (b-a)/N$, the number of equally distributed mesh points $t_m's$ is N:

$$a = t_0 < t_1 < \ldots < t_N = b,$$

w_n is the approximation to $y(t_n)$, the exact value of the solution $y(t)$ of (1) at the mess point t_n, for all $n = 0,1,\dots,N$.

We now consider a class of implicit Runge-Kutta method is constructed on the basis of Gaussian quadrature is introduced in [1-2], pp. 219. The specified method for a s-stages and p-order, where $s = 2, p = 4$ has the Butcher's table as follows.

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Assume that the equation (2) is presented into the matrix from as follows:

$$w_{n+1} = w_n + B\mathbf{k}$$
$$\mathbf{k} = h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\mathbf{k}) \qquad (3)$$

where

$$B = (b_1, b_2, \dots, b_s), C = (c_1, c_2, \dots, c_s)^T,$$
$$\mathbf{1} = (1,\dots,1)^{\mathrm{T}}, \mathbf{k} = (k_1, k_2, \dots, k_s)^{\mathrm{T}} \in \mathbb{R}^s,$$
$$\boldsymbol{F}(\boldsymbol{z},\boldsymbol{u}) = \boldsymbol{F}((z_1, z_2, \dots, z_s)^T, (u_1, u_2, \dots, u_s)^T) = \left(f(z_1,u_1), f(z_2,u_2), \dots, f(z_s,u_s)\right)^T.$$

In the equation (4), the unknown **k**can be solved in the iterative process

$$\mathbf{k}^{(q+1)} = h\boldsymbol{F}\left(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\mathbf{k}^{(q)}\right), \forall q \ge 0, \qquad (5)$$

to generate the sequence $\{\mathbf{k}^{(q)}\}_{q\geq 0}$ which converges to the true root $\mathbf{k}$ of the equation (5). The predictor-corrector approach with the initial term $\mathbf{k}_h^{(0)}$ at each step n chosen to be the solution of (5) is used to implement the method. The implementation are presented in Matlab code shown in the section below.

II. IMPLIMENTING THE PREDICTOR-CORRECTOR APPROACH

Input: Function $f(t,y)$, interval $[a,b]$, $y(a) = alpha$, maximum number of iteration in each step m, number of subinterval N.
Output: Approximation w_i to the value of the true solution $y(t)$ evaluated at each mesh point $t_i = a + ih = a + i(b-a)/N$.

The code written in an M-file function is given as follows.

```
function outp=IRK4(f,a,b,alpha,N,m)
h=(b-a)/N;
t0=a;
w0=alpha;
TW=[t0,w0];
%--------
A=[1/4,(1/4-sqrt(3)/6);(1/4+sqrt(3)/6),(1/4)];
syms tw;
for i=1:N
   j=1;
Q=eye(2)-h*double(subs(diff(f(t,w),w),[t,w],[t0,w0]))*A;
R=h*f(t0,w0)*[1;1]+h^2*double(subs(diff(f(t,w),t),[t,w],[t0,w0]))*[1/2-sqrt(3)/6;1/2+sqrt(3)/6];
   Z=linsolve(Q,R);
   k1=Z(1);
   k2=Z(2);
while j<=m
%------------------------------------------------
   U=A*[k1;k2];
   k1=h*f(t0+(1/2-sqrt(3)/6)*h,w0+U(1));
   k2=h*f(t0+(1/2+sqrt(3)/6)*h,w0+U(2));
%-------------------------------------------------
   j=j+1;
end
   t0=t0+h;
   w0=w0+1/2*k1+1/2*k2;
   TW=[TW;t0,w0];
end
%------------------------
outp=TW;
```

We can also implement other methods in the family of s-stage and p-order implicit Runge-Kutta method with the predictor-corrector approach in the same manner.

III. NUMERICAL EXPERIMENT

Consider the following numerical experiment.

Example([3]) Given the initial value problem

$$y' = (t + 2t^3)y^3 - ty, t \in [0,2], y(0) = 1/3. \qquad (6)$$

The exact solution of the problem is $y = \left(3 + 2t^2 + 6e^{t^2}\right)^{-1/2}$. The absolute error at $t_N = 2$ is shown in Table 1 for each method.

In this table, some methods are use to make the comparison including: IRK6_PC (implicite Runge-Kutta of order 6 with predictor-corrector approach), IRK4_PC (implicite Runge-Kutta of order 4 with predictor-corrector approach), IRK6 (implicite Runge-Kutta of order 6 with Newtons iteration approach), RK4 and RK6 (explicite Runge-Kutta of order 4 and 6, respectively).

IRK6_PC $N = 10, 20,30,70;$ $M = 10$	IRK4_PC $N = 10, 20,30;$ $M = 10$	IRK6 $N = 10, 20,30;$ $M = 10, tol = 0.001$	RK4 $N = 10, 20,30;$	RK6 $N = 10, 20,30;$
1.915×10^{-9}, 2.978×10^{-11}, 2.612×10^{-12}, 1.6×10^{-14}	1.82×10^{-7}, 1.064×10^{-8}, 2.075×10^{-9}	1.464×10^{-3}, 3.628×10^{-4} 1.606×10^{-4}	6.458×10^{-6}, 3.73×10^{-7}, 7.16×10^{-8}	1.982×10^{-4}, 1.033×10^{-4}, 6.991×10^{-5}
$0.92s, 1.27s, 1.59s, 9.1s$	$0.88s, 1.25s, 1.6s$	$0.98s, 1.33s, 1.69s$	$0.48s, 0.5s, 0.51s$	$1.52s, 1.58s, 1.6s$

Table 1 Absolute error to the approximation of the solution of (10) at the last mesh point $t_N = 2$ produced by the corresponding method and the time (in second) to perform the calculation corresponding to each number N in the list.

From this experiment, we could see the superiority of the implementation constructed to other approach. This especially true for the case of high stiffness.

IV. CONCLUSION

The approach to implement the implicit Runge-Kutta method constructed on Gaussian quadrature presented has an upper hand compared to other approaches. This new strategy makes benefit both in less computational cost and higher accuracy.

REFERENCES

[1]. Butcher, John. C., “Practical Runge - Kutta methods for scientific computation”, The ANZIAM Journal, Vol. 50 (03), pp. 333 – 342, 2009, doi: 10.1017/S1446181109000030.
[2]. Butcher, John. C., “Numerical Methods for Differential Equations and Applications”, 2nd Edition, John Wiley (Chichester), 2008.
[3]. Richard L. Burden, J. Douglas Faires, “Numerical Analysis”, 9th Edition, Brooks/Cole, Cengage Learning, 2011.
[4]. Luther, H. A., “An explicit sixth-order Runge-Kutta formula”, Mathematics of Computation 22, pp. 434-436, 1968.