PREDICTOR-CORRECTOR TECHNIQUE FOR IMPLEMENTING AN SIXTH ORDER IMPLICIT RUNGE-KUTTA METHOD

Pham Thi Thu Hang
TNU - University of Technology

ARTICLE INFO

Received: 04/11/2021
Revised: 30/11/2021
Published: 30/11/2021

KEYWORDS

Implicit Runge-Kutta method
Initial value problem
Predictor-Corrector technique
Implementation
Linear one step method

ABSTRACT

In this article, a new approach to implement an implicit Runge-Kutta method is developped. The implicit Runge-Kutta method in the study was constructed on the basis of Gauss-Legendre polynomials, first appeared in the paper of J.C. Butcher (2009). The improvement produced by this approach is much helpful. This is because it takes both advantages from an implicit one-step method of only three stages to approximate the stiff problems with fewer number of calculations and from a high accuracy of an order sixth method which is quite high order of convergence under the consistency. The proof for the convergence of the technique is also shown. This approach can also be used to implement an implicit Runge-Kutta method presented of even less order constructed basing on Gauss-Legendre polynomials. A combination of the implementation and the sixth order Backward Difference Formula Off-step Continuous block can shed light on the therapy to the stiffness and be worthy. This is also studied in the paper. Afterward, a comparison is made to show the improvement achieved.

KỸ THUẬT DỰ BÁO-HIỆU CHỈNH ĐỂ XÂY DỰNG TRÌNH THỰC THI CHO MỘT PHƯƠNG PHÁP BẬC SÁU RUNGE-KUTTA DẠNG ẨN

Phạm Thị Thu Hằng
Trường Đại học Kỹ thuật Công nghiệp - ĐH Thái Nguyên

THÔNG TIN BÀI BÁO

Ngày nhận bài: 04/11/2021
Ngày hoàn thiện: 30/11/2021
Ngày đăng: 30/11/2021

TỪ KHÓA

Runge-Kutta dạng ẩn
Bài toán giá trị ban đầu
Phương pháp dự báo-hiệu chỉnh
Trình thực thi
Phương pháp đơn bước tuyến tính

TÓM TẮT

Bài báo này đưa ra hướng tiếp cận mới đối với bài toán xây dựng trình thực thi cho một lớp các phương pháp Runge-Kutta dạng ẩn. Phương pháp Runge-Kutta dạng ẩn được nghiên cứu cụ thể ở đây được phát triển dựa trên các đa thức Gauss-Legendre, phương pháp xuất hiện đầu tiên trong bài báo của J. C. Butcher (2009). Sự cải tiến mà hướng tiếp cận mới mang lại là rất hữu ích. Điều này có được do những lợi thế của phương pháp một bước dạng ẩn chỉ có ba bước, đặc biệt phù hợp với các bài toán stiff, với khối lượng tính toán nhỏ mà độ chính xác cao của một phương pháp bậc sáu, một bậc tương đối cao của sự hội tụ mà vẫn đảm bảo điều kiện bền vững. Chứng minh cho sự hội tụ của phương pháp này được ra. Hướng tiếp cận này cũng có thể áp dụng cho một phương pháp Runge-Kutta dạng ẩn khác được đưa ra với bậc thấp hơn được xây dựng dựa trên các đa thức Gauss-Legendre. Sự kết hợp giữa hướng tiếp cận mới và phương pháp sai phân dạng khối Off-step bậc sáu có thể mang đến sự hợp lý trong việc xấp xỉ các bài toán stiff. Phương pháp này cũng được nghiên cứu trong bài báo. Sau cùng, các so sánh thực nghiệm đưa ra nhằm minh họa cho sự ưu việt của hướng tiếp cận đạt được.

DOI: https://doi.org/10.34238/tnu-jst.5230

Email: phamthuhang0201@gmail.com

1. Introduction

Consider the initial value problem

$$y' = f(t,y), a \leq t \leq b, y(a) = \alpha. \quad (1)$$

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

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

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 $n = 0,1,\dots,N$.

The Butcher's table [1, p. 94] of the method (2) is presented as follows

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

A Runge-Kutta method is explicit if the matrix $A = \left(a_{ij}\right)_{1 \leq i,j \leq s}$ of (2) is lower triangular, and one is implicit if the matrix A is not so. The benefit of an explicit one is from the fact that it can be easy to implement. But its drawback is that it needs very small step size to dial with the stiffness. This makes the number of functional evaluations raising a lot. This drawback is overcome with the use of an implicit Runge-Kutta method. Moreover, an implicit one need only fewer stages s to get an order p than that of an explicit one of the same order. Normally, for a s-stages and p-order implicit Runge-Kutta method, s may be less than p. However, for an explicit one, s must be greater than p for $p \geq 5$ ([1]). So, an implicit Runge-Kutta method is much suitable to treat the stiffness. A class of implicit Runge-Kutta method is constructed on the basis of Gaussian quadrature is introduced in [1], [2], pp. 219. Two specified methods of this class were given by

$s = 2, p = 4$:

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

$s = 3, p = 6$:

$$\begin{array}{c|ccc} \frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\ \frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array} \quad (4)$$

In fact, two methods mentioned have $p = 2s$. This is quite good in the sense of less computational cost of the functional evaluation but obtaining a high order of convergence which can not be seen from an explicit Runge-Kutta method of the same orders [3], [4].

However, the weakness of an implicit Runge-Kutta method having in common is from the difficulty of extracting the mediate elements at each stage. Concretely, in the difference equation (2), such obstacle is caused by finding the terms $k_i's$. The usual approach to overcome this situation is to make use of the Newton-Raphson iteration technique to approximate the terms $k_i's$ in the last r nonlinear equations of (2). This is a feasible schema and is somewhat efficient if we have a good function f and an appropriate initial approximation at each iteration process. However, the drawback of this approach comes from the fact that the solution of the nonlinear system is approximated by the solution of its linearized system, this in turn inflacts the truncation error. The new approach we are going to introduce here is then to make use directly the equations defining the terms $k_i's$ for iterative process. Because of the difference between the entered terms $k_i's$ and the generated terms $k_i's$, a predictor-corrector process is naturally obtained. But another question arises on the correction is that if the generated terms $k_i's$ gaining from this process is trusted. Fortunately, thank to the matrix A of Gaussian quadrature, appearing in (3) and (4), having the norm less than one, the aforementioned iterative process posses a fixed point which is just the right terms $k_i's$. This makes the success in this new approach. An upper hand of the new approach comparing to some innovative noticeable techniques presented in [6]-[8] can be seen from the illustration in section 4 . We are going to describe this achievement in the following section.

2. Iteration process for the predictor-corrector approach to implement an implicit Runge-Kutta method

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

$$\begin{aligned} w_{n+1} &= w_n + B\mathbf{k} \\ \mathbf{k} &= h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\mathbf{k}) \end{aligned} \qquad (5)$$

where

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

In the equation (5), the unknown $\mathbf{k}$ can be solve in the iterative process

$$\mathbf{k}^{(q+1)} = h\boldsymbol{F}\big(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\mathbf{k}^{(q)}\big), \forall q \geq 0, \qquad (6)$$

to generate the sequence $\{\mathbf{k}^{(q)}\}_{q\geq 0}$ which converges to the true root $\mathbf{k}$ of the equation (5). This is stated in the following result.

Theorem Given $n \in \mathbb{N}$, the transformations $\boldsymbol{G}_{n,h}: \mathbb{R}^s \to \mathbb{R}^s$ given by

$$\boldsymbol{G}_{n,h}(\mathbf{x}) = h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\mathbf{x})$$

has a unique fixed point by choosing N sufficiently large. Moreover, we could construct the formula for an initial approximation for the iteration process to find such fixed point by adding more requirement on how large N are.

Proof. We can choose N sufficiently large such that h is sufficiently small in order for

$$G_{n,h}(\{\mathbf{z} \in \mathbb{R}^s | \|\boldsymbol{z}\| \leq 1\}) \subset \{\mathbf{z} \in \mathbb{R}^s | \|\boldsymbol{z}\| \leq 1\}.$$

Therefore, $G_{n,h}: \bar{B}_{\mathbb{R}^s}(\mathbf{0}, 1) = \{\mathbf{z} \in \mathbb{R}^s | \|\boldsymbol{z}\| \leq 1\} \to \bar{B}_{\mathbb{R}^s}(\mathbf{0}, 1)$ is contraction mapping since

$$\|\boldsymbol{G}_{n,h}(\mathbf{x}) - \boldsymbol{G}_n(\mathbf{y})\| \leq \|\boldsymbol{\nabla}\boldsymbol{G}_{n,h}(\boldsymbol{\theta})\| \|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \bar{B}_{\mathbb{R}^s}(\mathbf{0}, 1),$$

for some $\boldsymbol{\theta} \in \{t\mathbf{x} + (1-t)\mathbf{y} | 0 \leq t \leq 1\}$, and by choosing N large enough one more time such that

$$\|\boldsymbol{\nabla}\boldsymbol{G}_{n,h}(\boldsymbol{\theta})\| = h\|\boldsymbol{D_u}\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\boldsymbol{\theta})A\| \leq hL_{n,h}\|A\| < \|A\| < 1, \forall \boldsymbol{\theta} \in \bar{B}_{\mathbb{R}^s}(\mathbf{0}, 1), \quad (7)$$

where

$$L_{n,h} = \sup_{\boldsymbol{\theta} \in \bar{B}_{\mathbb{R}^s}(\mathbf{0},1)} \|\boldsymbol{D_u F}(t_n\mathbf{1} + hC, w_n\mathbf{1} + A\boldsymbol{\theta})\|.$$

By Fixed point Theorem in the complete banach space, $G_{n,h}$ has a unique fixed point, say $\mathbf{x}_h^* \in \bar{B}_{\mathbb{R}^s}(\mathbf{0},1)$. Taking initial term $\mathbf{x}_h^{(0)} \in \bar{B}_{\mathbb{R}^s}(\mathbf{0},1)$, then generating the sequence $\left\{\mathbf{x}_h^{(q)}\right\}_{q\geq 0}$ by $\mathbf{x}_h^{(q+1)} = \boldsymbol{G}_{n,h}\left(\mathbf{x}_h^{(q)}\right)$, we have $\lim_{q\to\infty} \mathbf{x}_h^{(q)} = \mathbf{x}_h^*$.

To choose the right initial term which fulfils that $\left\|\mathbf{x}_h^{(0)}\right\| < 1$, we use linearization

$$\boldsymbol{G}_{n,h}(\mathbf{z}) = h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1}) + h^2\boldsymbol{D_u F}(t_n\mathbf{1} + hC, w_n\mathbf{1})A\mathbf{z} + o(\mathbf{z}^2), \forall \mathbf{z} \in \bar{B}_{\mathbb{R}^s}(\mathbf{0},1),$$

to reasonably take the initial term $\mathbf{x}_h^{(0)}$ to be the solution of the linear equation

$$\mathbf{z} = h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1}) + h^2\boldsymbol{D_u F}(t_n\mathbf{1} + hC, w_n\mathbf{1})A\mathbf{z},$$

or equivalently to the equation

$$[\mathbf{I}_s - h^2\boldsymbol{D_u F}(t_n\mathbf{1} + hC, w_n\mathbf{1})A]\mathbf{z} = h\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1}). \qquad (8)$$

In fact, if (7) is fulfilled, and that $\mathbf{z} = \mathbf{x}_h^{(0)}$ is the solution of (8) then

$$(1 - h\|A\|)\left\|\mathbf{x}_h^{(0)}\right\| \leq \|\mathbf{I}_s - h^2\boldsymbol{D_u F}(t_n\mathbf{1} + hC, w_n\mathbf{1})A\| \left\|\mathbf{x}_h^{(0)}\right\| = h\|\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1})\|,$$

$$\left\|\mathbf{x}_h^{(0)}\right\| \leq \frac{h\|\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1})\|}{1 - h\|A\|}. \qquad (9)$$

So, if we somewhat tightent the condition in (7) by adding the assumption on the right hand side in (9) to satisfy

$$\frac{h\|\boldsymbol{F}(t_n\mathbf{1} + hC, w_n\mathbf{1})\|}{1 - h\|A\|} < 1,$$

we obtain the appropriate initial term $\mathbf{x}_h^{(0)}$, that is $\left\|\mathbf{x}_h^{(0)}\right\| < 1$, for the iteration process to generate the sequence $\left\{\mathbf{x}_h^{(q)}\right\}_{q\geq 0}$ which converges to the fixed point $\mathbf{x}_h^*$. □

3. Implementation to the Implicit Runge-Kutta methods Based On Gaussian Quadrature

The predictor-corrector approach with the initial term $\mathbf{x}_h^{(0)}$ at each step n chosen to be the solution of (8) is used to implement the method (3) and (4). The implementation are presented in Matlab code shown below.

Implementation to the method (3)

```
function outp=RKI4(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 t w;
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;
end
```

Implementation to the method (4)

```
function outp=RKI6(f,a,b,alpha,N,m)
h=(b-a)/N;
t0=a;
w0=alpha;
TW=[t0,w0];
%--------
A=[5/36,(2/9-sqrt(15)/15),(5/36-
sqrt(15)/30);(5/36+sqrt(15)/24),(2/9),(5/36-
sqrt(15)/24);(5/36+sqrt(15)/30),(2/9+sqrt(15)/15),(5/36)];
syms t w;
for i=1:N
    j=1;
Q=eye(3)-h*double(subs(diff(f(t,w),w),[t,w],[t0,w0]))*A;
R=h*f(t0,w0)*[1;1;1]+h^2*double(subs(diff(f(t,w),t),[t,w],[t0,w0]))*[1/
2-sqrt(15)/10;1/2;1/2+sqrt(15)/10];
    Z=linsolve(Q,R);
    k1=Z(1);
    k2=Z(2);
    k3=Z(3);
    while j<=m
   %------------------------------------------------
    U1=5/36*k1+(2/9-sqrt(15)/15)*k2+(5/36-sqrt(15)/30)*k3;
    U2=(5/36+sqrt(15)/24)*k1+(2/9)*k2+(5/36-sqrt(15)/24)*k3;
    U3=(5/36+sqrt(15)/30)*k1+(2/9+sqrt(15)/15)*k2+(5/36)*k3;
    k1=h*f(t0+(1/2-sqrt(15)/10)*h,w0+U1);
    k2=h*f(t0+1/2*h,w0+U2);
    k3=h*f(t0+(1/2+sqrt(15)/10)*h,w0+U3);
    %------------------------------------------------
    j=j+1;
    end
    t0=t0+h;
    w0=w0+5/18*k1+4/9*k2+5/18*k3;
    TW=[TW;t0,w0];
end
%------------------------
outp=TW;
end
```

4. Numerical experiment and Comparison

We present here the comparison of the researched methods of order four and six, denoted as IRK4_PC and IRK6_PC respectively, and some referenced methods including the Implicit Runge-Kutta method of order six implented in the usual appoach, the usual explicit Runge-Kutta method of order four ([3]), the explicit Runge-Kutta method of order six ([4]), the sixth order Backward Difference Formula Off-step block method ([5]) with the initial approximation produced by the above Runge-Kutta method of order four, and the Backward Difference Formula Continuous Block method of order three ([6]-[8]). These referenced methods are denoted as IRK6, RK4, RK6, BDFO6, BDFblock3. We also introduce a new implementation, denoted as BDFO6_IRK6PC, which makes use of IRK6_PC to initiate BDFO6. The experimental results shown in each table below present the absolute error at the last mesh point $t_N = b$ for the corresponding problem. The parameters for considered methods are introduced also in each table.

Example 1 ([3], pp. 321) Given the initial value problem

$$y' = (t + 2t^3)y^3 - ty, t \in [0,2], y(0) = 1/3. \qquad (10)$$

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.

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*

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** $N = 10, 20,30;$
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}
$0.92s, 1.27s, 1.59s, 9.1s$	$0.88s, 1.25s, 1.6s$	$0.98s, 1.33s, 1.69s$	$0.48s, 0.5s, 0.51s$
RK6 $N = 10, 20,30;$	**BDFO6** $N = 10, 20,30,70;$ $M = 10, tol = 0.001$	**BDFblock3** $N = 9, 21,30,69;$ $M = 10, tol = 0.001$	**BDFO6_IRK6PC** $N = 10, 20,30,70;$ $M = 10, tol = 0.001$
1.982×10^{-4}, 1.033×10^{-4}, 6.991×10^{-5}	5.314×10^{-8}, 1.542×10^{-7}, 2.143×10^{-8}, 1.22×10^{-10}	1.181×10^{-4}, 1.132×10^{-5}, 4.174×10^{-6}, 4.04×10^{-7}	2.836×10^{-8}, 1.536×10^{-7}, 2.138×10^{-8}, 1.217×10^{-10}
$1.52s, 1.58s, 1.6s$	$3.58s, 4.91s, 5.92s, 11s$	$3.82s, 5.8s, 7.49s, 15.25s$	$5.88s, 6.87s, 8.04s, 13.4s$

Example 2 Given the initial value problem

$$y' = \left(\frac{1}{t} - 40\right)y + 40t^2 + t, t \in [\ln 2\,, 5], y(\ln 2) = \frac{\ln 2}{2^{40}} + \ln^2 2. \qquad (11)$$

The exact solution of the problem is $y = t^2 + te^{-40t}$. The absolute error at $t_N = 5$ is shown in Table 2 for each method.

Table 2. *Absolute error to the approximation of the solution of (11) at the last mesh point $t_N = 5$ produced by the corresponding method and the time (in second) to perform the calculation corresponding to each number N in the list*

IRK6_PC $N = 10,30,40,70;$ $M = 10$	**IRK4_PC** $N = 10, 30,40,70;$ $M = 10$	**IRK6** $N = 10, 20,30,40,70;$ $M = 10, tol = 0.001$	**RK4** $N = 10, 30,40;$
7.7×10^{33}, 1.31×10^{-1}, 3.698×10^{-3}, 3.483×10^{-6}	2.35×10^{39}, 5.12×10^{-1}, 1.62×10^{-2}, 1.964×10^{-5}	1.324×10^{-1}, 3.46×10^{-2}, 1.443×10^{-2}, 7.486×10^{-3}, 2.021×10^{-3}	2.143×10^{32}, 1.167×10^{39}, 2.574×10^{30}, 2.895×10^{-3}
$2.5s, 4.9s, 5.84s, 9.21s$	$2.47s, 5.08s, 5.8s, 9.3s$	$3.1s, 4.3s, 5.3s, 6.3s, 9.9s$	$1.33s, 1.4s, 1.42s, 1.43s$
RK6	**BDFO6**	**BDFblock3**	**BDFO6_IRK6PC**

$N = 10, 40,70;$	$N = 10, 20,30;$ $M = 10, tol = 0.001$	$N = 9;$ $M = 10, tol = 0.001$	$N = 10, 20;$ $M = 10, tol = 0.001$
3.34×10^{51}, 2.97×10^{45}, 2.915×10^{-2}	1.934, 8.829×10^{-9}, 1.4×10^{-14}, 1.22×10^{-10}	0.7×10^{-14}	1.283×10^{-12}, 0.7×10^{-14}
$1.2s, 1.35s, 1.36s$	$3.08s, 4.4s, 5.74s$	$3.39s$	$5.18s, 6.73s$

Example 3 Given the initial value problem

$$y' = -10y + 10\cos t - \sin t\,, t \in [0,4], y(0) = 2. \qquad (12)$$

The exact solution of the problem is $y = \cos{(t)} + e^{-10t}$. The absolute error at $t_N = 4$ is shown in Table 3 for each method.

Table 3. *Absolute error to the approximation of the solution of (12) at the last mesh point $t_N = 4$ produced by the corresponding method and the time (in second) to perform the calculation corresponding to each number N in the list*

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** $N = 10, 20,30;$
1.004×10^{-2}, 1.538×10^{-6}, 3.801×10^{-9}	4.24×10^{-2}, 1.628×10^{-5}, 5.123×10^{-6}	3.972×10^{-2}, 7.104×10^{-3} 2.727×10^{-3}	9.517×10^{6}, 3.982×10^{-3}, 4.607×10^{-4}
$0.83s, 1.16s, 1.55s$	$0.81s, 1.18s, 1.53s$	$1.14s, 1.37s, 1.72s$	$0.45s, 0.41s, 0.42s$
RK6 $N = 10, 20,30;$	**BDFO6** $N = 10, 20,30,70;$ $M = 10, tol = 0.001$	**BDFblock3** $N = 9, 21,30;$ $M = 10, tol = 0.001$	**BDFO6_IRK6PC** $N = 10, 20,30;$ $M = 10, tol = 0.001$
1.275×10^{10}, 3.065×10^{-4}, 6.712×10^{-4}	1.09×10^{-4}, 1.284×10^{-9}, 7.73×10^{-11}	0.971×10^{-3}, 3.25×10^{-5}, 8.425×10^{-6}	1.345×10^{-8}, 1.284×10^{-9}, 7.729×10^{-11}
$0.42s, 0.44s, 0.46s$	$0.97s, 1.45s, 1.94s$	$1.1s, 1.93s, 2.65s$	$1.75s, 2.21s, 2.67s$

We observe from three examples that the research method IRK6_PC and IRK4_PC and the composition method BDFO6_IRK6 between IRK6_PC and BDFO6 take very higher advantage in treating the non-stiff problem (equation (10)) fairly stiff problem (equation (11)) and very stiff problem (equation (12)) both in the exactness and the computational cost. In fact, it could also be seen easily basing on the global truncation error.

5. Conclusion

A good quality approach to implement the implicit Runge-Kutta method constructed on Gaussian quadrature is presented. This new strategy makes benefit both in less computational cost and higher accuracy. It still has the useful combination to initial the method BDFO6 to dial with the stiffness which is the strong property of this method. This approach could be expected to bring more efficiency for other developments which would be studied in the upcoming reseaches.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] J. C. Butcher, *Numerical Methods for Differential Equations and Applications*, 2nd Edition, John Wiley (Chichester), 2008.

[2] J. C. Butcher, "Practical Runge-Kutta methods for scientific computation," *The ANZIAM Journal*, vol. 50, no. 03, pp. 333-342, 2009, doi: 10.1017/S1446181109000030.

[3] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th Edition, Brooks/Cole, Cengage Learning, 2011.

[4] H. A. Luther, "An explicit sixth-order Runge-Kutta formula," *Mathematics of Computation*, vol. 22, pp. 434-436, 1968.

[5] A. A. Nasarudin, Z. B. Ibrahim, and H. Rosali, "On the Integration of Stiff ODEs Using Block Backward Differentiation Formulas of Order Six," *Symmetry,* vol. 12, 2020, Art. no. 952, doi: 10.3390/sym12060952.

[6] V. T. Dinh and T. T. H. Pham, "Constructing the implimentation to the continuous block BDF methods," (in Vietnamese), *TNU Journal of Science and Technology*, vol. 204, no. 11: Natural Sciences - Engineering - Technology, pp. 23-30, 2019.

[7] O. A. Akinfenwa, S. N. Jator, and N. M. Yao, "Continuous block backward differentiation formula for solving stiff ordinary differential equations," *Computers & Mathematics with Applications*, vol. 65, no. 7, pp. 996-1005, 2013.

[8] O. A. Akinfenwa, S. N. Jator, and N. M. Yao, "On The Stability of Continuous Block Backward Differentiation Formula For Solving Stiff Ordinary Differential Equations," *J. of Mod. Meth*. in Numer. Math., vol. 3, no. 2, pp. 50-58, 2012.