

Numerical Approximation of Riccati Type Differential Equations

Ahmad Fadly Nurullah bin Rasedee¹, Mohamad Hasan Abdul Sathar², Norizarina Ishak^{3*},
Siti Raihana Hamzah³ and Nur Amalina Jamaludin⁴

¹*Faculty of Economics and Muamalat, Universiti Sains Islam Malaysia,
71800 Bandar Baru Nilai, Negeri Sembilan, Malaysia.*

²*Centre of Foundation Studies for Agricultural Science,
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.*

³*Faculty of Science and Technology, Universiti Sains Islam Malaysia,
71800 Bandar Baru Nilai, Negeri Sembilan, Malaysia.*

⁴*Centre for Defence Foundation Studies, Universiti Pertahanan Nasional Malaysia, Selangor, Malaysia.*

Riccati differential equations are one of the most common type of non-linear differential equation used to model real life applications from various fields. The issue when dealing with non-linear differential equations is obtaining their exact solutions. In this research, a three-point block multi-step method in backward difference form is introduced to provide approximated solutions for these Riccati differential equations. The accuracy of the proposed three-point block method will be tested against known numerical methods. The efficiency of the method will apparent when compared with another multi-step method.

Keywords: Riccati equations, Block method, ODEs

I. INTRODUCTION

Applications of Riccati differential equations have become a common occurrence in social and natural sciences. Mathematical models in the form of Riccati differential equations ranges from stochastic realization theory, financial mathematics, network synthesis to random process and diffusion problems. The general Riccati differential equation (RDE) has the form

$$y'(t) = \alpha(t)y(t) + \beta(t)y^2(t) + \gamma(t), \quad t_0 \leq t \leq T,$$

$$y(t_0) = C$$

where the functions $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ are given.

Direct approach using multistep method have become a trend for solving higher order ordinary differential equations. This is because compared to reduction of order

methods, direct methods have shown to be not only accurate but with the added advantage of being cost effective (computational cost). Previous approximation methods for ODEs were considered to be robust because of their efficiency but, due to authors such as Krogh (1973), Lambert (1973) and Suleiman (1989) the interest of researchers was revived. Among the more avid researchers in developing these direct methods includes authors such as Suleiman (2011), Rasedee *et. al.*, (2016) and Ijamet *al.* (2018).

In the current work, a three-point block multistep method is introduced to obtain the numerical approximation for Riccati differential equations. Inspired by research conducted in Suleiman (1989), the three-point block method is infused with a variable order step size algorithm (3PBVOS) to reduce computational cost. The 3PBVOS method is formulated using an Adams-like code to

*Corresponding author's e-mail: norizarina@usim-edu.my

overcome the drawback of calculating integration coefficients at every step change as required when implementing a divided difference code (Suleiman, 1989). By extending the works of Rasedee *et. al.*, (2014) and Ijam *et. al.*, (2014), we established the 3PBVOS formulation in predictor-corrector form for solving Riccati type differential equations.

II. INTEGRATION COEFFICIENTS

This section provides derivation of the three-point integration coefficients. For the derivation of the one-point integration coefficients, refer to Rasedee *et. al.*, (2014) and for the two-point integration coefficients see Ijam *et. al.*, (2014), Consider the higher order ordinary differential equation

$$y^{(d)} = f(t, y, y', y'', \dots, y^{(d-1)}), \quad 1$$

where we denote $(y, y', y'', \dots, y^{(d-1)})$ by

$$Y = (y, y', y'', \dots, y^{(d-1)}),$$

with the initial value conditions

$$Y(a) = \eta,$$

given that

$$\eta = (\eta, \eta', \eta'', \dots, \eta^{(d-1)}), \quad a \leq t \leq b.$$

Obtaining the predictor-corrector formulae begins with the derivation of the explicit integration coefficients. For the explicit coefficients, we first consider the higher order ODE, $y^{(d)}$. By integrating $y^{(d)}$ once, yields

$$y^{(d-1)}(t_{n+3}) = y^{(d-1)}(t_n) + \int_{t_n}^{t_{n+3}} f(t, Y) dt.$$

Next, by substituting the Gregory-Newton polynomial,

$$P_n(t) = \sum_{i=0}^{k-1} \binom{-s}{i} \nabla^i f_n,$$

into the equation above by interpolating the k values $(t_n, f_n), \dots, (t_{n-(k-1)}, f_{n-(k-1)}), (t_{n-k}, f_{n-k})$ provides the following estimate

$$y^{(d-1)}(t_{n+3}) = y^{(d-1)}(t_n) + \int_0^3 \sum_{i=0}^{k-1} \binom{-s}{i} \nabla^i f_n ds,$$

where

$$t = t_n + sh.$$

Let $\phi_{3,1,i}$ be a set of coefficients denoted by the generating function,

$$\Phi_{3,1}(t) = \sum_{i=0}^{\infty} \phi_{3,1,i} t^i, \quad 2$$

given

$$\phi_{3,1,i} = (-1)^i \int_0^3 \binom{-s}{i} ds. \quad 3$$

By substituting the coefficients from (3) into (2), yields

$$\Phi_{3,1}(t) = \sum_{i=0}^{\infty} (-1)^i \int_0^3 \binom{-s}{i} ds.$$

Solving the integral above provides a simpler notation for the generating function, $\Phi_{3,1}(t)$ as follows

$$\Phi_{3,1}(t) = - \left[\frac{(1-t)^{-3}}{\log(1-t)} - \frac{1}{\log(1-t)} \right]$$

and its corresponding integration coefficients

$$\phi_{3,1,0} = 3, \quad \phi_{3,1,k} = 3 - \sum_{i=0}^{k-1} \left(\frac{\phi_{3,1,i}}{k-i+1} \right).$$

The integration process is repeated by integrating (1) from 2 up to d number of times, resulting in a general formulation for the predictor:

$$y^{(d-j)}(x_{n+3}) = \sum_{i=0}^{j-1} \frac{(3h)^i}{i!} y^{(d-j+i)}(x_n) + h^j \sum_{i=0}^{k-1} \phi_{3,j,i} \nabla^i f_n,$$

Through mathematical deduction, a general formula for the explicit integration coefficients is obtained as follows,

$$\phi_{3,d,0} = \phi_{3,d-1,1}, \quad \phi_{3,d,k} = \phi_{3,d-1,k+1} - \sum_{i=0}^{k-1} \left(\frac{\phi_{3,d,i}}{k-i+1} \right),$$

where $k = 1, 2, \dots$ whereas, the implicit integration coefficients are derived in a similar manner as the explicit coefficients, but with a slight difference of changing the limit of integration by the following

$$t = t_{n+3} + sh.$$

This provides the subsequent corrector formulae

$$y^{(d-j)}(t_{n+3}) = \sum_{i=0}^{j-1} \frac{(3h)^i}{i!} y^{(d-j+i)}(t_n) + h^j \sum_{i=0}^{k-1} \phi_{3,j,i}^* \nabla^i f_{n+3}.$$

The derivation will also show that the implicit integration coefficients can be written as

$$\phi_{3,d,0}^* = \phi_{3,d-1,1}^*, \quad \phi_{3,d,k}^* = \phi_{3,d-1,k+1}^* - \sum_{i=0}^{k-1} \left(\frac{\phi_{3,d,i}^*}{k-i+1} \right).$$

A connection between the explicit and implicit integration coefficients is then established. These coefficients are shown to correspond together in the following recursive relationship

$$\sum_{i=0}^{\infty} \phi_{3,j,i}^* t^i = (1-t)^3 \sum_{i=0}^{\infty} \phi_{3,j,i} t^i.$$

III. ERROR ESTIMATION AND STEP CHANGE CRITERION

This research adopts a modified error estimation based on a predict-evaluate correct-evaluate (PeCe) algorithm as suggested by Hall and Watt (1976). In establishing this PeCe algorithm, we have the predictor which takes the form

$$pr y_{n+3}^{(d-j)} = \sum_{i=0}^{j-1} \frac{(3h)^i}{i!} y_n^{(d-j+i)} + h^j \sum_{i=0}^{k-1} \phi_{3,j,i} \nabla^i f_n,$$

with $j = 0, 1, \dots, d$. and its corresponding corrector

$$cr y_{n+3}^{(d-j)} = \sum_{i=0}^{j-1} \frac{(3h)^i}{i!} y_n^{(d-j+i)} + h^j \sum_{i=0}^{k-1} \phi_{3,j,i}^* \nabla^i f_{n+3}.$$

With the advantage of obtaining a recursive relationship between the explicit and implicit integration coefficient, the corrector can be represented by way of the predictor

$$cr y_{n+3}^{(d-j)} = pr y_{n+3}^{(d-j)} + h^j \phi_{3,j,i}^* \nabla_{pr}^k f_{n+3},$$

to reduce computational cost. From here, the local truncation error (LTE) is obtained from corrector of different orders. For purpose of this research, the difference between the corrector of order k and $k-1$ is considered hence, giving the estimate

$$LTE_k^{(j)} = h^j \phi_{3,j,i}^* \nabla_{pr}^k f_{n+3}, \quad j = 0, 1, \dots, d.$$

Because this research implements a variable order step size algorithm, strategies and selection criteria as discussed in Rasedee *et. al.*, (2014) are employed, resulting in the following LTE,

$$LTE_k^{(d-p)} = h^{d-p} \phi_{3,d-p,i}^* \nabla_{pr}^k f_{n+3}, \quad p = 0, \dots, d.$$

which is monitored by selecting the appropriate p to provide less computational time with minimal loss of accuracy.

As mentioned in Rasedee *et. al.*, (2014), a crucial aspect when practicing a variable order step size algorithm is the

acceptance criteria. The decision to accept an integration step effects the reliability of a variable order step size (VOS) algorithm. In this research, the acceptance criteria are based on the local accuracy

$$\frac{1}{(A + B * P_n)} \cdot |Err_k^{(d-p)}| < TOL.$$

where A, B determines the type of test that is selected.

This research adopts a step size changing technique as proposed by Shampine and Gordon (1975) combined with the doubling and halving step size algorithm established by Krogh (1973).

The standard Adam's variable order code algorithm relies entirely on the amount of back values stored. The order of an Adam algorithm can be increased or decreased by simply retaining or discarding the appropriate amount of back values stored.

Variable order strategies for a multistep method simply depends on the back values stored. The order can be increased if the previous back values remains and can be decreased simply by discarding the appropriate amount of back values. Literature shows that there many strategies for implementing variable order algorithm in an Adam based code. In this research, similar strategies suggested by Shampine and Gordon (1975) was adopted. The variable step size strategy chosen for this research adopted the step size changing technique from Shampine and Gordon (1975) coupled with the doubling and halving step size algorithm derived in Krogh (1973).

IV. THREE POINT BLOCK ALGORITHM

For the sake of clarity, we present the algorithm for the for the three-point block method.

Step 1. Calculation of integration coefficients for block one, two, and three.

Step 2. Using the k back values to obtain the predictor and $k+1$ back values to obtain the corrector.

Step 3. Obtain a recursive relationship for integration coefficients between predictor and corrector also for coefficients of different orders.

Step 4. Determine whether E_k satisfies the local accuracy requirements (Tolerance) to decided VOS strategy.

Step 5. If the current step size h , is less than the distance

between the current and end point ($h < |x_{end} - x|$), repeat **Steps 2 - 4**. If not, the ratio r where $r = |x_{end} - x|/h$ then recalculate integration coefficients. Repeat **Steps 2 - 4** and exit the program.

V. NUMERICAL SIMULATIONS AND DISCUSSIONS

Research conducted by authors Ghorbani and Momani (2010), Mokhtarzadeh (2010), Mabood *et. al.*, (2013), Opanuga *et al.* (2015) and Rasedee *et. al.*, (2018) focuses on numerical solutions for Riccati differential equations. The 3PBVOS method is tested with first and higher order Riccati differential problems. Test problems 1 and 2 consist applied RDE problems. The maximum error of the proposed method is then compared with other Adam's type

multistep method to validate its accuracy. For test problems 3 and 4, the exact solutions are unknown. The approximated solution of the 3PBVOS method is then analysed parallel with known methods at various point to show its reliability. The solutions are also estimated with error types suggest in Suleiman (2011). The following are abbreviations used throughout this section:

- MAXERR: the overall maximum error
- MTHD: the method used,
- TOL : tolerance,
- TTS : Truncated Taylor Series,
- RTA : Rational Approximation,
- DI : Direct Integration,
- 2PBVOS: 2-Point Block Variable Order Stepsize,
- 3PBVOS: 3-Point Block Variable Order Stepsize,

Table I: Test Problems.

No.	Problem	Initial conditions	Exact solution
1.	$y'(t) = -y^2(t) + 2y(t) + 1$ $0 \leq t \leq 100$	$y(0) = 0$	$y(t) = 1 + \sqrt{2} \tanh\left(\sqrt{2}t + \frac{1}{2} \log\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)\right)$
2.	$y'(t) = y(t) + a(t)y^2(t) + \sin x$ $a(t) = -\frac{28(784 + \cos t + 29 \sin t)}{(784 + \cos t)^2}$ $0 \leq t \leq 5$	$y(0) = \frac{785}{28}$	$y(t) = \frac{1}{28}(784 + \cos t)$
3.	$y''(t) = 6y^2(t) + \lambda t, \lambda = 1$ $0 \leq t \leq 5$	$y(0) = 1$ $y'(0) = 0$	None
4.	$y''(t) = 2y^3(t) + ty(t) + \mu, \mu = 1$ $0 \leq t \leq 5$	$y(0) = 1$ $y'(0) = 0$	None

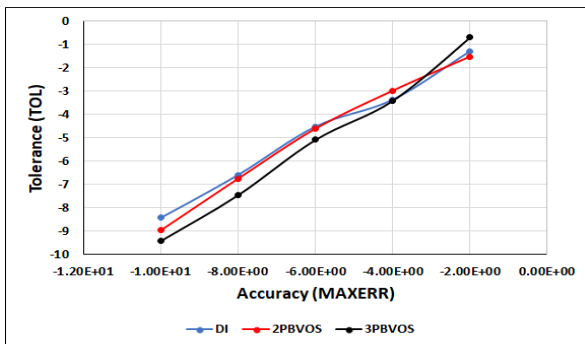


Figure 1: Accuracy of DI, 2PBVOS and 3PBVOS method for Problem 1.

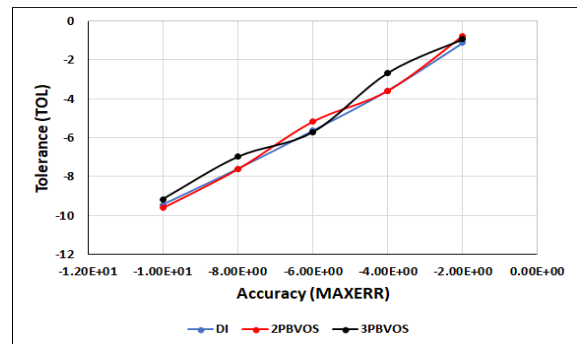


Figure 2: Accuracy of DI, 2PBVOS and 3PBVOS method for Problem 2.

TABLE I and II consists of test problems in the form of Riccati differential equations to validate the accuracy and efficiency of the proposed method. The competitive nature of the proposed 3PBVOS method is illustrated in TABLES II-III and FIGURES 1-2. Numerical results in TABLES II-III compares accuracy of the 3PBVOS against DI and 2PBVOS. Results in TABLE III shows the competitiveness of the 3PBVOS in comparison with 2PBVOS and DI whereas, TABLE II exemplifies the advantage of the

3PBVOS over other methods, with the exception of $TOL = 10^{-2}$. Another obvious advantage 3PBVOS will have over the DI and 2PBVOS method is, when implementing a parallel programming algorithm. Due to its block algorithm, the computational workload of the 3PBVOS method can then be distributed to 3 different processors which will reduce computational cost significantly whereas the 2PBVOS can only distribute its workload to 2 different process.

Table II: Numerical result for Problem 1.

TOL	MTHD	MAXERR	AVER
10^{-2}	DI	5.01213(-2)	3.89184(-3)
	2PBVOS	2.99323(-2)	5.71488(-3)
	3PBVOS	2.04926(-1)	1.11497(-2)
10^{-4}	DI	4.21037(-4)	8.71137(-5)
	2PBVOS	1.03746(-3)	2.80303(-4)
	3PBVOS	3.95254(-4)	4.97861(-5)
10^{-6}	DI	2.94671(-5)	5.56654(-6)
	2PBVOS	2.45586(-5)	2.93877(-6)
	3PBVOS	8.30493(-6)	8.52755(-7)
10^{-8}	DI	2.50331(-7)	5.25393(-8)
	2PBVOS	1.76946(-7)	1.43320(-8)
	3PBVOS	3.52116(-8)	9.68788(-9)
10^{-10}	DI	3.65933(-9)	1.26729(-10)
	2PBVOS	1.09322(-9)	1.14302(-10)
	3PBVOS	3.79234(-10)	9.72491(-11)

Table III: Numerical result for Problem 2.

TOL	MTHD	MAXERR	AVER
10^{-2}	DI	5.01213(-2)	3.89184(-3)
	2PBVOS	2.99323(-2)	5.71488(-3)
	3PBVOS	2.04926(-1)	1.11497(-2)
10^{-4}	DI	4.21037(-4)	8.71137(-5)
	2PBVOS	1.03746(-3)	2.80303(-4)
	3PBVOS	3.95254(-4)	4.97861(-5)
10^{-6}	DI	2.94671(-5)	5.56654(-6)
	2PBVOS	2.45586(-5)	2.93877(-6)
	3PBVOS	8.30493(-6)	8.52755(-7)
10^{-8}	DI	2.50331(-7)	5.25393(-8)
	2PBVOS	1.76946(-7)	1.43320(-8)
	3PBVOS	3.52116(-8)	9.68788(-9)
10^{-10}	DI	3.65933(-9)	1.26729(-10)
	2PBVOS	1.09322(-9)	1.14302(-10)
	3PBVOS	3.79234(-10)	9.72491(-11)

Test problems 3 and 4, are problems without any exact solutions (refer TABLES IV and V). Approximated solution for these problems validates the accuracy of the 3PBVOS method. The Truncated Taylor series (TTS) and Rational Approximation (RTA) are known methods that have been proven to show high accuracy. The reason for comparison with these methods is to show the reliability (in accuracy) of the 3PBVOS method against more established algorithms. Numerical approximation from TABLES IV and V shows that by a point to point comparison of the 3PBVOS rivals with approximations obtained by Truncated Taylor series and Rational Approximation. When a finer tolerance is used, the accuracy of 3PBVOS becomes more evident.

VI. ACKNOWLEDGEMENTS

This article was supported by the Ministry of Education (MoE) and Universiti Sains Islam Malaysia (USIM) under the Fundamental Research Grant Scheme (FRGS), project number USIM/FRGS/FEM/055002/51517 and Universiti Putra Malaysia under Grant Putra (GP), project number GP-IPM/2017/9589600.

Table IV: Numerical result for Problem 3.

t	3PBVOS			TTS	RTA
	TOL = 1×10^{-1}	TOL = 1×10^{-5}	TOL = 1×10^{-10}		
0.0	1.00000 (o)	1.00000 (o)	1.00000 (o)	1.00000 (o)	1.00000 (o)
0.2	1.12673 (o)	1.12670 (o)	1.12637 (o)	1.12640 (o)	1.12640 (o)
0.4	1.59121 (o)	1.58484 (o)	1.58305 (o)	1.58310 (o)	1.58310 (o)
0.6	2.67815 (o)	2.72634 (o)	2.72125 (o)	2.72120 (o)	2.72120 (o)
0.8	6.59993 (o)	6.05708 (o)	6.03835 (o)	6.03830 (o)	6.03830 (o)
1.0	2.52538 (1)	2.34075 (1)	2.33937 (1)	2.33936 (1)	2.33860 (1)

Table V: Numerical result for Problem 4.

t	3PBVOS			TTS	RTA
	TOL = 1×10^{-1}	TOL = 1×10^{-5}	TOL = 1×10^{-10}		
0.0	1.00000 (o)	1.00000 (o)	1.00000 (o)	1.00000 (o)	1.00000 (o)
0.2	1.06336 (o)	1.06269 (o)	1.06261 (o)	1.0620 (o)	1.06260 (o)
0.4	1.30608 (o)	1.27459 (o)	1.27415 (o)	1.27420 (o)	1.27420 (o)
0.6	1.72803(o)	1.73044 (o)	1.72538 (o)	1.72540 (o)	1.72540 (o)
0.8	2.71372(o)	2.74206 (o)	2.73694 (o)	2.73690 (o)	2.73690 (o)
1.0	1.52444 (1)	6.32029 (o)	6.31100 (o)	6.31100 (o)	6.31040 (o)

VII. REFERENCES

- de Araujo, A. L., Lemos, A., Alves, A. M. and Pedroso, K. M. 2018, Some results on Riccati equations, Floquet theory and applications. *Journal of Fixed Point Theory and Applications*,20(1),43.
- Ghorbani, A. and Momani, S.. 2010, An effective variational iteration algorithm for solving Riccati differential equations. *Applied Mathematics Letters*,23(8),922-927.
- Hall, G. and Watt, J. M. 1976, *Modern Numerical Methods for Ordinary Differential Equations*. Clarendon Press, Urbana.
- Ijam, H. M., Suleiman, M., Rasedee, A. F. N., Senu, N., Ali, A. and Salahshour, S. 2014, Solving nonstiff higher-order ordinary differential equations using 2-point block method directly. *Abstract and Applied Analysis*, 2014, 867095, 13 pages.
- Ijam, H. M., Ibrahim, Z. B., Senu, N., Suleiman, M. and Rasedee, A. F. N. 2018, Order and stability of 2-point block backward difference method. *AIP Conference Proceedings*,1974,020054.
- Krogh, F. T. 1973, Algorithms for changing the step size. *SIAM Journal on Numerical Analysis*,10(5),949-965.
- Lambert, J. D. 1973, *Computational methods in ordinary differential equations*. Wiley.
- Mabood, F., Md Ismail, A. I. and Hashim, I. 2013, Application of optimal homotopy asymptotic method for the approximate solution of Riccati equation. *Sains Malaysiana*,42(6),863-867.
- Mokhtarzadeh, M. R., Pournaki, M. R. and Razani, A. 2010, A note on periodic solutions of Riccati equations. *Nonlinear Dynamics*,62(1),119-125.
- Opanuga, A. A., Edeki, S. O., Okagbue, H. I. and Akinlabi, G.O. 2015, Numerical solution of two-point boundary value problems via differential transform method. *Global Journal of Pure and Applied Mathematics*,11(2),801-806.
- Rasedee, A. F. N., Suleiman, M. and Ibrahim Z. B. 2014, Solving nonstiff higher order odes using variable order step size backward difference directly, *Mathematical Problems in Engineering*, 2014, 565137, 10 pages.
- Rasedee, A. F. N., Ijam, H. M., Abdul Sathar, Deraman, F., Ijam, H. M., Suleiman, M. and Saaludin, N. and Rakhimov, A. 2016, 2 point block backward difference method for solving Riccati type differential problems. *AIP Conference Proceedings*,1775,030005.
- Rasedee, A. F. N., Ijam, H. M., Abdul Sathar, M. H., Ishak, N., Hamzah, S. R. and Ismail, I. 2018, Solution for nonlinear Riccati equation by block method. *AIP Conference Proceedings*,1974,020071.
- Shampine, L. F. and Gordon, M. K. 1975, *Computer solution of ordinary differential equations: the initial value problem*. Freeman.
- Suleiman, M. 1989, Solving nonstiff higher order ODEs directly by the direct integration method. *Applied Mathematics and Computation*,33(3),197-219.
- Suleiman, M., Ibrahim Z. B. and Rasedee, A. F. N. 2011, Solution of higher-order ODEs using backward difference method. *Mathematical Problems in Engineering*,2011,810324, 18 pages.