Chapter 14 Variable Order Step Size Algorithm for Solving Second Order ODEs

Chapter · December 2019

CITATIONS
0

READS
37

6 authors, including:

Ahmad Fadly Nurullah Rasedee
USIM | Universiti Sains Islam Malaysia
50 PUBLICATIONS **90** CITATIONS
SEE PROFILE

Mohammad Hasan Abdul Sathar
Universiti Putra Malaysia
29 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Tze Jin Wong
Universiti Putra Malaysia
22 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Koo Lee Feng
Universiti Putra Malaysia
9 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Chapter 14

Variable Order Step Size Algorithm for Solving Second Order ODEs

Ahmad Fadly Nurullah Rasedee[1,*], Mohamad Hassan Abdul Sathar[2], Wong Tze Jin[3], Koo Lee Feng[3]

[1] Faculty of Economics and Muamalat, Universiti Sains Islam Malaysia, 78100 Nilai, Negeri Sembilan, Malaysia.
[2] Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.
[3] Department of Basic Sciences and Engineering, Faculty of Agriculture and Food Science, Universiti Putra Malaysia, Bintulu Campus, 97008 Bintulu, Sarawak, Malaysia.

*Corresponding author: fadlynurullah@usim.edu.my

Abstract

Previous multi step method using a divided difference formulation for solving higher order ordinary differential equations (ODEs) requires calculating the integration coefficients at every step. In the current research, a multi step method in backwards difference form is established. The backward difference formulation offers a solution to the tedious calculation of integration coefficients. Rather than calculating integration coefficients at every step change, a backward difference formulation requires calculating integration coefficients only once in the beginning and if required once more at the end. The proposed method will also be equipped with a variable order step size algorithm to reduce computational cost (calculation time). Both linear and nonlinear second order ODEs will used to validate the accuracy and efficiency of the proposed method.

Keywords: ODE, backward difference, multistep method.

14.1 Introduction

Various of science and engineering problems are found in the form of higher order Ordinary Differential Equations (ODEs). A few examples where these problems can be found are, in the motion of projectiles, the bending of a thin clamped beam and growth population. Previously, it was common practice to solve these higher order ODEs by reducing them to a system of first order equations. These methods worked, so that methods for solving higher order ODEs were disregarded as robust

codes. In this research, an efficient algorithm for solving higher order, or in the current case second order ODEs with Initial Value Conditions (IVCs) directly using variable order step size method in backward difference formulation is developed. The advantages of solving second order systems directly compared to reducing it to first order systems will be apparent.

The approach for solving ODEs using multistep methods was made popular by authors such as [1–4]. Beginning from reduction to first order method to the current method of solving higher order ODEs directly. Suleiman in [4] initially proposed solving higher order ODEs using a divided difference multistep method. This lead to the current interest of using multistep methods for solving higher ODEs directly. Suleiman [4] designed a multistep code for solving stiff and nonstiff higher order ODEs directly without the need for reducing the problems to first order. This method was regarded as the Direct Integration (DI) method. The drawback of the proposed DI method was the tedious calculations of the divided differences in computing integration coefficients at every step change. Current research influenced by the works of [4] includes research by authors such as [5], [6, 7], [8–10], [11, 12] and [13–16].

In the current research, a multistep method based on backward difference formulation in predictor-corrector mode is established with variable order step size capability. The derivation of the proposed method begins as follow.

14.2 Derivation of The Predictor-Corrector Formulation

First consider the second order ordinary differential equation (ODE) in the general form

$$y'' = f(x, y, y'), \tag{14.1}$$

with $\widetilde{Y}(\alpha) = \widetilde{\eta}$ as the initial solution in the interval $\alpha \leq x \leq \beta$, and the initial value conditions are given by

$$\widetilde{Y}(x) = (x, y, y'), \qquad \widetilde{\eta} = (\eta, \eta'). \tag{14.2}$$

In order to obtain the predictor, the explicit integration coefficients need to be established.

14.2.1 Explicit Coefficients

For the evaluation of the explicit integration coefficients, y_{n+1} consider the second order ODE in (14.1). The derivation begins by integrating equation (14.1), once as follows

$$\int_{x_n}^{x_{n+1}} y''(x_{n+1})dx = \int_{x_n}^{x_{n+1}} f(x, y, y')dx. \tag{14.3}$$

This yields

$$y'(x_{n+1}) = y'(x_n) + \int_{x_n}^{x_{n+1}} f(x, y, y')dx. \tag{14.4}$$

Next, $f(y, y')$ is interpolated by the Newton-Gregory backward difference polynomial, $P_n(x)$

$$P_n(x) = \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n, \qquad s = \frac{x - x_n}{h}. \tag{14.5}$$

and substituting $dx = hds$ changes the limit of integration, thus giving

$$y'(x_{n+1}) = y'(x_n) + \int_0^1 \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n \, hds. \tag{14.6}$$

By denoting, $\gamma_{1,i}$ by

$$\gamma_{1,i} = (-1)^i \int_0^1 \begin{pmatrix} -s \\ i \end{pmatrix} ds.$$

and substituting $\gamma_{1,i}$ into equation (14.6) yields

$$y'(x_{n+1}) = y'(x_n) + h \sum_{i=0}^{k-1} \gamma_{1,i} \nabla^i f_n \, ds, \tag{14.7}$$

This is followed by defining the generating function, $G_1(t)$ of the coefficients $\gamma_{1,i}$ as

$$G_1(t) = \sum_{i=0}^{\infty} \gamma_{1,i} t^i. \tag{14.8}$$

Then by substituting $\gamma_{1,i}$ in the generating function as defined in (14.8) and solving the integral gives

$$G_1(t) = -\left[\frac{(1-t)^{-1}}{\log(1-t)} - \frac{1}{\log(1-t)} \right]. \tag{14.9}$$

The 2^{nd} order generating function, $G_2(t)$ is obtained by integrating equation (14.1) twice, followed by repeating steps from equation (14.3) to (14.8), hence

$$G_2(t) = \left[\frac{1}{\log(1-t)} - \frac{-1}{\log(1-t)} \left[\frac{(1-t)^{-1}}{\log(1-t)} - \frac{1}{\log(1-t)} \right] \right]. \tag{14.10}$$

The generating function, $G_2(t)$ then can be rewritten in terms of $G_1(t)$

$$G_2(t) = \left[\frac{1}{\log(1-t)} - \frac{G_1(t)}{\log(1-t)} \right].$$

14.2.2 Implicit Coefficients

The implicit integration coefficients can be obtained in a similar manner as the explicit coefficients with some subtle differences. As the prior, the derivation of the implicit coefficients also begins with equation (14.3). Again, using the Newton-Gregory backward difference polynomial to interpolate, $f(y, y')$ with the difference of substituting

$$s = \frac{x - x_{n+1}}{h},$$

thus, resulting in

$$y'(x_{n+1}) = y'(x_n) + \int_{-1}^{0} \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} -s \\ i \end{pmatrix} \nabla^i f_n \, h ds. \tag{14.11}$$

Now, denote $\gamma^*_{1,i}$ by

$$\gamma^*_{1,i} = (-1)^i \int_{-1}^{0} \begin{pmatrix} -s \\ i \end{pmatrix} ds$$

which gives

$$y'(x_{n+1}) = y'(x_n) + h \sum_{i=0}^{k-1} \gamma^*_{1,i} \nabla^i f_n \, ds, \tag{14.12}$$

The implicit generating function, $G^*_1(t)$ can be mathematically deduced as the following formulation

$$G^*_1(t) = -\left[\frac{1-(1-t)}{\log(1-t)}\right] \tag{14.13}$$

which can be generalized as

$$G^*_{(d)}(t) = \frac{(1-t)}{(d-1)!}\left[\frac{1}{\log(1-t)} - \frac{(d-1)! G^*_{(d-1)}(t)}{\log(1-t)}\right] \qquad d = 1, 2. \tag{14.14}$$

Calculation of the explicit and implicit coefficients directly involving large numbers of integration can be tedious and time consuming. To overcome this drawback, a recurrence relationship between integration coefficients is provided. This enables for a more efficient code when programming the algorithm. The recurrence relationship can expressed as follows

$$G^*_{(d)}(t) = (1-t) G_{(d)}(t), \qquad d = 1, 2. \tag{14.15}$$

From the generating function, its corresponding integration coefficients is represented as

$$\sum_{i=0}^{k} \gamma^*_{(d),i} = \gamma_{(d),k} \tag{14.16}$$

14.3 Order and step size criteria

The order and step size selection of a variable order step size algorithm is based on its acceptance criteria. This acceptance criteria will determine whether to increase the order and step size. When handling a variable order step size algorithm, determining the success of an integration step is crucial. The threshold for each integration step is predetermined by setting an acceptable tolerance level (TOL). The success of an integration step depends on whether the estimated error, $|E_k^{(d-p)}|$ satisfies the following local accuracy requirements

$$\text{TOL} > \frac{|E_k^{(d-p)}|}{A + B + P_n} \tag{14.17}$$

where A and B determines the type of error test used. Hence, every estimated error that satisfies the local accuracy condition also fulfills the acceptance criteria.

The variable order in a multistep method relies on the back values stored. The order may be increased if back values from the previous step are retained and may be decreased simply by relinquishing the appropriate amount of back values. The order strategies adopted here are similar to strategies proposed in [2].

When implementing a variable step size algorithm, Shampine and Gordon suggests restrictions on ratio of successive step size due to convergence and stability issues of variable step size techniques to ensure stability. Because the proposed method is based on the Adams-Bashforth formulation as predictor and Adams-Moulton formulation as corrector (PECE mode), we adopt the doubling or halving the step size algorithm from [1] which is implements a step size changing technique from [17].

14.4 Numerical Results

The tables and figures below show the numerical results for Problems 1 to 3 which were solved using Direct Integration and Backwards Difference method directly. Numerical result for methods that reduces second order ODES to first order systems are also included as a benchmark. In this section, numerical result includes the evaluation of maximum and average values of the error in the computed solution y. The efficiency of the 1PBD will be validated by comparing results of obtain by Suleiman's DI method.

Following are abbreviation used in this section
TOL: Tolerance level
MTHD: Method
TS: Total steps
FS: Fail steps
MAX: Maximum error (exponent of 10)
AVE: Average error (exponent of 10)
TIME: Execution time in micro seconds
D1: Reduction to first order divided difference method
D1: Reduction to first order backward difference method
DI: Direct Integration method
1PBD: 1 Point Backward Difference method
TTS: Truncated Taylor Series
RTA: Rational Approximation

Problem 1:(source: Suleiman [18])

$$y_1''(x) = -\frac{y_1}{r^3}, \quad , y_2''(x) = -\frac{y_2}{r^3}, \quad r = \left(y_1^2 + y_2^2\right)^{\frac{1}{2}} \quad 0 \le x \le 16\pi.$$

Initial condition

$$y_1(0) = 1 \quad y_1'(0) = 0, \quad y_2(0) = 0, \quad y_2'(0) = 1.$$

Exact Solution

$$y_1(x) = \cos x, \quad y_2(x) = \sin x.$$

Problem 2:(source: Rasedee [19])

$$y''(t) = 2y^3(t) + ty(t) + \mu, \quad 0 \leq x \leq 5$$

Initial condition

$$y\,(0) = 1, \quad y'\,(0) = 0.$$

Exact Solution

$$unknown$$

Problem 3:(source: Rasedee [20])

$$y''(x) + y(x) + y'(x) + y^2(x)y'(x) = 2\cos x - \cos^3 x, \quad 0 \leq x \leq 100$$

Initial condition

$$y(0) = 0, \quad y'(0) = 1,$$

Exact Solution

$$y(x) = \sin x$$

Table 1 and 3 displays numerical results for problem 1 and 3. The results displayed compares steps taken, accuracy and computational cost between the D1, B1, DI and 1PBD method. The D1 and B1 method are the traditional reduction to first order method, where as the DI and 1PBD method are direct solution methods. Reduction to first order methods are used as benchmarks to validate the viability of the direct method. Table 2 on the other hand, provide approximation of a problem without any known solution. In this table, the accuracy of the 1PBD method is test against more established methods.

Numerical result shown in Table 1 are results for nonlinear second order ODE with periodic solution (two body problem). Result shows the superiority of the 1PBD method in terms of accuracy and computational cost and its competitiveness in steps taken, especially at tolerance 10^{-10} where the difference in steps required is more than 100 compared to its nearest rival.

Table 14.1: Numerical results of D1, B1, DI and 1PBD method for problem 1.

TOL	MTHD	TS	FS	MAX	AVE	TIME
10^{-2}	D1	113	5	8.80309(-2)	1.19400(-1)	2078
	B1	89	3	2.07488(-1)	6.87734(-2)	1628
	DI	76	1	8.78700(-2)	2.56159(-2)	969
	1PBD	71	0	1.17774(-1)	1.70700(-2)	969
10^{-4}	D1	151	2	4.30130(-3)	5.45887(-3)	3024
	B1	170	1	5.37539(-4)	1.06165(-4)	3169
	DI	94	1	4.25614(-3)	1.53472(-3)	1284
	1PBD	149	0	5.05592(-6)	1.21782(-3)	1951
10^{-6}	D1	275	3	1.80685(-6)	1.81711(-6)	5227
	B1	121	0	1.38184(-5)	2.65555(-6)	4077
	DI	179	1	3.15166(-4)	1.29162(-4)	2294
	1PBD	176	0	5.88468(-6)	2.26518(-6)	2272
10^{-8}	D1	348	2	3.49871(-8)	2.75009(-8)	6310
	B1	424	2	4.59227(-7)	1.19307(-7)	5367
	DI	205	0	6.69118(-5)	2.76364(-5)	8665
	1PBD	209	0	6.96499(-8)	2.35911(-8)	2664
10^{-10}	D1	526	9	3.45300(-9)	2.61398(-9)	9866
	B1	475	10	7.88269(-9)	1.39352(-9)	8822
	DI	370	0	1.44822(-7)	5.81042(-8)	4639
	1PBD	248	0	4.13390(-9)	8.63144(-10)	3109

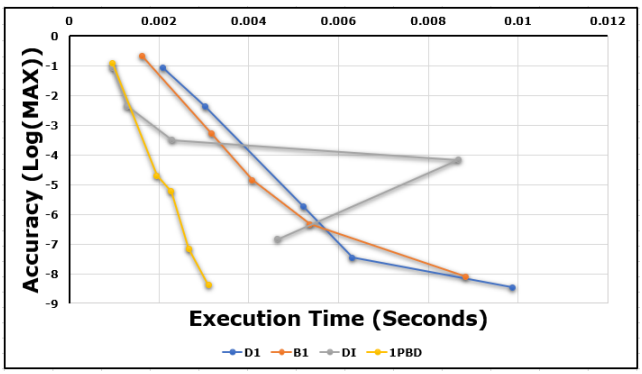

Figure 14.1: Efficiency of D1, B1, DI and 1PBD method for problem 1.

Table 2 contains results for a Riccatti type second order ODE without any exact solution. This problem was selected because of its level of difficulty. The 1PBD method is then compared against TTS ([21]) and RTA ([22]) methods which accuracy has been established for solving second order ODEs. Results in the current table reflects the accuracy of the proposed 1PBD method. The accuracy of the 1PBD increases when using smaller tolerances, to the point of matching accuracy provided by the TTS and RTA methods.

Table 3 provide results of approximated solution for Problem 3. Problem 3 is a Duffing type second order ODE and was selected for this research to test real life application problems. The selected problem has similar features with oscillatory problems with damping force. Numerical results provided in Table 3 shows the advantage of the 1PBD method in terms of accuracy and total step size over its counterpart, the DI method. The 1PBD out performs the DI method in least number of steps required at every tolerance. In terms of accuracy, the 1PBD method shows to be superior at almost every tolerance level with the exception of TOL= 10^{-2}.

Table 14.2: Comparison of accuracy for problem 2.

t	1PBD			TTS	RTA
	TOL$= 1 \times 10^{-2}$	TOL$= 1 \times 10^{-5}$	TOL$= 1 \times 10^{-10}$		
0.0	1.00000(0)	1.00000(0)	1.00000(0)	1.00000(0)	1.00000(0)
0.2	1.07160(0)	1.06262(0)	1.06261(0)	1.06260(0)	1.06260(0)
0.4	1.27252(0)	1.27417(0)	1.27415(0)	1.27420(0)	1.27420(0)
0.6	1.72688(0)	1.72542(0)	1.72538(0)	1.72540(0)	1.72540(0)
0.8	2.80714(0)	2.73708(0)	2.73694(0)	2.73690(0)	2.73690(0)
1.0	6.89972(0)	6.31186(0)	6.31100(0)	6.31100(0)	6.31040(0)

Table 14.3: Comparison of total steps and accuracy for problem 3.

TOL	MTD	STEPS	MAXE	AVER
10^{-2}	DI	254	8.49079(-2)	2.04794(-2)
	1PBD	217	1.07600(-1)	3.03894(-2)
10^{-4}	DI	332	1.54704(-3)	4.72630(-4)
	1PBD	284	1.24649(-3)	1.92899(-4)
10^{-6}	DI	382	4.24089(-5)	1.56849(-5)
	1PBD	330	1.28039(-5)	3.26641(-6)
10^{-8}	DI	651	7.93605(-7)	1.55130(-7)
	1PBD	499	7.27324(-7)	1.46368(-7)
10^{-10}	DI	772	7.83863(-9)	2.03721(-9)
	1PBD	702	9.05773(-9)	1.01381(-9)

Figure 1 illustrates the efficiency of the D1, B1, DI and 1PBD method where as, Figure 2 provides a clear comparison of efficiency between the DI and 1PBD method. Efficiency of the methods is adopted from definition in [16], where the efficiency is illustrated by undermost curve of the provided graphs. The efficiency of the proposed method is clearly presented in both figures where the 1PBD method is the under most curve of all four methods.

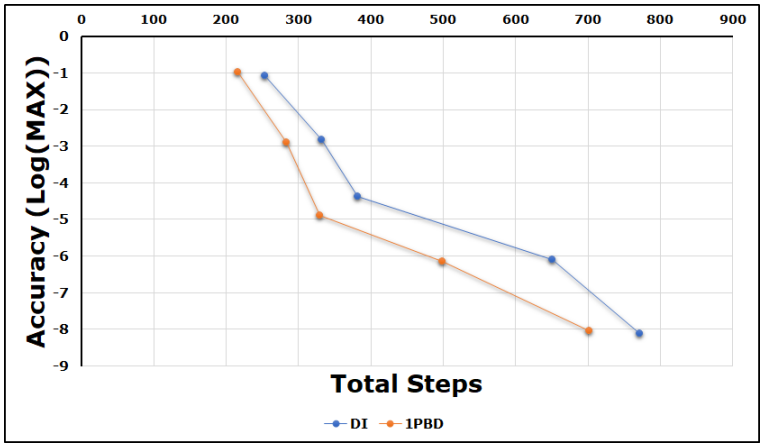

Figure 14.2: Efficiency of DI and 1PBD method for problem 3.

14.5 Conclusion

By justifications above, the 1PBD method proves to be a viable option for solving second order ODEs.

Acknowledgment

The research conducted in this article has been supported by 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.

Bibliography

[1] Krogh, F. T. (1973), *Algorithms for changing the step size*, SIAM Journal on Numerical Analysis, Vol 10(5), 949–965.

[2] Shampine, L. F. and Gordon M. K. (1975), *Local error and variable order Adams codes*, Applied Mathematics and Computation, Vol 1(1), 47–66.

[3] Hall, G. and Suleiman, M. (1981), *Stability of Adams-Type Formulae for Second-Order Ordinary Differential Equations*, IMA Journal of Numerical Analysis, Vol 1(4), 427–438.

[4] Suleiman, M. B. (1989), *Solving nonstiff higher order ODEs directly by the direct integration method*, Applied Mathematics and Computation, Vol 33(3) 197–219.

[5] Majid, Z. A. and Suleiman, M. (2007), *Two point fully implicit block direct integration variable step method for solving higher order system of ordinary differential equations*, World Congress on Engineering, 812–815.

[6] Mohd Ijam, H., Suleiman, M., Rasedee, A. F. N., Senu, N., Ahmadian, A. and Salahshour, S. (2014), *Solving nonstiff higher-order ordinary differential equations using 2-point block method directly*, Abstract and Applied Analysis, Vol 2014, Article ID 867095, 13 pages.

[7] Mohd Ijam, H., 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, Vol 1974(1), 020054.

[8] Waeleh, N., Abdul Majid, Z., Ismail, F. and Suleiman, M. (2012), *Numerical Solution of Higher Order Ordinary Differential Equations by Direct Block Code*, Journal of Mathematics and Statistics, Vol 8(1), 77-81.

[9] Waeleh, N. and Abdul Majid, Z. (2016), *A 4-Point Block Method for Solving Higher Order Ordinary Differential Equations Directly*, Mathematical Problems in Engineering, Vol 2016, Article ID 9823147, 8 pages.

[10] Waeleh, N. and Abdul Majid, Z. (2017), *Numerical Algorithm of Block Method for General Second Order ODEs using Variable Step Size*, Malaysiana, Vol 46(5), 817-824.

[11] Alkasassbeh, M. and Omar, Z. (2015), *hybrid one step block method for the solution of fourth order initial value problems of ordinary differential equations*, International Journal of Pure and Applied Mathematics, Vol 104(2), 159-169.

[12] Alkasassbeh, M. and Omar, Z. (2018), *hybrid one-step block fourth derivative method for the direct solution of third order initial value problems of ordinary differential equations*, International Journal of Pure and Applied Mathematics, Vol 119(1), 207-224.

[13] Rasedee, A. F.N. (2009), *Direct method using backward difference for solving higher order ordinary differential equations*, Selangor: University Putra of Malaysia.

[14] Rasedee, A. F. N., Suleiman, M. B and Ibrahim, Z. B. (2014), *Solving nonstiff higher order odes using variable order step size backward difference directly*, Mathematical Problems in Engineering, Vol 2014, Article ID 565137, 10 pages.

[15] Rasedee, A. F. N., Abdul Sathar, M. H., Deraman, F., Mohd Ijam, H., Suleiman, M., Saaludin, N., and Rakhimov, A. (2016), *2 point block backward difference method for solving Riccati type differential problems*, AIP Conference Proceedings, Vol 1775(1), 030005.

[16] Rasedee, A. F. N., Abdul Sathar, M. H., Ishak, N., Kamarudin, N. S., Nazri, M. A., Ramli, N. A., Ismail, I. and Sahrim, M. (2017), *Solution for nonlinear Duffing oscillator using variable order variable stepsize block method*, Matematika, Vol 33(2): 165–176.

[17] Lambert, J.D. (1973), *Computational Methods in Ordinary Differential Equations*, Wiley, New York.

[18] Suleiman, M. B., Ibrahim, Z. B. and Rasedee, A. F. N. (2011), *Solution of higher-order ODEs using backward difference method*, Mathematical Problems in Engineering, Vol 2011, Article ID 810324, 18 pages.

[19] Rasedee, A. F. N., Ishak, N., Hamzah, S. R., Mohd Ijam, H., Suleiman, M., Ibrahim, Z. B., Abdul Sathar, Ramli, N. A. and Kamarudin, N. S.(2017), *Solution for nonlinear riccati equation by block method*, Variable order variable stepsize algorithm for solving nonlinear Duffing Oscillator. *Journal of Physics: Conference Series*. Vol 890(1): 012045.

[20] Rasedee, A. F. N., Ijam, H. M., Abdul Sathar, M. H., Ishak, N., Hamzah, S. R., Sahrim, M., and Ismail,I. (2018), *Solution for nonlinear riccati equation by block method*, *AIP Conference Proceedings*. Vol 1974(1), 020071.

[21] Simon, W. E. (1965), *Numerical Technique for Solution and Error Estimate for the Initial Value Problem*, Mathematics of Computation, Vol 19(91): 387–393.

[22] Fair, W. and Luke, Y. L. (1966), *Rational approximations to the solution of the second order Riccati equation*, Mathematics of Computation, Vol 20(96): 602–606.