KYUNGPOOK Math. J. 62(2022), 729-736
https://doi.org/10.5666/KMJ.2022.62.4.729
pISSN 1225-6951 eISSN 0454-8124

Series Solution of High Order Abel, Bernoulli, Chini and Riccati Equations

Henk Koppelaar*
Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands
e-mail : Koppelaar.Henk@gmail.com

Peyman Nasehpour
Department of Engineering Science, Golpayegan University of Technology, Golpayegan, Iran
e-mail : nasehpour@gut.ac.ir *and* nasehpour@gmail.com

Abstract To help solving intractable nonlinear evolution equations (NLEEs) of waves in the field of fluid dynamics we develop an algorithm to find new high order solutions of the class of Abel, Bernoulli, Chini and Riccati equations of the form $y' = ay^n + by + c, n > 1$, with constant coefficients a, b, c. The role of this class of equations in NLEEs is explained in the introduction below. The basic algorithm to compute the coefficients of the power series solutions of the class, emerged long ago and is further developed in this paper. Practical application for hitherto unknown solutions is exemplified.

1. Introduction

There is as yet no general solution f for the equation

$$f' = af^n + bf + c \tag{1.1}$$

where f is a function dependent on t, $n > 1$ is a positive integer, and a, b, and c are given constants unequal to 0 in F of characteristic zero. The design of the equation is by D. Bernoulli [2] with $c = 0, n = 2$, as studied by him to predict the effect of smallpox vaccination. If $c \neq 0, n = 2$ then (1.1) is named the Riccati equation with

* Corresponding Author.
Received June 17, 2021; revised May 13, 2022; accepted May 31, 2022.
2020 Mathematics Subject Classification: 34A34, 65Y99.
Key words and phrases: Abel equation, Bernoulli equation, Chini equation, JCPMiller algorithm, Riccati equation, series solution.
The second author is supported by the Department of Engineering Science at the Golpayegan University of Technology and his special thanks go to the Department for providing all necessary facilities available to him for successfully conducting this research.

constant coefficients [20], and $n = 3$ is a special case of the classical Abel equation discussed on p. 24 in [8]. The importance of this equation nowadays originates from the need in the physics of fluid dynamics to develop e.g. accurate weather prediction, neurological implants for deaf and blind people, or control fluids in nano capacities (for medicine administration through the brain-body blood barrier), etc. Fluid dynamics models have the format of partial differential equations (PDEs) of a high order, most often in the class of non-linear evolution Equations (NLEEs). Nature often has a complicated relationship between the phase speed of traveling waves (for instance tsunamis) and the parameters in the PDEs predicting them. The order of an NLEE becomes higher if more precision is required, e.g. dispersion effects are taken into account, but most often these model equations then become unsolvable. Kudryashov [12] designed the 'Simplest Equation Method' (SEM) to reduce the order of intractable partial differential equations (PDEs) by assuming a series solution such that it solves a lower order Riccati equation.

An example of such use of a lower-order differential equation is in [9] for the solution of otherwise unsolvable double dispersion equations, i.e. the Sharma-Tasso-Olver (STO) equation. To glimpse other examples of the application of the work reported in this paper is the Benjamin-Ono PDEs in [23] which hit upon a Chini equation. Also, Chinis method is still under study in applications [17] to Gross-Pitaevskii PDEs.

In general, the field of fluid dynamics progresses if the SEM becomes wider applicable by raising the order of the SEM. This paper gives an automated technique, such that high order solutions become feasible for the SEM.

The main result of this paper is to aid application fields with a computer-assisted method for symbolic differential solutions of (1.1), if a, b, and c are constants. To this end, we devise an algorithmic i.e. a constructive, method to find series solutions of any power $n > 1$, with illustrative applications from [8], [18], [21]. There is as yet no general solution for this differential equation (1.1), if $n > 1$ and a, b, c are general, i.e. non-constant.

2. Power Series Solutions

Recall the formal derivative of power series over a field F with characteristic zero with

$$f = A_0 + A_1 t + A_2 t^2 + \cdots + A_k t^k + \cdots$$

an element of the power series ring $F[[t]]$, where t is an indeterminate over F. Then is the formal derivative f' of f defined as

$$f' := A_1 + 2A_2 t + \cdots + (k+1)A_{k+1} t^k + \cdots . \tag{2.1}$$

The following lemma is known for a long time (see Formula 6.361 in [1], Formula 0.314 in [5], Theorem 1.6c in [7], [14] Ch. 17 (1st edn.), and Ch. 21 (2nd edn.). We bring it for reference.

Lemma 1. *Let F be a field of characteristic zero and $\sum_{k=0}^{\infty} A_k t^k$ a formal power series in $F[[t]]$ with the indeterminate t. Then for each positive integer n,*

$$\Big(\sum_{k=0}^{\infty} A_k t^k\Big)^n = \sum_{k=0}^{\infty} C_k t^k,$$

where $C_0 = (A_0)^n$, and

$$C_m = \frac{1}{mA_0}\sum_{k=1}^{m}(kn - m + k)A_k C_{m-k}, \qquad \textit{for all } m \geq 1.$$

Optimization of computation of power series expansions has been coined JCP-Miller Algorithm in [6]. Later publications of the algorithm are in [7], [14], [22] though the algorithm has a long history. It was apparently known to Euler, according to Henrici [7], p. 65, but Wimp in [22] p. 2 refers to Lord Rayleigh [19] in 1910. Knopp [11], however, attributes it to Glaisher in 1875.

Now we prove the main theorem of this section:

Theorem 2. *Let F be a field of characteristic zero and $f = \sum_{k=0}^{\infty} A_k t^k$ be a power series solution for the differential equation (1.1)*

$$f' = af^n + bf + c$$

where $n > 1$ is a positive integer, and a, b, c, and A_0 are given elements in F. If $C_0 = (A_0)^n$ and $m \geq 1$,

$$C_m = \frac{1}{mA_0}\sum_{k=1}^{m}(kn - m + k)A_k C_{m-k},$$

then

$$A_1 = aC_0 + bA_0 + c,$$

and

$$A_m = \frac{1}{m}(aC_{m-1} + bA_{m-1}), \qquad \textit{for all } m > 1.$$

Proof. Assume that a, b, c, and $f(0) = A_0$ are given elements in F and let $f = \sum_{k=0}^{\infty} A_k t^k$ be a power series solution for the differential equation (1.1). It is clear that $f' = \sum_{k=0}^{\infty}(k+1)A_{k+1}t^k$. On the other hand, by Lemma 1, a power series raised to the power n is calculated with the formula

$$\Big(\sum_{k=0}^{\infty} A_k t^k\Big)^n = \sum_{k=0}^{\infty} C_k t^k,$$

where $C_0 = (A_0)^n$ and

$$C_m = \frac{1}{mA_0}\sum_{k=1}^{m}(kn - m + k)A_k C_{m-k}, \qquad \text{for all } m \geq 1. \tag{2.2}$$

Finally, from the differential equation (1.1), we obtain

$$A_1 = aC_0 + bA_0 + c, \quad A_m = \frac{1}{m}(aC_{m-1} + bA_{m-1}), \qquad \text{for all } m > 1. \tag{2.3}$$

This completes the proof. □

Let us recall the following Theorem 2.4.2 in [3]:

Theorem 3. (Existence and Uniqueness for First-Order Nonlinear Equations)
Let the functions f and $\partial f/\partial y$ be continuous in some rectangle $n < t < \beta$ and $\gamma < y < \delta$ containing the point (t_0, y_0). Then, in some interval $t_0 - h < t < t_0 + h$ contained in $n < t < \beta$ there is a unique solution $y = \varphi(t)$ of the initial value problem $y' = f(t, y)$ and $y(t_0) = y_0$.

Corollary 4. *Let $a, b,$ and c be arbitrary real constants with $a \neq 0$ and let $n > 1$ be integer. Then, there is a unique solution $y = \varphi(t)$ satisfying equation (1.1)*

$$\frac{dy}{dt} = y' = ay^n + by + c \tag{2.4}$$

with $y(t_0) = y_0$ on an open interval of real numbers including x_0.

Proof. Let us define $f(t, y) = ay^n + by + c$. Then it is obvious that the functions f and $\partial f/\partial y$ are continuous and so by Theorem 3, the solution is unique. □

3. Automating Power Series Solutions

The first recurrence in Theorem 2 expresses $C_m, m = 0, 1, 2, \ldots$ in terms of a, b, c, and A_m. The A_m are external for this algorithm, i.e. 'global' in Maple's terminology. Rewriting this by a recursive program in the mathematical symbolic programming language Maple, we obtain the listing below of Miller's algorithm with the shortcut name C.

```
C := proc(m,n,A0)  local k; global A;  option remember;
     if m=0 then (A0)^n
   else expand(simplify(add((k(n+1)-m)*A[k]*
              C(m-k,n,A0),k=1..m)/m/A0)
end if end proc:
```

Maple's symbolic output of successive calls of Miller's algorithm $C(m, 4, A_0)$ for $m = 0, 1, \dots, 5$ is

$C(0, 4, A_0) = A_0^4$

$C(1, 4, A_0) = 4A_0^3 A_1$

$C(2, 4, A_0) = 4A_0^3 A_2 + 6A_0^2 A_1^2$

$C(3, 4, A_0) = 4A_0^3 A_3 + 12A_0^2 A_1 A_2 + 4A_0 A_1^3$

$C(4, 4, A_0) = 4A_0^3 A_4 + 12A_0^2 A_1 A_3 + 6A_0^2 A_2^2 + 12A_0 A_1^2 A_2 + A_1^4$

$C(5, 4, A_0) = 4A_0^3 A_5 + 12A_0^2 A_1 A_4 + 12A_0^2 A_2 A_3 + 12A_0 A_1^2 A_3 + 12A_0 A_1 A_2^2 + 4A_1^3 A_2$

Remark 5. The proof of Theorem 2 also holds if the assumed solution is a formal Laurent series $\sum_{k=q}^{\infty} A_k t^k$, where $q \neq 0$ is an integer number and may or may not be negative. A polynomial or power series over a field with a nonzero constant term, i.e. $A_0 \neq 0$ is named 'unit' in [7]. Note that the formal Laurent series over a field form a field again. The algorithm CLaurent is the Laurent extension of Miller's algorithm C at p. 55 in [7], made available in Remark 6 hereafter.

Remark 6. The recurrences (2.2) and (2.3) of Theorem 1 are such that each A_m can be expressed in terms of a, b, c, and A_0, for each $m \geq 1$. Both A and C do necessarily execute in this order: $A_0 \Rightarrow C_0 \Rightarrow A_1 \Rightarrow C_1 \Rightarrow \cdots \Rightarrow A_{m-1} \Rightarrow C_{m-1} \Rightarrow A_m$. The full computation is given in Theorem 2. The first two steps are $A_0 \Rightarrow C_0$ and subsequently $C_0 \Rightarrow A_1$. Thereafter follow $A_m = (aC_{m-1} + bA_{m-1})/m$, for all $m > 1$.

An algorithm - embedding previous tools - to find all symbolic solutions of the equation (2.4), whenever the constants a, b, c, and A_0 are given, is such that if these constants are numerically substituted, the algorithm solves correctly Bernoulli, Riccati, Abel as well as Chini's equations [8]. Because of this generality, we name our algorithm ABCR in alphabetic order of the names of the original scholars Abel, Bernoulli, Chini and Riccati. The body of the algorithm ABCR first sets the initial values A_0, A_1, C_1 and thereafter it alternates - as said - between updating C_k and A_k until the stop condition is satisfied.

```
ABCR := proc(a,b,c,m,n,A0) local k; global A,C;
     A[0]:=A0;                    C[0]:=C(0,n,A0);
     A[1]:=a*C[0]+b*A[0]+c;       C[1]:=C(1,n,A0);
     for k from 2 to m do
          C[k]:=C(k,n,A0);
          A[k]:=expand(simplify((a*C[k-1]+b*A[k-1])/k))
end do end proc
```

This algorithm has to be changed slightly if non-units solutions are needed, i.e. the assumed power series f has $A_0 = 0$. Then the call to Miller's C algorithm needs to replaced by a call to the CLaurent algorithm, as follows.

```
CLaurent:=proc(m,n,q,Aq); local k,l; global A; option remember;
        l:=m-n * q;
      if l<0 then 0
   elif l=0 then (Aq)^n
   else expand(simplify(add((k(n+1)-l)*A[k+q]*
           CLaurent(m-k,n,q,Aq),k=1..m)/m/Aq))
 end if end proc:
```

4. Applications

Below we give applications by showing outputs of our ABCR algorithm.

Example 7. The solution of Bernoulli's equation $c = 0$ in (2.4) is known in general

$$\Big(\frac{1}{Ce^{-(n-1)t}-1}\Big)^{\frac{1}{n-1}},$$

where C is a constant depending on a, b and a boundary condition $y(0)$.

For the particular case $y'(t) = (y(t))^2 + y(t), y(0) = 1$, let $n = 2, a = 1, b = 1, c = 0$ in (1.1), and collate the known solution

$$\frac{1}{2e^{-t}-1} = -\frac{1}{2}(1 + \coth(\frac{t}{2} - \frac{1}{2}\ln(2)))$$

to the result below, by executing ABCR$(1, 1, 0, 7, 2, 1)$: by Maple. This gives the coefficients of the series solution by $seq(A_k t^k, k = 0 \ldots 7)$;. The output is the exact sequence of coefficients of expansion of the known solution.

$$1, 2t, 3t^2, \frac{13}{3}t^3, \frac{25}{4}t^4, \frac{541}{60}t^5, \frac{1561}{120}t^6, \frac{47293}{2520}t^7, \ldots$$

Example 8. All test cases in the table below, taken from the literature, give corect results:

n	a	b	c	$y(0)$	reference
2	1	1	0	1	example 7 above
2	-1	1	0	1/2	example 4 in [4]
3	1	1	0	1	example 3.1 in [10]
2	-1	-2	-1	1/2	example 1 in [15], [16]
2	1	-2	1	2	example 4.1 in [10]

Example 9. Solution of an hitherto unknown Chini equation with $n = 9, a = 1, b = 1, c = 1$ in (1.1):

$$1, 3t, \frac{15}{2}t^2, \frac{61}{2}t^3, \frac{1061}{8}t^4, \frac{23917}{40}t^5, \frac{133105}{48}t^6, \frac{22012957}{1680}t^7, \ldots$$

.

Example 10. Execution of the Laurent version of the algorithm

$$seq(\text{CLaurent}(m, 4, 1, A1), m = 0 \cdots 7)$$

on a truncated non-unit series for a fourth order equation, for example with $q = 1$, hence $A_1 \neq 0$, and $a_1 t + a_2 t^2 + \cdots + a_7 t^7$. The Claurent algorithm gives coefficients of $C_m, m = 0 \ldots 7$, as follows.

$$0, 0, 0, 0, A_1^4, 4A_1^3 A_2, 4A_1^3 A_3 + 6A_1^2 A_2^2,\ 4A_1^3 A_4 + 12A_1^2 A_2 A_3 + 4A_1 A_2^3.$$

5. Conclusion

An advantage of our method for equations with constant coefficients is the result of a series expansion without effort to analyse whether it has periodic solutions, or not. Our method grants solutions of all types. The series of solutions are obtained by the property discovered by Euler: a particular solution f_1 generates a series of solutions u via the substitution $f = u + 1/f_1$.

For the molten slag problem in metallurgy - a special case of the celebrated NLEEs in fluid flow dynamics - no general solution for the slag temperature $y(t)$ flowing out of the furnace was found thus far, see [13]. Our method solves this problem in full generality. The slag outflow temperature is fully known and predictable by our symbolic output. Our result is new to this metallurgy application field.

Also, by our method, we generate the needed expansion of the solution of Kudryashov's [12] SEM (Simplest Equation Method), to simplify otherwise intractable partial differential equations (PDEs) in the fluid dynamics field.

The technique is general, easy to implement via the two algorithms above, and yields exact expansion results.

References

[1] E. P. Adams and R. L. Hippisley, *Smithsonian Mathematical Formulae and Tables of Elliptic Functions*, Miscellaneous Collections, Smithsonian Institution, Washington(1922).

[2] D. Bernoulli, *Essai dune nouvelle analyse de la mortalit cause par la petite vrole, et des advantage de linoculation pour la prvenir*, Die Werke von Daniel Bernoulli, Analysis und Wahrscheinlichkeitsrechnung(1766).

[3] W. E. Boyce, R. C. DiPrima and D. B. Meade, *Elementary Differential Equations and Boundary Value Problems*, John Wiley and Sons, Hoboken(2017).

[4] F. Ghomanjani and E. Khorram, *Approximate solution for quadratic Riccati differential equation*, J. Taibah Univ. Sci., **11(2)**(2017), 246–250.

[5] I. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, Amsterdam(2017).

[6] P. Henrici, *Automatic Computations with Power Series*, J. Assoc. Comput. Mach., **3(1)**(1956), 10–15.

[7] P. Henrici, *Applied and Computational Complex Analysis Vol. 1*, Wiley, New York(1988).

[8] E. Kamke, *Differentialgleichungen. Losungsmethoden Und Losungen. I. Gewohnliche Differentialgleichungen*, Teubner, Stuttgart(1977).

[9] K. Khan, H. Koppelaar, A. M. Akbar and T. Mohyud-Din, *Analysis of travelling wave solutions of double dispersive sharma-Tasso-Olver equation*, J. Ocean Eng. Science, **3(18)**(2022), 1–14.

[10] S. Khuri and A. M. Wazwaz, *The successive differentiation computer-assisted method for solving well-known scientific and engineering models*, Int. J. Numer. Methods Heat Fluid Flow, **28(12)**(2018), 2862–2873.

[11] K. Knopp, *Theorie und Anwendung der Unendlichen Reihen*, Springer Verlag, Berlin(1964).

[12] N. Kudryashov, *Simplest equation method to look for exact solutions of nonlinear differential equations*, Chaos Solitons Fractals, **24**(2005), 1217–1231.

[13] C. Liu, H. Wu, and J. Chang, *Research on a Class of Ordinary Differential Equations and Application in Metallurgy*, in Zhu, R. et al. (Ed.), International Conference on Information Computing and Applications ICICA 2010, **II**, Springer-Verlag(2010).

[14] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms*, Academic Press, Elsevier(1978).

[15] Y. Pala and M. O. Ertas, *An Analytical Method for Solving General Riccati Equation*, Int. Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, **11(3)**(2017), 125–130.

[16] Y. Pala and M. O. Ertas, *A New Analytical Method for Solving General Riccati Equation*, Universal Journal of Applied Mathematics, **5**(**2**)(2017), 11–16. This paper is identical to the one just above.

[17] A. da Silva Pinto, P. N. da Silva, A. L. C. dos Santos *A note about a new method for solving Riccati differential equations*, Int. J. Innov. Educ. Res. **10**(2022), 123–129.

[18] A. D. Polyanin and V. F. Zaitsev, *Handbook of Ordinary Differential Equations*, Handbook of Ordinary Differential Equations, CRC Press, Taylor and Francis Group, Oxford(2018).

[19] Lord Rayleigh, *The Incidence of Light upon a Transparent Sphere of Dimensions Comparable with the Wave-Length*, Proceedings of the Royal Society of London. Series A, **84(567)**(1910), 25–46.

[20] W. T. Reid, *Riccati Differential Equations*, Academic Press, Amsterdam(1972).

[21] G. N. Watson, *Treatise on the Theory of Bessel Functions*, Cambridge University Press(1944).

[22] J. Wimp, *Computation with Recurrence Relations*, Pitman Publishing, Boston(1984).

[23] W. Zhao, M. Munir, G. Murtaza and M. Athar, *Lie symmetries of Benjamin-Ono equation*, Math. Biosci. Eng., **18**(2021), 9496–9510.