NEW BASIC RATIONAL APPROXIMATION METHOD FOR SOLVING SINGULAR INITIAL VALUE PROBLEMS OF ORDINARY DIFFERENTIAL EQUATIONS

by

M. O. Ogunniran and R. B. Adeniyi

Department of Mathematical Sciences, Osun State University Osogbo Nigeria. & Department of Mathematics, University of Ilorin, Ilorin Nigeria.

muioxygen@gmail.com, raphade@unilorin.edu.ng

Abstract
Most numerical methods for solving Initial Value Problems (IVPs) of Ordinary Differential Equations (ODEs) are based on the local representation of the theoretical solution to problems near singular or singular IVPs by polynomials in h and this presents poor integration of the IVPs. Rational methods are found suitable for numerical solution of such problems thus, in this paper we derive and implement a new numerical method based on the rational approximation of the theoretical solution of singular IVPs. Numerical examples are presented.

Keywords: Singularity, Singular Initial Value Problems, Rational Approximation, Stability.

Introduction
Some differential equations of the form:

\[ y' = f(x, y), \quad y(a) = y_0 \]

\[ y, f \in \mathbb{R}^n \quad \text{and} \quad x \in [a, b], a, b \in \mathbb{R} \]  

(1)

possess some kind of properties which make them very difficult to obtain a solution or the numerical solution may present very poor integration. Such properties are the property of Singularity and Stiffness. The conventional one-step scheme is given by:

\[ y_{n+1} = y_n + h \phi_n \]  

(2)

where \( \phi_n \) is the incremental function and the conventional Linear Multistep Method (LMM) is described by:

\[ \sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} \]  

(3)

Since the usual formulation in (2) is exclusively based on the local representation of the theoretical solution to problem (1) by polynomials in h, the resultant algorithms generally perform poorly when the IVP is stiff or when its solution possess singularity.

In general, the theory of ordinary nonlinear differential equations offers no clue as to the singularities of the solutions of such equations. Thus, the detection of singularities must be accomplished heuristically. Obviously the usual numerical integration techniques fail in the region of such singularity, but also the location of such a point evades detection (Luke et al., 1975). Hence, new techniques must be developed which will deal effectively with the problem of singularities of solutions to nonlinear differential equations.
Over the years, several studies have been carried out and until now, three classes of methods have been used successfully in the numerical solution of singularity IVPs. The categories of methods include: perturbed polynomial methods due to Lambert (1974); rational methods given by Lambert et al. (1965), Luke et al. (1965), Fatunla (1982, 1986), Van Niekerk (1987, 1988) Otunta et al. (1999), Ikhile (2002), Odekunle et al. (2004), Okosun et al. (2007), Teh (2014), and Garwood et al. (2016); extrapolation methods used by Fatunla (1986), and Ikhile (2002, 2004). Rational methods are found suitable for the numerical solution of singular IVPs when the zeros of the denominator are the singularities of the IVPs. The use of rational functions as approximants has been studied by many authors, but the main concern of most of this work has been direct approximation of a given function.

Method of Study

Given that the IVP as defined below has the property of singularity,

\[ y' = f(x, y), \quad y(a) = y_a, \]

\[ y, f \in \mathbb{R}^n \quad \text{and} \quad x \in [a, b], a, b \in \mathbb{R} \]  \hspace{1cm} (4)

We suggest an approximation to the theoretical solution \( y(x_{n+1}) \) of (4) by:

\[ y_{n+1} = y(x_{n+1}) = y(x_n + h) = \frac{\sum_{i=0}^{m} a_i h^i}{1 + bh} \]  \hspace{1cm} (5)

where \( a_i; (0 \leq i \leq m), b \) are parameters to be determined and they contain approximation of \( y_n \) and higher derivatives of \( y_n \).

From (5), we define the difference operator as:

\[ [y(x); h] = y(x_n + h)(1 + bh) - \sum_{i=0}^{m} a_i h^i \]  \hspace{1cm} (6)

where \( y(x) \) is a continuous and differentiable function on \( x \in [a, b] \subset \mathbb{R} \). Expanding \( y(x_n + h) \) in Taylor’s series and collecting terms in (6), we obtain:

\[ [y(x); h] = c_0 + c_1 h + c_2 h^2 + \ldots + c_{k-1} h^{k-1} + c_k h^{k} + \ldots \]  \hspace{1cm} (7)

where \( c_i (i = 0, 1, 2, \ldots, k - 1, k, \ldots, m) \) contain corresponding parameters which need to be determined.

Definition

A numerical scheme is said to be of order \( p = k \) if in the difference equation (7),

\[ c_0 = c_1 = c_2 = \ldots = c_k = 0, \]

and

\[ c_{k+1} \neq 0 \]

and the local truncation error, \( \text{LTE} = c_{k+1} h^{k+1} + o(h^{k+2}) \).

We now expand (5) by Taylor’s series and collect terms.

CASE I: \( m = 1 \) (SECOND ORDER RATIONAL METHOD)

With \( m = 1 \) in (5) together with (6), we have:
\[ y_{n+1} = \frac{\sum_{i=0}^{n} a_i h^i}{1 + bh} = \frac{a_0 + a_1 h}{1 + bh} \]  

(8)

\[ c_n = a_n - y_n, \]

\[ c_1 = a_1 - y'_n - by_n, \]

\[ c_2 = \frac{y''_n}{2} + by'_n, \]

\[ c_3 = \frac{y'''_n}{6} + \frac{by''_n}{2} \ldots \]

Setting \( c_n = c_1 = c_2 = 0 \), we have:

\[ a_n = y_n \]

\[ a_1 = \frac{2(y'_n)^2 - y_n y''_n}{2y''_n}, \]

\[ b = -\frac{y''_n}{2y''_n}, \]

and substituting into (8), we obtain the corresponding one-step second order formula:

\[ y_{n+1} = \frac{2y_n y'_n + 2h(y'_n)^2 - hy_n y''_n}{2y''_n - hy''_n} \]

(9)

with the difference equation, the local truncation error is obtained as:

\[ \text{LTE} = \frac{(2y_n + (y'_n)^2 - hy_n y''_n)}{64y''_n} + o(h^4) \]

(10)

**CASE II: \( m = 2 \) (THIRD ORDER RATIONAL METHOD)**

With \( m = 2 \) in (5),

\[ y_{n+1} = \frac{\sum_{i=0}^{3} a_i h^i}{1 + bh} = \frac{a_0 + a_1 h + a_2 h^2 + a_3 h^3}{1 + bh} \]

(11)

Thus by virtue of (6) and (7), we have:

\[ c_n = a_n - y_n, \]

\[ c_1 = a_1 - y'_n - by_n, \]

\[ c_2 = \frac{y''_n}{2} + by'_n - a_2, \]

\[ c_3 = \frac{y'''_n}{6} + \frac{by''_n}{2} \]

\[ c_4 = \frac{y'''}{24} + \frac{by'''}{6} \ldots \]

Setting \( c_n = c_1 = c_2 = c_3 = 0 \), we have:

\[ a_n = y_n, \]

\[ a_1 = \frac{3y_n y'' - y_n y'''}{3y''_n}, \]

\[ a_2 = \frac{3y_n y''' - y_n y''''}{3y''_n}, \]

121
\[ a_2 = \frac{3(y''')^2 - 2y'y'''}{6y''}, \]
\[ b = -\frac{y'''}{3y''} \]
and substituting into (11), we obtain the corresponding one-step third order formula:
\[ y_{n+1} = \frac{6y_n y_n'' + 6hy_n y_n''' - 2hy_n y_n'' + 3h^2 (y_n')^2 - 2h^2 y_n' y_n'''}{6y_n'' - 2hy_n'''} \quad (12) \]
with the difference equation, the local truncation error is obtained as:
\[ \text{LTE} = \frac{y_n''}{24} - \frac{(y_n''')^2}{18y_n''}h^2 + \mathcal{O}(h^3) \quad (13) \]

**CASE III: \( m = 3 \) (FOURTH ORDER RATIONAL METHOD)**

With \( m = 3 \) in (5),
\[ y_{n+1} = \frac{\sum_{i=0}^{4} c_i h^i}{1 + bh} = \frac{a_n + a_i h + a_i h^2 + a_i h^3}{1 + bh} \quad (14) \]
By virtue of (6) and (7), we have:
\[ c_0 = a_n - y_n, \]
\[ c_1 = a_i - y_i' - by_n, \]
\[ c_2 = \frac{y''}{2} + by' - a_2, \]
\[ c_3 = \frac{y'''}{6} + \frac{by''}{2} - a_3, \]
\[ c_4 = \frac{y^{iv}}{24} + \frac{by'''}{6}, \]
\[ c_5 = \frac{y^{iv}}{120} + \frac{by'''}{24} ... \]
Setting \( c_3 = c_1 = c_2 = c_4 = 0 \), we have:
\[ a_n = y_n, \]
\[ a_i = \frac{4y_i'y''' - y_i y'''}{4y''}, \]
\[ a_2 = \frac{2y''y''' - y''y'''}{4y''}, \]
\[ a_4 = \frac{4(y''')^2 - 3y''y'''}{24y''}, \]
\[ b = -\frac{y^{iv}}{4y''} \]
and substituting into (14), we obtain the corresponding one-step fourth order formula:
\[ y_{n+1} = \frac{24y_n y'' + 24hy_n y''' - 6hy_n y'' + 12h^2 y_n y'' + 6h^2 y'' - 4h^3 y_n y'''}{24y'' - 6hy_n'''} \quad (15) \]
with the difference equation, the local truncation error is obtained as:

\[
\text{LTE} = \frac{y''}{120} - \frac{(y'''^2)}{96y''^2}h^5 + o(h^6)
\]  

(16)

**Stability of Methods**

**Definition (Lambert 1991):** A numerical method is said to be L-stable if it is A-stable and in addition, when applied to the scalar test problem:

\[ y' = \lambda y, \text{Re}(\lambda) < 0, \]

it yields

\[ y_{n+1} = R(z)y_n, z = h\lambda. \]

where

\[ |R(z)| \to 0 \text{ as } \text{Re}(z) \to -\infty. \]

The obtained formulae in cases I, II and III are used to solve the scalar test problem:

\[ y' = \lambda y, \text{Re}(\lambda) < 0 \]

as described by Dahlquist to test for stability and they all satisfy the definition of Lambert (1991). It was proven to be stable with L-stability.

**General Formula of the Method**

The generalization of the method is of course needful, so we therefore obtained a generalized form as:

\[
y_{n+1} = \frac{p!y_n y^{(m)}}{p!} + \sum_{r=1}^{m} \frac{p!}{r!} \frac{y_{n+r} y^{(m)}}{r!} (\frac{(p-1)!}{(r-1)!} y^{(r-1)} y^{(m+1)}) h^r - (p-1)!h y^{(m+1)}
\]

(17)

we take \( y_n^{(0)} = y_n \) and \( p = m + 1 \) where \( p \) = order of the method, \( m \) = the approximation term and \( n \) = iteration number.

This method has an advantage of estimating the error apriori and the general form is as follow:

\[
\text{LTE} := \left(\frac{y_n^{(m+2)}}{(m+2)!} - \frac{(y^{(m+1)}^2)}{(m+1)^2} \frac{y^{(m+2)}}{m! y^{(m)}} h^2 + o(h^{(m+3)})\right)
\]  

(18)

**Results and Discussion**

In this section, we implement the fourth order method obtained in (15) to illustrate the accuracy of the method. All computations were carried out with a written MATLAB code. Let \( y(x_n) \) be the theoretical solution and \( y_n \) the approximate solution in the range \( x \in [0,1] \). We find the maximum absolute error by \( |y(x_n) - y_n| \).

**Example 1:** Consider the non-singular IVP:

\[
y' = \frac{1}{2} - 2x; y(0) = 1
\]

with theoretical solution given by:

\[
y(x) = e^x + \frac{x}{2}
\]

Example 2: \[ y' = 1 + y^2; \ y(0) = 1 \]
with the theoretical solution given by: \[ y(x) = \tan(x + \pi/4) \]
Also, we compare the performance of NBRAM of order 4 with Lambert (1965) of order 4, 
RK4 and Garwooe et al. (2016).

**Table 1.** Absolute errors for \( y' = \frac{1}{2} - x^2 y; \ y(0) = 1, h = 0.1 \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>( y(x_n) )</th>
<th>RK4</th>
<th>Lambert (1965)</th>
<th>Niekerk (1988)</th>
<th>NBRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000000000</td>
<td>1.0000000000</td>
<td>0.000 \times 10^0</td>
<td>0.000 \times 10^0</td>
<td>0.000 \times 10^0</td>
<td>0.000 \times 10^0</td>
</tr>
<tr>
<td>0.1000000000</td>
<td>1.2714027582</td>
<td>1.107 \times 10^{-2}</td>
<td>1.107 \times 10^{-2}</td>
<td>2.989 \times 10^{-2}</td>
<td>1.107 \times 10^{-2}</td>
</tr>
<tr>
<td>0.2000000000</td>
<td>1.5918246976</td>
<td>2.460 \times 10^{-2}</td>
<td>2.459 \times 10^{-2}</td>
<td>6.726 \times 10^{-3}</td>
<td>2.459 \times 10^{-2}</td>
</tr>
<tr>
<td>0.3000000000</td>
<td>1.9721188004</td>
<td>4.112 \times 10^{-2}</td>
<td>4.110 \times 10^{-2}</td>
<td>1.146 \times 10^{-1}</td>
<td>4.110 \times 10^{-2}</td>
</tr>
<tr>
<td>0.4000000000</td>
<td>2.4255409285</td>
<td>6.130 \times 10^{-2}</td>
<td>6.127 \times 10^{-2}</td>
<td>1.750 \times 10^{-1}</td>
<td>6.127 \times 10^{-2}</td>
</tr>
<tr>
<td>0.5000000000</td>
<td>2.9682818285</td>
<td>8.594 \times 10^{-2}</td>
<td>8.591 \times 10^{-2}</td>
<td>2.520 \times 10^{-1}</td>
<td>8.591 \times 10^{-2}</td>
</tr>
<tr>
<td>0.6000000000</td>
<td>3.6201169227</td>
<td>1.116 \times 10^{-1}</td>
<td>1.527 \times 10^{-1}</td>
<td>3.501 \times 10^{-1}</td>
<td>1.160 \times 10^{-1}</td>
</tr>
<tr>
<td>0.7000000000</td>
<td>4.4051999668</td>
<td>1.528 \times 10^{-1}</td>
<td>1.534 \times 10^{-1}</td>
<td>4.751 \times 10^{-1}</td>
<td>1.528 \times 10^{-1}</td>
</tr>
<tr>
<td>0.8000000000</td>
<td>5.3530324244</td>
<td>1.977 \times 10^{-1}</td>
<td>1.976 \times 10^{-1}</td>
<td>6.341 \times 10^{-1}</td>
<td>2.176 \times 10^{-1}</td>
</tr>
<tr>
<td>0.9000000000</td>
<td>6.4996474644</td>
<td>2.526 \times 10^{-1}</td>
<td>2.525 \times 10^{-1}</td>
<td>8.359 \times 10^{-1}</td>
<td>2.525 \times 10^{-1}</td>
</tr>
<tr>
<td>1.0000000000</td>
<td>7.8890560989</td>
<td>3.196 \times 10^{-1}</td>
<td>3.191 \times 10^{-1}</td>
<td>1.091 \times 10^{0}</td>
<td>3.194 \times 10^{-1}</td>
</tr>
</tbody>
</table>

**Table 2.** Absolute errors for \( y' = 1 + y^2; \ y(0) = 1, h = 0.05 \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>( y(x_n) )</th>
<th>Lambert (1965)</th>
<th>Garwood et al. (2016)</th>
<th>NBRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000000000</td>
<td>1.0000000000</td>
<td>0.000 \times 10^0</td>
<td>0.000 \times 10^0</td>
<td>0.000 \times 10^0</td>
</tr>
<tr>
<td>0.1000000000</td>
<td>1.2230488804</td>
<td>7.534 \times 10^{-8}</td>
<td>2.153 \times 10^{-8}</td>
<td>2.368 \times 10^{-5}</td>
</tr>
<tr>
<td>0.2000000000</td>
<td>1.5084976471</td>
<td>1.829 \times 10^{-7}</td>
<td>2.797 \times 10^{-7}</td>
<td>7.689 \times 10^{-6}</td>
</tr>
<tr>
<td>0.3000000000</td>
<td>1.8957651229</td>
<td>3.578 \times 10^{-7}</td>
<td>5.213 \times 10^{-7}</td>
<td>7.134 \times 10^{-5}</td>
</tr>
<tr>
<td>0.4000000000</td>
<td>2.4649627567</td>
<td>6.869 \times 10^{-7}</td>
<td>3.631 \times 10^{-6}</td>
<td>2.826 \times 10^{-4}</td>
</tr>
<tr>
<td>0.5000000000</td>
<td>3.4080223442</td>
<td>1.997 \times 10^{-4}</td>
<td>2.598 \times 10^{-5}</td>
<td>8.502 \times 10^{-4}</td>
</tr>
<tr>
<td>0.6000000000</td>
<td>5.3318552235</td>
<td>3.815 \times 10^{-6}</td>
<td>2.914 \times 10^{-4}</td>
<td>2.766 \times 10^{-3}</td>
</tr>
<tr>
<td>0.7000000000</td>
<td>11.681373800</td>
<td>1.981 \times 10^{-3}</td>
<td>1.336 \times 10^{-2}</td>
<td>1.537 \times 10^{-1}</td>
</tr>
<tr>
<td>0.8000000000</td>
<td>28.238252850</td>
<td>1.207 \times 10^{-4}</td>
<td>5.436 \times 10^{-1}</td>
<td>8.548 \times 10^{-2}</td>
</tr>
<tr>
<td>0.8500000000</td>
<td>-15.457896136</td>
<td>3.961 \times 10^{-3}</td>
<td>1.400 \times 10^{-2}</td>
<td>8.964 \times 10^{-3}</td>
</tr>
<tr>
<td>0.9000000000</td>
<td>-8.6876295465</td>
<td>1.316 \times 10^{-3}</td>
<td>2.693 \times 10^{-3}</td>
<td>5.730 \times 10^{-3}</td>
</tr>
<tr>
<td>0.9500000000</td>
<td>-6.0202997164</td>
<td>6.672 \times 10^{-3}</td>
<td>9.525 \times 10^{-3}</td>
<td>3.935 \times 10^{-3}</td>
</tr>
<tr>
<td>1.0000000000</td>
<td>-4.5880378250</td>
<td>4.109 \times 10^{-3}</td>
<td>5.704 \times 10^{-1}</td>
<td>2.926 \times 10^{-3}</td>
</tr>
</tbody>
</table>
Computational Details

The computational experiments was implemented via MATLAB 8.0 version on a personal computer with the following specifications.

- System name: Acer Aspire E15
- Processor: Intel(R) Pentium(R) CPU N3530 @ 2.16GHz
- Installed memory (RAM): 4.00GB
- System Type: 64-bit Operating System, x64-based processor
- Operating system: 3.9 Windows Experience Index.

Thus, the CPU time for computing solution for different methods is given as:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1000</td>
<td>0.7031</td>
<td>0.6719</td>
<td>0.7031</td>
<td>NA</td>
<td>0.5625</td>
</tr>
<tr>
<td>2</td>
<td>0.05000</td>
<td>0.7101</td>
<td>0.5156</td>
<td>NA</td>
<td>0.5287</td>
<td>0.4531</td>
</tr>
</tbody>
</table>

The numerical results in Tables 1, 2 and 3 dearly demonstrate the power of rational approximations in dealing with a function which has singular points within the range of definition. Since the theory of ordinary nonlinear differential equations offers no clue as to the singularities of the solutions of such equations, the detection of singularities must be accomplished heuristically. For Example 2, we consider a singular example which has singular point at $x = \pi/4 \approx 0.7854$, It was observed that at the interval of discontinuity $[\pi,1]$, RK4 fails while NBRAM outperforms the method of Garwood et al. (2016) within the region of definition. Computational experience, while observing the Tables of Errors and the CPU time, shows that the new method perform favourably for both non-singular and singular when compared with with existing methods of IVPs in ODEs.
Acknowledgments: This reasearch work was motivated by Prof. S. N. Jator of Austin Peay State University, USA. The Authors are really indebted to him.

References
Lambert, J. D. and Shaw, B. (1965), On the numerical solution of \( y' = f(x, y) \) by a class of formulae based on rational approximation, Maths Comp. 19, 456-462.