



---

## **A COMPARISON OF THE SOLUTIONS OF NONLINEAR DIFFERENTIAL EQUATIONS BY ADOMIAN DECOMPOSITION AND GENERALIZATION OF MILNE METHODS**

**N. TAGHIZADEH, S. KETABCHI and M. T. SADIDI**

Department of Mathematics  
Faculty of Science  
University of Guilan  
Rasht, Iran  
e-mail: taghizadeh@guilan.ac.ir; sketabchi@guilan.ac.ir  
msadidi@gmail.com

### **Abstract**

In this paper, we compare the numerical solutions of nonlinear ordinary differential equations (N.O.D.Es) obtained by Adomian decomposition, Milne and Generalization of Milne methods. The numerical results obtained by this way have been compared with the exact solution to show the efficiency of the method. Also, for comparison, we use two factors: Precision of convergence or value of error and Time of computing. Naturally, a method in the comparison among other methods is said to be better when the obtained solution undergoes least error and the time of computation be lowest.

### **1. Introduction**

The Adomian decomposition method is useful method for obtaining the numerical solutions of many equations without using restrictive

---

2000 Mathematics Subject Classification: 34K28.

Keywords and phrases: Adomian decomposition method, Milne method, generalization of Milne method, predictor formula, corrector formula.

Received May 14, 2008

assumption or discrimination or linearization. This method can be an effective procedure to obtain analytic and approximate solution of linear and nonlinear, ordinary, partial, deterministic or stochastic differential equations.

The Generalization of the Milne method is useful method for obtaining the numerical solution of the first order differential equations  $y' = f(x, y)$  with the initial condition  $y(x_0) = y_0$  and we use the predictor formula:

$$y_{n+1} = y_{n-5} + \frac{3}{10} \Delta x (11y'_{n-4} - 14y'_{n-3} + 26y'_{n-2} - 14y'_{n-1} + 11y'_n)$$

and corrector formula:

$$\begin{aligned} y_{n+1} &= \frac{60}{157} \Delta x (y'_{n-4} + y'_{n-3} + y'_{n-2} + y'_{n-1} + y'_n + y'_{n+1}) \\ &\quad + \frac{1}{157} (157y_{n-4} - 275y_{n-3} + 400y_{n-2} - 400y_{n-1} + 275y_n), \end{aligned}$$

where  $y(x_k) = y_k$  [5].

## 2. Adomian Decomposition Method

Assume there are the following linear and nonlinear operators:

$$Ly + Ry + Ny = g, \quad (1)$$

where  $L$  and  $R$  are linear and  $N$  is nonlinear. Assume  $L$  operator is of the highest order of derivation, i.e.,  $L = \frac{d^n}{dx^n}$ . Hence

$$Ly = g - Ry - Ny. \quad (2)$$

The inverse operator  $L^{-1}$  is therefore considered an  $n$ -fold integral operator defined by

$$L^{-1}(\cdot) = \int_0^x \int_0^x \cdots \int_0^x (\cdot) dx dx \cdots dx$$

is then operated on relation (2) to yield

$$L^{-1}Ly = L^{-1}g - L^{-1}Ry - L^{-1}Ny. \quad (3)$$

The left side of the relation (3) will be obtained as

$$L^{-1}Ly = y - Q_x, \quad (4)$$

where  $Q_x$  is a function in terms of  $x$ . From relations (3) and (4), we can write

$$y = Q_x + L^{-1}g - L^{-1}Ry - L^{-1}Ny. \quad (5)$$

Suppose the answer to relation (1) is as follows

$$y = \sum_{n=0}^{\infty} y_n \quad (6)$$

and the nonlinear part of relation (1) is as follows

$$Ny = \sum_{n=0}^{\infty} A_n(y_0, y_1, \dots, y_n), \quad (7)$$

where  $A_n$ 's are the Adomian polynomials given by

$$A_n = \frac{1}{n!} \left[ \frac{d^n}{d\lambda^n} N \left( \sum_{i=0}^{\infty} y_i \lambda^i \right) \right]_{\lambda=0}, \quad n = 0, 1, 2, \dots$$

Then the substitution of relations (6) and (7) for relation (5) will yield

$$\sum_{n=0}^{\infty} y_n = Q_x + L^{-1}g - L^{-1}R \sum_{n=0}^{\infty} y_n - L^{-1} \sum_{n=0}^{\infty} A_n. \quad (8)$$

From the decomposition of relation (8), we will have the following assumptions:

$$\begin{cases} y_0 = Q_x + L^{-1}g \\ y_{n+1} = -L^{-1}Ry_n - L^{-1}A_n \quad (n \geq 0). \end{cases} \quad (9)$$

Using relations (9), we can rewrite them as follows to obtain  $y$ :

$$y_0 = Q_x + L^{-1}g,$$

$$y_1 = -L^{-1}Ry_0 - L^{-1}A_0,$$

$$\begin{aligned}
y_2 &= -L^{-1}R(-L^{-1}Ry_0 - L^{-1}A_0) - L^{-1}A_1 \\
&= (-L^{-1}R)^2 y_0 + (L_{-1}R)L^{-1}A_0 - L^{-1}A_1, \\
y_3 &= -L^{-1}R((-L^{-1}R)^2 y_0 + (L^{-1}R)L^{-1}A_0 - L^{-1}A_1) - L^{-1}A_2 \\
&= (-L^{-1}R)^3 y_0 - (L^{-1}R)^2 L^{-1}A_0 + (L^{-1}R)L^{-1}A_1 - L^{-1}A_2 \\
&= (-1)^3(L^{-1}R)^3 y_0 + \sum_{k=0}^2 (-1)^{k+1}(L^{-1}R)^{2-k}L^{-1}A_k \\
&\vdots \\
y_n &= (-1)^n(L^{-1}R)^n y_0 + \sum_{k=0}^{n-1} (-1)^{k+1}(L^{-1}R)^{n-k-1}L^{-1}A_k.
\end{aligned}$$

If we consider the sum of the  $m$  terms  $y_0, y_1, y_2, \dots, y_{m-1}$  in relation (6), that is  $\Phi_m = \sum_{k=0}^{m-1} y_k$ , when  $m \rightarrow \infty$ ,  $\Phi_m$  tends to  $y$ . This means that  $\Phi_m$  is an appropriate approximation of  $y$ . The terms in the above series soon tend to zero, where  $\frac{1}{(mn)!}$  has been the coefficient of calculations derived from integration,  $m$  is the number of terms and  $n$  is the order of the operator derivation. Therefore, it has a rapid convergence [1].

### 3. Numerical Results

We present here, a comparison between Adomian decomposition method and Milne method and Generalization of Milne method by offering two examples of nonlinear differential equations with initial condition. In these examples, we consider the sum of the 10 terms of  $y_i$ 's in the Adomian decomposition method.

#### Example 1

For finding the numerical solution of  $y' = e^{-y}$  with initial condition  $y(0) = 1$  when  $0 < x < 1$  by above methods, we have

This is the Matlab code for Adomian decomposition method:

```
%tic;

%syms u0 u1 u2 u3 u4 u5 u6 u7 u8 u9 l x

%G=@(u) exp(-u);

%ul=u0+u1*l+u2*exp(2*log(l))+u3*exp(3*log(l))+u4*exp(4*log(l))
%+u5*exp (5*log(l)) ...
%+u6*exp(6*log(l))+u7*exp(7*log(l))+u8*exp(8*log(l))+u9*exp(9*log(l));

%for i=1:10
% a(i) = (diff(G(ul),'l', i-1))/factorial(i-1);
%
% l=0;
%
% A=subs(a);
%
%end

%u0=1;

%u1=int(subs(A(1)), x, 0, x);
%
%u2=int(subs(A(2)), x, 0, x);
%
%u3=int(subs(A(3)), x, 0, x);
%
%u4=int(subs(A(4)), x, 0, x);
%
%u5=int(subs(A(5)), x, 0, x);
%
%u6=int(subs(A(6)), x, 0, x);
%
%u7=int(subs(A(7)), x, 0, x);
%
%u8=int(subs(A(8)), x, 0, x);
%
%u9=int(subs(A(9)), x, 0, x);
%
%u=inline(sum(u0+u1+u2+u3+u4+u5+u6+u7+u8+u9));
%
%t2=toc;
```

```
%g=inline(log(x+exp(1)));
%h=input('Please input a number : h= ');
%xe=(0 : h : 1);
%y=g(xe)'; exact solution
%z=u(xe)'; Adomian solution
%E=abs(y-z); Error
%format short e;
% ErrorAdomian=max(E)
% format short;
% TimeAdomian=t2
```

This is the Matlab code for the Milne's method:

```
%function yp=salam1(x, y)
% yp=exp(-y);
%function g=kalam1(x, y)
% g=log(x+exp(1));
%function[maxem, maxer, timem, timer]=milne(fxy, gxy, xo, xf, yo, h);
%xspan=(xo : h : xf);
%s=size(xspan);
%-----Runge-Kutta Solution-----
%tic;
%[x, z]=ode45(fxy, xspan, yo);
%toc1=toc;
%-----Milne Solution-----
%y=z(1 : 4);
%for i=4 : s(2) -1;
```

```
% y(i+1)=y(i-3)+((4*h)/3)*(2*feval(fxy, x(i-2), y(i-2)) ...
% -feval(fxy, x(i-1), y(i-1))+2*feval(fxy, x(i), y(i)); Pishgoo
% % p(i+1)=y(i-1)+(h/3)*(feval(fxy, x(i-1), y(i-1)) ...
% +4*feval(fxy, x(i), y(i))+feval(fxy, x(i+1), y(i+1))); Eslahgar
% % while y(i+1) =p(i+1);
% y(i+1)=p(i+1);
% p(i+1)=y(i-1)+(h/3)*(feval(fxy, x(i-1), y(i-1)) ...
% +4*feval(fxy, x(i), y(i))+feval(fxy, x(i+1), p(i+1)));
%end
%
% -----
% ex(i+1)=feval(gxy, x(i+1), y(i+1)); Exact solution
% em(i+1)=abs(ex(i+1)-p(i+1)); Error Milne
% er(i+1)=abs(ex(i+1)-z(i+1)); Error
%end
%
%-----
%toc2=toc; Milne Time
%format short e;
%maxem=max(em);
%maxer=max(er);
%format short;
%timem=toc2;
%timer=toc1;
```

This is the Matlab code for generalization of Milne's method:

```
%function [maxegm, maxer, timegm, timer]=gmilne(fxy, gxy, xo, xf, yo, h);
%xspan=(xo : h : xf);
```

```
%s=size(xspan);
%-----Runge-Kutta Solution-----
%tic;
%[x, z]=ode45(fxy, xspan, yo);
%toc1=toc;
%-----Generalization of Milne Solution-----
%y=z(1 : 6);
%for i=6 : s(2)-1;
%   y(i+1)=y(i-5)+((3*h)/10)*(11*feval(fxy, x(i-4), y(i-4))-14*feval(fxy,
x(i-3), y(i-3)) ...
% +26*feval(fxy, x(i-2), y(i-2))-14*feval(fxy, x(i-1), y(i-1))+11*feval(fxy,
x(i), y(i))); Pishgoo
% %p(i+1)=((60*h)/157) * (feval(fxy, x(i-4), y(i-4))+feval(fxy, x(i-3),
y(i-3)) ...
% +feval(fxy, x(i-2), y(i-2)) ...
% +feval(fxy, x(i-1), y(i-1))+feval(fxy, x(i), y(i))+feval(fxy, x(i+1), y(i+1)))
...
% +(1/157)*(157*y(i-4)-275*y(i-3)+400*y(i-2)-400*y(i-1)+275*y(i));
Eslahgar
% %while y(i+1) =p(i+1);
% y(i+1)=p(i+1);
% p(i+1)=((60*h)/157)*(feval(fxy, x(i-4), y(i-4))+feval(fxy, x(i-3), y(i-3)) ...
% +feval(fxy, x(i-2), y(i-2)) ...
% +feval(fxy, x(i-1), y(i-1))+feval(fxy, x(i), y(i))+feval(fxy, x(i+1), p(i+1)))
...
% +(1/157)*(157*y(i-4)-275*y(i-3)+400*y(i-2)-400*y(i-1)+275*y(i));
%end
%
```

```

% ex(i+1)=feval(gxy, x(i+1), y(i+1)); Exact solution
% egm(i+1)=abs(ex(i+1)-p(i+1)); Error Generalization of Milne
% er(i+1)=abs(ex(i+1)-z(i+1)); Error Runge-Kutta
%end
%-----
%toc2=toc; Generalization of Milne Time
%format short e;
%maxegm=max(egm);
%maxer=max(er);
%format short;
%timegm=toc2;
%timer=toc1;

```

**Table 1**

| Step size | Adomian decomposition   |               | Milne                    |               | Generalization of Milne  |               |
|-----------|-------------------------|---------------|--------------------------|---------------|--------------------------|---------------|
|           | <i>h</i>                | Maximum error | Time                     | Maximum error | Time                     | Maximum error |
| .13       | $1.3561 \times 10^{-6}$ | 32.1283       | $5.4169 \times 10^{-8}$  | 0.3616        | $1.4484 \times 10^{-8}$  | 0.3631        |
| .02       | $3.4040 \times 10^{-6}$ | 32.0664       | $1.3658 \times 10^{-10}$ | 0.3747        | $1.3769 \times 10^{-10}$ | 0.3834        |
| .009      | $3.3710 \times 10^{-6}$ | 31.7148       | $3.9022 \times 10^{-12}$ | 0.3868        | $2.7318 \times 10^{-12}$ | 0.4042        |
| .003      | $3.3710 \times 10^{-6}$ | 31.4826       | $3.5527 \times 10^{-14}$ | 0.4251        | $1.8208 \times 10^{-14}$ | 0.4613        |
| .0005     | $3.4040 \times 10^{-6}$ | 32.2915       | $1.7764 \times 10^{-15}$ | 0.7076        | $5.8398 \times 10^{-14}$ | 0.9950        |

In the above table, for instance in row 3, compute errors in 100 points among (0, 1), then appear maximum of those errors in column 2 and the time in the above table is per second.

**Example 2**

For finding the numerical solution of  $y' + y = \frac{x}{y}$  with initial condition  $y(0) = 1$  when  $0 < x < 1$ , we have

**Table 2**

| Step size | Adomian decomposition   |               | Milne                    |               | Generalization of Milne  |               |
|-----------|-------------------------|---------------|--------------------------|---------------|--------------------------|---------------|
|           | $h$                     | Maximum error | Time                     | Maximum error | Time                     | Maximum error |
| .13       | $9.2672 \times 10^{-4}$ | 28.8794       | $1.2156 \times 10^{-5}$  | 0.0151        | $7.0525 \times 10^{-5}$  | 0.3657        |
| .08       | $1.1732 \times 10^{-3}$ | 28.1815       | $1.9098 \times 10^{-6}$  | 0.3657        | $4.4063 \times 10^{-6}$  | 0.3708        |
| .009      | $1.1833 \times 10^{-3}$ | 28.2661       | $6.5241 \times 10^{-10}$ | 0.3921        | $5.8862 \times 10^{-10}$ | 0.4186        |
| .002      | $1.1837 \times 10^{-3}$ | 28.3661       | $7.3708 \times 10^{-13}$ | 0.4700        | $2.8744 \times 10^{-13}$ | 0.5412        |
| .0005     | $1.1837 \times 10^{-3}$ | 28.5227       | $2.4425 \times 10^{-15}$ | 0.7560        | $2.3759 \times 10^{-14}$ | 0.9770        |

In the above table, it can be seen that Adomian decomposition method is weaker in accuracy than Milne and Generalization of Milne methods and also require higher time for computing solution.

**4. Conclusion**

From the above observations, it can be concluded that Adomian decomposition method, despite its greater stability in solving equations, suffers certain weaknesses. It can also be concluded that due to its affinity to Taylor series, Adomian decomposition method has flaws of its own and it is seen that its stability is lower than other numerical methods. Also, it is important that Adomian decomposition method unlike the most numerical techniques provides a close form of the solution. Then the Adomian decomposition method is weaker in accuracy than Milne and Generalization of Milne methods for all nonlinear, (linear), differential equations.

**References**

- [1] Y. Cherrault, Convergence of Adomian's method, *Kybernetes* 8(2) (1988), 31-38.
- [2] G. H. Keitel, An extension of Milne's three-point method, *J. ACM* 3 (1956), 212-222.
- [3] J. D. Lambert, Computational Method in Ordinary Differential Equation, John Wiley and Sons, 1973.
- [4] L. F. Shampine, Numerical Solution of Ordinary Equations, Chapman and Hall, 1994.
- [5] N. Taghizadeh, Generalization of the Milne's method in the numerical solution of the differential equations, *Korean Annals of Mathematics* 22(1) (2005), 37-42.
- [6] A. M. Wazwaz, A comparison between Adomian decomposition method and Taylor series method in the series solutions, *Appl. Math. Comput.* 97 (1998), 37-44.
- [7] A. M. Wazwaz, The numerical solution of fifth-order boundary value problems by the decomposition method, *J. Comput. Appl. Math.* 136 (2001), 259-270.

■