A Modified Method to Identification of Lagrange Multipliers

P Karimi, N Eshghi, D D Ganji, M Rostamian, H M Daniali
Department of Mechanical Engineering, Mazandaran University, P.O. Box 484, Babol, Iran
E-mail: ddg_davood@yahoo.com

Abstract. Exact identification of Lagrange multipliers in the variational iteration method is very important for obtaining highly accurate solutions, on the other hand, it is complicated to determine the multipliers for strongly nonlinear equations. This paper overcomes completely the problem, and results are helpful for solving nonlinear equations.

1. Introduction
One of the problems with the VIM is the calculation of $\lambda$, Lagrange multiplier. This parameter plays the important role in the convergence of the VIM solution. For the first time, the calculation of $\lambda$ was implemented by He [1-6], then was developed future by Dehghan [7], Wazwaz [8,9], D. D. Ganji [10-13].

In all of the implemented works, the process of achieving $\lambda$ was not discussed clearly, even in some of the sources, $\lambda$ has been calculated wrongly, but, because of the high ability of the VIM, the results came to convergence with more iteration.

In this paper, we studied about modifying the traditional method to achieve $\lambda$ with the fewer steps which would be much easier and faster than the previous accomplishment. Because of needing the traditional method to leibnitz, we are not able to write a computer program to obtain $\lambda$ by means of traditional method. We came to the modified method in which we do not need to calculate with hands and in order to obtain $\lambda$, it is sufficient to locate the linear part of differential equation at the disclosed program that it is written by Maple software package. Therefore, there are advantages with new modified method. Firstly, we can avoid a long process to achieve $\lambda$ and secondly, this above mentioned method can be used to achieve $\lambda$ in some higher order nonlinear differential equations that these are impossible to be calculated with hands.

2. The basic of the new modified method
Studying about the $\lambda$ at the previous articles [1-13], we can observe that the $\lambda$ is obtained in the three shape which are polynomial, exponential and trigonometry. All of these types are achieved from the solution of a homogeneous ordinary differential equation (ODE). Therefore, having the coefficients of ODE and its boundary conditions, we are able to calculate the $\lambda$ conveniently. In this paper, we are go to propose the modified procedure about the obtainment of these coefficients and boundary conditions. Consider the linear part of the nonlinear differential equation in the following form:

1 To whom any correspondence should be addressed.
**Linear Part**: \[ a_m y^{(m)}(t) + a_{m-1} y^{(m-1)}(t) + \cdots + a_1 y'(t) + a_0 y(t) \] (1)

writing the VIM integration, we have:

\[
y_{n+1}(t) = y_n(t) + \int_0^t \lambda(s)[a_m y_n^{(m)}(s) + a_{m-1} y_n^{(m-1)}(s) + \cdots + a_1 y_n'(s) + a_0 y_n(s)]ds
\] (2)

Then, Eq. (2) is expanded:

\[
y_{n+1}(t) = y_n(t) + \int_0^t \lambda(s)a_m y_n^{(m)}(s)ds + \int_0^t \lambda(s)a_{m-1} y_n^{(m-1)}(s)ds + \cdots + \int_0^t \lambda(s)a_1 y_n'(s)ds + \int_0^t \lambda(s)a_0 y_n(s)ds
\] (3)

We can convert the terms which have derivations with higher order to the first order by means of using from integration by parts. Finally, we are able to locate the two sides of this equation be equal to zero. Therefore, calculations would be similar to the following form:(See Table 1)

| \( \lambda(s) \) | \( a_m y_n^{(m)}(s) \) |
|---|---|
| \( -\lambda'(s) \) | \( a_m y_n^{(m-1)}(s) \) |
| \( +\lambda''(s) \) | \( a_m y_n^{(m-2)}(s) \) |
| \( \vdots \) | \( a_m y_n^{(m-3)}(s) \) |
| \( \pm\lambda^{(m)}(s) \) | \( \vdots \) |
| \( \int a_m y_n(s)ds \) |

After substituting the achieved terms in the equation and implementing the \( \delta \) performance and leibnitz, the two sides of equation are equal to zero. Then we have:
\pm a_m \lambda^{(m)}(s) \mp a_{m-1}\lambda^{(m-1)}(s) \pm \cdots - a_1 \lambda'(s) + a_0 \lambda(s) = 0
\pm a_m \lambda^{(m-1)}(s) \mp a_{m-1}\lambda^{(m-2)}(s) \pm \cdots - a_2 \lambda'(s) + a_1 \lambda(s) + 1 = 0
\pm a_m \lambda^{(m-2)}(s) \mp a_{m-1}\lambda^{(m-3)}(s) \pm \cdots - a_2 \lambda'(s) + a_1 \lambda(s) = 0
\pm a_m \lambda^{(m-3)}(s) \mp a_{m-1}\lambda^{(m-4)}(s) \pm \cdots - a_3 \lambda'(s) + a_2 \lambda(s) = 0
\vdots
a_m \lambda(s)

\lambda is obtained form the above equations. Observing these equations, we can realize that with a diagonal motion, the order of derivation is not change on every line (similar to the motion shown in Eq. (4)). Also, the coefficients index is increased and the sign of coefficients is changed from line to line. Notice that, \( a_m, a_{m-1}, \ldots \) are achieved from Eq(1). There exists only one line which is needed to be corrected by adding the number 1 to it. Therefore, by means of defining such as this algorithm, we are able to get final homogeneous ODE directly from the nonlinear differential equation without needing to leibnitz and integration by parts. Finally, \( \lambda \) is achieved from the solution of this homogeneous ODE.

In order to obtain \( \lambda \), this algorithm is written by Maple software package in the following part:

> restart;
> with(PDEtools):
> linear_part:=diff(u(t),t$3); m:=difforder(linear_part):
equ[m]:=linear_part:
for h from m by -1 to 1 do
a[h]:=coeff(equ[h],diff(u(t),t$h));
equ[h-1]:=equ[h]-a[h]*diff(u(t),t$h):
end do:
a[0]:=coeff(equ[0],u(t)):

\[ \text{linear_part} = \frac{d^3}{dt^3} u(t) \]

> for i1 from 1 by 1 to m+1 do
b[i1]:=0:
end do:
k:=m+1:
n1:=1:
for r1 from 0 by 1 to m-1 do
k:=k-1:
n1:=n1*(-1):
for r from 1 by 1 to k do
b[r]:=b[r]+a[r+1]*n1*diff(lambda(s),s$(r+1))
end do:
end do:
> for r2 from 1 by 1 to m+1 do
b[r2]:=b[r2]+a[r2-1]*lambda(s):
end do:
b[2]:=b[2]+1:
> for r3 from 1 by 1 to 3 do
b[r3];
end do;
> for r4 from 2 by 1 to m+1 do
b[r4]:=eval(b[r4],s=t)=0;
end do:
> dsolve([b[1],b[2],b[3],b[4]],lambda(s));
\[ \lambda(s) = K \frac{1}{2} s^2 \cos t s K \frac{1}{2} t^2 \]
> factor(%);
\[ \lambda(s) = K \frac{1}{2} (sK t)^2 \]

Notice that, to use this program, one has to enter the linear part of equation in front of linear part sentence (on the first yellow line) and enter b[1] to b[m+1] front of dsolve in the last line. m is the order of the linear part of nonlinear differential equation. b[1] is final ODE and b[2] to b[m+1] are boundary conditions that are used to solve the ODE. For example, in the third order equation, b[1], b[2], b[3] and b[4] are located in front of dsolve on the last yellow line. Finally, \( \lambda \) is obtained by solving this ODE with its boundary conditions.

| No. | equation | Linear part of equation | \( \lambda \) in article | \( \lambda \) obtain from program | Ref |
|-----|----------|-------------------------|--------------------------|--------------------------------|-----|
| 1   | \( y'' + \omega^2 y = f(t) \) | \( y'' + \omega^2 y = f(t) \) | \( \lambda = \frac{1}{\omega} \sin \omega (\tau - t) \) | \( \lambda = -\sin(\omega(t - \tau)) \omega \) | [1] |
| 2   | \( y'' + \omega^2 \dot{y} = 0 \) | \( y'' \) | \( \lambda = \tau - t \) | \( \lambda = \tau - t \) | [1] |
| 3   | \( T'' + T + x = 0 \) | \( T'' + T + x \) | \( \lambda = \sin(s - x) \) | \( \lambda = -\sin(x - s) \) | [2] |
| 4   | \( [1 + \varepsilon u] u'' + u = 0 \) | \( u'' + u \) | \( \lambda = -e^{(\varepsilon - t)} \) | \( \lambda = -e^{(\varepsilon - t)} \) | [10] |
| 5   | \( u_i + i u_{si} = 0 \) | \( u_i \) | \( \lambda = -1 \) | \( \lambda = -1 \) | [8] |
| 6   | \( i u_i + u_{si} + 2\rho \dot{u} u = 0 \) | \( i u_i \) | \( \lambda = i \) | \( \lambda = -\frac{1}{i} \) | [8] |
| 7   | \( u'' + \frac{1}{2} uu'' = 0 \) | \( u'' \) | \( \lambda = -\frac{1}{2} (\zeta - x)^2 \) | \( \lambda = -\frac{1}{2} (x - \zeta)^2 \) | [9] |
| 8   | \( x' - \ddot{x} (a - by) = 0 \) | \( x' \) | \( \lambda_1 = -1 \) | \( \lambda_1 = -1 \) | [11] |
|     | \( y' + \ddot{y} (c - dx) = 0 \) | \( y' \) | \( \lambda_2 = -1 \) | \( \lambda_2 = -1 \) |
3. Results
In order to check the program, we compared the result of many articles with \( \lambda \) that obtain from our Maple program. Some of these comparisons have been collected into Table 2. Also the reference of each equation is indicated. This program is able to obtain \( \lambda \) from nonlinear differential equation, nonlinear partial differential equation (PDE) and nonlinear system differential equation.

At another approach, we tested the mentioned program with some higher order equations and respective \( \lambda \) to each equation were calculated readily. The results of these tests have been listed in Table 3.

| No | Linear part of equation | \( \lambda \) obtain from program |
|----|-------------------------|---------------------------------|
| 1  | \( u''(t) - 2u(t) \)      | \( \lambda = \frac{1}{4} \sqrt{2} \left( e^{(-\sqrt{2}(\xi t + r))} - e^{(\sqrt{2}(\xi t + r))} \right) \) |
| 2  | \( u''(t) + u'(t) \)      | \( \lambda = -1 + \cos(t - \xi) \) |
| 3  | \( 5u''(t) - 4u''(t) \)  | \( \lambda = \frac{1}{4}t + \frac{5}{16} - \frac{1}{4} \xi - \frac{5}{16} e^{(\frac{5}{4} - \frac{5}{2} \xi^2)} \) |
| 4  | \( u^{(4)}(t) + u''(t) \) | \( \lambda = -t + \xi + \sin(t - \xi) \) |
| 5  | \( 2u^{(4)}(t) + 3u''(t) \) | \( \lambda = -\frac{1}{6} t^2 + \frac{2}{9} t - \frac{4}{27} + \frac{1}{3} \xi t - \frac{2}{9} \xi - \frac{1}{6} \xi^2 + \frac{4}{27} e^{(-\frac{3}{2} + \frac{3}{2} \xi^2)} \) |
| 6  | \( u^{(5)}(t) + 3u''(t) \) | \( \lambda = \frac{1}{9} - \frac{1}{6} t^2 + \frac{1}{3} \xi t - \frac{1}{6} \xi^2 - \frac{1}{9} \cos(\sqrt{3}(t - \xi)) \) |
| 7  | \( u^{(5)}(t) + u^{(4)}(t) + u''(t) \) | \( \lambda = t - \frac{1}{2} t^2 - \xi + \xi t - \frac{1}{2} \xi^2 - \frac{2}{3} \sin(\sqrt{3}(t - \xi)) \sqrt{3} e^{(-\frac{1}{2} + \frac{1}{2} \xi^2)} \) |
| 8  | \( u^{(7)}(t) \)          | \( -\frac{1}{720} (t - \xi)^6 \) |

4. Conclusion
In this study, the lagrange multiplier (\( \lambda \)) was achieved by using modified method without any intricate manually calculations, like traditional method. This method is able to compute \( \lambda \) by writing program on computer. Therefore, the new modified method to obtain \( \lambda \) has advantages that listed below:
1. Computer implementation of modified method is possible by using symbolic mathematical package software like Maple software.
2. Achieving exact \( \lambda \) without any error.
3. For cases of encountering higher order nonlinear equations, calculation of \( \lambda \) is too complicated manually. By means of present modified method, lagrange multiplier are acquired readily.

References
[1] J.H. He 1999 Int. J. Nonlinear Mech. 34 699
[2] J.H. He 2000 Appl. Math. Comput. 114 115
[3] J.H. He, X.H. Wu 2006 Chaos Solitons & Fractals 29 108
[4] J.H. He Variational iteration method—some recent results and new interpretations J Comput. Appl. Math. in press (doi:10.1016/j.cam.2006.07.009).
[5] Shu-Qiang Wang, Ji-Huan He Variational iteration method for solving integro-differential equations, Physics Letters.
[6] Shou DH, He JH 2007 Int. J. Nonlinear Sci. 8 121
[7] Mehdi Tatari, Mehdi Dehghan 2007 Chaos, Solitons and Fractals 33 671
[8] A.M. Wazwaz 2006 A study on linear and nonlinear Schrodinger equations by the variational iteration method, Chaos, Solitons and Fractals. (in press).
[9] A.M. Wazwaz 2006 The variational iteration method for solving two forms of Blasius equation on a half-infinite domain. (in press).
[10] H. Tari, D.D. Ganji, H. Babazadeh 20007 Phys. Lett. A 363 213.
[11] M. Rafei, H. Daniali, D.D. Ganji, 2007 Appl. Math. Comput. 186 1701
[12] Tari H, Ganji DD, Rostamian M 2007 Int. J. Nonlinear Sci. 8 203
[13] Ganji DD, Sadighi A 2007 Int. J. Nonlinear Sci. 7 411