



REVIEW OF THE NEW PERTURBATION-ITERATION METHOD

Mehmet Pakdemirli

Department of Mechanical Engineering, Celal Bayar University 45140, Muradiye, Manisa Turkey. mpak@cbu.edu.tr

Abstract-The new perturbation iteration method developed by Pakdemirli and coworkers are reviewed. First, applications of the method to algebraic equations are discussed and some new root-finding algorithms developed by this method are given. Next, the applications of the new method to first order, second order and systems of first order ordinary differential equations are discussed. Three sample problems are selected. Results are compared with analytical solutions, solutions by other methods and numerical solutions. The new perturbation iteration method is an effective method that does not require small parameter assumption as in classical perturbation methods.

Key Words-Perturbation Method, Perturbation-Iteration Algorithm, Algebraic Equations, Ordinary Differential Equations, Integral Equations.

1. INTRODUCTION

Perturbation methods have been effectively used for over a century in construction of approximate solutions of algebraic equations, differential equations, integro-differential equations. The solutions obtained by this method opened a new era in understanding the physics of many applied problems. The method and its variants are subject to development even today by the researchers.

One major limitation of the perturbation methods is the requirement of a small parameter. The small parameter may appear as an original physical parameter of the equation or can be introduced as an artificial parameter. This small parameter assumption restricts the range of validity of the results. For nonlinear problems, the solutions are valid for weakly nonlinear cases.

In a recent effort to overcome the major limitation of small parameter assumption, Pakdemirli and co-workers developed a new perturbation-iteration algorithm. The method was originally developed for algebraic equations and then generalized to differential equations. The pioneering work on algebraic equations was due to Pakdemirli and Boyacı [1] where they obtained some well-known root finding algorithms as well as new ones using the new systematic approach. Single point iteration algorithms up to third order derivatives were presented in that study. Fourth order derivative algorithms [2] as well as fifth order derivative algorithms [3] followed that study.

With an inspiration from the work on algebraic equations, the systematic approach of combining perturbations and iterations was applied to ordinary differential equations also. Pakdemirli *et al.* [4] derived the perturbation-iteration algorithms for first order differential equations. The new method was applied to Bratu-type second order equations by Aksoy and Pakdemirli [5]. The algorithms developed were tested

with success on some nonlinear heat equations also [6]. The most recent studies include the analysis of Fredholm and Volterra integral equations [7] and ordinary differential equation systems [8] by the developed method. In the mentioned work [4-8], the perturbation-iteration solutions were contrasted with exact solutions, if available, solutions obtained by other methods such as variational iteration method and with numerical solutions. The convergence of the iterations to real solutions were discussed and shown in detail. It seems that the method can be effectively applied to many physical problems yielding satisfactory results.

2. ALGEBRAIC EQUATIONS

In this section, a brief summary of the results in [1-3] will be given. For the roots of the nonlinear equation

$$f(x) = 0 \tag{1}$$

one may assume a perturbation expansion of the below form with n correction terms

$$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \dots + \varepsilon^n x_n \tag{2}$$

Inserting (2) into (1) and expanding in a Taylor series up to m'th order derivative terms yields

$$f(x_{0} + \varepsilon x_{1} + \varepsilon^{2} x_{2} + ...\varepsilon^{n} x_{n}) \cong f(x_{0}) + f'(x_{0})(\varepsilon x_{1} + \varepsilon^{2} x_{2} + ...\varepsilon^{n} x_{n}) + \frac{f''(x_{0})}{2!}(\varepsilon x_{1} + \varepsilon^{2} x_{2} + ...\varepsilon^{n} x_{n})^{2} + ... + \frac{f^{(m)}(x_{0})}{m!}(\varepsilon x_{1} + \varepsilon^{2} x_{2} + ...\varepsilon^{n} x_{n})^{m} = 0$$
(3)

Note that since n terms in the perturbation expansion and m'th order derivatives in the Taylor series are considered, the perturbation iteration algorithm developed will be named PIA(n,m). n should always be less than or equal to m, otherwise, the unknowns (correction terms in the perturbation expansion) cannot be determined. Equation (3) should be grouped with respect to the same orders of ε , then separated and solved for the unknown correction terms. Substituting back the correction terms into (2) yields an iteration algorithm for solution of (1). Note that separations may not be unique and there may be more than one way of separation of (3). Leaving the details of the algebra to [1-3], some final results are given below

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$
(PIA(1,1)) (4)

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \mp \frac{1}{f''(x_n)} \sqrt{(f'(x_n))^2 - 2f(x_n)f''(x_n)}$$
(PIA(1,2)) (5)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(x_n)(f(x_n))^2}{2(f'(x_n))^3}$$
(PIA(2,2)) (6)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2}{6(f'(x_n))^2} \frac{3f'(x_n)f''(x_n) - f(x_n)f'''(x_n)}{(f'(x_n))^2 - f(x_n)f''(x_n)}$$
(PIA(2,3)) (7)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(f(x_n))^2 f''(x_n)}{2(f'(x_n))^3} + (f(x_n))^3 \frac{f'(x_n) f'''(x_n) - 3(f''(x_n))^2}{6(f'(x_n))^5}$$
(PIA(3,3)) (8)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} + \frac{2f(x_n)f'(x_n)f''(x_n) - 2f'(x_n)^3 - f(x_n)^2 f'''(x_n)}{2f'(x_n)} + \frac{1}{2\sqrt{3}f'(x_n)^2 f''(x_n)}$$

$$\sqrt{12 \ f'(x_n)^6 + 3 \ f(x_n)^4 f'''(x_n)^2 - 24 \ f(x_n)f'(x_n)^4 f''(x_n) + 12 \ f(x_n)^2 f'(x_n)^3 f'''(x_n) - 8 \ f(x_n)^3 f'(x_n) f''(x_n) f'''(x_n) - f(x_n)^4 f''(x_n) f''(x_n) f''(x_n) - 8 \ f(x_n)^3 f'(x_n) f''(x_n) f'''(x_n) f''(x_n) f''(x_n) f''(x_n) f'''(x_n) f''(x_n) f''(x_n) f''$$

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2} \frac{f(x_n)^2 f''(x_n)}{f'(x_n)^3} \\ &+ f(x_n)^3 \frac{4f'(x_n)^3 f'''(x_n) - 12f'(x_n)^2 f''(x_n)^2 - 3f(x_n)f''(x_n)^3 + 6f(x_n)f'(x_n)f''(x_n)f'''(x_n) - f(x_n)f'(x_n)^2 f^{iv}(x_n)}{24f'(x_n)^5 [f'(x_n)^2 - f(x_n)f''(x_n)]} \end{aligned}$$
(PIA(3,4)) (10)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)^2 f''(x_n)}{2f'(x_n)^3} + f(x_n)^3 \frac{f'(x_n) f'''(x_n) - 3f''(x_n)^2}{6f'(x_n)^5} - f(x_n)^4 \frac{15f''(x_n)^3 - 10f'(x_n) f''(x_n) f'''(x_n) + f'(x_n)^2 f^{iv}(x_n)}{24f'(x_n)^7}$$
(PIA(4,4)) (11)

$$x_{n+1} = x_n - \frac{f}{f'} - \frac{f^2 f''}{2f'^3} + \frac{f^3}{6f'^5} [f''' f' - 3 f''^2] - \frac{f^4}{24f'^7} [15 f''^3 - 10 f' f'' f''' + f'^2 f^{iv}] + \frac{f^5}{120f'^9} [-105 f''^4 + 90 f' f''^2 f''' - 15 f'^2 f'' f^{iv} + 15 f' f''^2 f^{iv} - 10 f'^2 f'''^2 + f'^3 f^{v}] (PIA(5,5))$$
(12)

141

Note that PIA(1,1) algorithm is the well-known Newton Raphson method. For n=m, the algorithms were given in the literature which were derived by some other methods [9]. To the best of the authors' knowledge, the algorithms PIA(2,3), PIA(2,4), PIA(3,4) were not reported in other studies before [1-3].PIA(1,2) can be derived by purely geometrical considerations. In geometric derivation of Newton-Raphson method, tangent lines are employed to converge to the root. Similarly, if tangent parabolas are used, PIA(1,2) algorithm may also be developed. For convergence and performance of the algorithms, as well as some comparisons with the equivalent ones derived by Adomian decomposition method, see detailed discussions in [1-3]. If m is fixed, n=m algorithms usually perform better than n<m algorithms. As the number m increases for PIA(m,m) algorithms, the convergence is usually faster. However, the gain does not compensate for the algebraic complexity after m=4. Therefore PIA(3,3) or at most PIA(4,4) may be optimal for finding roots.

3. FIRST ORDER DIFFERENTIAL EQUATIONS

For first order differential equations, results of [4] will be briefly discussed. Results of PIA(1,1) and PIA(1,2) algorithms will be presented.

3.1. PIA(1,1) Algorithm

Consider the general first order differential equation

$$F(u, \dot{u}, \varepsilon) = 0$$
 (13)

with u=u(t) and ε the perturbation parameter. Only one correction term is taken in the perturbation expansion

$$u_1 = u_0 + \varepsilon u_c + \dots \tag{14}$$

Upon substitution of (14) into (13) and expanding in a Taylor series with first derivatives only yields

$$F(u_0, \dot{u}_0, 0) + F_u(u_0, \dot{u}_0, 0) \, \varepsilon u_c + F_{\dot{u}}(u_0, \dot{u}_0, 0) \, \varepsilon \dot{u}_c + F_{\varepsilon}(u_0, \dot{u}_0, 0) \, \varepsilon = 0 \, (15)$$

where subscripts denote differentiation with respect to the variable. Note that, in our method, the function and its derivatives are considered to be independent variables. Reorganizing the equation

$$\dot{u}_{c} + \frac{F_{u}}{F_{u}}u_{c} = -\frac{F_{\varepsilon} + F/\varepsilon}{F_{u}}$$
(16)

and keeping in mind that all derivatives are evaluated at $\varepsilon = 0$, it is readily observed that the above equation is a variable coefficient first order differential equation whose solution is

$$u_{c} = c \exp\left(-\int \frac{F_{u}}{F_{u}} dt\right) - \left(\int \frac{F_{\varepsilon} + F/\varepsilon}{F_{u}} \exp\left(\int \frac{F_{u}}{F_{u}} dt\right) dt\right) \exp\left(-\int \frac{F_{u}}{F_{u}} dt\right)$$
(17)

Substitution of (17) into (14) and constructing the iteration scheme yields

$$u_{n+1} = u_n + \varepsilon c_n \exp\left(-\int \frac{F_u(u_n, \dot{u}_n, 0)}{F_{\dot{u}}(u_n, \dot{u}_n, 0)} dt\right) - \varepsilon \left(\int \frac{F_\varepsilon(u_n, \dot{u}_n, 0) + F(u_n, \dot{u}_n, 0)/\varepsilon}{F_{\dot{u}}(u_n, \dot{u}_n, 0)} \exp\left(\int \frac{F_u(u_n, \dot{u}_n, 0)}{F_{\dot{u}}(u_n, \dot{u}_n, 0)} dt\right) dt\right) \exp\left(-\int \frac{F_u(u_n, \dot{u}_n, 0)}{F_{\dot{u}}(u_n, \dot{u}_n, 0)} dt\right)^{(18)}$$

Example Problem 1

Consider the differential equation with the condition

$$\dot{u} + \varepsilon u^2 = 0$$
 $u(0) = 1$ (19)
for which the exact solution is

$$u = \frac{l}{l + \varepsilon t} \tag{20}$$

Equation (18) reduces to

$$u_{n+1} = \varepsilon(c_n - \int u_n^2 dt) \qquad n = 0, 1, 2...$$
 (21)

In applying the iteration formula, an initial guess satisfying the initial condition should be selected and at each step c_n coefficients have to be determined from the initial condition. Selecting

$$u_0 = 1$$
 (22)

and using the formula, the approximate solutions at each step are $u_1 = 1 - \varepsilon t$ (23)

$$u_{2} = 1 - \varepsilon t + \varepsilon^{2} t^{2} - \frac{1}{3} \varepsilon^{3} t^{3}$$
(24)

$$u_{3} = 1 - \varepsilon t + \varepsilon^{2} t^{2} - \varepsilon^{3} t^{3} + \frac{2}{3} \varepsilon^{4} t^{4} - \frac{1}{3} \varepsilon^{5} t^{5} + \frac{1}{9} \varepsilon^{6} t^{6} - \frac{1}{63} \varepsilon^{7} t^{7}$$
(25)

Although, iterations up to u_7 are calculated, they are not given here for brevity. In fact, using a symbolic manipulation algorithm, iterations can be calculated up to any arbitrary order. In Figure 1, exact solution and Taylor series solution with 30 terms is given for ε =2. It is also shown how the successive iterations converge to the exact solution. An excellent match is observed for 7th iteration in the range considered. Note that using regular perturbation analysis, the Taylor series solution is retrieved. This means that with 7 terms in the perturbation-iteration, much better solution can be obtained than the 30 term regular perturbation expansion.



Figure 1. Comparison of PIA(1,1) solutions and Taylor series solution (30 terms) with exact solution (ϵ =2, Example Problem 1)

3.2. PIA(1,2) Algorithm

As in the previous case, again only one correction term in the perturbation expansion is taken

$$u_1 = u_0 + \varepsilon u_c + \dots \tag{26}$$

which upon substitution into (13) and expanding in a Taylor series up to second order derivatives yields after arrangement

$$(F_{\dot{u}} + \varepsilon F_{\dot{u}\varepsilon})\dot{u}_{c} + (F_{u} + \varepsilon F_{u\varepsilon})u_{c} + \frac{1}{2}\varepsilon F_{\dot{u}\dot{u}}\dot{u}_{c}^{2} + \varepsilon F_{u\dot{u}}u_{c}\dot{u}_{c} + \frac{1}{2}\varepsilon F_{uu}u_{c}^{2} = -\frac{F}{\varepsilon} - F_{\varepsilon} - \frac{1}{2}\varepsilon F_{\varepsilon\varepsilon}$$
(27)

After solving u_c from above, the iteration scheme is constructed as below

$$u_{n+1} = u_n + \varepsilon(u_c)_n$$
 n=0,1,2... (28)

Note that, as mentioned before all functions and derivatives are evaluated at $\varepsilon=0$. Since, the equation is nonlinear in u_c , a general solution cannot be given as in the previous case.

Example Problem 1

Consider the same example as in the previous section

$$\dot{u} + \varepsilon u^2 = 0 \qquad \qquad u(0) = 1 \tag{29}$$

Then (27) takes the simplified form

$$(\dot{u}_c)_n + 2\varepsilon u_n (u_c)_n = -\frac{\dot{u}_n}{\varepsilon} - u_n^2$$
(30)

For the initial assumed function, one may take

$$u_0 = 1 \tag{31}$$

Substituting this function to (30) yields

$$(\dot{u}_c)_0 + 2\varepsilon (u_c)_0 = -1 \tag{32}$$

Solving (32), substituting into (28) and applying the initial condition yields

$$u_1 = 1 + \frac{1}{2}(e^{-2a} - 1) \tag{33}$$

This solution is substituted into (30)

$$(\dot{u}_c)_1 + 2\varepsilon \left(1 + \frac{1}{2}(e^{-2\varepsilon t} - 1)\right)(u_c)_1 = e^{-2\varepsilon t} - \left(1 + \frac{1}{2}(e^{-2\varepsilon t} - 1)\right)^2$$
(34)

Since the equation to be solved is a variable coefficient equation which is involved, the function in the parentheses of second term is approximated as 1 for simplicity. Solving (34), substituting into the iteration expansion and applying the boundary condition yields

$$u_{2} = 1 + \frac{1}{2}(e^{-2a} - 1) + \frac{1}{2}\epsilon t e^{-2a} + \frac{1}{8}(e^{-4a} - 1)$$
(35)

A further iteration yields

$$u_{3} = e^{-2\varepsilon t} \left(\frac{161}{384} + \frac{5}{8} \varepsilon t + \frac{5}{16} \varepsilon^{2} t^{2} \right) + e^{-4\varepsilon t} \left(\frac{15}{64} + \frac{3}{8} \varepsilon t + \frac{1}{8} \varepsilon^{2} t^{2} \right) + e^{-6\varepsilon t} \left(\frac{5}{128} + \frac{1}{32} \varepsilon t \right)$$

$$+ \frac{1}{384} e^{-8\varepsilon t} + \frac{39}{128}$$
(36)

Using a symbolic manipulation program, up to 7th order iterations are calculated. Since the solutions are very complicated, they are not presented here. Again the seventh iteration compares well with the exact solution whereas the 30 term Taylor expansion which would be obtained by a regular perturbation expansion of 30 correction terms explode and do not represent the real solution after t ≥ 0.2 (See Figure 2).



Figure 2. Comparison of PIA(1,2) solutions and Taylor series solutions (30 terms) with exact solution (ϵ =4, Example Problem 1)

For more examples and detailed discussions on first order equations, see [4, 6].

145

4. SECOND ORDER DIFFERENTIAL EQUATIONS

In this section, a brief summary and some results of [6] are given. A general second order differential equation can be written as follows $F(u'',u',u,\varepsilon) = 0$ (37) with u=u(y) and ε is the perturbation parameter. Development of PIA(1,2) algorithm

will be displayed only. Taking one correction term in the perturbation expansion
$$(38)$$

$$u_{n+1} = u_n + \varepsilon(u_c)_n$$
 (30)
substituting into (37) expanding in a Taylor series up to second order derivatives from

substituting into (37), expanding in a Taylor series up to second order derivatives, from which after reorganization, the determining equation for the correction is obtained

$$(\varepsilon F_{u''} + \varepsilon^{2} F_{u'c})(u_{c}'')_{n} + (\varepsilon F_{u'} + \varepsilon^{2} F_{u'c})(u_{c}')_{n} + (\varepsilon F_{u} + \varepsilon^{2} F_{u\varepsilon})(u_{c})_{n} + \frac{1}{2} F_{u''u'} \varepsilon^{2} (u_{c}'')_{n}^{2} + \frac{1}{2} F_{u'u'} \varepsilon^{2} (u_{c}')_{n}^{2} + \frac{1}{2} F_{uu} \varepsilon^{2} (u_{c})_{n}^{2} + F_{u''u} \varepsilon^{2} (u_{c}'')_{n} (u_{c})_{n} + F_{u''u'} \varepsilon^{2} (u_{c}'')_{n} (u_{c})_{n} + F_{u'u} \varepsilon^{2} (u_{c}')_{n} (u_{c})_{n} + F_{u'u} \varepsilon^{2} (u_{c}')_{n} (u_{c})_{n} + F_{u'u} \varepsilon^{2} (u_{c}')_{n} (u_{c})_{n}$$
(39)
$$= -F - F_{\varepsilon} \varepsilon - \frac{1}{2} F_{\varepsilon\varepsilon} \varepsilon^{2}$$

Note that, as mentioned before all functions and derivatives are evaluated at $\varepsilon = 0$. It is readily observed that the above equation is a variable coefficient nonlinear second order differential equation. With the aid of (39) and (38), an iterative scheme is constructed for the specific equation under consideration.

Example Problem 2

Consider the dimensionless nonlinear boundary value problem describing temperature distribution in a uniformly thick rectangular fin with radiation to free space [10]

$$\frac{d^2 u}{dy^2} - \varepsilon u^4 = 0 \qquad u'(0) = 0, \quad u(1) = 1$$
(40)

where ε is our dimensionless radiation parameter. Second iteration analytical solutions will be given by PIA(1,2) and variational iteration methods.

4.1. PIA(1,2) Algorithm

Equation (39) for the above equation takes the following simple form

$$(u_c'')_n - 4\varepsilon u_n^3 (u_c)_n = -\frac{u_n''}{\varepsilon} + u_n^4$$
(41)

Initial function satisfying the boundary conditions exactly and the equation approximately is

(42)

$$u_0 = 1$$

The first iteration solution is then obtained by first substituting u_0 into (41) and then the result for $(u_c)_0$ into equation (38)

$$u_{I} = \frac{3}{4} + \frac{Cosh[2\sqrt{\varepsilon}y]}{4Cosh[2\sqrt{\varepsilon}]}$$
(43)

Since the iteration equation for the second step is a variable coefficient equation, for simplicity the function u_1^3 in the second term of equation (41) is approximated as u_0^3 . Using equation (43), (41) and (38) and applying the boundary conditions yields

$$u_{2} = \frac{3}{4} + \frac{Cosh[2\sqrt{\varepsilon}y]}{4Cosh[2\sqrt{\varepsilon}]} + \frac{Sech^{5}[2\sqrt{\varepsilon}]}{491520} (-34200Cosh[2\sqrt{\varepsilon}] - 15390Cosh[6\sqrt{\varepsilon}] \\ -2430Cosh[10\sqrt{\varepsilon}] + 2Cosh[(2-8y)\sqrt{\varepsilon}] + 45Cosh[(4-6y)\sqrt{\varepsilon}] + 1660Cosh[(2-4y)\sqrt{\varepsilon}] \\ +540Cosh[(6-4y)\sqrt{\varepsilon}] + 1843Cosh[(-8+2y)\sqrt{\varepsilon}] + 11795Cosh[(-4+2y)\sqrt{\varepsilon}] \\ +20160Cosh[2y\sqrt{\varepsilon}] + 90Cosh[6y\sqrt{\varepsilon}] + 11795Cosh[(4+2y)\sqrt{\varepsilon}] + 1843Cosh[(8+2y)\sqrt{\varepsilon}] \\ +1660Cosh[(2+4y)\sqrt{\varepsilon}] + 540Cosh[(6+4y)\sqrt{\varepsilon}] + 45Cosh[(4+6y)\sqrt{\varepsilon}] + 2Cosh[(2+8y)\sqrt{\varepsilon}] \\ -4440\sqrt{\varepsilon}Sinh[(-8+2y)\sqrt{\varepsilon}] - 4440y\sqrt{\varepsilon}Sinh[(-8+2y)\sqrt{\varepsilon}] - 7800\sqrt{\varepsilon}Sinh[(-4+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(-4+2y)\sqrt{\varepsilon}] - 24480y\sqrt{\varepsilon}Sinh[2y\sqrt{\varepsilon}] + 7800\sqrt{\varepsilon}Sinh[(4+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(4+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] - 4440y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(4+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] - 4440y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(4+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] - 4440y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] + 4440\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -16680y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -1600y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{\varepsilon}] \\ -1600y\sqrt{\varepsilon}Sinh[(8+2y)\sqrt{$$

Note that after a few iterations, algebraically involved equations appear. Symbolic manipulation programs are necessary for successive calculations.

4.2. Variational Iteration Method

Tari [11] have already treated the same problem with variational iteration method. The first iteration they found was

$$u_1 = \frac{2 + \varepsilon y^2}{2 + \varepsilon} \tag{45}$$

The second iteration was

$$u_{2} = \frac{10080 + 5040(3 + y^{2})\varepsilon + 1680(4.5 + y^{4})\varepsilon^{2} + 504(2.5 + y^{6})\varepsilon^{3} + 90y^{8}\varepsilon^{4} + 7y^{10}\varepsilon^{5}}{10080 + 20160\varepsilon + 9240\varepsilon^{2} + 1764\varepsilon^{3} + 90\varepsilon^{4} + 7\varepsilon^{5}}$$
(46)

PIA(1,2) and VIM solutions are compared with the numerical solutions. As can be seen from Figure 3, PIA(1,2) solution much better agrees with the numerical solution.

M. Pakdemirli



Figure 3. Comparison of two iteration solutions of PIA(1,2) and VIM with numerical solution. ($\epsilon = 0.5$,Example Problem 2)

For more examples and detailed discussions, see [5, 6].

5. SYSTEMS OF FIRST ORDER DIFFERENTIAL EQUATIONS

In this section, a brief summary of [8] will be given. The perturbation-iteration algorithm PIA(1,m) is constructed first by taking one correction term in the perturbation expansion and correction terms of m'th order derivatives in the Taylor Series expansion.

Consider the following system of first order differential equations $F_k(\dot{u}_k, u_j, \varepsilon, t) = 0 \ k = 1, 2, ..., K$ j = 1, 2, ..., K (47) where K represents the number of differential equations in the system and the number of dependent variables. K=1 for a single equation. In the open form, the system of equations is

$$F_{1} = F_{1}(\dot{u}_{1}, u_{1}, u_{2}, \dots u_{K}, \varepsilon, t) = 0$$

$$F_{2} = F_{2}(\dot{u}_{2}, u_{1}, u_{2}, \dots u_{K}, \varepsilon, t) = 0$$
(48)

 $F_K = F_K(\dot{u}_K, u_1, u_2, ..., u_K, \varepsilon, t) = 0$ Assume an approximate solution of the system

$$u_{k,n+1} = u_{k,n} + \varepsilon \, u_{k,n}^c \tag{49}$$

with one correction term in the perturbation expansion. The subscript n represents the n'th iteration over this approximate solution. The system can be approximated with a Taylor series expansion in the neighborhood of $\varepsilon=0$

$$F_{k} = \sum_{m=0}^{M} \frac{1}{m!} \left[\left(\frac{d}{d\varepsilon} \right)^{m} F_{k} \right]_{\varepsilon=0} \varepsilon^{m} \quad k=1,2,\dots,K$$
(50)

where

$$\frac{d}{d\varepsilon} = \frac{\partial \dot{u}_{k,n+1}}{\partial \varepsilon} \frac{\partial}{\partial \dot{u}_{k,n+1}} + \sum_{j=1}^{K} \left(\frac{\partial u_{j,n+1}}{\partial \varepsilon} \frac{\partial}{\partial u_{j,n+1}} \right) + \frac{\partial}{\partial \varepsilon}$$
(51)

is defined for the n+1'th iterative equation

$$F_k(\dot{u}_{k,n+1}, u_{j,n+1}, \varepsilon, t) = 0$$
(52)

Substituting (51) into (50), one obtains an iteration equation

$$F_{k} = \sum_{m=0}^{M} \frac{1}{m!} \left[\left(\dot{u}_{k,n}^{c} \frac{\partial}{\partial \dot{u}_{k,n+1}} + \sum_{j=1}^{K} u_{j,n}^{c} \frac{\partial}{\partial u_{j,n+1}} + \frac{\partial}{\partial \varepsilon} \right)^{m} F_{k} \right]_{\varepsilon=0} \varepsilon^{m} = 0 \qquad k=1,2... \text{ K}$$
(53)

which is a first order differential equation and can be solved for the correction terms $u_{k,n}^c$. Then using (49), the n+1'th iteration solution can be found. Iterations are terminated after a successful approximation is obtained.

Note that for a more general algorithm, n correction terms instead of one can be taken in expansion (49) which would then be a PIA(n,m) algorithm. The algorithm can also be generalized to a differential equation system having arbitrary order of derivatives.

Applications of the theory developed will be outlined in this section.

Example Problem 3

Two coupled equation system will now be considered. Solutions will be obtained by PIA(1,1) algorithm. The coupled system is

$$\dot{u}_1 = -1002u_1 + 1000u_2^2 \tag{54}$$

$$\dot{u}_2 = u_1 - u_2 - u_2^2 \tag{31}$$

with the initial conditions

$$u_1(0) = 1, \quad u_2(0) = 1$$
 (55)

for which exact solutions are available

$$u_1 = e^{-2t}, \qquad u_2 = e^{-t}$$
 (56)

An artificial perturbation parameter is inserted as below

$$F_1 = \dot{u}_{1,n+1} + 1002u_{1,n+1} - \varepsilon 1000u_{2,n+1}^2 = 0$$
(57)

$$F_2 = \dot{u}_{2,n+1} + u_{2,n+1} - u_{1,n+1} + \varepsilon u_{2,n+1}^2 = 0$$

For (57), equation (53) reduces to

$$\varepsilon \, \dot{u}_{1,n}^c + 1002 \, \varepsilon \, u_{1,n}^c = -\dot{u}_{1,n} - 1002 u_{1,n} + 1000 u_{2,n}^2 \tag{58}$$

$$\varepsilon \dot{u}_{2,n}^c + \varepsilon u_{2,n}^c = -\dot{u}_{2,n} - u_{2,n}^2 - u_{2,n}^2 + u_{1,n+1}$$
(59)

If the initial trial functions are taken as

 $u_{1,0} = 1$ (60)

$$u_{2,0} = 1$$
 (61)

thesuccessive iterations are

$$u_{1,1} = \frac{500}{501} + \frac{e^{-1002t}}{501} \tag{62}$$

149

$$\begin{split} u_{2,1} &= -\frac{1}{501} - \frac{e^{-1002t}}{501501} + \frac{1003 \, e^{-t}}{1001} \end{split} \tag{63} \\ u_{1,2} &= \frac{500}{125751501} - \frac{500 \, e^{-2004t}}{126003129753501} + \frac{2006000 \, e^{-1003t}}{502002501} + \frac{1006009 \, e^{-2t}}{1002001} \\ &- \frac{2006000 \, e^{-t}}{502002501} + e^{-1002t} \bigg(-\frac{1007012008}{251503253001} + \frac{2000 \, t}{251252001} \bigg) \\ u_{2,2} &= -\frac{1}{125751501} + \frac{e^{-2004t}}{252132136759503} - \frac{1003 \, e^{-1003t}}{251252001} \\ &+ e^{-1002t} \bigg(\frac{335670670}{83918252084667} - \frac{2000 \, t}{251503253001} \bigg) \end{aligned} \tag{65}$$

These results are identical with the results of thevariational iteration method given in [12]. For more worked examples, see [8].

6. CONCLUDING REMARKS

Based on the reviewed work [1-8], the following conclusions can be stated

- 1) The systematic development of PIA(n,m) algorithms (n \leq m) yield infinitely many root finding algorithms for algebraic equations, some of which do not exist in the literature.
- 2) n=m yields the best algorithms and there is no gain in taking n greater than 3 or at most 4.
- 3) For differential equations, usually PIA(1,1) and PIA(1,2) algorithms are employed because the iteration equations to be solved introduces much complexity for higher order terms.
- 4) PIA(1,1) may yield identical results with the variational iteration method and the solutions are usually in the polynomial form.
- 5) PIA(1,2) produces better results compared to PIA(1,1) and the convergence to the real solution is faster. Successive iteration solutions are in functional form.
- 6) Perturbation iteration algorithms do not require small parameter assumption as in classical perturbation approach and therefore extends the range of validity of results.

7. REFERENCES

1. M. Pakdemirli and H. Boyacı, Generation of root finding algorithms via perturbation theory and some formulas, *Applied Mathematics and Computation* **184**, 783-788, 2007. 2. M. Pakdemirli, H. Boyacı and H. A. Yurtsever, Perturbative derivation and comparisons of root-finding algorithms with fourth order derivatives, *Mathematical and Computational Applications* **12**, 117-124, 2007. 3. M. Pakdemirli, H. Boyacı and H. A. Yurtsever, A root finding algorithm with fifth order derivatives, *Mathematical and Computational Applications* **13**, 123-128, 2008.

4. M. Pakdemirli, Y. Aksoy and H. Boyacı, A New Perturbation-Iteration Approach for First Order Differential Equations, *Mathematical and Computational Applications* **16**, 890-899, 2011.

5. Y. Aksoy and M. Pakdemirli, New Perturbation-Iteration Solutions for Bratu-type Equations, *Computers & Mathematics with Application* **59**, 2802-2808, 2010.

6. Y. Aksoy, M. Pakdemirli, S. Abbasbandy and H. Boyacı, New Perturbation-Iteration Solutions for Nonlinear Heat Transfer Equations, *International Journal of Numerical Methods for Heat & Fluid Flow* **22**, 814-828, 2012.

7. I. T. Dolapçı, M. Şenol and M. Pakdemirli, New perturbation iteration solutions for Fredholm and Volterra integral equations, *Journal of Applied Mathematics*, 2013, *Article ID:* 682537, 5 pages, http://dx.doi.org/10.1155/2013/682537.

8. M. Şenol, İ. T. Dolapçı, Y. Aksoy and M. Pakdemirli, Perturbation-Iteration method for first order differential equations and systems, *Abstract and Applied Analysis*, 2013, Article ID: 704137, 6pages, http://dx.doi.org/10.1155/2013/704137.

9. A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.

10. M. T. Darvishi, F. Khani and A. A. Soliman, The numerical simulation for stiff systems of ordinary differential equations, *Computers & Mathematics with Applications* **54**, 1055-1063, 2007.

11. H. Tari, Modified variational iteration method, *Physics Letters A* 369, 290-293, 2007.