CHAPTER IV

ORDER-BY-ORDER EXTRACTION TECHNIQUE

Considering Adomian polynomials A_n, we compute them using Order-by-order extraction technique. Programmed by manipulation languages, this method is simpler to generate Adomian polynomials.

From the idea, we are using manipulation languages such as Mathematica or other languages to expand function in complicated series.Therefore, we extract them order by order, calling λ as the extraction parameter, defined as the extraction tool.Consequently, we shall have Adomian polynomials A_n for the nonlinear term. By this method, if one order got wrong, it would affect the integrity of all higher orders.Hence this technique enables us to justify the correctness of order-by-order extraction technique against others.

Order-By-Order Extraction Technique (OBOET) is a new algorithm for generating Adomian polynomials. According to parameterization method, we decompose y , the solution of nonlinear ODE, as follows:

$$y(\lambda)=\sum_{n=0}^{\infty} y_n \lambda^n, \tag{4.1}$$

Where λ plays role as the expansion parameter. Accordingly, F is now a functional of λ *i.e.*

$$\sum_{n=0}^{\infty} A_n \lambda^n \equiv F\left[y(\lambda)\right]=F\left(\sum_{n=0}^{\infty} y_n \lambda^n\right). \tag{4.2}$$

Obviously, the RHS of equation (4.2) is in the form of inner series as the argument of outer series-since F can in principle be expanded in power series of y, and y itself can be viewed as a power series of λ . If we can successfully expand and manipulate it in single summation form, by equating it with the RHS of equation (4.2) we should obtain the final expression of the Adomian polynomial A_n. To achieve this laborious task, we use manipulation languages for expanding series after series as described in the flowchart (see Figure 4).

Since performing complicated series expansion up to too high orders is formidable in practice, we begin with expanding F very briefly just to the order of λ^0. By this, we get A_0.

After that, we expand F up to the order of λ^1. Then subtract it with A_0, then divide the result by λ.

$$\left(\sum_{n=0}^{\infty} A_n\lambda^n - A_0\right)\Big/\lambda = A_1 + A_2\lambda^1 + A_3\lambda^2 + \cdots,$$

After this, letting $\lambda \to 0$ we obtain A_1.

Next, finding A_2 is similar. We expand F series up to the order of λ^1 or a bit higher order. Subtract it by the already known A_0 and $A_1\lambda$ and divide the result by λ^2, *i.e.*

$$\left(\sum_{n=0}^{\infty} A_n\lambda^n - A_0 - A_1\lambda\right)\Big/\lambda^2 = A_2 + A_3\lambda^1 + A_4\lambda^2 + \cdots,$$

this time, by letting $\lambda \to 0$ we obtain A_2.

This process can in principle be successively performed to any higher order at which $m \geq 1$; $m \in I^+$,

$$\left(\sum_{n=0}^{\infty} A_n\lambda^n - \sum_{n=0}^{m-1} A_n\lambda^n\right)\Big/\lambda^m = A_m + A_{m+1}\lambda^1 + A_{m+2}\lambda^2 + \cdots, \qquad (4.3)$$

As usual, letting $\lambda \to 0$, we obtain A_m .

Somehow it has to be emphasized that expanding F in λ^k up to a certain high order just once may not conveniently yield all A_ks at one time, since the large series may not be simply grouped due to its extreme complication. That is why they have to be extracted order by order.

We are going to display the algorithm of how to evaluate Adomian polynomials in case of $F(y) = y^{3/2}$ as follows:

Keeping equation (4.1) in mind, we write $F(y)$ in the form

$$F(y)=y^{3/2};y=\sum_{k=0}^{p}y_k\lambda^k$$

$$F(y)=\left(y_0+y_1\lambda+y_2\lambda^2+\cdots\right)^{3/2}$$

$$F(y)=\left(y_0+\sum_{k=1}^{p}y_k\lambda^k\right)^{3/2}$$

$$F(y)=y_0^{3/2}\left(1+\frac{1}{y_0}\sum_{k=1}^{p}y_k\lambda^k\right)^{3/2}$$

$$F(y)=y_0^{3/2}(1+w)^{3/2};w\equiv\frac{1}{y_0}\sum_{k=1}^{p}y_k\lambda^k$$

$$F(y)=y_0^{3/2}\left(1+\frac{3w}{2}+\frac{3w^2}{8}-\frac{w^3}{16}+\cdots\right)$$

where we have estimated the summation upper index as p instead of infinity without loss of accuracy. For low orders, low values of$p$may be ascribed to obtain low-order Adomian polynomials. Of course, manipulation languages does the job of series-after-series expansion. For the other forms of$F(y)$, the algorithms of Adomian polynomials evaluation are still similar.

To be illustrative enough, we show the algorithms in flow chart represented as Figure 4 below.

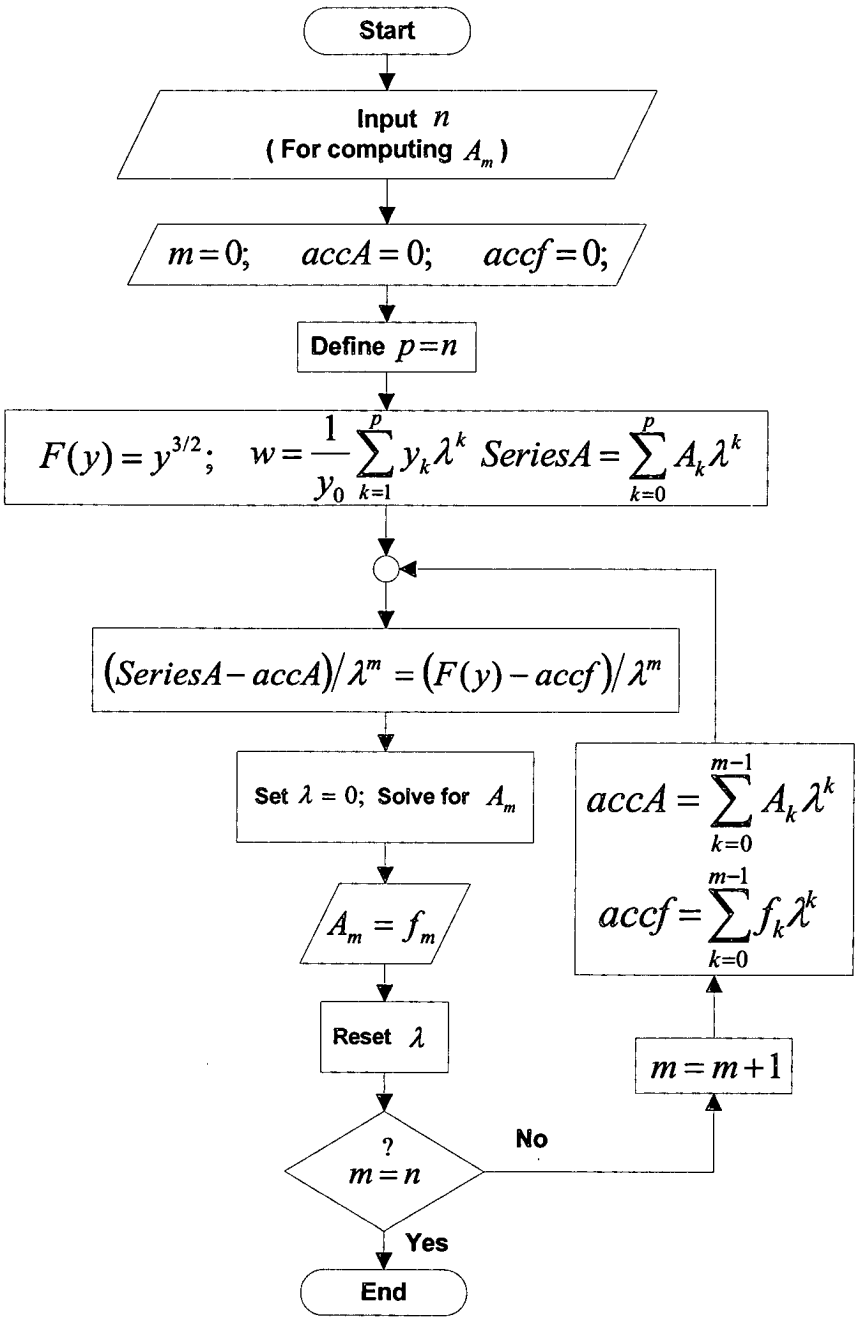

Figure 4 Flowchart for computing A_n in case $F(y)=y^{3/2}$,via OBOET