

# Derivation of Adams Method for the Numerical Solution of Ordinary Differential Equations Via Computer Algebra

Mustafa A., M. M. Hamza

Department of Mathematics, Usmanu Danfodiyo University, Sokoto, Nigeria

## Email address:

mustafa.aminu@udsok.edu.ng (Mustafa A.)

## To cite this article:

Mustafa A., M. M. Hamza. Derivation of Adams Method for the Numerical Solution of Ordinary Differential Equations Via Computer Algebra. *Applied and Computational Mathematics*. Vol. 6, No. 4, 2017, pp. 167-170. doi: 10.11648/j.acm.20170604.11

**Received:** September 26, 2017; **Accepted:** March 3, 2017; **Published:** July 4, 2017

---

**Abstract:** In this paper, a well-known computer algebra system (CAS) was considered for the derivation of Numerical method for the solution of initial value problems. This was achieved by the use of maple software. Numerical methods were derived through Lagrange interpolation method. Both the implicit and explicit method was derived with the help of the Computer algebra system. In particular, a review of Maple's functional role in the derivation of numerical methods was also presented. The main challenge was that the efficient handling and simplifying of very long expressions, which was met by the power of Maple's build-in functionality. The use of the maple procedure had significantly reduced the errors and hence improved efficiency in derivation of higher order Adams Methods.

**Keywords:** Computer Algebra, Lagrange, Initial Value, Linear Multistep Method

---

## 1. Introduction

The most popular multistep formulas in use today are based on numerical integration, by means of suitably chosen polynomial interpolation formulas. Adams methods are well-known methods that give the approximate solution to the initial value problem (IVPs) for Ordinary Differential Equations (ODEs) of the form

$$\begin{aligned} y'(x) &= f(x, y(x)), \quad \text{for } x \in [x_0, x_e], \\ y(x_0) &= y_0. \end{aligned} \quad (1)$$

This yields

$$y(x_{n+1}) = y(x_{n-r}) + \int_{x_{n-r}}^{x_{n+1}} f(x, y(x)) dx \quad (2)$$

Which approximate  $y'(x) = y(x_{n-r}) + f(x, y(x))$  in a polynomials interpolant and then integrate this polynomial to approximate the integral. The most satisfactory formulas have been found by doing the case

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx \quad (3)$$

Through integration method high order method derivation require, a lot of mathematical manipulation, such as solving of linear system of equation, as well as the long simplification which prone to a lot of errors if to be derived manually with paper and pencil. Therefore, in this paper, a Maple procedure is written to overcome some of the difficulty mentioned earlier. Many researchers have proposed some method of deriving high order methods, such as method of undetermined coefficients and numerical integration.

The Maple system has originally developed in 1980, by symbolic computation group (SCG) in University of Waterloo, Canada. [1] is a general purpose system, designed to handle a wide variety of problems Maple has known capability amongs are the symbolic computation, polynomial operations, symbolic differentiation, integration and solving algebraic equations, exact solution for ODEs/PDEs. Several researchers have uses Maple code to derived or manipulate some symbolic expression, [2], uses Maple to investigated on how computer algebra system can be employ for research and development study in the biology, also [3], uses somes Maple capabilities to derived a kinematic relations for robot. However, another

contribution was that of [4], but the focus there is on the dynamics and how dynamic model of robots. A brief report on the use of the symbolic manipulation language Maple, [4], for construction of stochastic numerical schemes was also illustrated. Also several researchers were involved in employing the symbolic algebra system, where [5] investigated the symbolic computation for use in stochastic numerics has been developed. A lot of books were published on how to use Maple such as [6]

Moreover, using computer algebra, rather complex numerical formulas can be derived far more complex than what can be done by hand. For example, all useful Lagrange interpolation formula can be computed without difficulty, in contrast to hand derivatives, which usually end with only few points. This paper is organized into five sections, section 2, introduced General form of Linear Multistep method, section 3, outline the derivation of the Implicit method, while in section 4, the Explicit method is also considered. In section 5, application and concluding remark will be presented.

The Algorithm in Appendix 1 and 2, sample statements serve to illustrate some of Maple's syntax.

## 2. The General form of Linear Multistep Method

The general form of a customary form of a linear multistep method is given by [7]

$$y_{n+1} = y_n + h \sum_{k=1}^{s-1} \phi_{-k} y'_{n-k} \quad (4)$$

Now the polynomial of degree  $n-1$  that passes through  $n$  given data points

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (5)$$

Then the Lagrange interpolation formula for the points in (3) is given as

$$\eta(x) = \sum_{k=1}^n y_k \left( \prod_{\substack{j=1 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j} \right) \quad (6)$$

The  $k$ th term of the sum in (4) is the product of  $(n-1)$  factors  $(x-x_k)/(x_i-x_k)$  namely of all those factors except for the one in which  $j=k$ . In our case where the points  $x_1, x_2, \dots, x_n$  are chosen to be equally space, and also suppose that

$$x_j = jh \quad \text{for } j = 1, 2, \dots, n, \quad (7)$$

after substituting (5) into (4) we now have

$$\eta(x) = \sum_{k=1}^n y_k \left( \prod_{\substack{j=1 \\ j \neq k}}^n \frac{x - jh}{(k-j)h} \right) \quad (8)$$

finally  $x$  is replace by  $xh$  and substitute in (8) to obtain

$$\eta(xh) = \sum_{k=1}^n \frac{(-1)^{n-k}}{(k-1)!(n-k)!} y_k \left( \prod_{\substack{j=1 \\ j \neq k}}^n (x-j) \right) \quad (9)$$

Now the result was used to obtain a collection of excellent method for differential equations, but it is well known that the so called Adams formula is given by

$$y((s+1)h) = y(ph) + h \int_s^{s+1} y'(\theta) d\theta \quad (10)$$

Instead of integrating the exact function  $y'$  as with Picard's iteration formula, then  $y'$  was replace by the Lagrange interpolating polynomial that agrees with it at the  $s+1$  points  $h, 2h, \dots, (s+1)h$ .

## 3. The Implicit Method: (Adams-Moulton)

AM methods are implicit methods; in other words, they use information at  $x_{n+1}$  to compute  $y^{n+1}$ . The derivation of the AM methods proceeds almost identically to that of AB methods, so we present them with little comment. Note that the upper boundary of the sums and products in these equations is  $i+1$ , in contrast to the  $i$  limit of the AB method formulae, but we adopt the algorithm by [3]

$$y_{s+1} = y_s + \sum_{k=1}^{s+1} \frac{(-1)^{s-k+1}}{(k-1)!(s-k+1)!} y'(kh) j \int_s^{s+1} \left( \prod_{\substack{j=1 \\ j \neq k}}^{s+1} (x-j) \right) dx \quad (11)$$

where  $k$  is replaced by  $s-k$  in (9), so numbers  $\phi_{-k}$  from (1) is given as

$$\phi_{-k} = \sum_{k=1}^{s+1} \frac{(-1)^{k+1}}{(s-k-1)!(k+1)!} \int_s^{s+1} \left( \prod_{\substack{j=1 \\ j \neq k}}^{s+1} (x-j) \right) dx \quad k = -1, 0, \dots, s-1 \quad (12)$$

Now to obtain the implicit Adams method for any step  $s$  in (12) all we have to do is to specify  $s$ . to obtain the formula. but for a higher scheme this method seem to be tedious specially in integrating the expression in (12) simplifying, instead we now used some built-in Maple functionality call *int*, which can be use to ease the construction of the schemes.

Also by studying the expression or the formula (12), it involve integration, interpolation, substitution and so on, here what we did here is to break the formula into three as follows, first we set

$$G_k := \left( \frac{1}{(x + (k - (s+1)))} \right) \prod_{j=1}^{s+1} (x-j) \quad k = 0, 1, \dots, s \quad (13)$$

By setting the value of  $s$  which will determine the type of the method to be obtained, we will have all the interpolant points needed for the method, then the second part is given as

$$\tau_k := \text{int}(G_k, x = s..s+1) \quad (14)$$

[8] Now, applying Maple command in equation (14) will return the integrated interpolant, then the last part of the formula will be

$$\phi_{-k+1} := \frac{(-1)^k}{(s-k)!(k)!} \tau_k \quad (15)$$

Using (15) in Maple command will return all the coefficient of the given value of s i.e the step of the method. Then substituting (14) into (15) and factoring will produce the Implicit Adams.

Now the Maple program which construct the discrete Implicit Adams methods, integrating equation (15) through equation (17) we obtain a Maple program, which return the discrete method by just repeating change of s. Here the Maple Program for Implicit Adams Method:

*Algorithm 1 for IMPLICIT ADAMSMOULTON METHOD*

$$\begin{aligned} G[i] &:= \left( \frac{1}{(x + (i - (s+1)))} \right) \prod_{j=1}^{s+1} (x - j); \\ T[i] &:= \text{int}(G[i], x = s..s+1); \\ > b[-i+1] &:= \frac{(-1)^i}{(s-i)!} T[i] \text{ od}; \\ y[n+1] &:= y[n] + \text{factor} \left( h \sum_{j=-1}^{s-1} b[-j] f[n-j] \right); \end{aligned}$$

## 4. The Explicit Method (Adams-Bashforth)

The Adams-Bashforth methods are explicit multistep methods. This means that they use information from the current and previous time-steps to compute the solution at  $x_{n+1}$ . These are typically abbreviated by ABs, where s is the order of the method. For example, AB2 is the second-order Adams-Bashforth method. The Adams-Bashforth methods are also based on a Lagrange polynomial approximation to the derivative. Our method requires a new Lagrange polynomial at each integration step s, which we shall call AB i. we have the explicit Methods, otherwise non as matching predictor formulas of (1). we now return to (9), and for an explicit method of order s+1 we replace  $y^1$  by the interpolating polynomial that agrees with it at the data points 0, h, 2h, 3h.....ph. as in [9]. In place (9) we now have.

$$y_{s+1} = y_s + \sum_{k=1}^{s+1} \frac{(-1)^{s-k+1}}{(k-1)!(s-k+1)!} y^1(kh) j \int_s^{s+1} \left( \prod_{\substack{j=1 \\ j \neq k}}^{s+1} (x-j) \right) dx \quad (16)$$

And (1) becomes

$$y_{n+1} = y_n + h \sum_{k=0}^s \phi_{-k} y_{n-k}^1 \quad (17)$$

Where the numbers in (10) becomes

$$\phi_{-k} = \sum_{k=1}^{s+1} \frac{(-1)^k}{(s-k)!(k)!} \int_s^{s+1} \left( \prod_{\substack{j=0 \\ j \neq k}}^s (x-j) \right) dx \quad k=0,1,\dots,s \quad (18)$$

Now the Maple program which construct the discrete explicit Adams Methods, integrating equation (18) through equation (17) we obtain a simple Maple program, which return the discrete method by just repeating (s), Here the Algorithm for Explicit Adams Method is given:

*Algorithm 2 EXPLICIT ADAMSBASHFORTH METHOD*

$$\begin{aligned} G[k] &:= \left( \frac{1}{(x + (k - s))} \right) \prod_{j=0}^s (x - j); \\ \sigma[i] &:= \text{int}(G[k], x = s..s+1); \\ \phi[-k] &:= \frac{(-1)^k}{(s-i)!(k)!} \sigma[k] \text{ od}; \\ y[n+1] &:= y[n] + \text{factor} \left( h \sum_{j=0}^s \phi[-j] f[n-j] \right); \end{aligned}$$

## 5. Application

Algorithm 1 and 2 have shown how the computer algebra was used to construct both implicit and explicit schemes of an Adams formula. For approximating the solution to initial value problem for given any order, for instance Algorithm 1, shown how an Euler's method was constructed with help of the computer algebra systems called Maple, so also for any higher order. In this Maple worksheet, we also show how maple build-in fuctionallity is used to integrate, loop, factoring and simplifying, some steps involve in the constructing of Adams linear Multistep formula. The use of the maple procedure had significantly reduced the errors and hence improved efficiency in derivation of higher order Adams Methods.

## 6. Conclusion

Adams methods has been introduced in this paper, we also obtained an implicit and explicit Adams Method through the use of symbolic algebra system called Maple. The interesting part of paper is the symbolic construction of Adams Methods which was used to generate a set of all explicit and implicit Linear Multistep Adams methods, otherwise known as predictor and corrector method for approximate solution of initial value problem for ordinary differential equations.

Consequently, computer algebra system still needs to progress further. They are not yet a replacement for paper and pencil. We should also know that computer algebra

systems still under some assumption. It is also important to inform the uses that whenever a numerical method is derived through the computer algebra, the results must be compared with one's expectations. We have demonstrated that our maple procedure was able to handle higher methods derivations efficiently and exhibit the expected order of convergence with various levels of sophistication. Nevertheless, as some examples in this article demonstrated, a computer algebra system is a very powerful tool for use in academic work.

## Appendix 1

```
> # The Maple Command
> # Implicit Adams Moulton Method
> Restart:
> With(LinearAlgebra):
```

for i from 0 to s do

$$G[i] := \left( \frac{1}{(x + (i - (s + 1)))} \right) \prod_{j=1}^{s+1} (x - j);$$

$$T[i] := \text{int}(G[i], x = s..s + 1);$$

$$b[-i + 1] := \frac{(-)^i}{(s - i)!} T[i] \text{ od};$$

$$y[n + 1] := y[n] + \text{factor} \left( h \sum_{j=-1}^{s-1} b[-j] f[n - j] \right);$$

$$G_0 := (x - 1)(x - 2)$$

$$T_0 := \frac{5}{6}$$

$$b_1 := \frac{5}{12}$$

$$G_1 := (x - 1)(x - 3)$$

$$T_1 := -\frac{2}{3}$$

$$b_0 := \frac{2}{3}$$

$$G_2 := (x - 2)(x - 3)$$

$$T_2 := -\frac{1}{6}$$

$$b_{-1} := -\frac{1}{12}$$

$$y_{n+1} := y_n + \frac{1}{12} (5f_{n+1} + 8f_{n+1} - f_{n-1})$$

## Appendix 2

```
> Worksheet 2
> EXPLICIT ADAMSBASHFORTH METHOD
> RESTART
> #EXPLICIT ADAMSBASHFORTH METHOD
> S=3
```

for i from 0 to s do

$$G[k] := \left( \frac{1}{(x + (k - s))} \right) \prod_{j=0}^s (x - j);$$

$$\sigma[i] := \text{int}(G[k], x = s..s + 1);$$

$$\phi[-k] := \frac{(-)^k}{(s - i)!(k)!} \sigma[k] \text{ od};$$

$$y[n + 1] := y[n] + \text{factor} \left( h \sum_{j=0}^s \phi[-j] f[n - j] \right);$$

$$G_0 := (x - 1)(x - 2)$$

$$\sigma_0 := \frac{55}{4}$$

$$\phi_0 := \frac{55}{24}$$

$$G_1 := x(x - 1)(x - 3)$$

$$\sigma_1 := \frac{59}{12}$$

$$\phi_{-1} := -\frac{59}{24}$$

$$G_2 := x(x - 2)(x - 3)$$

$$\sigma_2 := \frac{37}{12}$$

$$\phi_{-2} := \frac{37}{12}$$

$$G_3 := (x - 1)(x - 2)(x - 3)$$

$$\sigma_3 := \frac{9}{4}$$

$$\phi_{-3} := -\frac{3}{8}$$

$$y_{n+1} := y_n + \frac{1}{24} (55f_{n+1} - 59f_{n-1} - 37f_{n-2} - 9f_{n-3})$$

## References

- [1] Maplesoft, Getting Started Guide, Waterloo: <http://www.maplesoft.com>, 2007.
- [2] T. Yamaguchi, Mathematical methods with maple., Universal Academy press, inc, 2005, pp. 151-155.
- [3] D. a. H. S. Dennis, "Lecture note on numerical Analysis.," pp. 74-79, 2002.
- [4] P. a. S. Kloeden, "Construction of stochastic numerical schemes through Maple," Maple Technical Newsletter, 1993.
- [5] W. Kendell, Computer algebra and stochastic calculus, Notices Amer. Math.science, 1990, pp. 1254-1256.
- [6] H. A., Introduction to maple, third edition ed., Germany: Sringer-Verlag, 2003.
- [7] J. Lambert, Numerical methods for ordinary Differential equations: The initial value problem, 2nd ed., London: Wiley, 2000.
- [8] P. Corke, "An automated symbolic and numeric procedure for manipulating rigid-body dynamical significance analysis and simplification," vol. 12, pp. 1018-1023, 1996.
- [9] J. Butcher, Numerical Methods for ordinary differential equations, Chchester, England: Wiley and son's, 2008.
- [10] J. Wallen, "On robot modelling using Maple.," Universitytext.sweden, sweden, 2007.