

Chapter 5

Nonlinear Initial Value Ordinary Differential Equations

Mohammad M. Aghdam, Ali Fallah, and Poorya Haghi

Keywords Multi-step method • Nonlinear differential equation • Initial value problem • Error analysis

5.1 Introduction

Ordinary differential equations (ODEs) present the mathematical models of many physical problems in different fields of science, engineering, and economy. ODEs are categorized into initial value problems (IVP) and boundary value problem (BVP). BVPs usually explain the equilibrium state of the systems such as steady state heat or mass transfer while IVPs present the dynamic behavior of the system such as motion of system particles subjected to external load or transient heat/mass transfer. In this chapter, IVPs are considered. An initial value ODE in the general case is in the form of:

$$\frac{dy}{dx} = f(x, y), \quad x \in [a, b], \quad y(a) = y_0 \quad (5.1)$$

where $f(x, y)$ can be linear or nonlinear function of x and y , and y_0 is initial condition. It is worth mentioning that, higher order ODEs and systems of ODEs can also be shown in the form of Eq. (5.1). Thus, the presented method is applicable to systems of ODEs of any order.

M.M. Aghdam (✉) • P. Haghi
Department of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran
e-mail: aghdam@aut.ac.ir; phaghi@purdue.edu

A. Fallah
Department of Mechanical Engineering, Sharif University of Technology, Tehran, Iran
e-mail: a_fallah@mech.sharif.edu

Except in some special cases such as constant coefficient linear ODEs, analytical solutions for ODEs are generally not available. Therefore, various numerical methods are proposed for the solution of initial value ODEs such as methods based on Taylor series, one-step methods such as Euler method, multi-stage single-step methods such as Runge–Kutta methods and linear multi-step methods. Single-step methods only require information about the solution at one previous point such as $x = x_{n-1}$ to compute the solution at an advanced point such as $x = x_n$. For instance, Euler method is a single-step formula for the solution of IVPs that predicts the solution of the problem at $x = x_n$, only from slope of the $y(x)$ at $x = x_{n-1}$ (Süli and Mayers 2003):

$$y'_n = \frac{y_n - y_{n-1}}{h} = f(x_{n-1}, y_{n-1}) = f_{n-1} \rightarrow y_n = y_{n-1} + hf_{n-1} \quad (5.2)$$

where h is the step size, i.e. $h = x_n - x_{n-1}$ and $y_{n-1} = y(x_{n-1})$, $f_{n-1} = f(x_{n-1}, y_{n-1})$. Equation (5.2) is called explicit formula because the right hand side of (5.2) can be explicitly computed from known values of y_{n-1} . In spite of the simplicity of the Euler method, (5.2), it is not generally employed in applied cases. There are two main reasons why Euler's method is *not* generally used in scientific computing. Firstly, the truncation error per step associated with this method is far larger than those associated with other, more advanced methods for a given value of h . Secondly, Euler's method is too prone to numerical instabilities. The main reason that Euler's method has such large truncation errors per step is that the method only evaluates derivatives at the beginning of the interval: i.e., at x_{n-1} to compute the solution from x_{n-1} to x_n . Therefore, the method is very *asymmetric* with respect to the beginning and the end of the interval.

Multi-stage methods such as Runge–Kutta were developed to overcome this shortcoming. In multi-stage methods, more symmetric integration is employed by evaluating the derivatives in some intermediate points in the considered interval. For instance, in the second-order Runge–Kutta method, derivatives are evaluated by using an Euler-like trial stage to the midpoint of the interval and then using the values of both x and y at the midpoint to make the actual step across the interval, see Fig. 5.1. The second-order Runge–Kutta formula is (Süli and Mayers 2003):

$$\begin{aligned} k_1 &= hf_{n-1} \\ k_2 &= hf\left(x_{n-1} + \frac{h}{2}, y_{n-1} + \frac{k_1}{2}\right) \\ y_n &= y_{n-1} + k_2 \end{aligned} \quad (5.3)$$

Therefore, the Euler's method is equivalent to a first-order Runge–Kutta method. Obviously, there is no need to stop at a second-order method and it is possible to achieve higher order of accuracy using more trial stages per interval. The most common Runge–Kutta method is the fourth-order Runge–Kutta (RK4). It is worth mentioning that multi-stage methods such as Runge–Kutta methods are still single-step methods because only the information of the previous point is used to obtain the current point value. In spite of higher order of accuracy of the multi-stage methods with respect to simple single-step methods, they need many derivative calculations in each step.

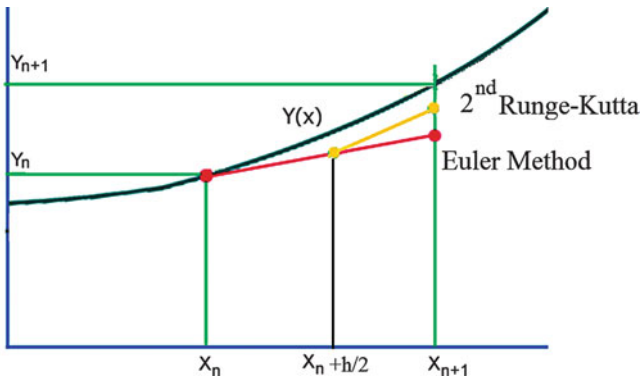


Fig. 5.1 Comparison between Euler method and second-order Runge–Kutta method

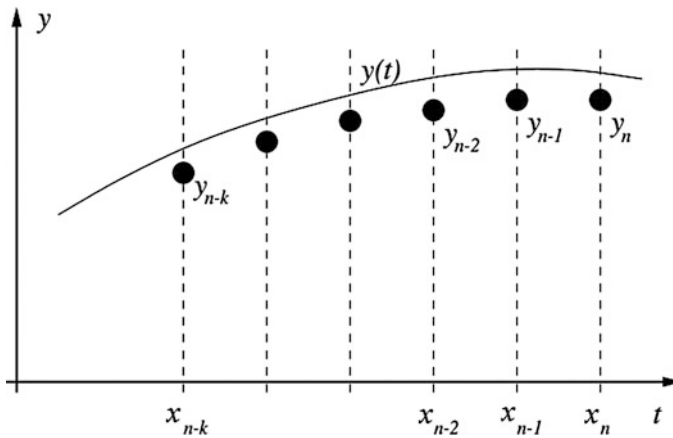


Fig. 5.2 Domain of k-step method

Multi-step methods use previous point information to increase the accuracy of the results instead of using trial stages in each interval, see Fig. 5.2.

Consider the general linear or nonlinear first-order ordinary differential equation (5.1). The integral form of (5.1) can be expressed as:

$$y(x) = y_0 + \int_a^x f(x, y) dx \tag{5.4}$$

In multi-step techniques, numerical solution is obtained by using a discretized version of (5.4) as:

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx \tag{5.5}$$

Various multi-step methods use different ways to approximate the integral term in (5.5). For instance, Adams–Moulton method, one of the most common multi-step methods, employs an approximation of $f(x, y)$ in (5.5) by using the Newton backward difference method to obtain a multi-step formulation.

Although the existing multi-step methods such as Adams–Moulton method are accurate and useful, they also have their own limitations in terms of accuracy and stability at large step sizes and weak performance in the case of stiff ODEs. Therefore, providing more accurate and/or stable predictions while increasing step size to reduce computational cost would be interesting for relevant researchers. In order to achieve this goal, in the present work, the well-known Bernstein polynomials as the basis for Bézier curves are employed. Bernstein polynomials (special case of B-spline) are piecewise polynomials that represent a great variety of functions. They can be differentiated and integrated without difficulty and construct spline curves to approximate functions with desired accuracy. There has been widespread interest in the use of these polynomials to obtain solutions for various types of differential equations (Caglar and Caglar 2006; Jator and Sinkala 2007; Khalifa et al. 2008; Caglar et al. 2008; Zerarka and Nine 2008; Bhattia and Brackenb 2007). In these studies, the solution is expanded in terms of a set of continuous Bernstein or B-spline polynomials over a closed interval and then by employing the weighted residual methods such as Galerkin or collocation to determine the expansion coefficients to construct a solution. However, in the current work, the Bernstein polynomials and Bézier curve are used to evaluate the integral in (5.5) by providing a general k -step formula.

Results of the presented formula show improved stability in comparison with Adams–Moulton method for the same step size. This leads to significant reduction of computational time. In addition, stability and convergence of the presented formula are discussed through common stability and error analysis of multi-step methods.

5.2 Bézier Curves

Before computer graphics ever existed, engineers designed aircraft wings and automobile bodies using splines. A spline is a long flexible piece of wood or plastic with a rectangular cross section held in place at various positions by heavy lead weights with a protrusion called *duck*, where the duck holds the spline in a fixed position against the drawing board (Beach 1991). The spline then conforms to a natural shape between the ducks. By moving the ducks around, the designer can change the shape of the spline. The drawbacks are obvious. Recording duck positions and maintaining the drafting equipment necessary for many complex parts will take up square footage in a storage facility, costs that would be absorbed by a consumer. A not so obvious drawback is that when analyzed mathematically, there is no closed form solution (Buss 2003).

In order to overcome these shortages, in the 1960s a mathematician and engineer named Pierre Bézier introduced a family of smooth curves based on the Bernstein basis polynomials. Bézier used the curves mainly to design various parts of automobile bodies such as exterior car panels for his work at Renault in the 1960s.

The advent of Bézier curves revolutionized the design process and helped the designers draw smooth looking curves on a computer screen, and use less physical storage space for design materials. Bézier's contribution to computer graphics has opened the road for computer-aided design (CAD) software like Maya, Blender, 3D Max, and Corel (Gerald et al. 1999; Lorentz et al. 1986; Farin 2002).

The n th-degree Bézier curve for $(n + 1)$ given points P_0, P_1, \dots, P_n can be written as:

$$B(u) = \sum_{i=0}^n B_{i,n}(u)P_i, \quad u \in [0, 1] \quad (5.6)$$

where the points P_i are called *control points* for the Bézier curves and $B_{i,n}$ are known as *Bernstein basis polynomials* of degree n defined as:

$$B_{i,n}(u) = \binom{n}{i} (1-u)^{n-i} u^i, \quad i = 0, 1, \dots, n \quad (5.7)$$

It is worth mentioning here that quadratic and cubic Bézier curves are the most common types of Bézier curves since higher degree curves are more computationally expensive to evaluate. Figure 5.3 and Table 5.1 show quadratic and cubic Bernstein basis polynomials.

The location of control points in the 2D Cartesian coordinate system is:

$$P_i = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \quad (5.8)$$

Using (5.8), one can rewrite (5.6) in the parametric form of:

$$x(u) = \sum_{i=0}^n B_{i,n}(u)x_i \quad (5.9.1)$$

$$y(u) = \sum_{i=0}^n B_{i,n}(u)y_i \quad (5.9.2)$$

Bézier curves are widely used in computer graphics and computer-aided design, mainly to provide smooth curves. It should be noted that similar to the least-square curves, Bézier curves are not really interpolating curves, as they do not normally pass through all of the control points. However, they have the important property of staying within the polygon determined by the given points as shown in Fig. 5.4.

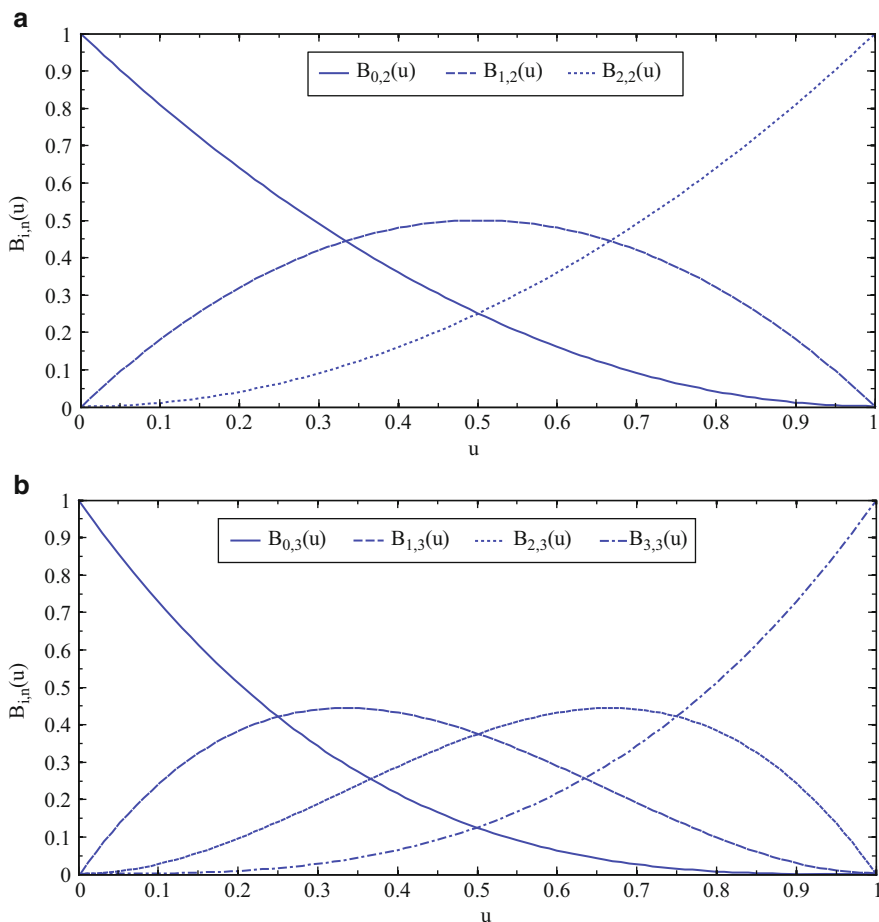


Fig. 5.3 Bernstein polynomials (a) quadratic $n = 2$, (b) cubic $n = 3$

Table 5.1 Linear, quadratic, and cubic Bernstein polynomials

n	$B_{i,n}(u)$
1	$B_{0,1}(u) = 1 - u, B_{1,1}(u) = u$
2	$B_{0,2}(u) = (1 - u)^2, B_{1,2}(u) = 2u(1 - u), B_{2,2}(u) = u^2$
3	$B_{0,3}(u) = (1 - u)^3, B_{1,3}(u) = 3u(1 - u)^2, B_{2,3}(u) = 3u^2(1 - u), B_{3,3}(u) = u^3$

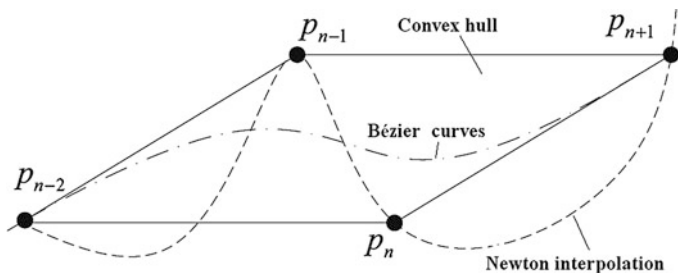


Fig. 5.4 Comparing Newton interpolation with Bézier curves and Convex hull property

More details of Bézier curves and their properties can be found elsewhere (Gerald et al. 1999; Lorentz et al. 1986; Farin 2002).

5.3 Technique

In the present study, Bézier curves of any order are used to develop a general k -step formula that becomes unstable at larger step sizes compared to the existing multi-step methods. To this end, in the integral of (5.5) is approximated by m th-degree Bézier curves using control points which are defined as:

$$P_{n-j} = \left\{ \begin{matrix} x_{n-j} \\ f_{n-j} \end{matrix} \right\} = \left\{ \begin{matrix} x_n - jh \\ f(x_{n-j}, y_{n-j}) \end{matrix} \right\}, \quad j = n - m \dots n \tag{5.10}$$

Using Eqs. (9) and (5.10), one can write the new parametric form of m th-degree Bézier curves as:

$$x(u) = \sum_{i=0}^m B_{i,m}(u)x_{n-m+i} = \varnothing(u) \tag{5.11.1}$$

$$f(x, y) = \sum_{i=0}^m B_{i,m}(u)f_{n-m+i} = \varphi(u) \tag{5.11.2}$$

Finally, after approximation of the integral term in (5.5), one may rewrite (5.5) as:

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx \cong y_n + \int_{\varnothing^{-1}(x_n)}^{\varnothing^{-1}(x_{n+1})} \varphi(u)\varnothing'(u)du \tag{5.12}$$

Using general formula (5.12) together with various orders of Bézier curves in (5.6) leads to different multi-step formulas. For instance, assuming quadratic Bézier

curve leads to three-step formula. Different degrees of Bézier curves correspond to different step formulas that are:

Two-step formula:

$$y_{n+1} = y_n + \frac{h}{2} (3f_n - f_{n-1}) \quad (5.13)$$

Three-step formula:

$$y_{n+1} = y_n + \frac{h}{12} (19f_n - 8f_{n-1} + f_{n-2}) \quad (5.14)$$

Four-step formula:

$$y_{n+1} = y_n + \frac{h}{108} (175f_n - 81f_{n-1} + 15f_{n-2} - f_{n-3}) \quad (5.15)$$

and general $m + 1$ -step formula:

$$y_{n+1} = y_n + h \sum_{i=0}^m \left(m \int_1^{\frac{m+1}{m}} \binom{m}{i} (1-u)^{m-i} u^i du \right) f_{n-m+i} \quad (5.16)$$

5.4 Consistency, Stability, and Convergence Analyses

The main concepts in the analysis of multi-step methods, and indeed any numerical method for differential equations, are consistency, stability, and convergence. The basic theory for analysis of multi-step methods has been developed by Dahlquist (1963) and is presented in the classical book of Henrici (1962).

It is useful to employ a general framework for the presentation of general $m + 1$ -step methods as:

$$a_{m+1}y_{n+1} + a_m y_n + \cdots + a_0 y_{n-(m+1)} = h [b_{m+1} f_{n+1} + \cdots + b_0 f_{n-(m+1)}] \quad (5.17)$$

Associated with (5.17) are also two *characteristics* polynomials of the $m + 1$ -step formula (5.17) as:

$$p(z) = a_{m+1}z^{m+1} + a_m z^m + \cdots + a_1 z + a_0 \quad (5.18)$$

$$q(z) = b_{m+1}z^{m+1} + b_m z^m + \cdots + b_1 z + b_0 \quad (5.19)$$

A multi-step formula is in the implicit form if $b_{m+1} \neq 0$ and is in explicit configuration if $b_{m+1} = 0$. Thus, all presented multi-step formulas in (5.13) through (5.16) are in explicit form.

Definition 1 For a given $m + 1$ -step formula Eq. (5.17) the polynomials

$$p(z) = \sum_{i=0}^{m+1} a_i z^i \tag{5.20}$$

$$q(z) = \sum_{i=0}^{m+1} b_i z^i$$

are called first and second characteristic polynomials, respectively.

A multi-step method is called consistent, if it reproduces the exact solution for the differential equation $y' = 1$, when exact starting approximations are used.

Definition 2 (Süli and Mayers 2003) The multi-step formula (5.17) is consistent if $p(1) = 0$ and $p'(1) = q(1)$.

Theorem 1 General $m + 1$ -step formula (5.16) is consistent.

Proof From (5.16) and (5.18), it is clear that for the $m + 1$ -step formula

$$p(z) = z^{m+1} - z^m \tag{5.21}$$

Using (5.21) to determine $p(1)$ and $p'(1)$, one can easily conclude for the $m + 1$ -step formula (5.16) that

$$\begin{cases} p(1) = 0 \\ p'(1) = 1 \end{cases} \tag{5.22}$$

Therefore, in order for $m + 1$ -step formula (5.16) to be consistent, the value of $q(1)$ must be equal to one. Using general $m + 1$ -step formula (5.16) in conjunction with (5.20), one can determine coefficients of $q(z)$ in (5.20) as:

$$b_i = m \int_1^{\frac{m+1}{m}} \binom{m}{i} (1-u)^{m-i} u^i du = \binom{m}{i} \left(\sum_{j=0}^{m-i} \binom{m-i}{j} (-1)^j \left(\frac{m}{i+j+1} \right) \left[\left(\frac{m+1}{m} \right)^{i+j+1} - 1 \right] \right), \quad i = 0 \dots m \tag{5.23}$$

Substitution of (5.23) in (5.20), one can derive $q(1)$ as:

$$q(1) = \sum_{i=0}^m b_i = \sum_{i=0}^m \left(\binom{m}{i} \left(\sum_{j=0}^{m-i} \binom{m-i}{j} (-1)^j \left(\frac{m}{i+j+1} \right) \left[\left(\frac{m+1}{m} \right)^{i+j+1} - 1 \right] \right) \right) \tag{5.24}$$

After some mathematical simplification, (5.24) can be rewritten as:

$$\sum_{i=0}^m b_i = \sum_{i=0}^m c_i \tag{5.25}$$

where

$$c_i = \binom{m}{m-i} (-1)^{m-i} \left(\frac{m}{m+1-i} \right) \left[\left(\frac{m+1}{m} \right)^{m+1-i} - 1 \right] (1 + (-1))^{m-i} \tag{5.26}$$

It can easily be shown that the last term in (5.26) is zero for $i=0 \dots m-1$. However, for $i=m$ the value of c_i in (5.26) becomes one. Therefore, considering (5.24) and (5.26) results in:

$$q(1) = \sum_{i=0}^m b_i = \sum_{i=0}^m c_i = c_m = 1 \tag{5.27}$$

Consequently, comparing (5.22) and (5.27) with Definition 2, it can be concluded that the general $m + 1$ -step formula (5.16) is consistent.

The numerical solution of a single-step method depends on the initial condition, y_0 while numerical solution of a k -step method depends on all the k starting values y_1, y_2, \dots, y_k . Thus, it is important to know whether the numerical solution is stable with respect to perturbations in the starting values or not. A multi-step formula is *stable* (zero-stable) for a certain differential equation on a given step size, h , if a perturbation in the starting values of size ε causes numerical solution over that interval to change by no more than $\alpha\varepsilon$ for some value of α which does not depend on the step size h . This is called “zero-stability” because it is enough to check the condition for the differential equation $y' = 0$ (Süli and Mayers 2003).

Definition 3 (Süli and Mayers 2003) The multi-step formula in general form of (5.17) is stable (zero-stable) if all roots of $p(z)$ lie in the disk $|z| \leq 1$ and if each root of modulus one is simple.

Theorem 2 General $m + 1$ -step formula (5.16) is zero-stable.

Proof Considering (5.21), roots of the first characteristic polynomial of general $m + 1$ -step formula (5.16) are:

$$z_i = \begin{cases} 0 & i = 1 \dots m \\ 1 & i = m + 1 \end{cases} \quad (5.28)$$

Using Definition 3, one may conclude that the $m + 1$ -step formula (5.16) is zero-stable.

Accuracy of results of the multi-step methods depends on the starting point values and the value of step size. It is important to be assured that as the step size value decreases, difference between the results of the multi-step formula and the exact solution also decreases, and prediction tends to the exact solution. Consider multi-step formula (5.17) is employed for the solution of IVP (5.1) in domain $a < x < b$ and the sequence of meshes as $\{a = x_0 < x_1 < x_2 \dots < x_n = b\}$. As the number of intervals increased, i.e., $n \rightarrow \infty$ the step size tends to zeros, $h \rightarrow 0$. The multi-step method is said to be convergent if as $h \rightarrow 0$ the $y_n \rightarrow y(t_n)$. Swedish mathematician Dahlquist (1963) proved the relation between the concepts of zero-stability, consistency, and convergence of a multi-step method by the theorem named after him as Dahlquist's equivalence theorem.

Theorem 3 (Dahlquist's Equivalence Theorem) (Dahlquist 1963) For the multi-step formula (5.17) to be convergent, it is necessary and sufficient that the formula (5.17) to be zero-stable and consistent.

Proof Proof of this theorem is beyond of scope of the presented chapter as details can be found in Dahlquist (1963), Henrici (1962), and Gautschi (1997).

Theorem 4 General $m + 1$ -step formula (5.16) is convergent.

Proof Since the general $m + 1$ -step formula (5.16) is both zero-stable and consistent, based on Dahlquist's equivalence theorem the general $m + 1$ -step formula (5.16) is also convergent.

Definition 4 (Süli and Mayers 2003) A convergent multi-step formula (5.17) is said to be strongly stable if $z = 1$ is the only root of modulus 1 of its first characteristic polynomial. If there is more than one such root, it is said to be relatively stable.

Theorem 5 General $m + 1$ -step formula (5.16) is strongly stable.

Proof According to (5.28), $p(z)$ have m zero roots while the only one nonzero root is $z = 1$, thus the general $m + 1$ -step formula (5.16) is strongly stable.

5.5 Numerical Results and Discussions

In this section, efficiency of the presented multi-step method in terms of accuracy, convergence, and computational cost is investigated using various numerical examples. It should be noted that all considered examples are nonlinear ODEs

and therefore, analytical solutions are not available. Predictions of the present work with four-step formula (5.15) are compared with numerical results of ODE45 MATLAB. ODE45 is a single-step solver based on an explicit fourth- or fifth-order Runge–Kutta formula for the solution of ordinary differential equations in MATLAB software (Guide et al. 2004). Furthermore, results of the well-known four-step Adams–Moulton method are also included mainly to compare stability of the presented method.

In examples, three different levels of the step size h are chosen. These step sizes are carefully selected to show three different stages. At the first stage, both presented and Adams–Moulton numeric methods are stable and accurate while in the second stage, instability initiates in the Adams–Moulton technique and finally at the third stage unstable behavior can be observed in the presented method. Furthermore, in order to examine accuracy of the method, error for both numeric techniques is reported in some examples.

In both methods, four-step formula is employed which indeed information of the first four points is needed to start solution procedure. Since y_0 is the only known initial condition, the ODE45 results are used to obtain the first three points, i.e., y_1, y_2, y_3 for both presented and the Adams–Moulton method.

5.5.1 Example 1

As first example, a nonlinear mass–spring–damper system as shown in Fig. 5.5 is considered. Governing equation of such system is the well-known Duffing equation (Nayfeh et al. 2007):

$$m \frac{d^2 y}{dt^2} + c \frac{dy}{dt} + K_L y + K_{NL} y^3 = 0 \quad (5.29)$$

where m , c , K_L , and K_{NL} are mass, damping coefficient, linear, and nonlinear spring stiffnesses, respectively.

Introducing new variables $y_1 = y$ and $y_2 = \frac{dy}{dt}$, the second-order governing nonlinear ODE (5.29) can be converted to a first order system of nonlinear ODE as:

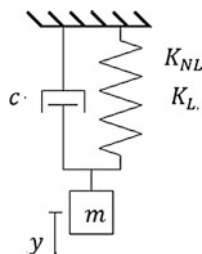


Fig. 5.5 Nonlinear mass–spring–damper system

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = -\frac{1}{m}(cy_2 + K_L y_1 + K_{NL} y_1^3) \end{cases} \quad (5.30)$$

Figure 5.6 shows predictions of presented and Adams–Moulton methods together with numerical solution obtained by ODE45 for different values of step sizes in the case of nonlinear mass–spring system with damping. Numerical results are obtained for $m = 1$, $c = 0.5$, $K_L = K_{NL} = 1$ and initial condition $y(0) = 1$, $y'(0) = 0$. In each case, errors of the Adams–Moulton ($y_{Adams} - y_{ODE45}$) and presented method ($y_{Bezier} - y_{ODE45}$) predictions are provided in the figures. Results show that for small values of step size ($h = 0.1$), both methods are accurate and stable while for higher step size ($h = 0.3$), error of the Adams–Moulton grows rapidly. For step sizes greater than $h = 0.3$, the Adams–Moulton becomes unstable whereas the presented method is still stable with reasonable level of accuracy for $h = 0.5$.

5.5.2 Example 2

In the second example, the Lotka–Volterra equation, which is, also known as the predator–prey equation is considered. This equation describes the dynamics of biological systems in which two species interact, one as a predator and the other as prey. The populations change through time according to the pair of equations (Berryman 1992):

$$\begin{cases} \frac{dx}{dt} = ax - bxy \\ \frac{dy}{dt} = -cy + dxy \end{cases} \quad (5.31)$$

where $x(t)$ and $y(t)$ denote population of the prey and predator species, respectively and a, b, c , and d are positive constants.

Solution of predator–prey equation with $a = 1$, $b = 0.03$, $c = 0.4$, $d = 0.01$ using presented and Adams–Moulton methods is shown in Fig. 5.7. For $h = 0.1$, results for both methods are accurate and stable. As the step size growth to $h = 0.3$, Adams–Moulton shows instability while the presented method become unstable in larger step size of $h = 0.5$.

5.5.3 Example 3

Consider a Van der Pol oscillator, which is a non-conservative oscillator with nonlinear damping and governing equation of motion as (Nayfeh et al. 2007):

$$\frac{d^2y}{dt^2} - \mu(1 - y^2)\frac{dy}{dt} + y = 0 \quad (5.32)$$

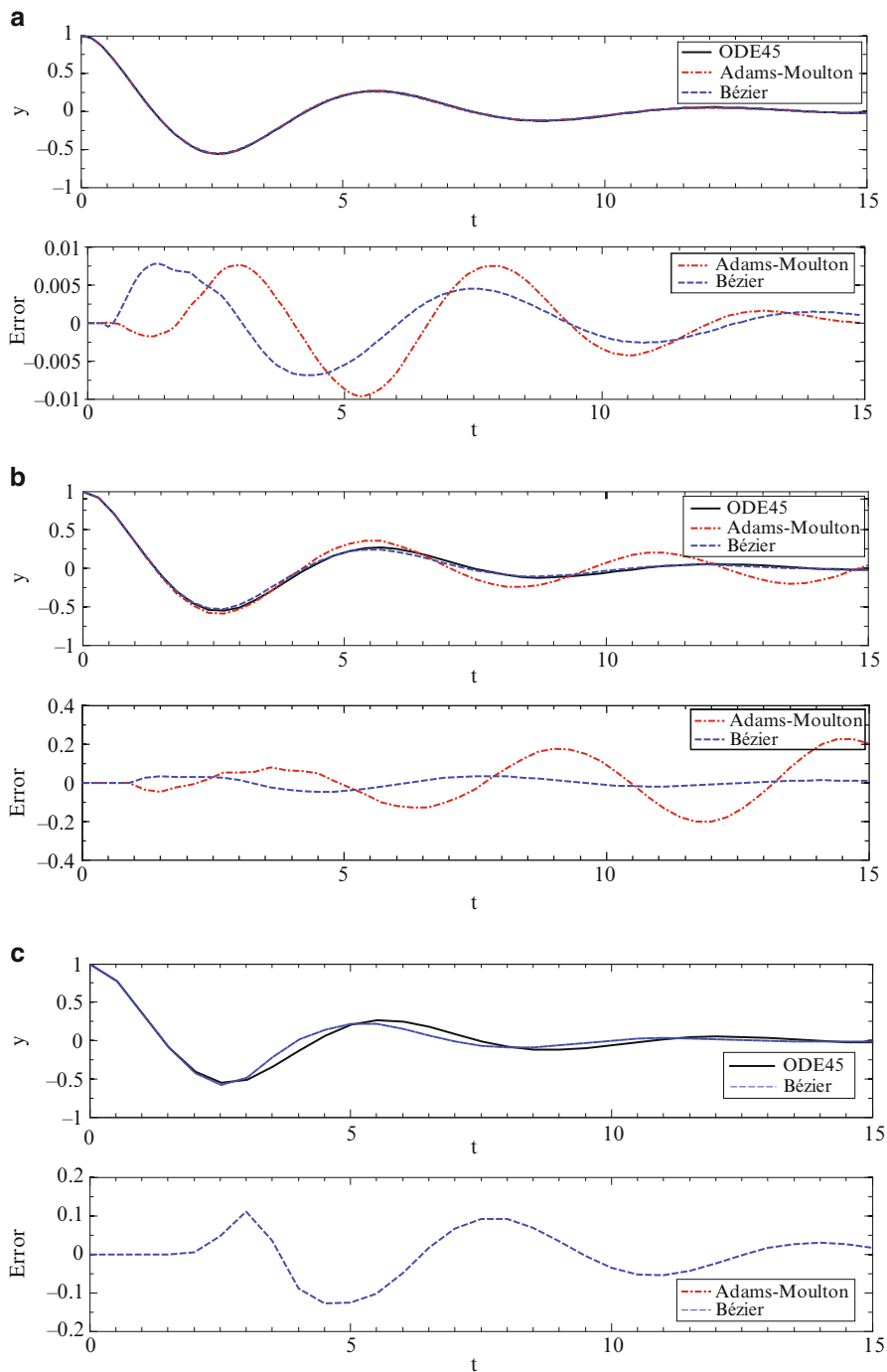


Fig. 5.6 Solution of nonlinear mass-spring system with damping for (a) $h = 0.1$, (b) $h = 0.3$, (c) $h = 0.5$

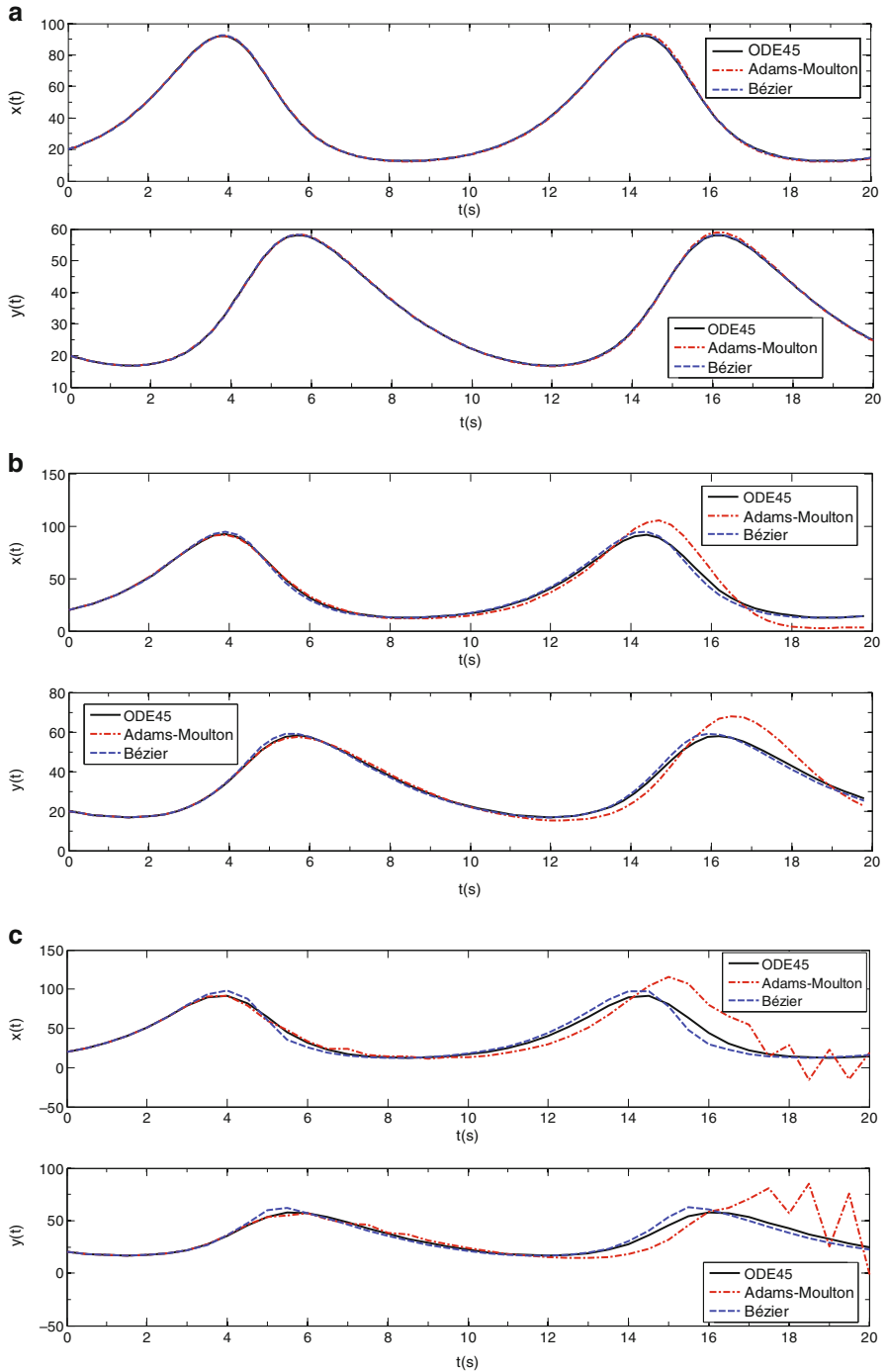


Fig. 5.7 Solution of Lotka–Volterra equation for (a) $h = 0.1$, (b) $h = 0.3$, (c) $h = 0.5$

where μ is a scalar parameter indicating the nonlinearity and the strength of the damping. In the case of $\mu = 0$, Eq. (5.32) reduces to $\frac{d^2y}{dt^2} + y = 0$ which has the solution $y = c_1 \sin(t) + c_2 \cos(t)$. For small values of μ , one obtain weekly nonlinear oscillation, i.e., oscillation which slightly differ from simple harmonic motion. On the other hand, however, for high values of μ relaxation oscillation can be observed which shows strongly nonlinear oscillations exhibiting sharp periodic jumps. In the case of high damping, the Van der Pol equation becomes stiff. Most techniques show weak performance for solution of stiff ODEs. Here the performance of the presented and Adams–Moulton methods in two different conditions of low and high damping is investigated.

Introducing new variables $y_1 = y$ and $y_2 = \frac{dy}{dt}$, the second order governing nonlinear ODE is converted to the following first order system of nonlinear ODE:

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = \mu(1 - y_1^2)y_2 + y_1 \end{cases} \quad (5.33)$$

Solution of the Van der Pol equation in the case of $\mu = 0.5$ and $\mu = 5$ using presented multi-step method and Adams–Moulton is presented in Figs. 5.8 and 5.9, respectively. Included in the figures are also results of the ODE45 for comparison. In the case of lower damping, i.e., $\mu = 0.5$, predictions of both presented and Adams–Moulton for $h = 0.1$ are stable and accurate. As the step size increases to $h = 0.2$, Adams–Moulton results become unstable while the presented method shows unstable behavior at $h = 0.4$.

Furthermore, for higher damping of $\mu = 5$, smaller step size is used. For $h = 0.001$, both methods are accurate and stable. As the step size growth to $h = 0.005$, instability initiates in the results of Adams–Moulton while the presented method becomes unstable at step sizes ten times greater, i.e., $h = 0.05$. Results show that even in the case of stiff ODEs, the presented multi-step method is more accurate and stable in comparison with the well-known Adams–Moulton method.

5.6 Conclusion

Multi-step methods such as Adams–Moulton are common techniques for solution of initial value ODEs. Besides their accuracy and stability, they also have their own limitations such instability in large step sizes or in the case of stiff differential equation. Thus, multi-step methods with better performance in terms of accuracy and convergence particularly in the case of stiff equations are appealing.

In the present study, a new general multi-step formula developed based on the Bézier curves. The consistency, stability, and convergence of the presented formula are proved by using the common stability and error analysis theorems of multi-step methods. The other advantage of the method is the presentation of a closed form general $m + 1$ -step formula.

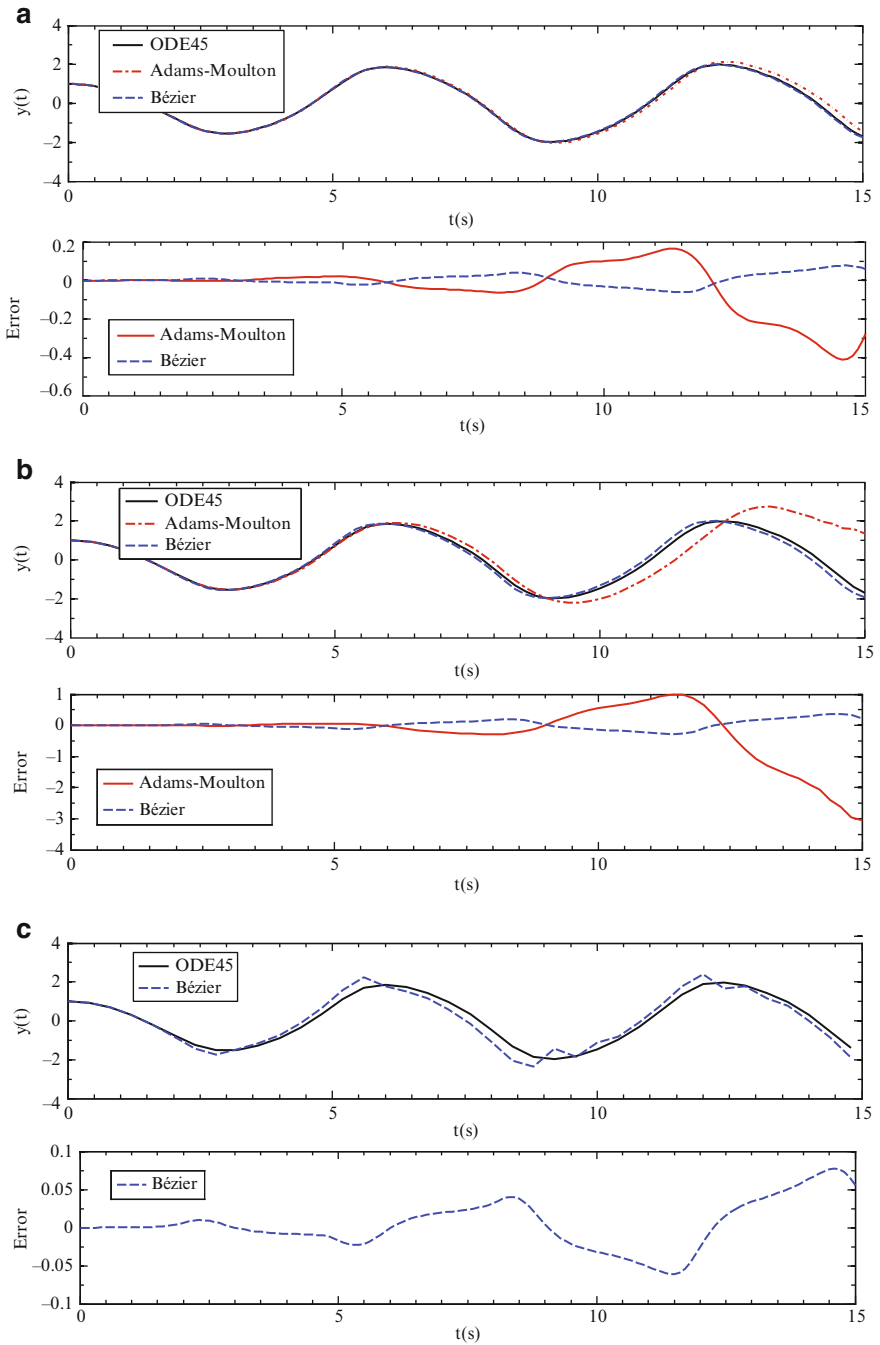


Fig. 5.8 Solution of Van der Pol equation in case of low damping $\mu = 0.5$ for (a) $h = 0.1$, (b) $h = 0.2$, (c) $h = 0.4$

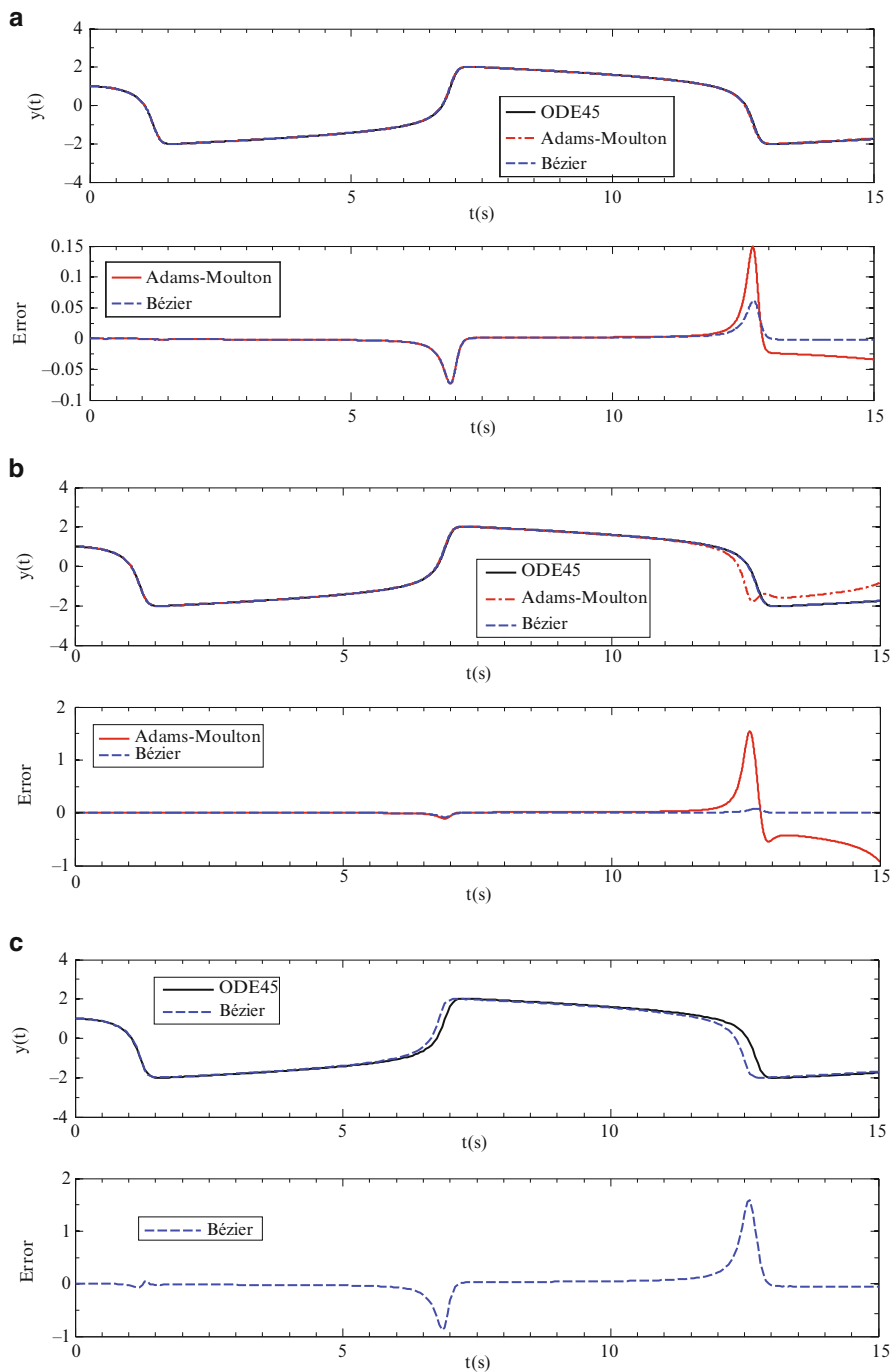


Fig. 5.9 Solution of Van der Pol equation in case of high damping $\mu = 5$ for (a) $h = 0.001$, (b) $h = 0.005$, (c) $h = 0.05$

Effects of the step size on the accuracy and stability of the presented method are examined through numerical results of various nonlinear and stiff systems. Results revealed that the presented method shows improved stability behavior in comparison with the well-known Adams–Moulton method, which yields to significant reduction of computational cost. Due to simplicity, minimum computational efforts, and accuracy, the presented multi-step method is expected to be used in nonlinear problems in various science and engineering applications.

Key Symbols

$B_{i,n}(u)$	The i th Bernstein basis polynomials of degree n
P_i	Bézier curve control point
$p(z)$	First characteristic polynomials of multi-step formula
$q(z)$	Second characteristic polynomials of multi-step formula
f_n	Value of $f(x, y)$ at point (x_n, y_n)
h	Step size
x_n	The n th discretized point x coordinate
y_n	The n th discretized point y coordinate and value of $y(x)$ at point (x_n)
x_0	x coordinate of the first point
y_0	Initial condition

References

- Beach RC (1991) An introduction to curves and surfaces of computer-aided design. Van Nostrand Reinhold, New York
- Beryman AA (1992) The origins and evolution of predator–prey theory. *Ecology* 73(5):1530–1535
- Bhattia MI, Brackenb P (2007) Solutions of differential equations in a Bernstein polynomial basis. *Comput Appl Math* 205:272–280
- Buss SR (2003) 3D computer graphics-a mathematical introduction with open GL. Cambridge University Press, Cambridge, UK
- Caglar N, Caglar H (2006) B-spline solution of singular boundary value problems. *Appl Math Comput* 182:1509–1513
- Caglar H, Ozer M, Caglar N (2008) The numerical solution of the one-dimensional heat equation by using third degree B-spline functions. *Chaos Solit Fract* 38:1197–1201
- Dahlquist GG (1963) A special stability problem for linear multi-step methods. *BIT Numer Math* 3(1):27–43
- Farin G (2002) Curves and surfaces for CAGD. A practical guide, 5th edn. Elsevier Ltd., San Francisco, CA, USA
- Gautschi W (1997) Numerical analysis: an introduction. Birkhäuser, Boston
- Gerald CF, Wheatley PO (1999) Applied numerical analysis, 6th edn. Addison-Wesley, Reading
- Henrici P (1962) Discrete variable methods in ordinary differential equations. Wiley, New York
- Jator S, Sinkala Z (2007) A high order B-spline collocation method for linear boundary value problems. *Appl Math Comput* 191:100–116

- Khalifa AK, Raslan KR, Alzubaidi HM (2008) A collocation method with cubic B-splines for solving the MRLW equation. *Comput Appl Math* 212:406–418
- Lorentz GG (1986) *Bernstein polynomials*, 2nd edn. Chelsea Publishing Co., New York
- MATLAB Reference Guide Version 7.0 (2004) The Math Works Inc
- Nayfeh AH, Mook DT (2007) *Nonlinear oscillations*. Wiley, New York
- Süli E, Mayers D (2003) *An introduction to numerical analysis*. Cambridge University Press, Cambridge, UK
- Zerarka A, Nine B (2008) Solutions of the Von Kàrmàn equations via the non-variational Galerkin-B-spline approach. *Commun Nonlinear Sci Numer Simulat* 13:2320–2327