Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 205 (2007) 272–280
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Solutions of differential equations in a Bernstein polynomial basis

M. Idrees Bhatti[a], P. Bracken[b,*]

[a] *Department of Physics and Geology, University of Texas Pan American, Edinburg, TX 78541-2999, USA*
[b] *Department of Mathematics, University of Texas Pan American, Edinburg, TX 78541-2999, USA*

Received 2 December 2005; received in revised form 28 April 2006

Abstract

An algorithm for approximating solutions to differential equations in a modified new Bernstein polynomial basis is introduced. The algorithm expands the desired solution in terms of a set of continuous polynomials over a closed interval and then makes use of the Galerkin method to determine the expansion coefficients to construct a solution. Matrix formulation is used throughout the entire procedure. However, accuracy and efficiency are dependent on the size of the set of Bernstein polynomials and the procedure is much simpler compared to the piecewise B spline method for solving differential equations. A recursive definition of the Bernstein polynomials and their derivatives are also presented. The current procedure is implemented to solve three linear equations and one nonlinear equation, and excellent agreement is found between the exact and approximate solutions. In addition, the algorithm improves the accuracy and efficiency of the traditional methods for solving differential equations that rely on much more complicated numerical techniques. This procedure has great potential to be implemented in more complex systems where there are no exact solutions available except approximations.

MSC: 32.10.Dk

Keywords: Bernstein polynomial; Galerkin method; Nonlinear differential equation

1. Introduction

Continuous or piecewise polynomials are incredibly useful mathematical tools as they are precisely defined, calculated rapidly on a modern computer system and can represent a great variety of functions. They can be differentiated and integrated without difficulty, and can be put together to form spline curves that can approximate any function to any accuracy desired.

There has been widespread interest in the use of B splines in complex atomic systems. These are special spline functions that are well adapted to numerical tasks and are being extensively used in production type programs for approximating data and representing functions of any degree. Many authors such as Bottcher and Strayer [2] applied the B splines to time-dependent problems, Johnson and co-workers to many-body perturbation theory [7,8] Fischer and co-workers [3–5] to Hartree–Fock calculations and continuum problems. Saperstein and Johnson provided more detailed discussion on the application of B splines in a review [12]. Bhatti and co-workers [1] introduced a procedure

* Corresponding author. Tel.: +1 956 381 3457.
E-mail address: bracken@panam.edu (P. Bracken).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2006.05.002

and implemented B splines of degree n in approximating the solution of the non-homogeneous second order differential equations for calculating static polarizabilities of the hydrogenic states. This algorithm improved the efficiency and accuracy over traditional methods and perhaps over the k-order B spline method. Qiu and Fischer [11] also introduced integration by cell algorithm for Slater integrals in a spline basis, which significantly improved the efficiency and accuracy over traditional methods [11].

In this paper, we introduce and implement a new algorithm based on a modification to the type of bases for the space of polynomials, the Bernstein basis [6]. We discuss a new form of polynomial, the so-called Bernstein polynomials (B-polynomials) which have many useful properties. The procedure takes advantage of the continuity and unity partition properties of the basis set of B-polynomials over an interval $[0, R]$. The B-polynomial bases vanish except the first polynomial at $x = 0$, which is equal to 1 and the last polynomial at $x = R$, which is also equal to 1 over the interval $[0, R]$. This provides greater flexibility in which to impose boundary conditions at the end points of the interval. It also ensures that the sum at any point x of all the B-polynomials is unity. In many applications, a matrix formulation for the B-polynomials of degree less than or equal to n is considered. These are straightforward to develop and are applied to solve the differential equations in this paper. The set of B-polynomials of degree n on an interval forms a complete basis for continuous $(n + 1)$ polynomials. We present the solutions to three linear differential equations as linear combinations of these polynomials $P(x) = \sum_{i=0}^{n} C_i B_{i,n}(x)$, and the coefficients C_i are determined using the Galerkin method [6]. In the following sections, we explain the procedure, define the polynomial basis and show graphs of approximated solutions to the differential equations. Finally, an example in which a nonlinear equation is solved is treated at the end.

2. Polynomial basis

The general form of the B-polynomials of nth-degree are defined by

$$B_{i,n}(x) = \binom{n}{i} \frac{x^i (R - x)^{n-i}}{R^n}, \quad 0 \leqslant i \leqslant n, \tag{1}$$

for $i = 0, 1, \ldots, n$, where the binomial coefficients are given by

$$\binom{n}{i} = \frac{n!}{i!\,(n - i)!}, \tag{2}$$

and R is the maximum range of the interval $[0, R]$ over which the polynomials are defined to form a complete basis. There are $n + 1$ nth-degree polynomials. For convenience, we set $B_{i,n}(x) = 0$, if $i < 0$ or $i > n$. A simple code written in *Mathematica* or *Maple* can be used to generate all the non-zero polynomials of any degree n supported over the interval. The first and last polynomial are generally related to the boundary conditions of the problem currently under investigation. As an example, a set of 10 B-polynomials of degree 9 is plotted in Fig. 1.

These polynomials are quite easy to write down and the binomial coefficients may be obtained from Pascal's triangle. One may observe that the exponents on the x-term increase by one as i increase, while the exponents on the $(R - x)$-term decrease by one as i increases. As an example, we provide a set of 10 polynomials of degree 9 below:

$$\begin{aligned}
&B_{0,9}(x) = \frac{(10 - x)^9}{1\,000\,000\,000}, \quad B_{1,9}(x) = \frac{9(10 - x)^8 x}{1\,000\,000\,000},\\
&B_{2,9}(x) = \frac{9(10 - x)^7 x^2}{250\,000\,000}, \quad B_{3,9}(x) = \frac{21(10 - x)^6 x^3}{250\,000\,000},\\
&B_{4,9}(x) = \frac{63(10 - x)^5 x^4}{500\,000\,000}, \quad B_{5,9}(x) = \frac{63(10 - x)^4 x^5}{500\,000\,000},\\
&B_{6,9}(x) = \frac{21(10 - x)^3 x^6}{250\,000\,000}, \quad B_{7,9}(x) = \frac{9(10 - x)^2 x^7}{250\,000\,000},\\
&B_{8,9}(x) = \frac{9(10 - x) x^8}{1\,000\,000\,000}, \quad B_{9,9}(x) = \frac{x^9}{1\,000\,000\,000}.
\end{aligned} \tag{3}$$

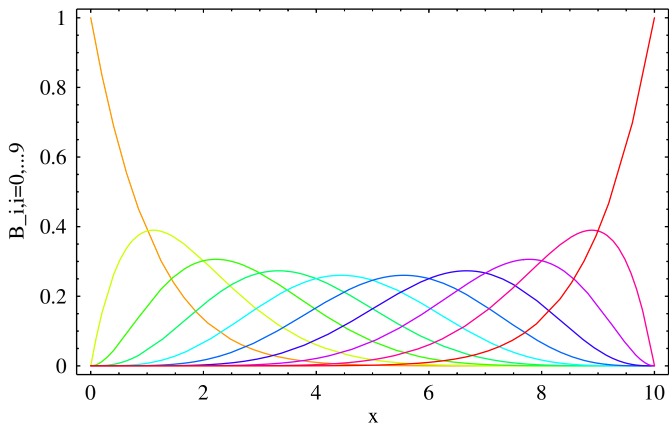

Fig. 1. The set of 10 B-polynomials of degree 9 are shown in the region $x = [0, 10]$. The quantities are dimensionless.

A recursive definition also can be used to generate the B-polynomials over this interval so that the ith nth degree B-polynomial can be written

$$B_{i,n}(x) = \frac{(R-x)}{R} B_{i,n-1}(x) + \frac{x}{R} B_{i-1,n-1}(x). \tag{4}$$

The derivatives of the nth degree B-polynomials are polynomials of degree $n-1$ and are given by

$$\frac{\mathrm{d}B_{i,n}(x)}{\mathrm{d}x} = \frac{n}{R} (B_{i-1,n-1}(x) - B_{i,n-1}(x)). \tag{5}$$

It can be readily shown that each of the B-polynomials is positive and also the sum of all the B-polynomials is unity for all real x belonging to the interval $[0, R]$, that is, $\sum_{i=0}^{n} B_{i,n}(x) = 1$. It can be easily shown that any given polynomial of degree n can be expanded in terms of a linear combination of the basis functions

$$P(x) = \sum_{i=0}^{n} C_i B_{i,n}(x), \quad n \geqslant 1. \tag{6}$$

Any set of functions having these properties is called a partition of unity on the interval $[0, R]$. B-splines have similar properties and have been extensively used in most programs for computer-aided design. It is perhaps instructive to look at the graphs of the individual B-polynomials of the sums and to visualize how they might add to approximate a function $y(x)$.

3. Approximation of solutions in the B-polynomial basis

We use the Galerkin method [6] to approximate the solutions to the differential equations we study. Four examples are chosen to demonstrate the B-polynomial basis method for approximating solutions. Three linear systems are studied and one nonlinear system.

Example 1. Consider the equation

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + y = x^2 \mathrm{e}^{-x}, \tag{7}$$

whose solution is sought on $0 \leqslant x \leqslant R$ with the boundary conditions $y(0) = y(R) = 0$. An approximation to the solution may be written as

$$y(x) = \sum_{i=0}^{n} C_i B_{i,n}(x), \quad n \geqslant 1, \tag{8}$$

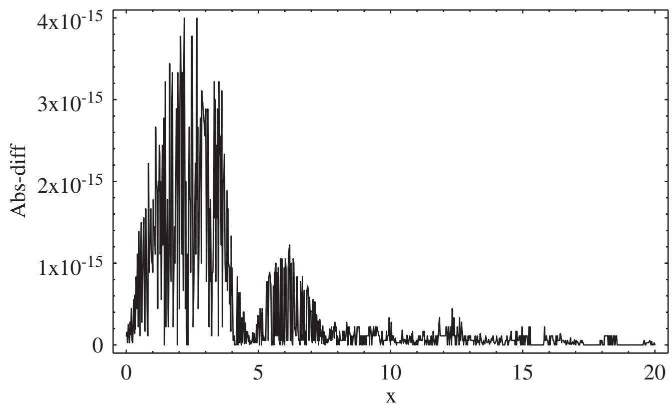

Fig. 2. A plot of absolute difference between exact and approximate solutions is shown. The absolute difference is obtained in a basis set of 45 B-polynomials over the interval [0, 20].

where C_i is the ith coefficient of the expansion. By substituting (8) into the differential equation (7) with small simplification and from the variational property with respect to the coefficients, we obtain

$$\sum_{i=0}^{n} \left\{ \int_0^R [-B'_{i,n}(x)B'_{j,n}(x) + B_{i,n}(x)B_{j,n}(x)]\,\mathrm{d}x \right\} C_i - \int_0^R x^2\,\mathrm{e}^{-x} B_{j,n}(x)\,\mathrm{d}x = 0, \tag{9}$$

where $B'_{i,n}(x)$ denotes the first derivative of the B-polynomial. This process leads to an $n+1$ by $n+1$ linear system of equations $\mathbf{BC} = \mathbf{b}$, in the variables $C_0, C_1, \ldots, C_n$, where the symmetric matrix $\mathbf{B}$ is given by

$$B_{i,j} = \int_0^R [-B'_{i,n}(x)B'_{j,n}(x) + B_{i,n}(x)B_{j,n}(x)]\,\mathrm{d}x, \tag{10}$$

and the column matrix $\mathbf{b}$ has elements

$$b_j = \int_0^R x^2\,\mathrm{e}^{-x} B_{j,n}(x)\,\mathrm{d}x. \tag{11}$$

Applying both boundary conditions, the matrix equation $\mathbf{BC} = \mathbf{b}$ is solved after imposing the boundary conditions on the matrix given by deleting the first row and first column and last row and last column. The expansion coefficients C_i and substituted into (8) to determine an approximate solution to the differential equation (7). The exact solution for (7) is also obtained analytically applying both boundary condition

$$y(x) = c_1 \cos(x) + c_2 \sin(x) + \frac{1}{2}\,\mathrm{e}^{-x}(1+x)^2,$$

$$c_1 = -\frac{1}{2}, \quad c_2 = \frac{1}{2\sin(R)}\,(\cos(R) - \mathrm{e}^{-R}(1+R)^2). \tag{12}$$

In Fig. 2, we display a plot of the absolute difference between approximate and exact solutions given in (7) and (12). Both solutions are overlapping, showing no appreciable differences. The solution of the differential equation is obtained using 45 B-polynomials. With a set of 21 B-polynomials, the difference between the respective solutions is of the order of 10^{-5} and the accuracy increases as the basis set is increased. It is also noticed that when the basis set of B-polynomials is doubled, the accuracy is tripled. In this particular example, the absolute difference between the exact and the approximate solutions diminishes nearly reaching the order of 10^{-15}, an indication that the approximate solution is excellent. Therefore, the number of polynomials were chosen to be 45 to obtain the desired accuracy.

Example 2. Consider another differential equation that of the forced harmonic oscillator

$$y''(x) + y(x) = \cos(x). \tag{13}$$

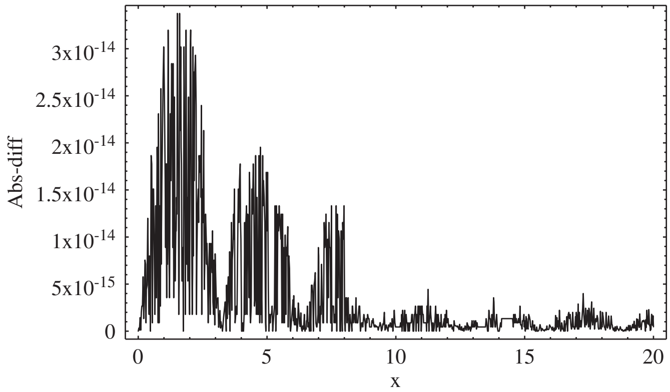

Fig. 3. A plot of absolute difference between exact and approximate solutions for Example 2 is shown. The absolute difference is obtained using a basis set of 45 B-polynomials continuous over the entire interval.

We follow the same procedure as in Example 1 to find the solution of (13) in the region $0 \leqslant x \leqslant R$ with boundary conditions $y(0)=y(R)=0$. Expanding (13) in terms of the basis set of B-polynomials, only vector **b** will change

$$b_j = \int_0^R \cos(x) B_{j,n}(x)\,\mathrm{d}x. \tag{14}$$

The exact solution of (13) is given by

$$\begin{aligned} y(x) &= c_1 \cos x + c_2 \sin x + \tfrac{1}{4}\,(2\cos^3 x + 2x\sin x + \sin x\ \sin 2x),\\ c_1 &= -\frac{1}{2}, \quad c_2 = -\frac{1}{\sin R}\left(-\frac{1}{2}\cos R + \frac{1}{4}(2\cos^3 R + 2R\sin R + \sin R \sin 2R)\right). \end{aligned} \tag{15}$$

A plot of the absolute difference between exact and approximate solutions is shown in Fig. 3. The accuracy is of the order of 10^{-14} with 45 B-polynomials.

Example 3. We consider a more complicated differential equation which is that of a critically damped harmonic oscillator to demonstrate that the B-polynomials are powerful to approximate the solution to desired accuracy. The equation we consider is

$$2y'' + 3y' + y = \cos(x), \tag{16}$$

whose exact solution under the above-mentioned boundary conditions is given by

$$\begin{aligned} y(x) &= c_1 \mathrm{e}^{-x/2} + c_2 \mathrm{e}^{-x} + \frac{1}{10}(3\sin x - \cos x),\\ c_1 &= \frac{1}{10}\left(1 + \frac{\mathrm{e}^{-R/2} + 3\sin R - \cos R}{\mathrm{e}^{-R} - \mathrm{e}^{-R/2}}\right), \quad c_2 = -\frac{\mathrm{e}^{-R/2} + 3\sin R - \cos R}{10(\mathrm{e}^{-R} - \mathrm{e}^{-R/2})}. \end{aligned} \tag{17}$$

We again use a basis set of 45 B-polynomials to approximate the solution of (16) and compare it with the exact solution following the procedure given in Examples 1 and 2. The procedure yields matrix **B** and the vector **b** as follows:

$$\begin{aligned} B_{i,j} &= \int_0^R [-2B'_{i,n}(x)B'_{j,n}(x) + 3B'_{i,n}(x)B_{j,n}(x) + B_{i,n}(x)B_{j,n}(x)]\,\mathrm{d}x,\\ b_j &= \int_0^R \cos(x) B_{j,n}(x)\,\mathrm{d}x. \end{aligned} \tag{18}$$

In the three examples considered, the inverse of matrix **B** is determined after imposing the boundary condition on the matrix by deleting the first row and first column and last row and last column. Finally, the equation **BC** = **b** is solved for

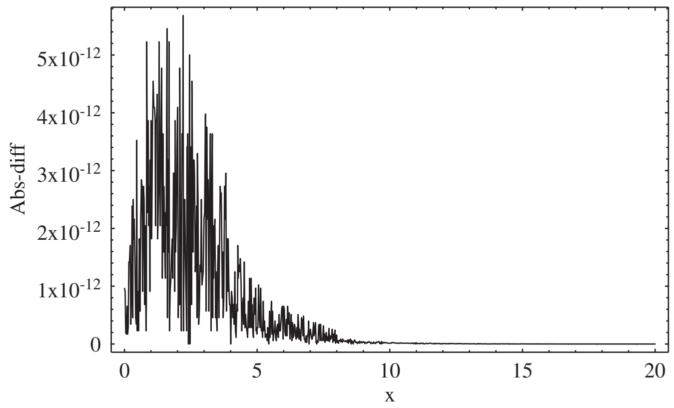

Fig. 4. A plot of absolute difference between exact and approximate solutions for Example 3 is shown. The absolute difference is obtained using a basis set of 45 B-polynomials continuous over the entire interval.

the coefficients to find the solution of the differential equation. The absolute difference between exact and approximate solutions is shown in Fig. 4. The absolute difference between both solutions is nearly of the order of 10^{-14}, yielding a very accurate solution. However, the desired accuracy of the solutions hinges on the size of the basis set chosen.

Example 4. This final example considers a nonlinear differential equation. It shows that these polynomials are suitable to tackle nonlinear equations by approximating the solution of the equation to reasonable accuracy. The equation we consider is a nonlinear Bernoulli equation given by

$$y' - y + 2y^2 = 0. \tag{19}$$

The exact solution subject to the initial condition $y(0) = \frac{1}{3}$ is given by

$$y(x) = \frac{1}{2 + \mathrm{e}^{-x}}. \tag{20}$$

Expand the solution of Eq. (19) in the approximate form

$$y(x) = a_0 + \sum_{i=1}^{n} a_i B_{i,n}(x). \tag{21}$$

Substitution of Eq. (21) into Eq. (19) yields an equation of the form

$$\sum_i a_i B'_{i,n}(x) - \sum_i a_i B_{i,n}(x) + 4a_0 \sum_i a_i B_{i,n}(x) + 2 \sum_i a_i B_{i,n}(x) \sum_j a_j B_{j,n}(x) = a_0 - 2a_0^2. \tag{22}$$

Here $a_0 = \frac{1}{3}$ as specified by the initial value of y at $x = 0$. Evaluation of the inner product of (22) with the polynomials $B_i(x)$ gives the system of equations

$$(D + C + B)A = G, \tag{23}$$

where an element of A is a_i and the elements of D, C and B are given by

$$d_{i,k} = \int_0^R B'_{i,n}(x) B_{k,n}(x)\, \mathrm{d}x,$$

Table 1
The initial values of the unknown expansion coefficients a_i with $N=10$ B-polynomials which have been obtained from Eqs. (26) to (27), and the final values of the expansion coefficients obtained from (23) to (25)

Coefficient	Initial a	Final a
1	0.02222	0.02222
2	0.042798	0.042798
3	0.061865	0.061867
4	0.079548	0.077561
5	0.095958	0.091545
6	0.11119	0.10344
7	0.12536	0.1135
8	0.13853	0.12195
9	0.15078	0.12904
10	0.16219	0.13497

$$c_{i,k} = (4a_0 - 1)\int_0^R B_{i,n}(x)B_{k,n}(x)\,\mathrm{d}x, \tag{24}$$

$$b_{i,k} = 2\sum_j a_j \int_0^R B_{j,n}(x)B_{i,n}(x)B_{k,n}(x)\,\mathrm{d}x.$$

The column matrix G has elements given by

$$g_i = (a_0 - 2a_0^2)\int_0^R B_{i,n}(x)\,\mathrm{d}x. \tag{25}$$

The interval over which these are calculated is $[0, 2]$ The dependence of $b_{i,k}$ on the summation of the unknown coefficients a_j manifest the nonlinearity of the problem. The initial values of these coefficients a_i were obtained by applying the Galerkin method to the initial data and neglecting the nonlinear term in (19)

$$(D + C)A = G, \tag{26}$$

whose matrices are constructed from

$$d_{i,k} = \int_0^R B'_{i,n}(x)B_{k,n}(x)\,\mathrm{d}x,$$

$$c_{i,k} = -\int_0^R B_{i,n}(x)B_{k,n}(x)\,\mathrm{d}x,$$

$$g_i = a_0\int_0^R B_{k,n}(x)\,\mathrm{d}x. \tag{27}$$

Once the initial values of the a_i are obtained, they are substituted into Eq. (23) to obtain new estimates for the a_i. This iteration process continues until the converged values of the unknowns are obtained. The typical run gave the initial values of the unknown coefficients which are given in Table 1 in the second column, and the final values of the coefficients given in the third column.

After eight iterations of (23)–(24), values for the coefficients which had converged were used to construct the solution using Eq. (21) for the nonlinear differential equation (19). The error was of the order 10^{-10} between the exact and approximate solutions to the nonlinear equation. It is hoped that the method can be extended to other types of nonlinear equation in future. In Fig. 5, a plot of the absolute difference between approximate and exact solutions is given. In Fig. 6 for Example 4, it is also shown that the convergence of the absolute difference between exact and approximate solutions improved as the number of B-polys were increased from 3 through 10.

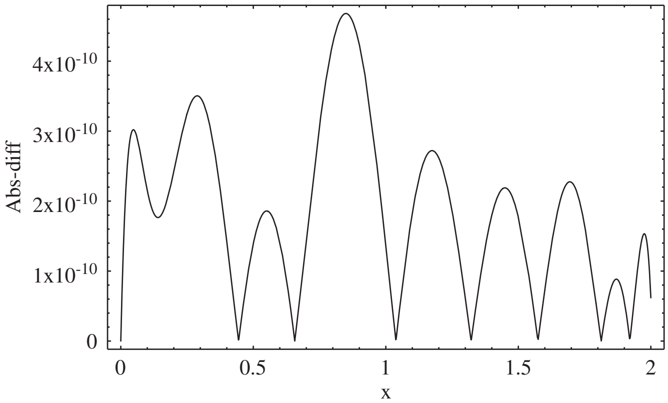

Fig. 5. A plot of absolute difference between exact and approximate solutions for nonlinear case 4 is shown. The absolute difference is obtained using a basis set of 10 B-polynomials continuous on the interval [0, 2].

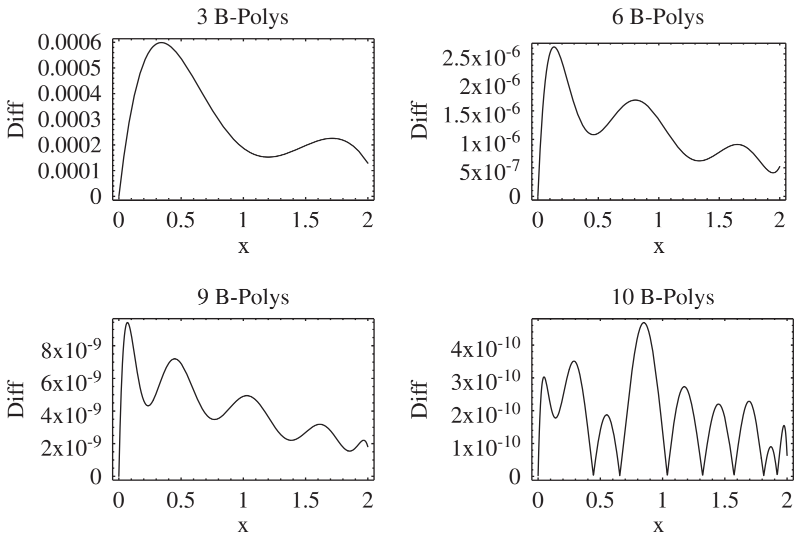

Fig. 6. Plots of absolute difference between exact and approximate solutions for nonlinear case 4 are shown to display the convergence of the solution as the number of B-polys are increased from 3 to 10 in the interval [0, 2].

4. Results and discussion

We have provided a detailed algorithmic description of how B-polynomial bases may be used to give highly accurate solutions to differential equations. To demonstrate this algorithm, we have laid down a procedure and implemented it to solve a few complicated examples with boundary conditions at the origin and at $x = R$. In each of the four cases, we compared the exact solutions with the approximate solutions obtained using the Galerkin method, and agreement was obtained to nearly 10^{-15} figures, as shown in Figs. 2–4. It has also proved effective in producing a solution to a simple nonlinear equation in Example 4. We have also noticed that by nearly doubling the number (22–45) of B-polynomials to approximate the solution nearly triples the accuracy of the desired solution. This happens in the linear case but not in the nonlinear case. In the linear case one is including more powers of x while keeping the linear structure of the equations to be solved. This probably has something to do with it. The basic shapes of the first 10 B-polynomials are displayed in Fig. 1. This merely explains how these polynomials may be added to represent complicated functions. All of these calculations and analytic integrations were carried out using a Mathematica program [9].

The current procedure, which makes use of a new form of continuous B-polynomials, may offer great promise over the piecewise B spline method for solving differential equations. We have shown that the new B-polynomial method will return a valid solution for a differential equation and is a powerful tool that we may utilize to overcome the difficulties associated with complex atomic and molecular systems, where there are no exact solutions available, with less computational effort. The B-polynomial method can probably be extended to the calculation of other atomic properties such as energies, many-body perturbation terms with infinite sums, and transition amplitudes [10].

References

[1] M.I. Bhatti, K.D. Coleman, W.F. Perger, Static polarizabilities of hydrogen in B splines basis, Phys. Rev. A 68 (2003) 044503-1.
[2] C. Bottcher, M.R. Strayer, Relativistic theory of fermions and classical field on a collocation lattice, Ann. Phys. (NY) 175 (1987) 64.
[3] C. Fischer, W. Guo, Spline algorithms for the Hartree–Fock equation for the helium ground state, J. Comput. Phys. 90 (1990) 486.
[4] C. Fischer, M. Idrees, Spline algorithms for continuum functions, Comput. Phys. 3 (1998) 53.
[5] C.W.A. Fletcher, Computational Galerkin Methods, Springer, New York, 1984.
[6] B.R. Gelbaum, Modern Real and Complex Analysis, Wiley, New York, 1995.
[7] W.R. Johnson, S.A. Blundell, J. Saperstein, Finite basis sets for the Dirac equation constructed from B-splines, Phys. Rev. A 37 (1988) 307.
[8] W.R. Johnson, M. Idrees, J. Saperstein, Second-order energies and third-order matrix elements of alkali atoms, Phys. Rev. 35 (1987) 3218.
[9] Mathematica, Wolfram Research Inc. 2004.
[10] W.F. Perger, M. Xia, K. Flurchick, M.I. Bhatti, Comput. Sci. Eng. 3 (2001) 38.
[11] Y. Qiu, C.F. Fischer, Integration by cell algorithm for Slater integrals in a spline basis, J. Comput. Phys. 156 (1999) 257.
[12] J. Saperstein, W.R. Johnson, The use of basis splines in theoretical atomic physics, J. Phys. B 29 (1996) 5213.