Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 192 (2006) 310–319

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Approximations of Sturm–Liouville eigenvalues using differential quadrature (DQ) method

Uğur Yücel*

Department of Mathematics, Faculty of Science and Art, Pamukkale University, Denizli 20020, Turkey

Received 25 September 2004; received in revised form 2 May 2005

Abstract

The polynomial-based differential quadrature (PDQ) and the Fourier expansion-based differential quadrature (FDQ) methods are applied in this work to compute eigenvalues of the Sturm–Liouville problem. It is demonstrated through some examples that accurate results for the first kth eigenvalues of the problem, where $k = 1, 2, 3, \ldots$, can be obtained by using (at least) $2k$ mesh points. The results of this work are compared with other published results in the literature.

Keywords: Eigenvalues; Differential quadrature method; Schrödinger equation; Sturm–Liouville problem

1. Introduction

Problems in the fields of elasticity and vibration, including applications of the wave equations of modern physics, fall into a special class of boundary-value problems known as characteristic-value problems. Certain problems in statics also reduce to such problems.

In this work, we consider a special type of boundary-value problem called Sturm–Liouville problem. A general Sturm–Liouville problem consists of the linear homogeneous second-order differential equation of the form

$$[p(x)y']' + [q_1(x) + \lambda r(x)]y = 0, \quad a \leqslant x \leqslant b, \tag{1}$$

* Tel.: +90 533 439 7409; fax: +90 258 212 5546.
E-mail address: uyucel@pamukkale.edu.tr.

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.05.008

with the boundary conditions

$$a_0 y(a) + a_1 y'(a) = 0,$$
$$b_0 y(b) + b_1 y'(b) = 0. \tag{2}$$

The general Sturm–Liouville problem (1) can be easily reduced to the so-called Liouville normal form (or equivalently, the one-dimensional, time independent form of the all-pervading equation of physics, the Schrödinger equation)

$$-y'' + q(x)y = \lambda y. \tag{3}$$

Here, we consider the above differential equation with essential boundary conditions of the type

$$y(0) = y(\pi) = 0. \tag{4}$$

In most cases, it is not possible to obtain the eigenvalues of the above problem analytically. However, there are various approximate methods as, for example, the weighted residual methods, the variational methods, finite difference and finite elements methods.

Pryce [9] has provided an excellent review of the mathematical back round of Sturm–Liouville problems and their numerical solution, as well as a detailed discussion of applications, till approximately 1993. He provides examples of Sturm–Liouville problems that have been considered by numerous authors. One particular benchmark problem is the set of Paine problems, which are the Schrödinger equations of the form (3), with two choices, $q(x) = \mathrm{e}^x$ and $q(x) = (x + 0.1)^{-2}$. A listing, to 11 significant figures, of the first 50 eigenvalues of these two choices is also given in his book.

There is a continued interest in the numerical solution of Sturm–Liouville problems and associated Schrödinger equations with the aim to improve convergence rates and ease of implementation of different algorithms.

In order to obtain more efficient numerical results, several ways have been devised in the last years [3,4,1,2,7,8,14,15]. For instance, Paine et al. [8] and Andrew and Paine [3] used some traditional difference schemes followed by an asymptotic correction technique. Vanden Berghe and De Meyer [14] have developed special two-step methods producing very accurate results. If more than one eigenvalue is required, the method of [14] requires more computational effort for a given N, number of grid points, than the method of [3], as the method of [14] uses different finite difference equations for the different eigenvalues. Vanden Berghe et al. [15] showed that this disadvantage could be reduced by using parallel computing, though it must be remembered that the method of [3] can also be speeded up considerably if parallel computing facilities are used. Ghelardoni [7] proposed some linear multistep methods, called boundary value methods, for discretizing a Sturm–Liouville problem and extended the correction techniques of [3] and [8] to these new methods. For a given N, the method of [7] requires even more computing effort than the method of [14], though as noted in [7], the potential exist for modifications to make that method more efficient also.

To our knowledge there is no study on the differential quadrature (DQ) applications to Sturm–Liouville problems. On the other hand, the DQ method is an efficient discretization technique for obtaining accurate numerical solutions using a considerably small number of grid points. Bellman et al. [5] introduced the method in the early seventies for solving linear and nonlinear partial differential equations. And then it was improved by [10,13,12]. The DQ method also has been shown to perform extremely well in solving initial and boundary value problems [6].

In the DQ method, derivatives of a function with respect to a coordinate direction are expressed as linear weighted sums of all the functional values at all mesh (grid) points along that direction. The weighting coefficients in that weighting sum are determined using test functions. Among the many kinds of test functions, the Lagrange interpolation polynomial is widely employed since it has no limitation on the choice of the grid points. This leads to polynomial-based differential quadrature (PDQ) which is suitable in most engineering problems. For problems with periodic behaviours, polynomial approximation may not be the best choice for the true solution. In contrast, Fourier series expansion can be the best approximation giving the Fourier expansion-based differential quadrature (FDQ). The ease for computation of weighting coefficients in explicit formulations [13,11] for both cases is based on the analysis of function approximation and linear vector space.

In this paper, both the PDQ and FDQ methods are used for obtaining eigenvalues of the considered Sturm–Liouville problem. The DQ method is summarized in Section 2. Applications to the Sturm–Liouville problems, numerical examples used by several authors [3,7,14], are given in Section 3 and results are summarized in Section 4.

2. Differential quadrature method

DQ method was presented for the first time by [5] as mentioned before for solving differential equations. The DQ method uses the basis of the quadrature method in deriving the derivatives of a function. It follows that the partial derivative of a function with respect to a space variable can be approximated by a weighted linear combination of function values at some intermediate points in that variable.

In order to show the mathematical representation of DQ method, we consider a single variable function $f(x)$ on the domain $a \leqslant x \leqslant b$; then the nth order derivative of the function $f(x)$ at an intermediate point (grid point) x_i can be written as

$$\left.\frac{\mathrm{d}^n f}{\mathrm{d}x^n}\right|_{x=x_i} = \sum_{j=1}^{N} r_j^{(n)}(x_i) f(x_j), \quad i = 1, 2, \ldots, N, \quad n = 1, 2, \ldots, N-1, \tag{5}$$

where N is the number of grid points in the whole domain ($a = x_1, x_2, \ldots, x_N = b$) and $r_j^{(n)}(x_i)$ are the weighting
coefficients of the nth derivative. As it can be seen from (5), two important factors control the quality of the approximation resulting from the application of DQ method. These are the values of weighting coefficients and the positions of the discrete variables. Once the weighting coefficients are determined, the bridge to link the derivatives in the governing differential equation and the functional values at the mesh points is established. In other words, with the weighting coefficients, one can easily use the functional values to compute the derivatives. Note that for multidimensional problems each derivative is approximated in the respective direction similarly.

In order to determine the weighting coefficients in (5), $f(x)$ must be approximated by some test functions. To select a suitable test function, one needs to satisfy the following conditions:

(i) Differentiability: The test function of the differential equation must be differentiable at least up to the nth derivative (here, n is the highest order of the differential equation).
(ii) Smoothness: $f(x)$ must be sufficiently smooth to allow one to write (i).

2.1. Polynomial-based differential quadrature (PDQ)

When the function $f(x)$ is approximated by a higher order polynomial, Shu [11] and Shu and Richards [13] presented some explicit formulations to compute the weighing coefficients within the scope of a higher order polynomial approximation and a linear vector space. It is supposed that the solution of a one-dimensional differential equation is approximated by a $(N-1)$th degree polynomial

$$f(x)=\sum_{k=0}^{N-1} c_k x^k. \tag{6}$$

If $r_k(x),\ k=1,2,\ldots,N$, are the base polynomials in V_N (N-dimensional linear vector space), then $f(x)$ can be expressed by

$$f(x)=\sum_{k=1}^{N} d_k r_k(x). \tag{7}$$

Here, the base polynomials $r_k(x),\ k=1,2,\ldots,N$, are chosen as the Lagrange interpolated polynomials

$$r_k(x)=\frac{M(x)}{(x-x_k)M^{(1)}(x_k)}, \tag{8}$$

where

$$M(x)=\prod_{j=1}^{N}(x-x_j),$$
$$M^{(1)}(x_k)=\prod_{\substack{j=1\\ j\neq k}}^{N}(x_k-x_j), \tag{9}$$

and $x_i,\ i=1,2,\ldots,N$, are the coordinates of grid points which may be chosen arbitrarily. Substituting (7) into (5) and using (8) results in the following weighting coefficients for the first- and second-order derivatives

$$r_j^{(1)}(x_i)=\frac{M^{(1)}(x_i)}{(x_i-x_j)M^{(1)}(x_j)} \quad \text{for } j\neq i, \quad i,j=1,2,\ldots,N,$$
$$r_i^{(1)}(x_i)=-\sum_{\substack{j=1\\ i\neq j}}^{N} r_j^{(1)}(x_i), \tag{10}$$

$$r_j^{(2)}(x_i)=2r_j^{(1)}(x_i)\left(r_i^{(1)}(x_i)-\frac{1}{x_i-x_j}\right) \quad \text{for } j\neq i, \quad i,j=1,2,\ldots,N,$$
$$r_i^{(2)}(x_i)=-\sum_{\substack{j=1\\ i\neq j}}^{N} r_j^{(2)}(x_i). \tag{11}$$

It can be seen from the above equations that the weighting coefficients of the second-order derivative can be completely determined from those of the first-order derivative.

2.2. Fourier expansion-based differential quadrature (FDQ)

When the function $f(x)$ is approximated by a Fourier series expansion in the form

$$f(x) = c_0 + \sum_{k=1}^{N/2} \left(c_k \cos \frac{k\pi x}{L} + d_k \sin \frac{k\pi x}{L} \right), \tag{12}$$

and Lagrange interpolated trigonometric polynomials are taken as

$$r_k(x) = \frac{\sin(x - x_0)/2L\pi \ldots \sin(x - x_{k-1})/2L\pi \sin(x - x_{k+1})/2L\pi \ldots \sin(x - x_N)/2L\pi}{\sin(x_k - x_0)/2L\pi \ldots \sin(x_k - x_{k-1})/2L\pi \sin(x_k - x_{k+1})/2L\pi \ldots \sin(x_k - x_N)/2L\pi}, \tag{13}$$

the FDQ approach is obtained for $f(x)$ and its first- and second-order derivatives. So, the weighting coefficients $r_j^{(n)}(x_i)$, $(n = 1, 2)$ to be determined by the FDQ method are given by

$$\begin{aligned} r_j^{(1)}(x_i) &= \frac{\pi}{2L} \frac{P(x_i)}{\sin(x_i - x_j)\pi/2LP(x_j)} \quad \text{for } j \neq i, \quad i, j = 1, 2, \ldots, N, \\ r_i^{(1)}(x_i) &= -\sum_{\substack{j=1 \\ i \neq j}}^{N} r_j^{(1)}(x_i), \end{aligned} \tag{14}$$

$$\begin{aligned} r_j^{(2)}(x_i) &= r_j^{(1)}(x_i) \left(2r_i^{(1)}(x_i) - \frac{\pi}{L} \mathrm{ctg} \frac{x_i - x_j}{2L} \pi \right) \quad \text{for } j \neq i, \quad i, j = 1, 2, \ldots, N, \\ r_i^{(2)}(x_i) &= -\sum_{\substack{j=1 \\ i \neq j}}^{N} r_j^{(2)}(x_i), \end{aligned} \tag{15}$$

where L is the length of the interval (physical domain) and

$$P(x_i) = \prod_{\substack{j=0 \\ j \neq i}}^{N} \sin \frac{x_i - x_j}{2L} \pi. \tag{16}$$

2.3. Choice of the grid point distributions

The selection of locations of the sampling points plays a significant role in the accuracy of the solution of the differential equations. Using equally spaced points (uniform grid) can be considered to be a convenient and easy selection method. For a domain specified by $a \leqslant x \leqslant b$ and discretized by N points, then the coordinate of any point i can be evaluated by

$$x_i = a + \frac{i-1}{N-1}(b - a). \tag{17}$$

Quite frequently, the DQ method delivers more accurate solutions with a set of unequally spaced points (nonuniform grid). The so-called Chebyshev–Gauss–Lobatto points, which were first used by [13] and whose advantage has been discussed by [6], are well accepted in the DQ method as follows:

$$x_i = a + \frac{1}{2}\left(1 - \cos\left(\frac{i-1}{N-1}\pi\right)\right)(b-a), \tag{18}$$

for a domain $a \leqslant x \leqslant b$ again.

2.4. Implementation of boundary conditions

Proper implementation of the boundary conditions is also very important for the accurate numerical solution of differential equations. Essential and natural boundary conditions can be approximated by DQ method. Using the DQ method for solving differential equations, we actually satisfy the governing equations at each sampling point of the domain, so we have one equation for each point, for each unknown. To satisfy the boundary conditions, at the boundary points, the boundary condition equations are satisfied instead of the governing equations. In other words, in the resulting system of algebraic equations from DQ method, each boundary condition replaces the corresponding field equation. This procedure is straightforward when there is one boundary condition at each boundary and when we have distributed the sampling points so that there is one point at each boundary.

3. Applications to Sturm–Liouville problems

The DQ method has been applied to solve the following regular Sturm–Liouville problem:

$$\begin{aligned} &-y'' + q(x)y = \lambda y, \quad 0 \leqslant x \leqslant \pi, \\ &y(0) = y(\pi) = 0. \end{aligned} \tag{19}$$

By applying the PDQ or FDQ method, (19) can be discretized as

$$-\sum_{j=1}^{N} r_j^{(2)}(x_i)y(x_j) + q(x_i)y(x_i) = \lambda y(x_i), \quad i = 1, 2, \ldots, N, \tag{20}$$

where N is the number of grid points in the x-direction ($0 = x_1, x_2, \ldots, x_N = \pi$) and $r_j^{(2)}(x_i)$ the weighting coefficients. When the PDQ method is used, $r_j^{(2)}(x_i)$ are computed by Eqs. (10)–(11), while for the FDQ approach, the weighting coefficients $r_j^{(2)}(x_i)$ are computed by Eqs. (14)–(16). It is obvious that the Dirichlet boundary conditions (homogeneous in our problem) can be directly substituted into (20). Then, applying (20) at all the interior points leads to the following $(N-2) \times (N-2)$ eigenvalue equation system:

$$[A]\{y\} = \lambda\{y\}. \tag{21}$$

From the above equation, the λ values can be obtained from the eigenvalues of matrix $[A]$. This can be done by using various methods. Here, we use a FORTRAN IMSL Routine called DEVLRG. Routine DEVLRG computes the eigenvalues of a real matrix. The matrix is first balanced. Elementary or Gauss

Table 1
Relative errors of the DQ results with $q(x)=0$ for several values of the number of grid points N

k	ε_k $(N=40)$		ε_k $(N=50)$	
	PDQ	FDQ	PDQ	FDQ
1	7.95×10^{-14}	8.88×10^{-15}	4.06×10^{-14}	8.99×10^{-14}
2	2.29×10^{-14}	1.33×10^{-14}	1.78×10^{-14}	4.11×10^{-15}
3	8.29×10^{-15}	2.96×10^{-15}	2.17×10^{-15}	3.55×10^{-15}
4	2.44×10^{-15}	2.22×10^{-15}	3.55×10^{-15}	1.78×10^{-15}
5	5.68×10^{-16}	2.13×10^{-15}	1.99×10^{-15}	2.84×10^{-16}
6	$0.00\times10^{+00}$	1.18×10^{-15}	3.55×10^{-15}	5.92×10^{-16}
7	1.42×10^{-19}	1.42×10^{-19}	2.03×10^{-15}	1.74×10^{-15}
8	4.44×10^{-16}	4.44×10^{-16}	1.55×10^{-15}	5.55×10^{-16}
9	1.76×10^{-16}	2.63×10^{-15}	3.51×10^{-16}	3.51×10^{-16}
10	1.71×10^{-15}	2.84×10^{-16}	4.26×10^{-16}	2.27×10^{-15}
11	5.90×10^{-14}	3.41×10^{-15}	1.29×10^{-15}	2.35×10^{-16}
12	6.34×10^{-13}	6.32×10^{-15}	3.95×10^{-16}	1.58×10^{-15}
13	1.66×10^{-11}	9.59×10^{-15}	6.73×10^{-16}	1.68×10^{-15}
14	1.28×10^{-10}	4.64×10^{-15}	1.45×10^{-16}	1.31×10^{-15}
15	1.61×10^{-09}	1.28×10^{-14}	2.53×10^{-15}	4.04×10^{-15}
16	9.90×10^{-09}	7.44×10^{-14}	3.89×10^{-14}	5.55×10^{-15}
17	7.17×10^{-08}	2.95×10^{-12}	8.01×10^{-13}	4.72×10^{-15}
18	1.52×10^{-07}	2.19×10^{-11}	5.72×10^{-12}	7.02×10^{-16}
19	1.45×10^{-06}	1.57×10^{-09}	6.67×10^{-11}	1.10×10^{-15}
20	3.76×10^{-05}	4.56×10^{-08}	3.93×10^{-10}	1.42×10^{-16}

similarity transformations with partial pivoting are used to reduce this balanced matrix to a real upper Hessenberg matrix. A hybrid double-shifted LR-QR algorithm is used to compute the eigenvalues of the Hessenberg matrix.

To demonstrate the efficiency and accuracy of the DQ method, the sample problem, i.e., $q(x)=0$, which has exact eigenvalues, namely $\lambda_k=k^2,\ k=1,2,3,\ldots,$ is first choosen for study. Both the PDQ and FDQ methods with the grid points distribution given by (18) are applied. The performance of the DQ approach is measured by the relative error ε_k which is defined as

$$\varepsilon_k=\left|\frac{\lambda_k-\lambda_k^{(\mathrm{DQ})}}{\lambda_k}\right|,\qquad k=1,2,3,\ldots, \tag{22}$$

where $\lambda_k^{(\mathrm{DQ})}$ indicates kth algebraic eigenvalues obtained by DQ method and $\lambda_k=k^2$, $k=1,2,3,\ldots$ are the exact eigenvalues.

Table 1 lists the relative errors of the DQ results with different number of mesh points N. Here, it is interesting to note that in order to have good approximations to the first kth eigenvalues, at least $2k$ grid points have to be used. In other words, $N=2k$. It is observed from Table 1 that the accuracy of the computed eigenvalues by the FDQ method is better than the PDQ approach. This is especially true for the higher eigenvalues. In both methods, the computed values for the lower eigenvalues have a better

Table 2
Comparison of errors in computed solutions for $q(x) = \mathrm{e}^x$

k	Λ_k	$\mathrm{e}_k = (\tilde{\Lambda}_k^{(h)} - \Lambda_k) \times 10^3$		$\mathrm{e}_k = (\tilde{\Lambda}_k^{(h)} - \Lambda_k) \times 10^3$		
		PDQ	FDQ	US	CN	VD2
1	4.8966694	0.0000	0.0000	−0.0139	−0.0026	0.0000
2	10.045190	0.0000	0.0000	−0.0753	−0.0326	0.0000
3	16.019267	0.0000	0.0000	−0.2940	−0.1111	
4	23.266271	0.0000	0.0000	−0.4927	−0.2318	0.0000
5	32.263707	0.0000	0.0000	−1.1286	−0.3878	
6	43.220020	0.0000	0.0000	−1.6781	−0.5823	0.0027
7	56.181594	0.0000	0.0000	−0.7448	−0.8158	
8	71.152998	0.0000	0.0000	−5.7267	−1.0917	−0.0138
9	88.132119	0.0000	0.0000	−5.7250	−1.4106	
10	107.11668	0.0000	0.0000	−2.2789	−1.7816	−0.0290
11	128.10502	0.0000	0.0000	−23.286	−2.1949	
12	151.09604	0.0000	0.0000	−21.656	−2.6672	−0.0673
13	176.08900	0.0000	0.0000	−10.082	−3.2113	
14	203.08337	0.0000	0.0000	−68.490	−3.8143	−0.0103
15	232.07881	0.0000	0.0000	−82.532	−4.4995	
16	263.07507	−0.010	0.0000	−26.684	−5.2801	−0.1343
17	296.07196	0.0200	0.0000	−117.02	−6.1622	
18	331.06934	0.0500	0.0000	−254.16	−7.1575	−0.1434
19	368.06713	0.5600	0.0000	−118.14	−8.3085	
20	407.06524	−16.01	−0.030	−51.722	−9.6295	−0.0538

accuracy than those for the higher eigenvalues. As the number of grid points further increased to above $2k$, the accuracy of the DQ results especially for the higher eigenvalues can be further improved as shown in Table 1.

In order to facilitate comparison with the results of [3,7,14], we also choose the same functions $q(x)$ for our numerical examples, i.e., $q(x) = \mathrm{e}^x$ and $q(x) = (x + 0.1)^{-2}$, and use the absolute error, to agree with their notations, defined by

$$\mathrm{e}_k = (\tilde{\Lambda}_k^{(h)} - \Lambda_k) \times 10^3, \quad k = 1, 2, 3, \ldots, \tag{23}$$

where Λ_k is the reference values taken from [7] and $\tilde{\Lambda}_k^{(h)}$ is the computed eigenvalues.

In Table 2, we compare the errors in computed solutions of the present work for $q(x) = \mathrm{e}^x$ with the uniderivative Simpson method [7], called US, the corrected Numerov method [3], called CN, and the method of [14] called VD2.

As a first step, we compare the results obtained by the various methods using the same N, namely $N = 40$. However, it is important to remember that, for a given N, the amount of work required by the DQ methods is much greater than that required by the methods of [3] and [14], and even (to a lesser extent) the method of [7]. The matrices arising in the methods of [3] and [14] are symmetric tridiagonal $(N-1) \times (N-1)$ matrices, whereas the DQ methods produce dense, nonsymmetric $(N-2) \times (N-2)$ matrices for the problems considered here. This tridiagonal structure greatly reduces the amount of time required for solution of the eigenvalue problem. The method of [7] also produces banded matrices.

Table 3
Comparison of errors in computed solutions for $q(x) = (x + 0.1)^{-2}$

k	Λ_k	$e_k = (\tilde{\Lambda}_k^{(h)} - \Lambda_k) \times 10^3$		$e_k = (\tilde{\Lambda}_k^{(h)} - \Lambda_k) \times 10^3$		
		PDQ	FDQ	US	CN	VDI
1	1.5198658	0.0000	0.0000	−0.0868	−0.0462	−0.023
2	4.9433098	0.0000	0.0000	−0.5548	−0.2931	−0.166
3	10.284663	0.0000	0.0000	−1.7289	−0.9037	
4	17.559958	0.0000	0.0000	−3.8803	−2.0115	−1.381
5	26.782863	0.0000	0.0000	−7.3116	−3.7173	
6	37.964426	0.0000	0.0000	−12.144	−6.0952	−3.056
7	51.113358	0.0000	0.0000	−18.296	−9.1986	
8	66.236448	0.0000	0.0000	−27.192	−13.071	−8.718
9	83.338962	0.0000	0.0000	−36.928	−17.753	
10	102.42499	0.0000	0.0000	−47.755	−23.289	−15.518
11	123.49771	0.0000	0.0000	−66.529	−29.720	
12	146.55961	0.0000	0.0000	−81.467	−37.101	−23.058
13	171.61264	0.0000	0.0000	−96.132	−45.486	
14	198.65837	0.0000	0.0000	−135.13	−54.968	−30.017
15	227.69803	0.0000	0.0000	−161.78	−65.627	
16	258.73262	0.0000	0.0000	−170.50	−77.573	−34.243
17	291.76293	0.0200	−0.007	−243.50	−90.918	
18	326.78963	0.0500	0.0100	−334.06	−105.82	−33.587
19	363.81325	0.6500	0.0500	−337.51	−122.45	
20	402.83424	−16.89	−0.060	−367.43	−141.02	−25.122

Moreover, the effort required to compute the matrix elements arising in the methods of [3] and [14] is negligible compared with the time required to evaluate expressions such as (15), especially when (18) is used. Also, results of [1] and [2] indicate that the accuracy of the results of the method of [3] can be improved significantly at negligible extra cost by a simple extrapolation technique, whereas the results of our Table 1 suggest that extrapolation may not be practical for the DQ methods. More work is required to determine whether, with the same computational effort (rather than with the same N), the DQ method will be superior to the other methods considered here. However, it is encouraging to note that, at least with the same N, the DQ methods produced considerably more accurate results than the other methods for the examples considered here.

Similar things can be observed from Table 3 which compares the errors in computed solutions for $q(x) = (x + 0.1)^{-2}$. Here, the errors in the last column, called VDI, are also from the method of [14].

4. Conclusions

Both the PDQ and FDQ methods have been applied to compute the eigenvalues of Sturm–Liouville problem. Through test examples which have exact solutions, it was found that in order to obtain accurate numerical results for the first kth eigenvalues of the problem, where $k = 1, 2, 3, \ldots$, the minimum number of grid points N must be equal to $2k$. It was also found that as the number of grid points is further increased to above $2k$, the accuracy of the DQ results for both approaches can be further improved. Generally, the FDQ approach obtains more accurate results than the PDQ approach.

Comparison with other published works in the literature showed that the DQ method produces highly accurate results for the eigenvalues of the Sturm–Liouville problem considered in this work. The method is very effective for the higher-index eigenvalues as well. This was tested through some examples which are not reported here. It will be also interesting to see how the method works for the Sturm–Liouville problem with other type of boundary conditions. This will be considered in a future work.

Acknowledgements

I would like to thank Prof. Peter B. Monk, principal editor, and anonymous referees of JCAM for their valuable comments and suggestions to improve the introduction and presentation of the numerical results at the end of Section 3 of this paper.

References

[1] A.L. Andrew, Asymptotic correction of computed eigenvalues of differential equations, Ann. Numer. Math. 1 (1994) 41–51.

[2] A.L. Andrew, Asymptotic correction of Numerov's eigenvalue estimates with natural boundary conditions, J. Comput. Appl. Math. 125 (2000) 359–366.

[3] A.L. Andrew, J.W. Paine, Correction of Numerov's eigenvalue estimates, Numer. Math. 47 (1985) 289–300.

[4] A.L. Andrew, J.W. Paine, Correction of finite element estimates for Sturm–Liouville eigenvalues, Numer. Math. 50 (1986) 205–215.

[5] R. Bellman, B.G. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations, J. Comput. Phys. 10 (1972) 40–52.

[6] C.W. Bert, M. Malik, Differential quadrature method in computational mechanics: a review, Appl. Mech. Rev. 49 (1996) 1–28.

[7] P. Ghelardoni, Approximations of Sturm–Liouville eigenvalues using Boundary Value Methods, Appl. Numer. Math. 23 (1997) 311–325.

[8] J.W. Paine, F.R. de Hoog, R.S. Anderssen, On the correction of finite difference eigenvalue approximations for Sturm–Liouville problems, Computing 26 (1981) 123–139.

[9] J.D. Pryce, Numerical Solution of Sturm–Liouville Problems, Oxford University Press, Oxford, 1993.

[10] C. Shu, Generalized differential-integral quadrature and application to the simulation of incompressible viscous flows including parallel computation, Ph.D. Thesis, University of Glasgow, UK, 1991.

[11] C. Shu, Differential Quadrature and Its Application in Engineering, Springer, Berlin, 2000.

[12] C. Shu, B.C. Khoo, Y.T. Chew, K.S. Yeo, Numerical studies of unsteady boundary layer flows past an impulsively started circular cylinder by GDQ and GIQ approaches, Comput. Methods Appl. Mech. Engrg. 135 (1996) 229–241.

[13] C. Shu, B.E. Richards, Application of generalized differential quadrature to solve two-dimensional incompressible Navier–Stokes equations, Internat. J. Numer. Methods Fluids 15 (1992) 791–798.

[14] G. Vanden Berghe, H. De Meyer, Accurate computation of higher Sturm–Liouville eigenvalues, Numer. Math. 59 (1991) 243–254.

[15] G. Vanden Berghe, H. De Meyer, M. Van Daele, A parallel approach to the modified Numerov-like eigenvalue determination for Sturm–Liouville problems, Comput. Math. Appl. 23 (12) (1992) 69–74.