Computers and Mathematics with Applications 62 (2011) 3183–3193

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

A note on three numerical procedures to solve Volterra integro-differential equations in structural analysis

S. Tomasiello *

DiSGG — Faculty of Engineering, University of Basilicata, C.da Macchia Romana, 85100 Potenza, Italy

ARTICLE INFO

Article history:
Received 27 November 2010
Received in revised form 5 August 2011
Accepted 8 August 2011

Keywords:
Iterative algorithms
Integro-differential equations
Grid points
Lagrange polynomials
Tubular tapered cantilever beam

ABSTRACT

In this paper, three numerical methods to solve Volterra integro-differential equations containing rational functions are discussed. The first one is the Differential Quadrature Method and, to the best knowledge of the author, it has never been applied to this kind of problem; the second one is a new version of the Iterative Differential Quadrature method, a method proposed by the author some years ago to solve problems in space–time domains, revised herein for the single space variable problem; the third one is a numerical Picard-like method, proposed herein to combine successive approximations with numerical integration. Stability and convergence of the second and the third method are discussed. The three methods have been applied to solve a real world problem in the field of the structural engineering and the numerical results compared.

1. Introduction

In this paper, the following integro-differential equation is considered

$$\frac{\mathrm{d}^r y}{\mathrm{d}z^r} + \beta(z)y(z) + \alpha(z)\int_0^z \gamma(z)y(z)\mathrm{d}z = q(z), \quad z\epsilon\,[0,h] \tag{1}$$

with $r \geq 2$ and with the boundary conditions

$$\frac{\mathrm{d}^i y}{\mathrm{d}z^i}(h) = a_i, \quad i = 0,\ldots,r-1$$

where a_i are real constants, $\alpha(z)$ and $q(z)$ are real continuous functions, $\beta(z)$ and $\gamma(z)$ are rational functions.

Many problems in science and engineering involve integro-differential equations often requiring efficient solution methods, e.g. heat conduction in materials with memory, viscoelasticity and reactor dynamics. The problem here considered deals with the bending moment equation for a tubular tapered cantilever beam, which depending on the load conditions, can represent a typical model for wind towers or for chimneys. The mathematical model is given by a second order Volterra integro-differential equation containing rational functions as expressed by Eq. (1). With regard to the numerical solution of general Volterra integro-differential equations, the current literature discuss the use of spline collocation methods (see [1,2]) or finite element methods [3]. In this paper, the aim is to investigate the use of methods not previously discussed or proposed. The first one is the Differential Quadrature Method (DQM). In recent years, the DQM has become an increasingly popular numerical technique for the rapid and efficient solution of a variety of science and engineering problems. The method was

* Tel.: +39 0 971 205089; fax: +39 0 971 205070.
E-mail addresses: tomasiello@unibas.it, stefania.tomasiello@gmail.com.

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.08.031

first developed by Bellman and Casti in the early 1970s [4] and can be regarded as a powerful alternative to the conventional low order Finite Difference Method (FDM) and Finite Element Method (FEM), since it yields very accurate numerical results using a considerably smaller number of grid points and hence requiring relatively little computational effort. So far hundreds of papers devoted to the DQM or its variants have been published. In 1996, Bert and Malik presented a comprehensive review of the chronological development and application of the DQM up to that year [5]. In 2000 appeared the book by Shu, describing systematically both the theoretical analysis and the application of the method [6]. A critical survey of the method according to the current achievements can be retrieved in [7].

The idea of the DQM is that, chosen some grid (or sampling) points in a direction (i.e. a grid line), the derivative at a grid point along a grid line is approximated by a linear weighted sum of the functional values at all the grid points along the grid line.

The second numerical method here discussed is the Iterative Differential Quadrature (IDQ) method, a method proposed by the author some years ago [8,9] to solve problems in space–time domains, by applying the same quadrature rules used in the DQM both in space and in time and computing the solution for a certain number M of subdomains one by one, i.e. by retrieving each time the initial conditions by the immediately preceding calculus step. This method is here proposed for the first time in a new version, named the Modified Iterative Differential Quadrature (MIDQ) method, to solve a single space variable problem. It should be pointed out that the method is different from the FEM, since, despite of the subdomains, the method does not require an assembly step.

Finally, the third method is a Picard-like method, i.e. a numerical adaptation of the Picard method which combines successive approximations with numerical integration.

Although the method of successive approximations can be used as a semi-analytical approach by computing the integrals exactly, some complications may arise when there are rational functions, even for the linear case, since computations become very time-expensive. To overcome this drawback, the r-fold inverse operator applied to the rth-order differential equation (1) is replaced by a weighted sum of the integrand values at N grid points, where the weights are obtained by integrating r times the Lagrange interpolating polynomials.

As one can see, all the three approaches use the Lagrange interpolating polynomials but differently, since the first one uses them to obtain a fully discretized system, the second one uses them to approximate solution by piecewise polynomials computed recursively one by one and the third one uses them to introduce the numerical integration in a recursive approximation scheme.

A comparison between the results obtained by the numerical methods here considered and the ones obtained by the standard FEM is proposed: differently from the FEM all the methods here considered present the advantage of allowing a one-dimensional model, by solving the problem by means of a small number of grid points. The three methods behave differently depending on the load conditions: for all the load conditions considered herein, the best results seem to be allowed by the MIDQ method.

It should be pointed out that, especially with regard to DQ based methods, the aim of the paper is not to propose a methodology for retrieving an optimized set of sampling points (this is the subject of an ongoing research) but to examine possible range of application of these methods to the problem to be solved, by means of a relatively small number of grid points. Finally, it is worth noting that the proposed methods are different from other iterative methods applied to nonlinear problems (e.g. see [10]).

The paper is structured in three theoretical sections in order to give an overview of the methods, also dealing with stability and convergence, followed by a section where the mathematical model is described and a section devoted to numerical results and discussion.

2. The Differential Quadrature Method

The DQM approximates the rth order derivative of a function $y(z)$ by a linear weighted sum of the functional values at given grid points along the grid line.

By writing the function $y(z)$ as:

$$y(z) = \sum_{k=1}^{N} l_k(z) y_k$$

where $l_k(z)$ are the Lagrange interpolation functions at the points z_k, the rth-order z-derivative of the function $y(z)$ at a point $z = z_i$ may be written as

$$\left[\frac{\mathrm{d}^r y}{\mathrm{d}z^r}\right]_{z=z_i} = \sum_{k=1}^{N} \frac{\mathrm{d}^r L_k}{\mathrm{d}z^r}(z_i) = \sum_{k=1}^{N} A_{ik}^{(r)} y_k \quad i = 1, 2, \ldots, N \tag{2}$$

where $A_{ik}^{(r)}$ are the weighting coefficients to be computed as follows [6].

The off-diagonal terms of the weighting coefficient matrix of the first-order derivative turns out to be:

$$A_{ij}^{(1)} = \frac{\prod_{\substack{v=1 \\ v\neq i}}^{N} (z_i - z_v)}{(z_i - z_j) \prod_{\substack{v=1 \\ v\neq j}}^{N} (z_j - z_v)} \quad i, j = 1, 2, \ldots, N \; j \neq i. \tag{3}$$

The off-diagonal terms of the weighting coefficient matrix of the higher-order derivative are obtained through the recurrence relationship:

$$A_{ij}^{(r)} = r \left[A_{ii}^{(r-1)} A_{ij}^{(1)} - \frac{A_{ij}^{(r-1)}}{(z_i - z_j)} \right] \quad i, j = 1, 2, \ldots, N \; j \neq i \tag{4}$$

where $2 \leq r \leq (N-1)$.

The diagonal terms of the weighting coefficient matrix are given by:

$$A_{ii}^{(r)} = -\sum_{\substack{v=1 \\ v\neq i}}^{N} A_{iv}^{(r)} \quad i = 1, 2, \ldots, N \tag{5}$$

where $1 \leq r \leq (N-1)$.

Assuming the Lagrange interpolated polynomials as test functions, there is no restriction in the choice of the grid coordinates. In the current literature, the choice of Gauss–Chebyshev–Lobatto (GCL) points is usual, but in what follows, two different distributions will be considered (not so frequent in the DQM literature): a graded distribution

$$z_i = \left(\frac{i-1}{N-1} \right)^p h, \quad p > 1 \tag{6}$$

and a geometric distribution

$$z_i = b^{N-i} h, \quad 0 < b < 1 \tag{7}$$

with $i = 1, \ldots, N$.

The distribution of the sampling points can also be obtained by computing the zeros of the first order derivative of Gegenbauer polynomials [11].

Gegenbauer polynomials can be seen as a particular case of the Jacobi polynomials. An explicit representation of the ultraspherical polynomials of degree m and order λ, $G_m^{\lambda}(z)$, can be retrieved in [12].

For $\lambda = 1/2$, Gegenbauer polynomials reduce to Legendre polynomials; for $\lambda \to 0$, Gegenbauer polynomials multiplied by λ^{-1} differ in the limit by a constant respect to the Chebyshev polynomials and the shifted Gauss–Chebyshev–Lobatto (GCL) points can be retrieved.

By applying the quadrature rules (2) to Eq. (1), one has the following discretized equations

$$\sum_{j=1}^{N} A_{ij}^{(r)} y_j + \beta_i y_i + \alpha_i \sum_{k=1}^{N} C_k(z_i) \gamma_k y_k = q_i, \quad i = 1, \ldots, N \tag{8}$$

where

$$C_k(z) = \int_0^z l_k(z) \mathrm{d}z$$

being $l_k(z)$ the Lagrange interpolating polynomials and with the boundary conditions

$$y_N = a_0, \qquad \sum_{j=1}^{N} A_{Nj}^{(p)} y_j = a_p, \quad p = 1, \ldots, r-1.$$

Discussion about the error related to a DQ solution can be retrieved in [6]. Numerical results for different distributions of sampling points are presented in Section 6.

3. The Modified Iterative Differential Quadrature Method

In this section, a new modified version of the IDQ method is presented to solve Eq. (1) with $r = 2$, according to the mathematical model presented in Section 5.

By means of this method, the quadrature grid can be regarded as a series of M sub-grids with N points and the solution will be computed successively for each sub-grid. So if $\Delta z^{[i]}$ is the length of the ith subdomain (or interval) one has $h = \sum_{i=0}^{M-1} \Delta z^{[i]}$ and the approximated solution over the entire domain is given by piecewise polynomials

$$y^{[i]} = \sum_{j=1}^{N} V_j^{[i]}(z) y_j^{[i]} \tag{9}$$

being

$$V_j^{[i]}(z) = \delta_{1j} + \sum_{r=1}^{N-1} \frac{A_{1j}^{(r)}}{r!} \left(\frac{z}{\Delta z^{[i]}}\right)^r$$

where $A_{1j}^{(r)}$ are the weighting coefficients computed with regard to a unitary interval.

It should be noted that the solution is computed not only at the end of each interval, but also through it.

In order to apply the MIDQ method, Eq. (1) is conveniently rewritten as follows, by introducing the reference change $\zeta = h - z$

$$\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}\zeta} = -\mathbf{q}(\zeta) + \mathbf{B}(\zeta)\mathbf{w} + \alpha(\zeta) \int_{h-\zeta}^{h} \mathbf{G}(\zeta)\mathbf{w}\mathrm{d}\zeta, \quad \zeta \epsilon [0, h] \tag{10}$$

where $\mathbf{w}^T = (y, \bar{y})$ represents the vector containing the solution values (i.e. solution y and its first-order ζ-derivative $\bar{y}$), $\mathbf{q}^T = (0, q(\zeta))$, being $\mathbf{B}$ and $\mathbf{G}$ 2×2 matrices defined as follows

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ -\beta(\zeta) & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 \\ \gamma(\zeta) & 0 \end{bmatrix}.$$

By dividing the space domain into subdomains and by applying the quadrature rules one has with regard to the ith subdomain (by omitting the index $[i]$ for the sake of simplicity):

$$\sum_{j=1}^{N} \mathbf{C}_{kj}\mathbf{w}_j = \mathbf{q}_k, \quad k = 1, \ldots, N_t \tag{11}$$

where

$$\mathbf{C}_{kj} = -\frac{A_{kj}^{(1)}}{\Delta z}\mathbf{I}_2 + \alpha_k c_j(\bar{\zeta}_k)\mathbf{G}(\bar{\zeta}_j) - \delta_{kj}\mathbf{B}(\bar{\zeta}_j) \tag{12}$$

with

$$c_j(\zeta) = \int_{h-\zeta}^{h} l_j(\zeta)\mathrm{d}\zeta$$

and for $\zeta_k^{[i]} \epsilon [0, \Delta z^{[i]}]$, $k = 1, \ldots, N$

$$\bar{\zeta}_k^{[i]} = \zeta_k^{[i]} + \sum_{p=0}^{i-1} \Delta z^{[p]}, \quad i > 0$$

which becomes

$$\bar{\zeta}_k^{[i]} = \zeta_k^{[i]} + i\Delta z, \quad i = 0, \ldots, M-1$$

if all the intervals have equal length. The elements c_j also depend on the interval length since one can write:

$$c_j(\zeta) = \sum_{r=1}^{N} \bar{b}_r \left(\frac{\zeta}{\Delta z}\right)^r \Delta z$$

where $\bar{b}_r$ are real parameters.

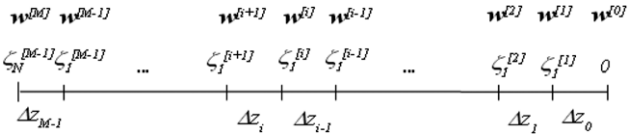

Fig. 1. The subdomains in the MIDQ scheme.

In order to retrieve a recursive formula involving the solution $\mathbf{w}^{[i]}$ at the abscissa $\zeta_1^{[i]}$ and the solution $\mathbf{w}^{[i+1]}$ at the abscissa $\zeta_1^{[i]} + \Delta z^{[i]}$ (see Fig. 1), the unknowns $\mathbf{w}_j$, with $j = 2, \ldots, N-1$ are eliminated and after some algebra one obtains:

$$\mathbf{w}^{[i+1]} = \mathbf{H}^{[i]}\mathbf{w}^{[i]} + \mathbf{Q}^{[i]} \tag{13}$$

where $\mathbf{H}$ is a 2×2 matrix constructed as follows

$$\mathbf{H} = -\mathbf{D}_N^{-1}\mathbf{D}_1$$
$$\mathbf{D}_N = \mathbf{C}_{NN} - \mathbf{C}_N\overline{\mathbf{C}}^{-1}\overline{\mathbf{C}}_N$$
$$\mathbf{D}_1 = \mathbf{C}_{N1} - \mathbf{C}_N\overline{\mathbf{C}}^{-1}\overline{\mathbf{C}}_1$$
$$\mathbf{C}_N = (\mathbf{C}_{N2}, \ldots, \mathbf{C}_{N(N-1)})$$
$$\overline{\mathbf{C}}_1^T = (\mathbf{C}_{21}, \ldots, \mathbf{C}_{(N-1)1})^T$$
$$\overline{\mathbf{C}}_N^T = (\mathbf{C}_{2N}, \ldots, \mathbf{C}_{(N-1)N})^T$$
$$\overline{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_{22} & \cdots & \mathbf{C}_{2(N-1)} \\ \vdots & \vdots & \vdots \\ \mathbf{C}_{(N-1)2} & \cdots & \mathbf{C}_{(N-1)(N-1)} \end{bmatrix}$$

and the vector $\mathbf{Q}^{[i]}$ turns out to be

$$\mathbf{Q}^{[i]} = -\mathbf{D}_N^{-1}(\mathbf{C}_N\overline{\mathbf{C}}^{-1}\overline{\mathbf{q}} - \mathbf{q}_N),$$
$$\overline{\mathbf{q}}^T = (\mathbf{q}_2, \ldots, \mathbf{q}_{(N-1)})^T.$$

By applying Eq. (13) recursively, it is possible to write:

$$\mathbf{w}^{[i+1]} = \prod_{j=0}^{i} \mathbf{H}^{[j]}\mathbf{w}^{[0]} + \mathbf{Q}^{[i]} + \sum_{k=0}^{i-1} \prod_{j=k+1}^{i} \mathbf{H}^{[j]}\mathbf{Q}^{[k]} \tag{14}$$

where $\mathbf{w}^{[0]}$ contains the terms a_i ($i = 0, 1$) given in the boundary conditions at $z = h$ (i.e. $\zeta = 0$).

Considerations about the local error are equivalent to the ones for the DQM [6], but here in addition one has to analyze stability. In order to achieve stability, the spectral radius of the matrix $\mathbf{H}$ has to not exceed the unit value (see Section 6). Since the elements $\mathbf{C}_{kj}$ can be written as:

$$\mathbf{C}_{kj} = \frac{1}{\Delta z^{N-1}} \sum_{r=0}^{N-1} \mathbf{a}_{kj}^{[r]} \Delta z^{N-r-1}$$

where $\mathbf{a}_{kj}^{[r]}$ are 2×2 matrices, more complicated polynomial forms (involving all the terms of $\mathbf{C}_{kj}$) are encountered in the $\mathbf{H}$ elements, so it is difficult to state an a priori condition on Δz.

The grid points can be generated by means of Gegenbauer polynomials as explained in the previous section.

4. A Picard-like numerical method

In this section, another numerical iterative method is presented.

To the scope, let us introduce the operator

$$L = \frac{\mathrm{d}^r}{\mathrm{d}z^r}$$

whose inverse is the r-fold operator

$$L^{-1}(\cdot) = \int_z^h \ldots r\text{-fold} \ldots \int_z^h (\cdot)\mathrm{d}z \ldots \mathrm{d}z$$

and applying L^{-1} on both sides of Eq. (1) one has

$$y(z) = \sum_{i=0}^{r-1} a_i \frac{(z-h)^i}{i!} + (-1)^r L^{-1}\left(q(z)\right) + (-1)^{r+1} L^{-1}\left(\beta(z) y(z)\right) + (-1)^{r+1} L^{-1}\left(\alpha(z) \int_0^z \gamma(z) y(z)\right). \tag{15}$$

In accordance with the idea of the successive approximations, the solution can be written by series of unknown functions y_i which can be determined recursively

$$y = \sum_{k=0}^{\infty} y_k \tag{16}$$

$$y_0 = \sum_{i=0}^{r-1} a_i \frac{(z-h)^i}{i!} + (-1)^r L^{-1}\left(q(z)\right) \tag{17}$$

$$y_{k+1} = (-1)^{r+1} L^{-1}\left(\beta(z) y_k\right) + (-1)^{r+1} L^{-1}\left(\alpha(z) \int_0^z \gamma(z) y_k\right). \tag{18}$$

By introducing numerical integration, one can write, with regard to a function $\overline{f}(z)$:

$$L^{-1}\left(\overline{f}(z)\right) = \sum_{i=1}^{N} C_i(z) \overline{f}(z_i) \tag{19}$$

where

$$C_i(z) = \int_z^h \dots \int_z^h l_i(z) \mathrm{d}z \dots \mathrm{d}z \tag{20}$$

being $l_i(z)$ the Lagrange interpolating polynomials and N the number of grid points at abscissae $0 = z_1 < z_2 < \dots < z_{N-1} < z_N = h$.

So one has (by assuming r as even for the sake of simplicity):

$$y_0 = \mathbf{C}_0(z)\mathbf{q}_0 \tag{21}$$

where

$$\mathbf{C}_0(z) = \left(1, (z-h), \dots, (z-h)^{r-1}, C_1(z), \dots, C_N(z)\right) \tag{22}$$

$$\mathbf{q}_0^T = \left(a_0, a_1, \dots, \frac{a_{r-1}}{(r-1)!}, q(z_1), \dots, q(z_N)\right) \tag{23}$$

and

$$y_{k+1} = \mathbf{D}(z)\mathbf{y}_k \tag{24}$$

where $\mathbf{y}_k^T = (y_k(z_1), \dots, y_k(z_N))$ and

$$\mathbf{D}(z) = -\mathbf{C}(z)\left[\mathbf{B} + \mathbf{A}\overline{\mathbf{C}}\mathbf{G}\right] \tag{25}$$

with $\mathbf{B}$, $\mathbf{A}$, $\mathbf{G}$ diagonal matrices whose ith element is given by $\beta(z_i)$, $\alpha(z_i)$, $\gamma(z_i)$, respectively, and

$$\mathbf{C}(z) = (C_1(z), \dots, C_N(z)) \tag{26}$$

$$\overline{\mathbf{C}} = \begin{bmatrix} \overline{C}_1(z_1) & \cdots & \overline{C}_N(z_1) \\ \vdots & \vdots & \vdots \\ \overline{C}_1(z_N) & \cdots & \overline{C}_N(z_N) \end{bmatrix} \tag{27}$$

with $\overline{C}_i(z) = \int_0^z l_i(z)\mathrm{d}z$.

If one consider the solution given by the first k series terms $y^{[k]}(z) = \sum_{i=0}^{k} y_i(z)$, the related error $e^{[k]}$, i.e. the error after k iterations is

$$e^{[k]}(z) = y(z) - y^{[k]}(z). \tag{28}$$

Since it is

$$\mathbf{y}_k = \overline{\mathbf{D}}\mathbf{y}_{k-1} = \overline{\mathbf{D}}^k \mathbf{y}_0 \tag{29}$$

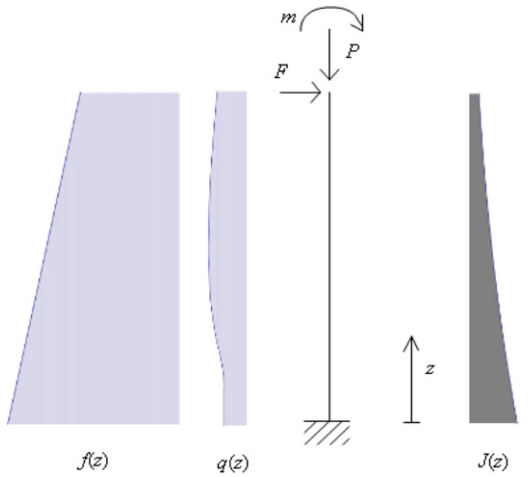

Fig. 2. The model: a tubular tapered cantilever beam subject to concentrated loads at the top and to variable distributed transversal loads over the span.

where

$$\overline{\mathbf{D}}^T = (\mathbf{D}(z_1), \ldots, \mathbf{D}(z_N)) \tag{30}$$

then Eq. (24) can be written as follows

$$y_{k+1} = \mathbf{D}(z)\overline{\mathbf{D}}^k \mathbf{y}_0. \tag{31}$$

So one has

$$y^{[k]}(z) = y_0 + \mathbf{D}(z) \sum_{i=0}^{k-1} \overline{\mathbf{D}}^i \mathbf{y}_0. \tag{32}$$

By considering the Taylor series expansion of $y(z)$ around h, one can write

$$y(z) = y_0 - \sum_{i=1}^{N} C_i q(z_i) + O((z-h)^r) \tag{33}$$

and it follows that

$$e^{[k]} = -\mathbf{D}(z)\left[(\mathbf{I} - \overline{\mathbf{D}})^{-1}(\mathbf{I} - \overline{\mathbf{D}}^k)\right]\mathbf{y}_0 - \sum_{i=1}^{N} C_i q(z_i) + O((z-h)^r) \tag{34}$$

with $\mathbf{I}$ the identity matrix of order N.

This error function converges to a quantity c for k which tends to infinity, if the spectral radius of the matrix $\overline{\mathbf{D}}$ is less than 1:

$$c = -\mathbf{D}(z)\left[(\mathbf{I} - \overline{\mathbf{D}})^{-1}\right]\mathbf{y}_0 - \sum_{i=1}^{N} C_i q(z_i) + O((z-h)^r). \tag{35}$$

Iterations can be ended if for any $0 < \varepsilon_i \ll 1$ one has:

$$|e^{[k]}(z_i) - e^{[k-1]}(z_i)| \le |y^{[k]}(z_i) - y^{[k-1]}(z_i)| \le |y_k(z_i)| \le \varepsilon_i \quad i = 1, \ldots, N \tag{36}$$

where $y_k(z_i)$ is the ith element of the vector $\mathbf{y}_k$.

Grid points can be equally spaced or not. In this second case, a general way to generate the grid points can be computing the zeros of the first order derivative of Gegenbauer polynomials as introduced in Section 2.

5. The model

A tubular tapered cantilever beam is considered, with height h and moment of inertia $J(z)$ subject at the top to a concentrated transversal force F, a concentrated bending moment m, a concentrated axial load P and over the span to a variable distributed transversal load $q(z)$ and the self-weight $f(z)$, as shown in Fig. 2. This is a typical model for wind towers, where $q(z)$ is the peak wind load, obtained by multiplying the peak wind pressures on the tower surface (according to the

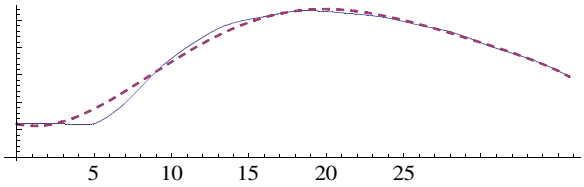

Fig. 3. Exact (continuous line) and approximated (dashed line) distributed wind load $q(z)$ (in N/m).

national code and Eurocode 1—part 1:4 [13]) by the diameter at abscissa z, and the concentrated loads represent the action of the turbine or, excluding these concentrated loads, this is the model usually assumed for a chimney. At the preliminary design stage of a precast concrete small-wind tower, a linear analysis can be performed, but changes in the shape may be frequent, so one has to compute repeatedly the bending moments, axial and shear forces. Whilst the axial force $N(z)$ and the shear force $T(z)$ can be easily obtained, bending moment $M(z)$ requires solving the following integro-differential equation (where negative axial loads are compressive loads)

$$\frac{\mathrm{d}^2M}{\mathrm{d}z^2} + \left(-P + \int_z^h f(z)\mathrm{d}z\right)\frac{M(z)}{EJ(z)} + f(z)\int_0^z \frac{M(z)}{EJ(z)}\mathrm{d}z = -q(z), \quad z\epsilon[0,h] \tag{37}$$

with the boundary conditions

$$M(h) = -m, \qquad \frac{\mathrm{d}M}{\mathrm{d}z}(h) = F.$$

It should be pointed out that the bending moments can also be deduced by deriving twice the displacements obtained by solving (in this case) a complete fourth-order ordinary differential equation (ODE) with variable coefficients, but in order to implement a general optimization procedure to detect the geometric characteristics (procedure not discussed here), an efficient numerical scheme to compute accurate enough bending moments is needed.

For a circular cross-section with depth s, maximum external diameter (at the tower base) $d_{\max}$, minimum external diameter (at the tower top) $d_{\min}$, one has:

$$f(z) = \eta\pi s\left[(d_{\max} - s) - \frac{1}{h}(d_{\max} - d_{\min})\,z\right], \qquad J(z) = \sum_{i=0}^{3} b_i z^i \tag{38}$$

where η is the specific weight of the material and

$$b_0 = \frac{1}{8}\pi s\left(d_{\max}^3 - 3d_{\max}^2 s + 4d_{\max}s^2 - 2s^3\right) \tag{39}$$

$$b_1 = -\frac{(d_{\max} - d_{\min})\pi s\left(3d_{\max}^2 - 6d_{\max}s + 4s^2\right)}{8h} \tag{40}$$

$$b_2 = \frac{3(d_{\max} - d_{\min})^2\pi s\,(d_{\max} - s)}{8h^2}, \qquad b_3 = \frac{(d_{\max} - d_{\min})^3\pi s}{8h^3} \tag{41}$$

are coefficients obtained by rearranging the expression of the moment of inertia $J(z)$.

Giving a formula for $q(z)$ is more complicated since it is constructed under certain conditions according to Eurocode 1—Part 1:4 [13]. In Fig. 3, the distributed wind load $q(z)$ (continuous line), for the case discussed in the next section, is compared with a possible approximation (dashed line) obtained by Lagrange interpolating polynomials with seven equally spaced grid points.

6. Numerical results and discussion

Several numerical experiments have been carried out, but just some significant results are here discussed for the sake of brevity. In what follows, the results are referred to three load conditions: the first one includes all the concentrated loads at the top (as depicted in Fig. 2), the second one has no concentrated loads at the top, the third one has no concentrated loads and the distributed load $q(z)$ is assumed to be constant over the span. A 36 m tower, with $s = 0.16$ m and $d_{\min} = 0.6$, $d_{\max} = 1.2$ m is here considered. Young's modulus is $E = 3.55 \times 10^4$ MPa. A finite element (FE) model, composed by 1240 four-nodes shell elements (accounting for bending and membrane effects), gives the reference solution. The values of the bending moments at different abscissae, i.e. $z = 0$, $z = h/3$, $z = 2h/3$, obtained by the FE analysis and the numerical methods, for different load conditions, are tabled in Tables 1–15 (bending moments are in kN m). The results are referred both to equally and not equally spaced points. With regard to the first load condition, the DQM does not work well: the smallest errors can be obtained by the graded distribution with $p = 2$ and $N = 5$ (Table 1), whereas by means of some distributions of sampling points, the numerical solution at $z = 0$ turns out to be one half enough of the reference solution

Table 1
First load condition: numerical results obtained by the DQM for the geometric and the graded distributions.

	FE	Geometric $b = 0.54$, $N = 5$	Error (%)	Graded $p = 2$, $N = 5$	Error (%)
$M(0)$	−3350.85	−3273.12	2.3197	−3390.07	−1.1704
$M(h/3)$	−1918.91	−1734.34	9.6185	−1943.27	−1.2695
$M(2/3h)$	−808.15	−512.44	36.5910	−814.26	−0.7560

Table 2
First load condition: the bending moment at $z = 0$, $M(0)$, obtained by the DQM for different distributions of grid points.

N	$\lambda = 0$	$\lambda = 0.5$	$\lambda = -1.4$	Equally spaced
5	−1606.09	−1601.81	−1625.42	−1591.95
8	−1510.15	−1509.38	−1510.85	−1503.70
12	−1510.29	−1510.72	−1509.95	−1555.09
20	−1510.64	−1510.71	−1510.55	−2666.08

Table 3
First load condition: MIDQ results at different abscissae z and for different values of λ (3 equal intervals with $N = 4$).

	FE	$\lambda = 0$	Error (%)	$\lambda = -0.2$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = -0.5$	Error (%)
$M(0)$	−3350.85	−3236.94	3.3994	−3273.17	2.3182	−3185.81	4.9253	−3368.70	−0.5327
$M(h/3)$	−1918.91	−1868.16	2.6447	−1884.84	1.7755	−1844.39	3.8835	−1928.59	−0.5045
$M(2/3h)$	−808.15	−794.62	1.6742	−799.21	1.1062	−788.01	2.4921	−811.17	−0.3737

Table 4
First load condition: MIDQ results at different abscissae z for λ closest to −0.5 (3 equal intervals with $N = 4$).

	FE	$\lambda = -0.45$	Error (%)	$\lambda = -0.46$	Error (%)	$\lambda = -0.455$	Error (%)
$M(0)$	−3350.85	−3347.32	0.1053	−3351.36	−0.0152	−3349.33	0.0454
$M(h/3)$	−1918.91	−1918.83	0.0042	−1920.68	−0.0922	−1919.75	−0.0438
$M(2/3h)$	−808.15	−808.51	−0.0445	−809.01	−0.1064	−808.76	−0.0755

Table 5
First load condition: spectral radius of the matrices H_i ($i = 1, 2, 3$) for different values of λ.

	$\lambda = 0$	$\lambda = 0.5$	$\lambda = -0.2$	$\lambda = -0.5$	$\lambda = -0.45$	$\lambda = -0.46$	$\lambda = -0.455$
$\rho 1$	0.9816	0.9746	0.9865	0.9991	0.9963	0.9968	0.9965
$\rho 2$	0.9847	0.9794	0.9884	0.9979	0.9958	0.9962	0.9960
$\rho 3$	0.9884	0.9846	0.9910	0.9978	0.9963	0.9966	0.9964

Table 6
First load condition: the Picard-like numerical results at different abscissae z and for different distributions of grid points with $N = 7$.

	FE	Equally spaced	Error (%)	$\lambda = -1.4$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = 0$	Error (%)
$M(0)$	−3318.87	−3346.53	−0.8334	−3346.78	−0.8409	−3348.97	−0.9069	−3349.23	−0.9148
$M(h/3)$	−1904.91	−1916.98	−0.6336	−1919.58	−0.7701	−1917.74	−0.6735	−1918.14	−0.6945
$M(2/3h)$	−804.39	−807.16	−0.3444	−807.69	−0.4102	−807.50	−0.3866	−807.58	−0.3966

Table 7
First load condition: the Picard-like numerical results at different abscissae z and for different distributions of grid points with $N = 8$.

	FE	Equally spaced	Error (%)	$\lambda = -1.4$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = 0$	Error (%)
$M(0)$	−3318.87	−3346.62	−0.8361	−3348.47	−0.8919	−3348.59	−0.8955	−3348.40	−0.8898
$M(h/3)$	−1904.91	−1917.01	−0.6352	−1917.41	−0.6562	−1918.21	−0.6982	−1918.19	−0.6971
$M(2/3h)$	−804.39	−807.32	−0.3643	−807.21	−0.3506	−807.73	−0.4152	−807.72	−0.4140

(Table 2). By using the MIDQ method, with three intervals of equal length $\Delta z = h/3$ and with $N = 4$ for each interval, one has good results for λ closest to −0.45 (Tables 3 and 4); the spectral radius of the matrices H_i, with $i = 1, 2, 3$, can be retrieved in Table 5.

With regard to the third method, i.e. the Picard-like numerical method, good results can be achieved for $N = 7$ without needing any suitable distribution of grid points: approximated solutions are reported in Table 6 for $N = 7$ and Table 7 for $N = 8$. Besides, five terms in the series solution are sufficient to obtain accurate enough solutions. In fact, for $N = 7$ and equally spaced grid points one has:

$$\mathbf{M}_5^T = \left(1.89 \times 10^{-3}, 1.22 \times 10^{-3}, 7.37 \times 10^{-4}, 3.93 \times 10^{-4}, 1.67 \times 10^{-4}, 4 \times 10^{-5}, 0\right).$$

Table 8
Second load condition: DQM results for the geometric and the graded distributions.

	FE	Geometric $b = 0.48$, $N = 5$	Error (%)	Graded $p = 2$, $N = 5$	Error (%)
$M(0)$	−1487.41	−1495.73	−0.5594	−1600.43	−7.5984
$M(h/3)$	−663.96	−600.77	9.5171	−728.93	−9.7852
$M(2/3h)$	−162.55	−33.58	79.3417	−185.10	−13.8727

Table 9
Second load condition: DQM bending moments at $z = 0$ for different distributions of grid points.

N	$\lambda = 0$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = -1.4$	Error (%)	Equally spaced	Error (%)
6	−1526.65	−2.6381	−1525.74	−2.5770	−1522.31	−2.3464	−1531.87	−2.9891
12	−1524.62	−2.5017	−1525.05	−2.5306	−1524.28	−2.4788	−1569.32	−5.5069
20	−1524.97	−2.5252	−1525.04	−2.5299	−1524.88	−2.5191	−2677.74	−80.0270

Table 10
Second load condition: DQM bending moments at $z = h/3$ for different distributions of grid points.

N	$\lambda = 0$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = -1.4$	Error (%)	Equally spaced	Error (%)
6	−680.32	−2.4640	−680.26	−2.4550	−679.70	−2.3706	−687.29	−3.5138
12	−680.61	−2.5077	−680.90	−2.5514	−680.25	−2.4535	−710.57	−7.0200
20	−680.67	−2.5167	−680.71	−2.5227	−680.62	−2.5092	−1447.49	−118.0086

Table 11
Second load condition: DQM bending moments at $z = 2h/3$ for different distributions of grid points.

N	$\lambda = 0$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = -1.4$	Error (%)	Equally spaced	Error (%)
6	−166.65	−2.5223	−166.61	−2.4977	−167.36	−2.9591	−169.82	−4.4725
12	−166.75	−2.5838	−166.88	−2.6638	−166.61	−2.4977	−181.37	−11.5780
20	−166.70	−2.5531	−166.75	−2.5838	−96.82	40.4368	−545.76	−235.7490

Table 12
Second load condition: MIDQ results at different abscissae z and for different values of λ (3 equal intervals with $N = 4$).

	FE	$\lambda = 0$	Error (%)	$\lambda = 0.1$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = 0.05$	Error (%)
$M(0)$	−1487.41	−1489.07	−0.1116	−1484.03	0.2272	−1470.11	1.1631	−1486.45	0.0645
$M(h/3)$	−663.96	−668.87	−0.7395	−666.94	−0.4488	−661.56	0.3615	−667.87	−0.5889
$M(2/3h)$	−162.55	−159.73	1.7349	−164.69	−1.3165	−163.55	−0.6152	−164.88	−1.4334

Table 13
Second load condition: spectral radius of the matrices H_i ($i = 1, 2, 3$) for different values of λ.

	$\lambda = 0$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.05$
$\rho 1$	0.9816	0.9747	0.9798	0.9907
$\rho 2$	0.9847	0.9794	0.9833	0.9840
$\rho 3$	0.9884	0.9846	0.9874	0.9879

Table 14
Second load condition: the Picard-like numerical results at different abscissae z and for different distributions of grid points with $N = 7$.

	FE	Equally spaced	Error (%)	$\lambda = -1.4$	Error (%)	$\lambda = 0.5$	Error (%)	$\lambda = 0$	Error (%)
$M(0)$	−1487.41	−1523.21	−2.4069	−1523.44	−2.4223	−1525.63	−2.5696	−1525.89	−2.5870
$M(h/3)$	−663.96	−679.90	−2.4007	−682.49	−2.7908	−680.65	−2.5137	−681.06	−2.5755
$M(2/3h)$	−162.55	−166.31	−2.3131	−166.83	−2.6330	−166.65	−2.5223	−166.72	−2.5654

Table 15
Third load condition: percentage errors for the three methods.

	$\lambda = 0$	$\lambda = -0.2$	$\lambda = 0.5$
DQM ($N = 6$)	0.4190	0.4190	0.4190
DQM ($N = 12$)	0.4190	0.4190	0.4190
MIDQ	0.4155	0.4120	0.4190
Picard-like ($N = 7$)	0.9661	0.9661	0.9661

Similar results hold for the other distributions of grid points, i.e. for $N = 7$ and $\lambda = 0$

$$\mathbf{M}_5^T = \left(1.89 \times 10^{-3}, 1.60 \times 10^{-3}, 9.62 \times 10^{-4}, 3.93 \times 10^{-4}, 9.2 \times 10^{-5}, 6.34 \times 10^{-6}, 0\right)$$

for $N = 7$ and $\lambda = 0.5$

$$\mathbf{M}_5^T = \left(1.89 \times 10^{-3}, 1.53 \times 10^{-3}, 9.17 \times 10^{-4}, 3.93 \times 10^{-4}, 1.04 \times 10^{-4}, 1 \times 10^{-5}, 0\right)$$

for $N = 7$ and $\lambda = -1.4$

$$\mathbf{M}_5^T = \left(1.89 \times 10^{-3}, 1.87 \times 10^{-3}, 1.18 \times 10^{-3}, 3.93 \times 10^{-4}, 4.7 \times 10^{-5}, 3.48 \times 10^{-8}, 0\right).$$

For all the cases considered, the spectral radius of the matrix **D** turns out to be less than 1.

With regard to the second load condition, the DQM does not work well again (Tables 8–11), even if the usual distributions allow errors less than the ones referred to the first load condition (Tables 9–11).

The MIDQ method, with three intervals of equal length $\Delta z = h/3$ and with $N = 4$ for each interval again, allows small percentage errors for λ closest to 0 (Table 12); the spectral radius of the matrices H_i, with $i = 1, 2, 3$, is tabled in Table 13.

The third numerical method does not work well: the percentage errors are comparable with the ones obtained by the DQM with the usual distributions (Table 13), but these errors cannot be decreased since the spectral radius of the matrix **D** is greater than 1 for any N.

Finally, the third load condition can be easily handled by the DQM, which allows in this case good results; equal results can be obtained by the MIDQ method, with three intervals of equal length $\Delta z = h/3$ and with $N = 4$ again (the spectral radius of the matrices H_i is less than one for any i); comparable results can be obtained by the Picard-like method.

7. Conclusions

In this paper, three numerical methods to solve integro-differential equations containing rational functions, with application to a real world problem in structural engineering, have been discussed. The first method is the DQM, the second one is a new version of the IDQ method, here named MIDQ, and the third one is a Picard-like method where successive approximations jointly with numerical integration are used. Three different cases were considered and a comparison between the results obtained by an FE analysis and by the present methods discussed. For all the three cases, the MIDQ method seems to allow the best results.

Acknowledgments

This work was funded by INPES S.p.A. The author acknowledges the technical staff at INPES for FE results.

References

[1] H. Brunner, The approximate solution of initial-value problems for general Volterra integro-differential equations, Computing 40 (1988) 125–137.
[2] E. Rawashdeh, D. Mcdowell, L. Rakesh, The stability of collocation methods for higher-order Volterra integro-differential equations, International Journal of Mathematics and Mathematical Sciences 19 (2005) 3075–3089.
[3] C. Chen, T. Shih, Finite Element Methods for Integrodifferential Equations, World Scientific, Singapore, 1998.
[4] R. Bellman, J. Casti, Differential quadrature and long-term integration, Journal of Mathematical Analysis and Applications 34 (1971) 235–238.
[5] C.W. Bert, M. Malik, Differential quadrature method in computational mechanics: a review, Applied Mechanics Reviews 49 (1) (1996) 1–28.
[6] C. Shu, Differential Quadrature and its Application in Engineering, Springer, New York, 2000.
[7] S. Tomasiello, Hershey: IGI Global DQ based methods: theory and application to engineering and physical sciences, in: J. Leng and W. Sharrock (Eds.), Handbook of Research on Computational Science and Engineering: Theory and Practice (2011) (in press).
[8] S. Tomasiello, Simulating non-linear coupled oscillators by an iterative differential quadrature method, Journal of Sound and Vibration 265 (3) (2003) 507–525.
[9] S. Tomasiello, Stability and accuracy of the iterative differential quadrature method, International Journal for Numerical Methods in Engineering 58 (2003) 1277–1296.
[10] S.K. Khattri, I.K. Argyros, Sixth order derivative free family of iterative methods, Applied Mathematics and Computation 217 (12) (2011) 5500–5507.
[11] S. Tomasiello, A DQ based approach to simulate the vibrations of buckled beams, Nonlinear Dynamics 50 (2007) 37–48.
[12] G. Szego, Orthogonal Polynomials, in: AMS Colloquium Publications, vol. 32, 1939, pp. 1277–1296.
[13] CEN/TC250. Eurocode 1: actions on structures—general actions—part 1–4: wind actions. EN 1991-1-4:2005, 2005.