Streamline Performance of Excel in Stepwise Implementation of Numerical Solutions

MAJID NIAZKAR, SEIED HOSEIN AFZALI

Department of Civil and Environmental Engineering, School of Engineering, Shiraz University, Shiraz, Iran

Received 20 December 2015; accepted 26 February 2016

ABSTRACT: Excel spreadsheet not only provides useful facilities to solve various kinds of engineering problems, but also can be simply utilized to enhance the curriculum of engineering courses. In this paper, first a summary of literature review on the application of spreadsheets was presented. Reviewing the literature of the application of Excel in different engineering fields shows that Excel was only applied for different individual applicants; while it can be utilized to implement the solution of all kinds of physical problems because of various facilities which Excel provides. In this regard, the step-by-step numerical solutions of several examples from all kinds of physical problems in civil engineering were implemented into Excel. The educational merit of this study is the stepwise implementation of Excel and Visual Basic Application (VBA) solution to simple examples of all kinds of physical problems without any advanced requirement of prior knowledge of the software. The robust performance of Excel in solving all the considered examples demonstrates that the Excel appraisal in solving engineering problems should be more addressed than what it is in the current literature.
 Comput Appl Eng Educ; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21731

Keywords: engineering education; excel spreadsheet; VBA code; physical problems; stepwise implementation; numerical solution

INTRODUCTION

In general, solution of most engineering problems can be treated as solving specific equations governing over the entire computational domain. These equations may commonly be solved using analytical, experimental, numerical, or any other approaches. Substantial features of each approach play essential key roles in approach selection based on the kind of problem under consideration. In this regard, the numerical methodology has become more popular in practice comparing with other techniques for some purposes after the arrival of micro-computers in the last decades. Therefore, by the conducted advancement in the micro-computers, the implementation of the numerical approaches in computer-aided programs for solving various engineering problems becomes one of major concerns.

In order to utilize computer-aided programs in solving engineering problems, numerous open-source and commercial softwares have been proposed for dealing with exclusive kind of problems. Along with these programs, spreadsheets programs, such as Excel, which potentially can be used for a variety of purposes, have already been utilized in many engineering branches.

Excel provides a suitable platform to handle many engineering problems. These spreadsheets are a scientific tool eliminating the tedious and repetitive computational tasks which may be conducted manually [1]. It becomes increasingly popular in engineering education because of their instinctive cell-based structure and simply applied capabilities. For instance, Excel facilitates the user with numerous numbers of cells which can be preferably linked and cooperated together. These cells accompanied with the built-in robust coding environment, that is, Visual Basic Application (VBA), can be desirably customized to implement the methodologies required for solving different problems. VBA is identified as an optimal programming language for quality control analysis since it is not only the most adequate language that reduces the programming time but also compatible with limited resources [2–3].

Correspondence to S. H. Afzali (afzali@shirazu.ac.ir).

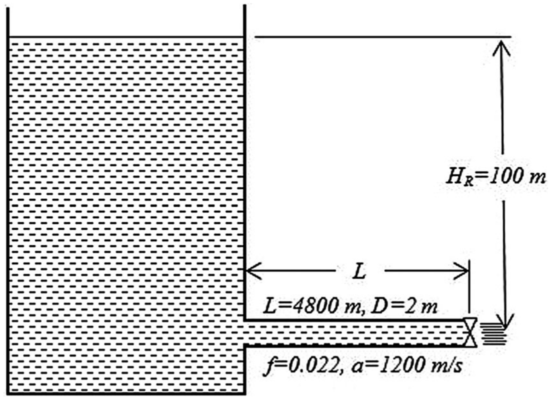

Figure 1 The constant-head reservoir-pipe system with a valve at its end.

The cells of Excel spreadsheet can be interpreted as the indices of a matrix and subsequently matrix-based codes can be implemented. In addition, the add-in solver along with the capability of interacting cells can be programed to solve the governing equations in engineering problems. Furthermore, the computational part of solving a problem with various built-in functions is preferably handed to computer which inevitably seems encouraging for applicants. The substantial capability of solving the problem for other probable scenarios or different values brings about encouraging experiments especially for undergraduate students. Graphical features in Excel also permit drawing variety kinds of plots. The capability of linking Excel to other solving softwares, provide an incredible opportunity for users to prepare either input or output data in the Excel format [4]. Furthermore, it powers the user to solve several kinds of equations numerically. To be more specific, one can implement different numerical techniques to: (1) determine roots of equations, (2) solve systems of simultaneous equations, (3) solve initial-value problem, (4) solve boundary-value problem, (5) solve partial differential equations, (6) conduct linear regression and curve fitting, (7) conduct nonlinear regression using the solver, and (8) generate random numbers using the Monte Carlo method [5].

When the educational purpose is one's exclusive goal, Excel spreadsheet can be utilized efficiently since: (1) the applicants were able to see the real procedure which operates at its background; (2) it provokes the innovative-oriented minds of students in the progress of learning. Consequently, Excel can provide students with the actual fulfillment of educational purposes.

In spite of all these facilities, the stepwise implementation of numerical approaches for all kinds of physical problems in Excel is not well-documented in the computer-aided literature. To be more specific, Excel was exclusively utilized for individual purposes in many engineering fields like civil Engineering and a framework for step-by-step implementation was missed in previous studies [6–13]. The potential of Excel spreadsheet, which was reported as the most suitable spreadsheet for some purposes [2,3,14–16], necessitates usage of this software for much more applications in engineering education than what it is in the current literature. Finally, investigation of Excel applicability in solving all kinds of physical problems can recommend Excel for several applications in different major of Civil Engineering fields.

In this paper, the assessment of Excel as a powerful tool in both educational engineering schools and practical projects is being revisited. First, a short review on appraisal of the spreadsheet such as Excel in several engineering fields is presented to introduce its widespread potential to be utilized in more engineering problems. Afterward, it was tried to implement three common numerical methods including: (1) finite difference method, (2) finite element method, and (3) method of characteristics (MOC), in Excel for step-by-step solving all kinds of physical problems in different areas of civil engineering field. The successful application of these numerical approaches in Excel obviously indicates that similarly numerous problems, which can be solved using similar methods, can probably be handled with Excel. Moreover, the simplicity of implementation of these methods in Excel shows that not only the Excel spreadsheet can be utilized by undergraduate students in civil engineering, but also civil engineers can efficiently solve their practical problems using Excel.

LITERATURE REVIEW ON APPLICATION OF EXCEL SPREADSHEET

The spreadsheets has been extensively applied for teaching undergraduate engineering students in mechanical, electrical

```
'Input data
f = Cells(2, 1) 'f=friction factor of Darcy-Weisbach
L = Cells(2, 2) 'L=pipe length(m)
D = Cells(2, 3) 'D=pipe diameter(m)
ws = Cells(2, 4) 'ws=wave speed(m/s)
HR = Cells(2, 5) 'RH=reservoir head
g = Cells(2, 7) 'g=gravity acceleration (m/s^2)
T = Cells(2, 8) 'T=maximum time interval for computation (s)
'Number of discretization of domain
N = Cells(2, 6) 'N=number of reach into which the pipe is divided
ReDim VC(T + 1, 2) 'VC=the matrix of valve cross section in respect to time
'The first column of VC-matrix is time
'The second column of VC-matrix is valve opening cross section
For i = 1 To T + 1
VC(i, 1) = Cells(i + 9, 1) 'VC(i,1)=time (s)
VC(i, 2) = Cells(i + 9, 2) 'VC(i,2)=valve opening cross section (m^2)
Next i
```

Figure 2 The VBA code of the second step in the propagation-type problem.

```
'Computing the essential parameters
AR = 0.25 * Application.Pi() * D ^ 2 'AR=pipe cross section (m^2)
B = ws / (g * AR)
NS = N + 1 'NS=number of computational nodes
Rf = f * L / (2 * g * D * AR ^ 2) 'Rf=resistance coefficient
Q0 = (HR / (Rf + (1 / (2 * g * VC(1, 2) ^ 2)))) ^ 0.5
H0 = (Q0 / VC(1, 2)) ^ 2 / (2 * g)
R = (HR - H0) / (N * Q0 ^ 2)
```

Figure 3 The VBA code of the third step in the propagation-type problem.

engineering, and mathematics regardless of their field of expertise. Clements [17] assessed spreadsheets as a tool for teaching elementary numerical analysis. Boye et al. [18] used spreadsheets to improve the problem-solving skills of educators. Doak et al. [19] utilized animated spreadsheets as a teaching resource. y Leon et al. [20] assessed Excel to implement numerical solution of ordinary differential equations for teaching the corresponding solution algorithms to students. Evans [21] described teaching simulation techniques using the Excel spreadsheet program. Baker and Sugden [22] reviewed the application of spreadsheets in education in the first 25 years after their first appearance for personal computers. Pecherska and Merkuryev [23] recommended a spreadsheet-based approach to teaching simulation, where the objective is to introduce spreadsheets as a powerful simulation tool for educational purposes. Aliane [24] reported the development of a control system teaching package using the Excel spreadsheet program. Ku and Fulcher [16] reported the achieved experience results of the usage of different software packages to solve the simulation problems of the Bachelor course, using Monte-Carlo technique while the preferred software was Excel in MS Office. These and similar studies indicate the applicability of spreadsheets in engineering education.

Although spreadsheets, such as Excel, has been widely utilized for different application in mechanical engineering [25,26], electrical engineering [27–28], chemical engineering [15,29], and civil engineering [6–13,30], the detailed implementation of numerical solutions to all kinds of physical problems is not being addressed. According to the numerous facilities, which Excel provides, it potentially can be more assessed for solving different engineering problems than what it is in the current literature. Finally, investigating the capability of Excel in solving all kinds of physical problems, as one of major interests of this study, may provide more opportunities for utilizing Excel in solving both educational and practical engineering problems.

In the next section, common numerical solutions to several civil engineering problems including all types of physical problems are implemented in Excel in full details.

IMPLEMENTATION OF COMMON NUMERICAL METHODS IN EXCEL

Physical problems can be classified in three general groups: (1) propagation problems, (2) equilibrium problems, and (3) eigen-

```
'Defining unknown matrices
ReDim Q(T + 1, NS), H(T + 1, NS)
'Q=flow discharge matrix for computational nodes
'H=head matrix for computational nodes
ReDim CP(T, N), BP(T, N), CM(T, N - 1), BM(T, N - 1)
'CP,BP,CM, and BM are matrices of constant values for each time
```

Figure 4 The VBA code of the fourth step in the propagation-type problem.

```
'Computing head and flow discharge at computational nodes at t=0
For i = 1 To NS
Q(1, i) = Q0
H(1, i) = HR - (i - 1) * R * Q0 ^ 2
Next i
```

Figure 5 The VBA code of the fifth step in the propagation-type problem.

problems. The first group comprises initial-value problems in which the known information is marched forward on time. The second category includes boundary-value problems with conditions specified at the extreme boundaries. The third one is a special type of problem in which the solution exists only for eigenvalues of a parameter of the problem [31].

Civil engineering, as one of engineering branches, is filled with various kinds of physical problems. In order to investigate the applicability of Excel in solving physical problems, several examples from all kinds of physical problems in civil engineering were numerically solved using the Excel spreadsheet. The focus is not only to use Excel as a sole problem solver but also to implement the step-by-step solution procedure in the spreadsheet. Since both Excel solution and the implementation are of major concerns, not only relatively simple but also common problems were selected to focus more on Excel application. For this purpose, common numerical approaches including: (1) finite difference method, (2) finite element method, and (3) MOC were step-by-step implemented in Excel and VBA environment to solve these civil engineering problems. For each category of physical problems, a typical example with their stepwise implementation in Excel is presented. The step-by-step implementation of Excel solution can be significantly beneficial for educators since it probably provides the opportunity to better capture the procedure of solution algorithms.

Application of Excel in Solving Water Hammer Problems

In fluid mechanics, the study of the consequences of any sudden change in a pressurized conduit flow is investigated as water hammer problems. Since the origin of this kind of phenomenon is time-dependent, the corresponding problems are unsteady and can be allocated as propagation problems in physical-problem classification. The governing equations for this problem consist of two partial differential equations: (1) equation of motion (Eq. 1) and (2) continuity equation (Eq. 2).

$$\frac{dV}{dt} + \frac{1}{\rho}\frac{\partial p}{\partial x} + g\ \sin\theta + \frac{fV|V|}{2D} = 0 \quad (1)$$

$$\frac{1}{A}\frac{dA}{dt} + \frac{1}{\rho}\frac{d\rho}{dt} + \frac{\partial V}{\partial x} = 0 \quad (2)$$

where V is velocity, t is time, ρ is fluid density, p is pressure, x is the length along the pipe, g is the gravity acceleration, θ is the angle between pipe and horizontal axis, f is the Darcy–Weisbach friction factor, D is pipe diameter, and A is pipe cross section.

Using MOC, the above system of partial differential equations was solved numerically with the aid of the Excel spreadsheet.

As a typical example chosen form Evett and Liu [32], the flow field of a pipe with a valve at its end connecting to a constant-head reservoir is being investigated. The reservoir height above the pipe (H_R), f, D, the wave speed (a), and pipe length (L) are specified in Figure 1. The initial valve opening is equal to 0.06 m^2 and varies at 5-s intervals. The valve opening becomes 0.03, 0.01, 0.003, 0.001, 0.0005, 0.0002, and 0.0 and then remained closed.

```
'Computing head and flow discharge at computational nodes for t>0
For i = 2 To T + 1
'Upstream node
H(i, 1) = HR
Q(i, 1) = (HR - H(i - 1, 2) + B * Q(i - 1, 2)) / (B + R * Abs(Q(i - 1, 2)))
'Downstream node
CP(i - 1, N) = H(i - 1, N) + B * Q(i - 1, N)
BP(i - 1, N) = B + R * Abs(Q(i - 1, N))
Q(i, NS) = -g * BP(i - 1, N) * VC(i, 2) ^ 2 + ((g * BP(i - 1, N) * VC(i, 2) ^ 2) ^ 2 + 2 * g * VC(i, 2) ^
2 * CP(i - 1, N)) ^ 0.5
H(i, NS) = CP(i - 1, N) - BP(i - 1, N) * Q(i, NS)
'Interior nodes
For j = 2 To 4
CP(i - 1, j - 1) = H(i - 1, j - 1) + B * Q(i - 1, j - 1)
BP(i - 1, j - 1) = B + R * Abs(Q(i - 1, j - 1))
CM(i - 1, j - 1) = H(i - 1, j + 1) - B * Q(i - 1, j + 1)
BM(i - 1, j - 1) = B + R * Abs(Q(i - 1, j + 1))
Q(i, j) = (CP(i - 1, j - 1) - CM(i - 1, j - 1)) / (BP(i - 1, j - 1) + BM(i - 1, j - 1))
H(i, j) = CP(i - 1, j - 1) - BP(i - 1, j - 1) * Q(i, j)
Next j
Next i
```

Figure 6 The VBA code of the sixth step in the propagation-type problem.

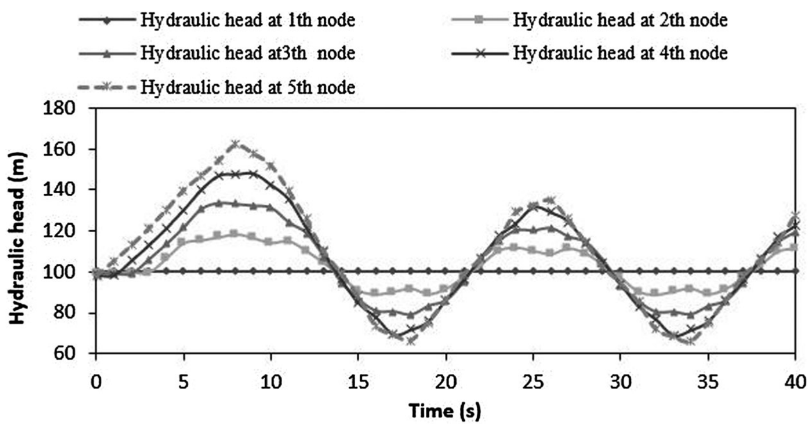

Figure 7 The temporal variation of hydraulic head at computational nodes.

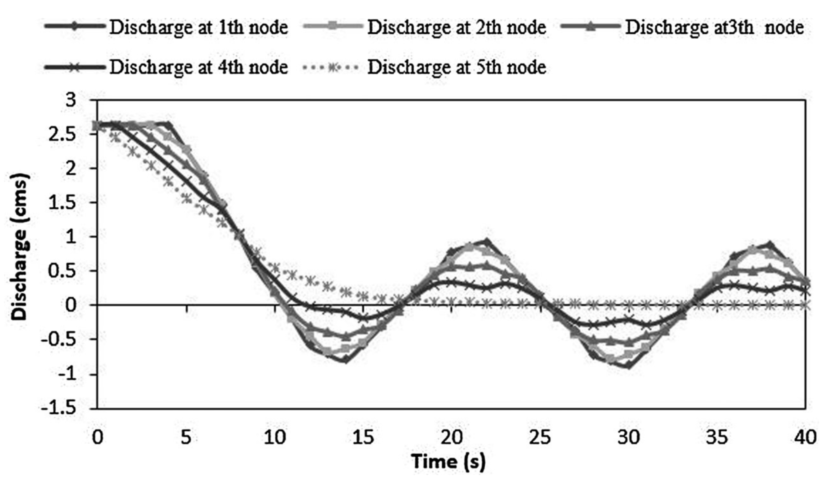

Figure 8 The temporal variation of flow rate at computational nodes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	f	L	D	wave speed	HR	N	g	T	CdA	VAt5	VAt10	VAt15	VAt20	VAt25	VAt30	VAt35	VAt40	
2	0.022	4800	2	1200	100	4	9.806001	40	0.06	0.03	0.01	0.003	0.001	0.0005	0.0002	0	0	
3																		
4	AR	B	NS	Rf	Q0	H0	R											
5	3.141593	38.95287	5	0.272779826	2.631903	98.11048	0.068195											
6																		
7				1					2						3			
8									N =	1					N =	2		
9	time (s)	CV	QA	HA	CP	BP	CM	BM	Q	H	CP	BP	CM	BM	Q	H	CP	BP
10	0	0.06	2.631903	100					2.631903	99.52762					2.631903	99.05524		
11	1	0.054	2.631903	100	202.5202	39.13235	-3.46494	39.13235	2.631903	99.52762	202.0478	39.13235	-3.93732	39.13235	2.631903	99.05524	201.5754	39.13235
12	2	0.048	2.631903	100	202.5202	39.13235	-3.46494	39.13235	2.631903	99.52762	202.0478	39.13235	-3.93732	39.13235	2.631903	99.05524	201.5754	39.13235
13	3	0.042	2.631903	100	202.5202	39.13235	-3.46494	39.13235	2.631903	99.52762	202.0478	39.13235	10.03618	39.12015	2.453744	106.027	201.5754	39.13235
14	4	0.036	2.631903	100	202.5202	39.13235	10.44664	39.1202	2.454534	106.4685	202.0478	39.13235	25.18641	39.10692	2.260519	113.5884	201.6074	39.1202
15	5	0.03	2.278682	100	202.5202	39.13235	25.53464	39.10702	2.262103	113.9988	202.0796	39.12025	41.6372	39.0926	2.051356	121.83	201.6421	39.10702
16	6	0.026	1.895221	100	188.7612	39.10826	41.92383	39.09276	1.877691	115.328	202.1142	39.10713	59.46666	39.07708	1.824505	130.763	201.7363	39.09276
17	7	0.022	1.479328	100	173.8243	39.08211	59.69327	39.07729	1.460234	116.7553	188.4694	39.08092	78.8152	39.06027	1.403283	133.6278	201.8327	39.07729
18	8	0.018	1.027465	100	157.6241	39.05375	78.96592	39.04856	1.007117	118.2924	173.6356	39.05245	92.62256	39.04831	1.037288	133.1269	188.2897	39.04856
19	9	0.014	0.536568	100	140.0227	39.02294	92.72156	39.02361	0.606064	116.3723	157.5225	39.02155	107.203	39.02376	0.644747	132.3634	173.5323	39.02361

Figure 9 Excel spreadsheet for water hammer example.

The objective is to find the velocity and pressure fields through the pipe during the first 40 s. The stepwise procedure of implementing the solution in Excel is illustrated as below:

Step 1. Inserting the input data into the cells and VBA code: In this example, the input data includes: (1) Darcy–Weisbach friction factor, (2) pipe length, (3) pipe diameter, (4) wave speed, (5) reservoir height from the center of the pipe, (6) gravity acceleration, (7) maximum time interval for computations, (8) temporal variation of the valve opening, and (9) number of reaches. Figure 2 shows introducing these parameters in the VBA environment. In Figure 2, the temporal variation of the valve opening is defined as a 41×2 matrix which the first and second columns of this matrix indicate the time and the corresponding valve opening values.

Step 2. Discretization of the pipe length into several reaches in which each reach extreme is one of computational nodes: In this example, the pipe is divided into four equal reaches giving five computational nodes. The first and fifth nodes are right at the point which flow is entering and exiting the pipe, respectively. The former node is called upstream node and latter is named the downstream node. The remained nodes, that is, interior nodes, are placed along the pipe. Ergo, the magnitudes of the velocity and pressure of the flow field are computed in these five nodes.

Step 3. Computing the essential parameters: In order to commence the MOC calculations, several parameters are required to be computed in advance including: (1) pipe cross section (AR), (2) resistance coefficient (Rf), (3) number of computational nodes (NS), (4) two constant coefficients in MOC equations, that is, B and R, and (5) the hydraulic head and discharge at steady state condition for the valve-opening at $t = 0$ (s). The detail of the computation of these parameters is depicted in Figure 3.

Step 4. Defining unknown matrices in the VBA code: Some matrices are required to be introduced for the velocity, pressure and some MOC parameters at the beginning of the solution process (Fig. 4). These indices of these matrices will be evaluated in next steps.

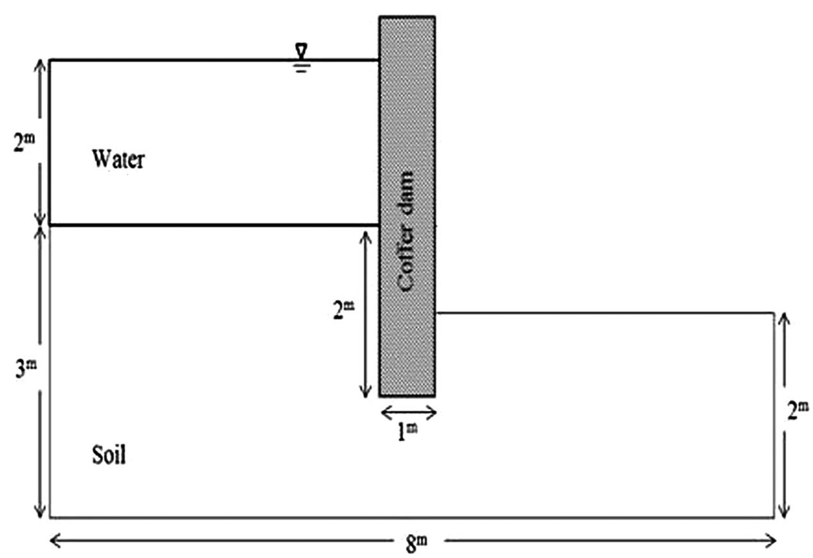

Figure 10 The coffer dam configuration.

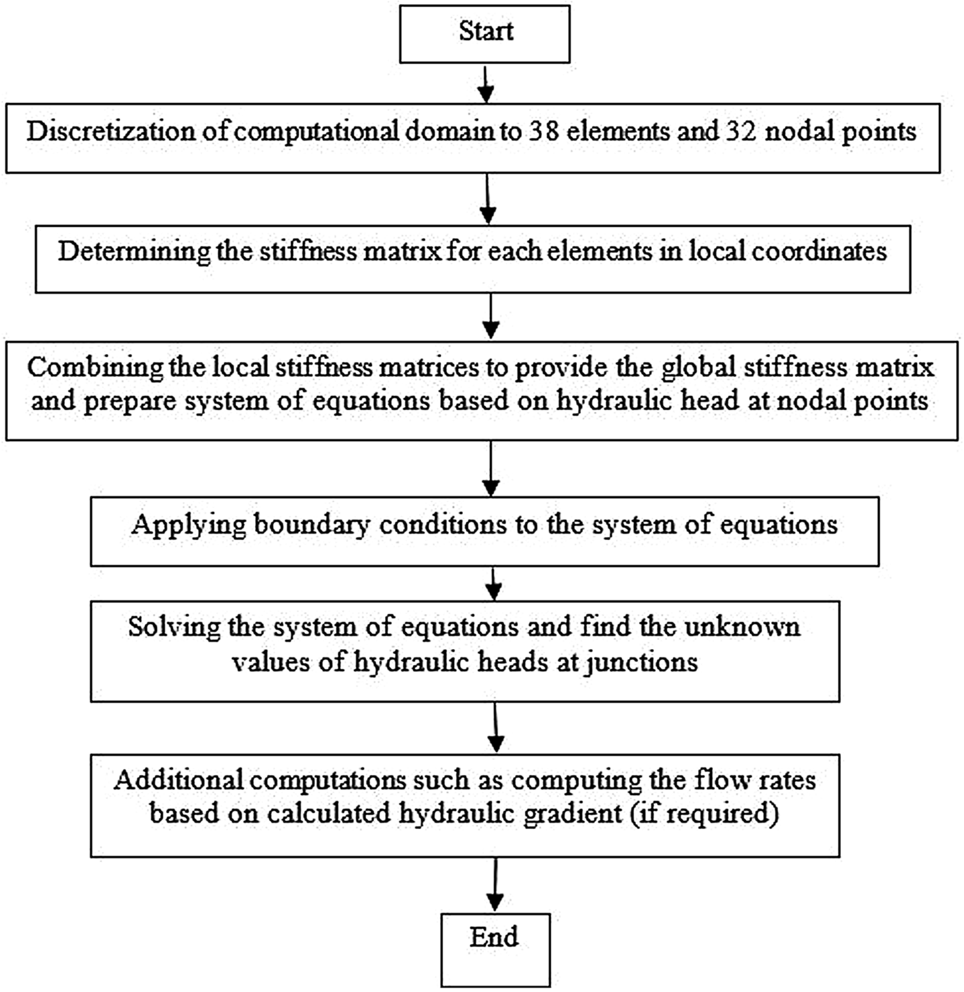

Figure 11 The flowchart of the finite element method.

Step 5. Computing hydraulic head and flow rate values at all computational nodes at $t = 0$ (s): The flow discharge at the beginning, taht is, $t = 0$ (s), is equal to the flow rate for the steady-state condition in all computational nodes. Consequently, all nodes possess the same flow rate at $t = 0$ (s). Similarly, the hydraulic head at each node is equal to the corresponding values in the steady-state condition. Since the major loss due to pipe friction is only taken into account, the hydraulic head at five nodal points differ with one another based on its location in the pipe (Fig. 5). The computed values for the flow rate and hydraulic head at the steady-state condition for all nodes are utilized as the initial condition for the unsteady water hammer equations in the next step.

```
'Inserting input data
n = Cells(2, 1) 'n=Number of nodes
m = Cells(2, 2) 'm=Number of elements
a = Cells(2, 3) 'a=Unit length along x-axis
b = Cells(2, 4) 'b=Unit length along y-axis
e = Cells(2, 5) 'e=Total length along x-axis
f = Cells(2, 6) 'f=Number of nodes with essential boundary condition
'Computing a typical stiffness matrix for triangular element
Cells(5, 14) = (1 / (2 * a * b)) * (a ^ 2 + b ^ 2) 'K1(1,1)
Cells(5, 15) = (1 / (2 * a * b)) * -b ^ 2 'K1(1,2)
Cells(5, 16) = (1 / (2 * a * b)) * -a ^ 2 'K1(1,3)
Cells(6, 14) = (1 / (2 * a * b)) * -b ^ 2  'K1(2,1)
Cells(6, 15) = (1 / (2 * a * b)) * b ^ 2 'K1(2,2)
Cells(6, 16) = (1 / (2 * a * b)) * 0 'K1(2,3)
Cells(7, 14) = (1 / (2 * a * b)) * -a ^ 2  'K1(3,1)
Cells(7, 15) = (1 / (2 * a * b)) * 0 'K1(3,2)
Cells(7, 16) = (1 / (2 * a * b)) * a ^ 2 'K1(3,3)
```

Figure 12 The VBA code of the fifth step in the equilibrium-type problem.

Step 6. Computing the hydraulic head and flow discharge values at all computational nodes for $0 < t \leq 40$: The water hammer governing equation is particularly solved using the computed steady-state initial conditions. As the unsteady condition is considered in this step, the specific matrix attributed to the MOC parameters will be evaluated along with finding the main unknowns of the problem. The VBA code for this step is illustrated in Figure 6. In this figure, the order of calculations is computing the hydraulic head and flow rate values at: (1) the upstream node, (2) the downstream node, and (3) the interior nodes for $0 < t \leq 40$, respectively. The hydraulic head at the upstream node is always equal to H_R while its flow rate depends on the hydraulic head and flow rate values of the second node in the

```
'The governing equation is KP=Q.
'Initiating the global stiffness
For i = 1 To n
For j = 1 To n
Cells(i + 4, j + 17) = 0
Next j
Next i
'Determining the global stiffness
For i = 1 To m
For j = 1 To 3
For K = 1 To 3
C = Cells(i + 4, j + 1)
d = Cells(i + 4, K + 1)
Cells(C + 4, d + 17) = Cells(C + 4, d + 17) + Cells(j + 4, K + 13)
Next K
Next j
Next i
```

Figure 13 The VBA code of the sixth step in the equilibrium-type problem.

previous time step. On the other hand, the hydraulic head and flow rate values at other nodal points can be calculated using the hydraulic head and flow rate values of other nodes in the previous time step and the MOC parameters. Since these parameters may have different values at each time and for each node, they have to be evaluated in each time step for each node. In the VBA code for this step, it is shown that the values of the MOC parameters are stored in the matrices which was previously defined in step 4 (Fig. 6).

These steps can be whether implemented directly in Excel spreadsheet or coded in the VBA environment linking with the spreadsheet. The results of solving this problem using Excel are presented in Figures. 7 and 8. In these figures, the temporal variations of hydraulic head and flow rate values for all nodes are shown respectively. Figure 9 depicts the cells of Excel spreadsheet evaluated in the water hammer example. Finally, the successful implementation of this typical initial-value problem indicates that similar procedure can be preceded in Excel to solve the propagation-type problems.

```
'Introducing the Q-vector
For i = 1 To n
Cells(4 + i, 13) = 0 'Setting zero
Next i
'Making the Q-vector
For i = 1 To f
C = Cells(i + 4, 6) 'K2(i,1)
For j = 1 To n
Cells(j + 4, 13) = Cells(j + 4, 13) - Cells(i + 4, 7) * Cells(j + 4, C + 17)
Next j
Next i
```

Figure 14 The VBA code of the seventh step in the equilibrium-type problem.

Application of Excel in Computing Pore Pressure Contour Lines in a Coffer Dam

In soil mechanics, the calculation of pore pressure contour lines in embankments or coffer dams is a major of interest from both analysis and design aspects. As water flows throughout the soil behind and under the coffer dam, pore water pressure drops slowly from the upstream to the downstream of the coffer dam. Finding the pore-water pressure distribution throughout the porous media behind and under the coffer dam is indeed required in the design of embankment dams.

In order to determine the variation of pore water pressure in the soil, the following governing equation should be considered:

$$\mathrm{div}(K.\mathrm{grad}\phi) - Q = S\frac{\partial \phi}{\partial t} \tag{3}$$

where div is the divergence operator, K is hydraulic conductivity, grad is the spatial derivative, ϕ is hydraulic head, Q is the applied source or sink, and S is the specific storage. In this problem, not only Q and S are equal to zero, but also isotropic and homogenous condition is presumed for simplicity. Because of the applied assumptions, Equation 3 simplifies to Equation 4 and the problem subsequently can be categorized as an equilibrium problem with boundary conditions.

$$K(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}) = 0 \tag{4}$$

```
'Applying boundary conditions to the right hand-side of system of equations
For i = 1 To f
C = Cells(i + 4, 6) 'K2(i,1)
Cells(C + 4, 13) = Cells(i + 4, 7)
Next i
'Define the Q-matrix
Dim vMtxA As Variant, vMtxB As Variant, vMtxC As Variant
vMtxA = Range("M5:M36") 'vMtxA is the Q-matrix
'Applying boundary conditions to the left hand-side of system of equations
For i = 1 To f
C = Cells(i + 4, 6) 'K2(i,1)
For j = 1 To n
Cells(j + 4, C + 17) = 0
Cells(C + 4, j + 17) = 0
Next j
Cells(C + 4, C + 17) = 1
Next i
```

Figure 15 The VBA code of the eighth step in the equilibrium-type problem.

```
'Computing the inverse of stiffness matrix
vMtxB = Range("R5:AW36") 'vMtxB is the K-matrix
ReDim vMtxC(n, n)
vMtxC = Application.MInverse(vMtxB) 'vMtxC is the inverse of K-matrix
For i = 1 To n
For j = 1 To n
Cells(i + 4, j + 50) = vMtxC(i, j)
Next j
Next i
'Computing the unknown vector
ReDim vMtxD(n, 1)
For i = 1 To n
For j = 1 To n
vMtxD(i, 1) = vMtxD(i, 1) + vMtxC(i, j) * vMtxA(j, 1) 'vMtxD is the P-matrix
Next j
Next i
```

Figure 16 The VBA code of the ninth step in the equilibrium-type problem.

As an example, a typical coffer dam was considered in which a constant hydraulic head equal to 2 meter was applied at the upstream of the coffer dam (Fig. 10).

The main objective of this problem is to determine the pore-water pressure contour lines in the coffer dam surroundings. For this purpose, Equation 4 is the sole governing equation of this problem, which was solved by implementing the finite element method in Excel accompanied with a VBA code.

The flow chart of the finite element method is shown in Figure 11. According to Figure 11, first the whole computational domain was discretized into 38 triangular linear elements, which subsequently provided 32 computational nodes. Each element has equal unit-length along the horizontal and vertical directions. In the next step in Figure 11, the governing equation for each element was built in the local coordinate system in the matrix format. Based on the discretization, a continuity table was prepared which informs Excel how each node is attached to others. Using this continuity table, the governing equations in the local coordinate system for each element were combined to build up the governing system of equations in the global coordinate system. One of the important steps in the finite element method is probably applying boundary conditions. As the specified boundary conditions are applied to both side of system of equations, the governing equations can be subsequently solved. The implementation of these steps in Excel is described as below:

Step 1. Inserting the input data into preferable cells of Excel and VBA environment: The input data for this problem includes: (1) number of elements ($m = 38$), (2) number of nodes ($n = 32$), (3) unit length of triangular element in the horizontal and vertical directions, (4) total length of the computational domain along the horizontal axis, and (5) number of nodes with essential boundary conditions.

```
'Exporting the obtained results
For i = 1 To n
Cells(i + 4, 12) = vMtxD(i, 1)
Next i
```

Figure 17 The VBA code of the tenth step in the equilibrium-type problem.

Step 2. Discretization of computational domain into elements: The computational domain in this problem is divided into 38 right equal-side triangular elements in which each one comprises three nodes. As all elements and nodes are numbered, the designated numbers should be inserted in the Excel spreadsheet as the continuity table. This table illustrates the way in which the elements are connected to each other. Each row of this table is associated with one element and the columns of this table represent the three corresponding numbers of nodes for each element in the global coordinate system. This continuity table is exclusively utilized for combining local stiffness matrices into the global stiffness matrix.

Step 3. Specifying nodes with essential boundary conditions: The nodes with essential boundary conditions should be introduced into cells. In this problem, the nodes with specified essential boundary condition are the ones with known hydraulic head, which locate at both the upstream and downstream ends of computational domain.

Step 4. Computing a typical stiffness matrix for triangular element: Since the right triangular element is selected as element-type for this problem, a typical stiffness matrix should be inserted into cells of Excel. This typical stiffness matrix depends on the unit-length of element whereas it is independent of the orientation of the element. This stiffness matrix is a symmetric 3×3 one which its indices are the unit-length of triangular element in the horizontal and vertical directions. Since the structure of this matrix remains constant for all elements in the local coordinate system, its definition significantly helps coding the finite element method.

Step 5. Inserting the input data into the VBA code: All input data, which have been already entered into Excel cells, should be imported to the VBA code. This process in the VBA environment is shown in Figure 12. According to Figure 12, the code utilizes the values of the input data which were inserted in specific cells of Excel spreadsheet in advance. It is obvious that by changing the values of the input data, this VBA code can solve the problem with new input data. This Excel facility brings about the chance of solving the same problem for any desirable conditions.

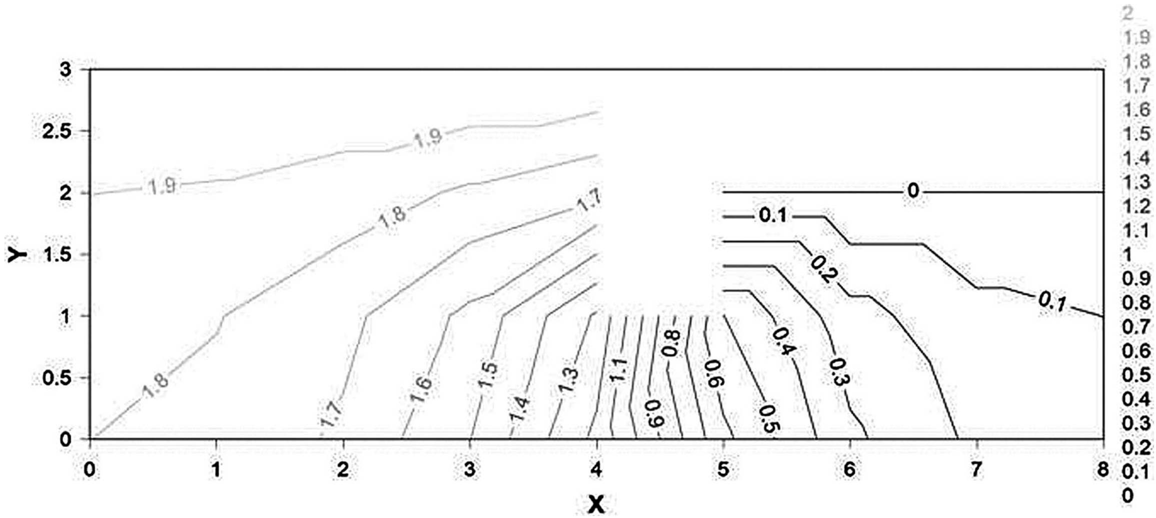

Figure 18 The pore water pressure contour lines.

Step 6. Determining the global stiffness matrix: In the VBA code for this step (Fig. 13), the global stiffness matrix, which is a symmetric 32×32 one, should be initiated as a zero-matrix. The evaluation process of this matrix is conducted using the previously defined continuity table and typical local stiffness matrix.

Step 7. Introducing the flux vector in the VBA code: Based on the governing equation, the matrix multiplication of the global stiffness matrix with the vector of the nodal hydraulic heads comprises the right hand-side of the matrix equation. The left hand-side of this matrix equation is the flux vector (Q-vector). In order to define this vector, a zero-value 32×1 vector is first introduced. For the nodes with specified essential boundary condition, the corresponding indices of this vector are evaluated equal to those specified values of the essential boundary conditions; while the remained indices are kept zero (Fig. 14). Hence, the governing equation in the matrix format in the global coordinate system is completely formed at the end of this step.

Step 8. Applying boundary conditions: In order to solve the obtained matrix equation, the specified boundary conditions should be applied to both side of this equation. The VBA code for applying boundary condition to both sides of the matrix equation is shown in Figure 15. Since Excel provides the opportunity to change the input data, this problem can be solved for any other desirable boundary conditions.

Step 9. Computing the unknown vector: As the boundary conditions are applied, the matrix equation becomes solvable for a single unique answer. Therefore, the unknown hydraulic-head vector can be computed by multiplying the inverse of the global stiffness matrix into the right-hand side of the Q-vector in the VBA code (Fig. 16). Since Excel enables matrix computation, such as inverse calculation of a matrix, it can be extensively utilized for this kind of purposes.

Step 10. Exporting the obtained results from VBA to the spreadsheet: As the hydraulic head values for all the nodal points in the computational domain were obtained, these values can simply be transferred from VBA environment into the cell spreadsheet. The detail of this process is shown in Figure 17.

According to the obtained result, a hydraulic head contour-line map is plotted and depicted in Figure 18 using one of the Excel add-ins. This figure clearly shows the way in which the pore water pressure varies throughout the soil in the surroundings of the coffer dam. Finally, the successful implementation of the finite element method for this equilibrium-type problem in Excel

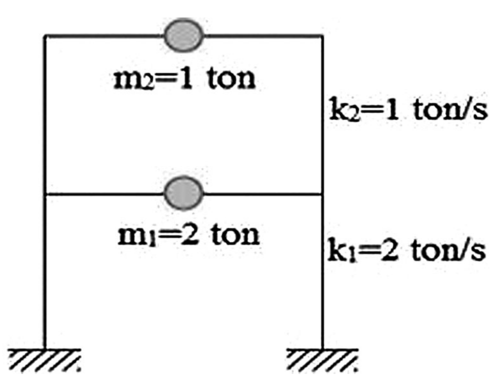

Figure 19 The two-story planar frame.

```
'Inserting input data
n = Cells(2, 1) 'n = No. of stories
Dim K(2, 2), M(2, 2), A As Double, B As Double, C As Double, w1 As
Double, w2 As Double
'Developing mass matrix (M)
For i = 1 To n
M(i, i) = Cells(i + 1, 3)
Next i
'Developing stiffness matrix (K)
K(1, 1) = Cells(2, 4) + Cells(3, 4)
K(1, 2) = -Cells(3, 4)
K(2, 1) = K(1, 2)
K(2, 2) = Cells(3, 4)
```

Figure 20 The VBA code of the first two steps in the eigenvalue-type problem.

```
'Determining the coefficients of equation which its unknowns are the eigenvalues of the problem
'Aw^2+Bw+C=0
'Computing A
A = M(1, 1) * M(2, 2) - M(1, 2) * M(2, 1)
'Computing B
B = -K(1, 1) * M(2, 2) - K(2, 2) * M(1, 1) + K(2, 1) * M(1, 2) + K(1, 2) * M(2, 1)
'Computing C
C = K(1, 1) * K(2, 2) - K(2, 1) * K(1, 2)
'Computing eigenvalues
w1 = (-B + (B ^ 2 - 4 * A * C) ^ 0.5) / (2 * A)
w2 = (-B - (B ^ 2 - 4 * A * C) ^ 0.5) / (2 * A)
Cells(5, 2) = w1
Cells(6, 2) = w2
```

Figure 21 The VBA code of the third step in the eigenvalue-type problem.

indicates that the Excel spreadsheet facilitates great opportunities for solving this kind of problems.

Application of Excel in Computing Free-Vibrating Modes of Two-Story Frame

In dynamic of structures, the response of structures under free or forced vibration is one of the important topics. In a planar structure, the equation of motion consists of inertial force, damping resisting force, elastic resisting force, and external dynamic forces. In a simplified situation, which no damping resisting force and external dynamic force exist, the governing equation for determining free-vibration modes is as below:

$$[k - \omega_n^2 m]\phi_n = 0 \quad (5)$$

where k and m are stiffness and mass matrices, ω_n and ϕ_n are natural frequencies and modes, respectively. Equation 5 represents a system of algebraic equations for n-story. If the trivial solution for ϕ_n, that is, $\phi_n = 0$, is put aside, the roots of the characteristic equation, that is, $det[k - \omega_n^2 m] = 0$, are the natural frequencies of the corresponding structure. In other words, the nontrivial solutions of Equation 5 are the eigenvalues of characteristic equation.

As an example, an idealized two-story shear building, which is shown in Figure 19, is considered as an eigenvalue problem to be solved by Excel [33]. The objective of this problem is to determine the two natural modes of the shown frame (Fig. 19). The Excel solution to this problem is described as below:

Step 1. Inserting input data into cells and the VBA code: The input data for this problem includes: (1) number of stories, (2) the values of the lumped mass of the first and second stories, and (3) the lateral stiffness of the first and second stories. These values can first be entered into cells and the corresponding cells should be addressed in the prepared VBA code. It should be noted that the frame under consideration has equal height for each story.

Step 2. Developing essential matrices in the VBA code: The 2×2 symmetric mass and lateral stiffness matrices should be built in this step. In this regard, the input data from the previous step is utilized to evaluate the aforementioned matrices. The VBA code for the first two steps is shown in Figure 20. In Figure 20, two matrices and five parameters are defined in the VBA environment. These matrices are used for the mass and lateral stiffness matrices while the parameters are the coefficients and the eigenvalues of the characteristic equation which will be computed in next steps.

Step 3. Solving the characteristic equation: In this problem, the characteristic equation is an algebraic parabolic one in respect to the squared of natural frequencies, that is, ω_n^2. Using the defined matrices for mass and lateral stiffness, the coefficients of the characteristic equation are computed in the VBA environment (Fig. 21). Hence, the eigenvalues can be obtained by solving the characteristic equation in the next step.

Step 4. Determining the natural modes: The two-story frame under investigation has two free-vibrating modes. The first and second modes are computed while it is assumed that the second story has a unit-value mode. In Figure 22, the VBA code for determining the first and second natural modes is shown.

This eigenvalue problem could be solved either by using both the spreadsheet and the VBA environment or by exclusively customizing the cells in the Excel spreadsheet. Figure 23 shows

```
'Determining matrix T = K-m(landa)^2
Dim T(2, 2), V(2, 2)
'Computing the first mode
T(1, 1) = K(1, 1) - M(1, 1) * w1
T(1, 2) = K(1, 2) - M(1, 2) * w1
'T(2, 1) = K(2, 1) - M(2, 1) * w1
'T(2, 2) = K(2, 2) - M(2, 2) * w1
Cells(5, 5) = -T(1, 2) / T(1, 1)
Cells(6, 5) = 1
'Computing the second mode
V(1, 1) = K(1, 1) - M(1, 1) * w2
V(1, 2) = K(1, 2) - M(1, 2) * w2
'V(2, 1) = K(2, 1) - M(2, 1) * w2
'V(2, 2) = K(2, 2) - M(2, 2) * w2
Cells(5, 7) = -V(1, 2) / V(1, 1)
Cells(6, 7) = 1
```

Figure 22 The VBA code of the fourth step in the eigenvalue-type problem.

	A	B	C	D	E	F	G	H
1	No. of stories		Mass (*m)	Stiffness (*k)				
2	2	First story	2	2				
3		Second story	1	1				
4								
5	First eigenvalue	2		First mode	-1	Second mode	0.5	
6	Second eigenvalue	0.5			1		1	
7								

Figure 23 Excel spreadsheet for determining free-vibrating modes of two-story frame.

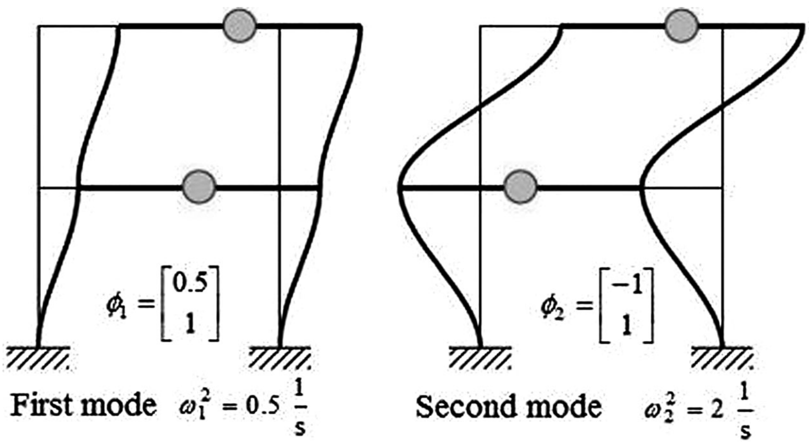

Figure 24 The two obtained free-vibrating modes of the frame.

the spreadsheet cells which are evaluated in this problem. Furthermore, the obtained natural modes were schematically shown in Figure 24. Finally, the Excel solution to these problems obviously demonstrates the efficient facilities that Excel provides for its users in implementation of numerical methods for solving all kinds of physical problems.

CONCLUSION

The successful application of various simple or complicated engineering problems in spreadsheets such as Excel reveals the inevitable potential of these programs for both educational and practical purposes. The ease of use, powerful add-ins, and robust environment of spreadsheets like Excel makes it favorable for solving different engineering problems. In this paper, first the previous appraisal of Excel spreadsheet not only as an educational tool but also as a powerful solver in mechanical, electrical, chemical, and civil engineering fields was reviewed. Although the literature is full of applications of spreadsheet for engineering problems, Excel was not addressed in literature for solving all kinds of physical problems. Therefore, the step-by-step Excel and VBA solutions of three examples from all kind of physical problems in civil engineering were presented to address the detail description of implementing common numerical approaches for applicants. In this regard, different numerical approaches including: (1) finite difference method, (2) finite element method, and (3) method of characteristics were utilized. For each step of the solution in each example, the VBA code is presented to better address the implementation of the methodologies in Excel. Finally, the efficiency and robustness of Excel solutions for all kinds of physical problems indicate that Excel should be much more utilized for solving all kinds of engineering problems than what it is in the current literature.

REFERENCES

[1] S. A. Oke, Spreadsheets applications in engineering education—A review, Int J Eng Educ 20 (2004), 893–901.

[2] J. A. Orosa García, Programming languages for marine engineers, Comput Appl Eng Educ 19 (2011), 591–597.

[3] J. A. Orosa and A. C. Oliveira, Improvement in quality control for applications used by marine engineers, Comput Appl Eng Educ 20 (2012), 187–192.

[4] J. R. Punuru and F. C. Knopf, Bridging Excel and C/C++ code, Comput Appl Eng Educ 16 (2008), 289–304.

[5] E. J. Billo, Excel for scientists and engineers: Numerical methods. John Wiley & Sons, New Jersey, 2007.

[6] D. H. Huddleston, V. J. Alarcon, and W. Chen, Water distribution network analysis using Excel, J Hydraul Eng 130 (2004), 1033–1035.

[7] H. Karahan and M. T. Ayvaz, Time-dependent groundwater modeling using spreadsheet, Comput Appl Eng Educ 13 (2005), 192–199.

[8] H. Karahan, Solution of weighted finite difference techniques with the advection–diffusion equation using spreadsheets, Comput Appl Eng Educ 16 (2008), 147–156.
[9] S. M. Hosseini, Application of spreadsheets in developing flexible multiple-reach and multiple-branch methods of Muskingum flood routing, Comput Appl Eng Educ 17 (2009), 448–454.
[10] R. K. Bhattacharjya, Solving groundwater flow inverse problem using spreadsheet solver, J Hydrol Eng 16 (2011), 472–477.
[11] H. Karahan, Predicting Muskingum flood routing parameters using spreadsheets, Comput Appl Eng Educ 20 (2012), 280–286.
[12] R. Barati, Application of Excel solver for parameter estimation of the nonlinear Muskingum Models, KSCE J Civ Eng 17 (2013), 1139–1148.
[13] A. Sana, Enhancing water resources and coastal engineering curricula using visual basic programs in MS-Excel. In InCIEC 2013. Springer, Singapore, pp 323–330.
[14] Z. L. Kahn-Jetter and P. A. Sasser, Using spreadsheets for studying machine design problems involving optimization, Comput Appl Eng Educ 5 (1997), 199–211.
[15] O. A. Iglesias and C. N. Paniagua, Use of spreadsheets in optimization problems involving iterations, Comput Appl Eng Educ 7 (1999), 227–234.
[16] H. Ku and R. Fulcher, Using computer software packages to enhance the teaching in Engineering Management Science: Part 3—simulation, Comput Appl Eng Educ 20 (2012), 547–552.
[17] R. R. Clements, Spreadsheets as a Tool for Teaching Elementary Numerical Analysis, University of Bristol, Department of Engineering Mathematics, Report EM/RRC/91/1 (1991).
[18] A. J. Boye, P. F. Williams, and R. J. Soukup, Using spreadsheets to teach problem solving in a first year class, IEEE Trans Educ 36 (1993), 68–71.
[19] R. B. Doak, J. McCarter, M. Green, S. Duerdeen, D. Evans, R. Roedel, and P. Williams, Animated spreadsheets as a teaching resource on the freshman level. In Frontiers in Education Conference, FIE'96. 26th Annual Conference, Proceedings of IEEE, 2, 1996, pp 914–918.
[20] B. y Leon, R. Ulfig, and J. Blanchard, Teaching the numerical solution of ordinary differential equations using excel 5.0, Comput Appl Eng Educ 4 (1996), 117–125.
[21] J. Evans, Spreadsheets as a tool for teaching simulation, IEEE Trans Educ 1 (2000), 1.
[22] J. E. Baker and S. J. Sugden, Spreadsheets in education–The first 25 years, Spreadsheets in Education (eJSiE) 1 (2007), 18–43.
[23] J. Pecherska and Y. Merkuryev, Teaching simulation with spreadsheets. Proceedings of 19th European on Modelling and Simulation, 2005, ISBN: 1-84233-112-4.
[24] N. Aliane, Control systems development using spreadsheets, World Trans Eng Technol Educ 6 (2007), 349–352.
[25] K. A. Woodbury, R. P. Taylor, J. Huguet, T. Dent, J. Chappell, and K. Mahan, Vertical integration of Excel in the Tthermal mechanical engineering curriculum, in ASME 2008 International Mechanical Engineering Congress and Exposition, Ame Soc Mech Eng (2008), 317–325.
[26] M. R. Schumack, Teaching heat transfer using automotive-related case studies with a spreadsheet analysis package, Int J Mech Eng Educ 25 (1996), 177–196.
[27] J. A. Svoboda, Using spreadsheets in introductory EE courses, IEEE Trans Educ 35 (1992), 226–229.
[28] D. Ibrahim, Teaching Science and Mathematics Subjects Using the Excel Spreadsheet Package. Frontiers in Science Education Research, Eastern Mediterranean University, 22–24 March, Famagusta, Cyprus, 2009.
[29] E. C. Ferreira, R Lima, and R. Salcedo, Spreadsheets in chemical engineering education—A tool in process design and process integration, Int J Eng Educ 20 (2004), 928–938.
[30] A. Rivas, T. Gomez-Acebo, and J. C. Ramos, The application of spreadsheets to the analysis and optimization of systems and processes in the teaching of hydraulic and thermal engineering, Comput Appl Eng Educ 14 (2006), 256–268.
[31] J. D. Hoffman and S. Frankel, Numerical methods for engineers and scientists, 2nd ed, CRC Press, New York, 2001, p 326.
[32] J. B. Evett and C. Liu, 2500 solved problems in fluid mechanics and hydraulics, McGraw-Hill (1989), 592–593.
[33] A. K. Chopra, Dynamics of structures: Theory and applications to earthquake engineering, 4th ed, Prentice Hall, Boston, 2012, pp 415–416.

BIOGRAPHIES

Majid Niazkar received his BS degree in civil engineering (2012) and MS degree in hydraulic structures (2014) both from the Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran. He is currently a PhD student of the Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran, in major of water resources management. His research interests include parameter estimation, water distribution systems, optimization problems, and river engineering. He has published several papers in international journals and conference proceedings.

Seied Hosein Afzali received the BS degree from the Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran, in 1986, and the MSc degree from the Department of Civil Engineering, Amir kabir University of Technology, Tehran, Iran, in 1988. His PhD is in hydraulic structures (2008) from the Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran. Dr. Afzali is currently an assistant professor of civil engineering at Shiraz University, Shiraz, Iran. His research interests include groundwater, hydraulic structures, river engineering, optimization problems, and fluid mechanics. He is the author of more than 40 papers in national and international journals and conference proceedings. He is also the first author of a textbook about open channel hydraulics translated in Persian.