Structural Engineering and Mechanics, Vol. 45, No.4 (2013) 521-542 521
DOI: http://dx.doi.org/10.12989/sem.2013.45.4.521

Nonlinear analysis of thin shallow arches subject to snap-through using truss models

H. Xenidis[1a], K. Morfidis[*2] and P.G. Papadopoulos[1b]

[1]Department of Civil Engineering, Aristotle University of Thessaloniki 54124, Greece
[2]Institute of Engineering Seismology and Earthquake Engineering, Thessaloniki 55535, Greece

(Received September 21, 2012, Revised January 8, 2013, Accepted January 11, 2013)

Abstract. In this study a truss model is used for the geometrically nonlinear static and dynamic analysis of a thin shallow arch subject to snap-through. Thanks to the very simple geometry of a truss, the equilibrium conditions can be easily written and the global stiffness matrix can be easily updated with respect to the deformed structure, within each step of the analysis. A very coarse discretization is applied; so, in a very simple way, the high frequency modes are suppressed from the beginning and there is no need to develop a complicated reduced-order technique. Two short computer programs have been developed for the geometrically nonlinear static analysis by displacement control of a plane truss model of a structure as well as for its dynamic analysis by the step-by-step time integration algorithm of trapezoidal rule, combined with a predictor-corrector technique. These two short, fully documented computer programs are applied on the geometrically nonlinear static and dynamic analysis of a specific thin shallow arch subject to snap-through.

Keywords: truss model; static nonlinear analysis; dynamic nonlinear analysis; snap-through; snap-back

1. Introduction

Thin shallow arches are often met in civil, mechanical and aeronautical engineering structures. When the ratios of arch span to its height as well as to its thickness are high, strong geometric nonlinearities due to large displacements appear. For a critical high value of the loading, the arch may be subject to snap-through. By unloading the arch snaps-back, following a different loading path. So, the generalized load-displacement curve exhibits a hysteresis loop (Bradford 2002, Pi 2002, 2008a, b, Chen 2009, Chandra 2009).

Because of the above strong geometric nonlinearities in the structural behavior of a thin shallow arch, its static analysis should be performed by incremental displacement control and its dynamic analysis by step-by-step time integration. Within each step of static or dynamic analysis, the equilibrium equations should be written and the global stiffness matrix should be updated with respect to the deformed structure.

*Corresponding author, Assistant Researcher, E-mail: kmorfidis@itsak.gr
[a]Associate Professor, E-mail: xharis@civil.auth.gr
[b]Assistant Professor, E-mail: panaggpapad@yahoo.gr

The usual finite elements used for the spatial discretization of a structure, present difficulties in describing geometric nonlinearities (Argyris 1978, Taylor 2008, Felippa 2009). Here a truss model is proposed as an alternative to the usual finite elements (Papadopoulos *et al*. 2008a, Papadopoulos *et al*. 2008b, Papadopoulos *et al*. 2011, Papadopoulos *et al*. 2012). Thanks to the very simple geometry of a truss, the equilibrium equations can be easily written and the global stiffness matrix can be easily updated with respect to the deformed truss within each step of static or dynamic analysis, so that to take into account the strong geometric nonlinearities.

Usually, a refined spatial discretization is applied to a structure by a large number of finite elements. As a consequence, in dynamic analysis very high frequencies appear which dictate a very small time steplength of the algorithm and we have to follow a lot of complicated very small vibrations, which are useless to the engineer. For this reason often some rather complicated techniques are developed, the so-called reduced-order techniques, in order to suppress the high vibration modes (Armero and Romero 2001a,b, Przekop and Rizzi 2006, 2007, Bathe 2007, Hollkamp and Gordon 2008, Spottswood *et al.* 2010).

In the present study as an alternative to the above reduced-order techniques, a very coarse spatial discretization of the structure is applied. So, the high frequency modes are suppressed in a very simple way from the beginning. In this way there is no more need to develop afterwards complicated reduced-order techniques to suppress them.

For the geometrically nonlinear static analysis of a thin shallow arch, an algorithm of incremental displacement control is used. Whereas for the dynamic analysis, the step-by-step time integration algorithm of trapezoidal rule is proposed, combined with a predictor-corrector technique with two corrections per step. So, there is no need for solving an algebraic system within each time step of the algorithm.

Based on the above proposed algorithms, two short special purpose computer programs have been developed for the geometrically nonlinear static and dynamic analysis of a plane truss model of a structure, with only approximately 150 and 100 Fortran90 instructions, respectively. These two short, fully documented computer programs, compared to the often used very large general purpose computer programs, exhibit the advantages of more transparency, simplicity and clarity of assumptions. The proposed computer programs for the geometrically nonlinear static and dynamic analysis of a plane truss model of a structure are applied on a specific thin shallow arch subject to snap-through.

2. Presentation of the problem

2.1 Structural system

A thin shallow arch is considered as shown in Fig. 1(a), with fixed both ends, with span L, height H and thickness d, subject to a concentrated vertical load P at its midpoint. In Fig. 1(b) is shown the rectangular thin plate cross-section with width b and depth d. The Young modulus E, as well as, the density ρ of the structural material are given. Linear elastic stress-strain behavior is assumed and only the geometric nonlinearity is taken into account.

2.2 Geometric nonlinearity

When the ratios of the span L of the arch to its height H and to its thickness d are high, large displacements result which imply geometric nonlinearities. So, the static analysis should be performed by incremental loading and the dynamic analysis by a step-by-step time integration algorithm. Within each step of static or dynamic analysis, the equilibrium equations should be written and the global stiffness matrix should be updated with respect to the deformed arch, so that to take into account the geometric nonlinearities.

2.3 Static analysis by displacement control

In Fig. 1(c) is shown a deformed state of the arch, where the ordinate of arch midpoint is noted as y_m. The geometrically nonlinear function $y_m(P)$ is multi-valued, that is to one value of the load P, one up to three values of the midpoint ordinate y_m correspond.

On the contrary, the inverse function $P(y_m)$ is single-valued. Namely, to one value of midpoint ordinate y_m, only one value of the load P corresponds. That is the reason that we prefer to perform the static analysis by displacement control. Then we impose an additional constraint preventing vertical displacement of the midpoint as shown in Fig. 1(d) and we perform the geometrically nonlinear static analysis of the arch, by incremental forced vertical displacement of this support. Where the vertical reaction at this imposed support gives each time the value of the external load P.

2.4 Snap-through

For a specific high value of the load $P = P_{cr}$, called critical load, the arch is subject to snap-through. Then by unloading, the arch following a different loading path may snap-back for a different critical load $P = P'_{cr}$. So, a hysteretic loop is formed on the load -displacement curve P-y_m, as shown in Fig. 1(e).

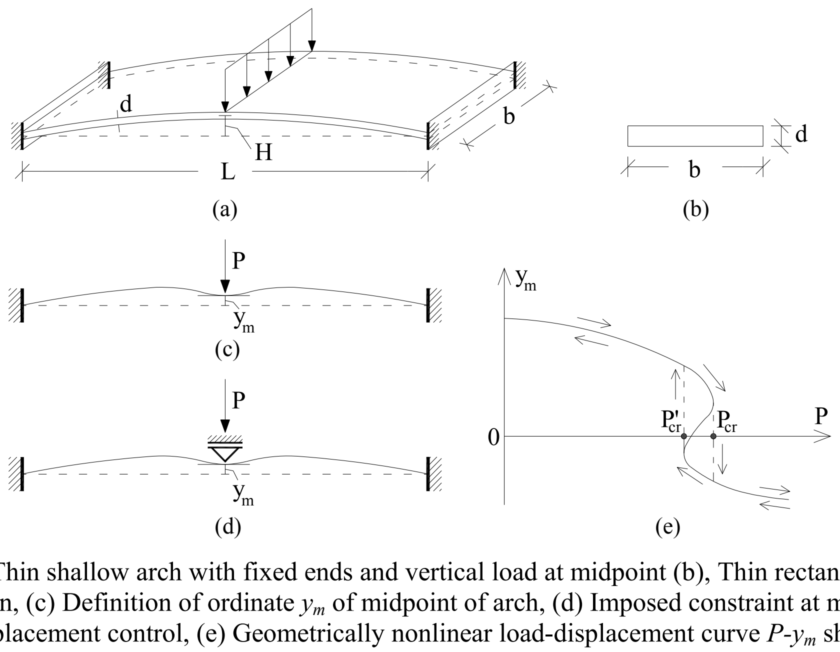

Fig. 1(a) Thin shallow arch with fixed ends and vertical load at midpoint (b), Thin rectangular plate cross-section, (c) Definition of ordinate y_m of midpoint of arch, (d) Imposed constraint at midpoint for vertical displacement control, (e) Geometrically nonlinear load-displacement curve P-y_m showing loading -unloading/snap-through, snap-back and corresponding critical loads

3. Modeling procedure

3.1 The proposed truss model

The whole arch is spatially discretized to only six plane quadrilateral truss elements as shown in Fig. 2(a). Each element results from a plane quadrilateral where all 4 sides and 2 diagonals are bars. Because the arch is very shallow all the six elements can be approximately considered as rectangular, horizontal and equal to each other.

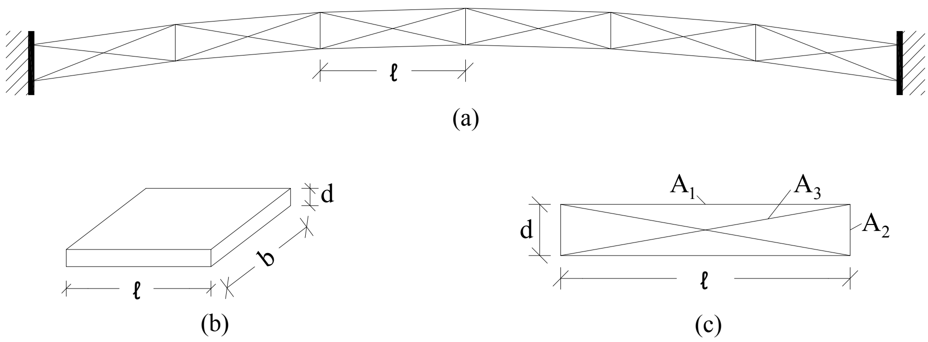

Fig. 2(a) Truss model of an arch, (b) an element of the arch under simulation, (c) a rectangular truss model element

In order to determine the cross-section areas of the bars, we have to compare the stress-strain behavior of an element of the arch as shown in Fig. 2(b), with the stress-strain behavior of the corresponding rectangular truss model as shown in Fig. 2(c). By considering the bending behavior of the element we find the cross-section areas of the longitudinal bars.

So for the arch element (Fig. 3(a)) is valid

$\Delta\varphi = \dfrac{M \cdot \ell}{EI}$ and with $I = \dfrac{b \cdot d^3}{12}$ result $\Delta\varphi = \dfrac{12 \cdot M \cdot \ell}{E \cdot (b \cdot d^3)}$

Respectively for the truss element (Fig. 3(b)) is valid

$\Delta\varphi = \dfrac{2 \cdot \Delta\ell}{d}$ and with $\Delta\ell = \dfrac{N \cdot \ell}{EA} = \dfrac{(M \cdot \ell)/d}{EA_1}$ result $\Delta\varphi = \dfrac{2 \cdot M \cdot \ell}{EA_1 \cdot d^2}$

Therefore

$$A_1 = (b \cdot d)/6 \tag{1}$$

Whereas, by additionally taking into account the axial behavior in longitudinal direction, we find the cross-section areas of the diagonal bars. For the arch element (Fig. 3(c)) is valid

$\Delta\varphi = \dfrac{N \cdot \ell}{E \cdot (b \cdot d)} \cdot (1 - v^2)$ where $(1 - v^2) = 1 - (1/3)^2 \approx 1 \rightarrow \Delta\ell = \dfrac{N \cdot \ell}{E \cdot (b \cdot d)}$

v = Poison’s ratio

Respectively for the truss element (Fig. 3(d)) is valid

$$\Delta\ell = \frac{(N/2)\cdot\ell}{E\cdot[A_3+(b\cdot d)/6]}$$

Therefore

$$A_3 = (b{\cdot}d)/3 \tag{2}$$

By considering the axial behavior in the transverse direction, we find a theoretical value for the cross-section areas of the transverse bars. So, for the arch element (Fig. 3(e)) is valid

$$\Delta\ell = \frac{N\cdot d}{E\cdot(b\cdot\ell)}$$

Respectively for the truss element (Fig. 3(f)) is valid

$$\Delta\ell = \frac{(N/2)\cdot d}{E\cdot A_2}$$

Therefore

$$A_2 = (b{\cdot}\ell)/2 \tag{3}$$

This value of A_2 implies a very high axial stiffness $K = E{\cdot}2A_2/d$ of the transverse bars, which creates in dynamic analysis, very high frequency vibration modes. However, numerical experiments show that, by drastically reducing the above theoretical value of A_2, the resulting error is very small whereas the attained simplification is significant by suppression of the high frequency modes.

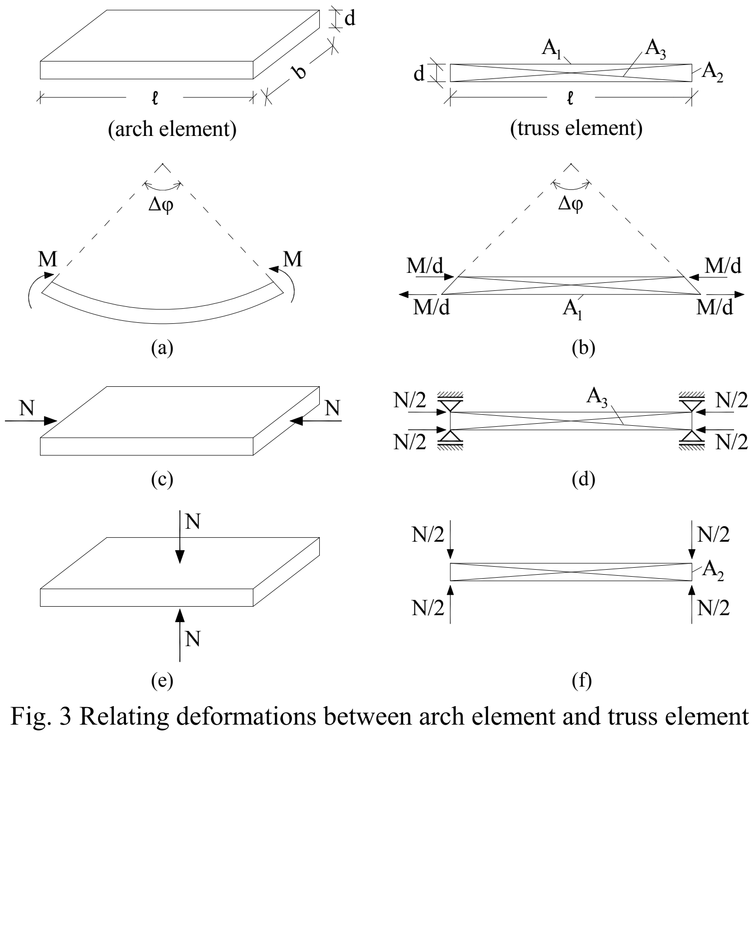

Fig. 3 Relating deformations between arch element and truss element

3.2 Algorithm for static analysis

An additional support, preventing vertical displacement is imposed at a node of truss model at the midpoint of the arch, at the application point of external load P, as shown in Fig. 4(a). An increment of forced vertical displacement $\Delta\upsilon$ of this support is performed, within each step of static analysis algorithm. By using the accurate geometrically nonlinear equations of the truss model, we find the out-of-balance forces $\Delta\mathbf{f}$ on the neighboring free nodes around the midpoint, due to the displacement increment $\Delta\upsilon$. We form the global tangential stiffness matrix $\mathbf{K}$ with respect to the deformed truss. We solve the algebraic system $\mathbf{K}\Delta\mathbf{u} = \Delta\mathbf{f}$ and find the nodal displacements $\Delta\mathbf{u}$ within the present step of the algorithm. We update the coordinates $\mathbf{x}$ of the nodes. Next we find by the nonlinear equations, the nodal forces $\mathbf{f}$ due to the above updated nodal coordinates $\mathbf{x}$. These $\mathbf{f}$ in the supports, give the reactions and in the additional imposed support give the external load P. Whereas in the free nodal DOFs, they had to be zero, however because of truncation error, they exhibit small nonzero values, which are taken as loads in the next step of the algorithm.

3.3 Algorithm for dynamic analysis

To each free node of the truss model as shown in Fig. 4(b), a lumped mass is assigned with value

$$m = (\rho\cdot\ell\cdot b\cdot d)/2 \tag{4}$$

The time-history of the external load *P(t)* is given in input, as shown in Fig. 4(c). Zero damping is assumed.

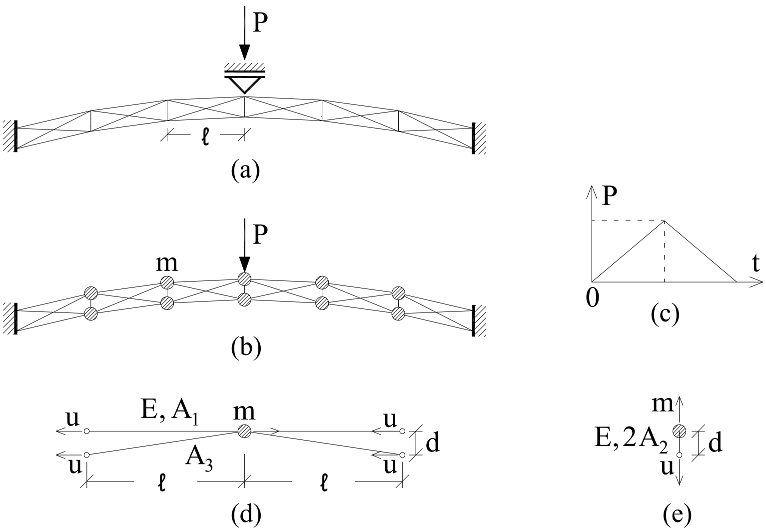

Fig. 4(a) Imposed constraint at midpoint of truss model for displacement control in static analysis, (b) masses m at the free nodes for dynamic analysis, (c) given time-history of external load, (d) vibration mode in longitudinal direction of arch, giving an upper bound for eigenfrequencies, (e) vibration mode in transverse direction of arch, giving an upper bound for eigenfrequencies.

The step-by-step time integration algorithm of trapezoidal rule is used, combined with a predictor-corrector technique with two corrections per step. So, there is no need to solve an algebraic system within each time step of the algorithm. The accuracy criterion of the proposed algorithm is (Papadopoulos *et al.* 2012)

$$\omega_{max}\Delta t < 0.5\text{rad, that is } \Delta t/T_{min} < 1/4\,\pi = 1/12.57 \tag{5}$$

From this criterion, the time steplength of the algorithm is determined.

An upper bound ω_0 for the normal frequencies can be found from the norm of matrix $\mathbf{M}^{-1}\mathbf{K}$, where M the mass matrix and K the stiffness matrix of the structure

$$\omega_0^2 = \left\| M^{-1}K \right\| = \max_i \frac{1}{m_1} \sum_k \left| K_{ik} \right|, \quad i,k = 1,\dots n \tag{6}$$

where $n = 2n_f$ is the number of DOFs and n_f the number of free nodes.

The above upper bound ω_0 for the normal frequencies, corresponds to one of the two vibration modes shown in Figs. 4(d), (e). In the longitudinal direction of the arch we have according to Fig. 4(d)

$$\omega_0^2 = \frac{2E\cdot[2\cdot(A_1 + A_3)/\ell]}{m} \quad \text{where} \quad A_1 = (b\cdot d)/6,\ A_3 = (b\cdot d)/3 \tag{7}$$

Whereas, in the transverse direction of the arch we obtain according to Fig. 4(e)

$$\omega_0^2 = \frac{2E\cdot[2\cdot A_2/d]}{m} \tag{8}$$

We assume equal ω_0 in longitudinal and transverse direction, in order to avoid high frequencies due to large sections of transverse bars. This yields

$$\left.\begin{aligned} \Delta\ell &= \frac{N\cdot\ell}{E\cdot(b\cdot d)}\ (\text{Fig. 3(c)}) \\ \Delta\ell &= \frac{(N/2)\cdot d}{E\cdot A_2}\ (\text{Fig. 3(f)}) \end{aligned}\right\} \Rightarrow A_2 = (b\cdot d^2)/(2\cdot\ell) \tag{9}$$

This value of A_2 is drastically reduced, compared with the theoretical value of A_2 given by Eq. (3). However, as mentioned previously in section 3.1 this causes a very small error, whereas the attained simplification is significant by suppression of high vibration modes.

3.4 The two short computer programs

Based on the two algorithms proposed in the above sections 3.2, 3.3, for the geometrically nonlinear static and dynamic analysis of a truss model of a structure, two short computer programs have been developed, with only about 150 and 100 Fortran90 instructions respectively. The listings of these programs are presented in Appendices A and B.

4. Numerical example

4.1 Given data

The above mentioned computer programs are applied to the geometrically nonlinear static and dynamic analysis of a specific thin shallow arch as shown in Fig. 5(a) with fixed both ends and a vertical concentrated load P at the midpoint. The span of the arch is L = 300mm and its height H = 9mm. The arch axis has a 2nd order parabolic shape described by the equation

$$y = H[1 - (4x^2/L^2)] = 9\text{mm}\ [1\text{-}(4x^2/300^2\text{mm}^2)] \quad (10)$$

as shown in Fig. 5(b), where the ordinates y of the axis are exaggerated with a scale five times larger than horizontal scale, because by the uniform scale of Fig. 5(a) they were hardly visible, as they are very small. In the following, the distorted scale with exaggerated five times larger ordinates y of arch axis will be used.

The thickness of the arch is d = 5mm as shown in Fig. 5(a), whereas the cross-section is a thin rectangular plate with width b = 120mm and depth d = 5mm (Fig. 5(c)). The Young modulus of the structural material (steel) is: $E = \sigma_y/\varepsilon_y = (40\ \text{kN/cm}^2)/0.002 = 2{\cdot}10^4\ \text{kN/cm}^2$, whereas the density of structural material is: ρ = 7.85 t/m^3. A linear elastic stress-strain behavior is assumed and only the geometric nonlinearity is taken into account.

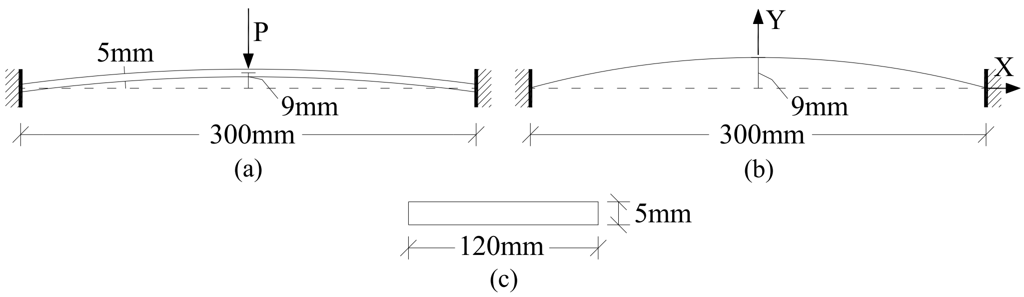

Fig. 5 Given data of the application (a) geometry and loading, (b) parabolic axis of arch with exaggerated ordinates, (c) cross-section

4.2 Cross-section areas of bars

The arch is spatially discretized to only six plane truss elements (Fig. 6(a)). Each element results from a 2D quadrilateral with all its 4 sides and 2 diagonals as bars. As the arch is very shallow, all the six elements can be approximately considered rectangular, horizontal and equal to each other. So, we have to simulate a 3D rectangular solid continuum element (Fig. 6(b)) by a plane rectangular truss (Fig. 6(c)). As mentioned in section 3.1, the cross-section areas are for the longitudinal bars: $A_1 = (bd)/6$ = (120mm·5mm)/6 = 100mm^2, for the diagonal bars: $A_3 = (bd)/3$ = (120mm·5mm)/3 = 200mm^2 and for the transverse bars: $A_2 = (bd^2)/2\ell$ = (120mm·5^{2}mm^2)/(2·50mm) = 30mm^2, which means equal stiffness in the longitudinal and transverse direction of a truss element.

4.3 Static analysis by forced displacement increment

An additional support, preventing the vertical displacement, is imposed at the midpoint of truss model of arch at the application point of the external load P as shown in Fig. 4(a). An incremental forced vertical displacement of this support is performed. Numerical trials show that, starting from $y_m = H = +9$mm and reaching gradually to $y_m \approx -7$mm is enough to give a general picture of the nonlinear S-shaped, load-displacement curve P-y_m and a constant displacement increment (steplength) $\Delta\upsilon = 0.04$mm is satisfactory to accurately describe this nonlinear P-y_m curve. That is, a total number of steps $n_s = [+9\text{mm} - (-7\text{mm})]/\Delta\upsilon = 16\text{mm}/0.04\text{mm} = 400$ are required.

4.4 Static nonlinear load-displacement curve

Based on the output of static analysis, the nonlinear load-displacement curve P-y_m is drawn in Fig. 7, where we can observe the loading procedure, the snap-through and its critical load $P_{cr} \approx$ 14kN, the unloading following a different loading path, the snap-back with its critical load $P'_{cr} \approx$ 13kN and the resulting hysteric loop in the loading-unloading procedure.

We also observe, in the nonlinear static load-displacement curve P-y_m of Fig. 7, that the function $P(y_m)$ is single valued, that is to one value of y_m only one value of P corresponds. Whereas, the inverse function $y_m(P)$ is multi-valued, that is to one value of P, one up to three values of y_m may correspond.

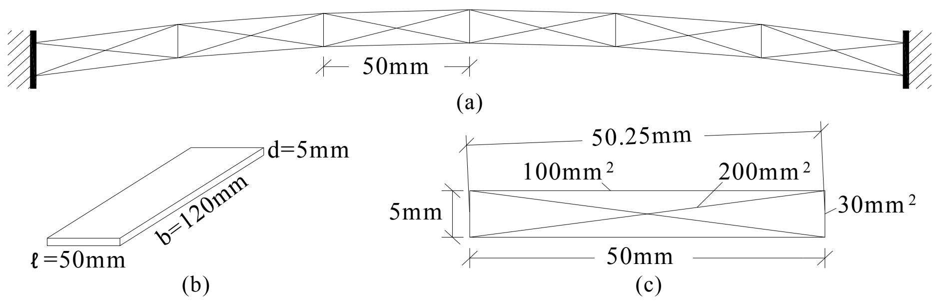

Fig. 6(a) Discretization of the arch by truss model, (b) element of the arch under simulation, (c) cross-section areas of bars of a truss element

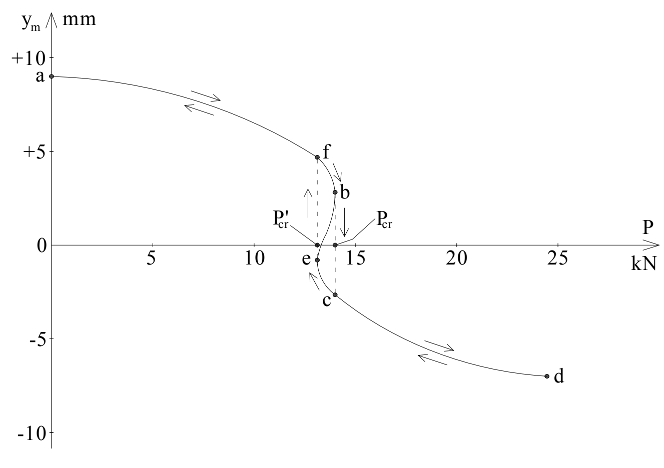

Fig. 7 Nonlinear static load-displacement curve P-y_m

4.5 Characteristic states of static analysis

On the nonlinear static load-displacement curve P-y_m of Fig. 7, six characteristic states of the loading-unloading procedure are noted by the letters (a) up to (f). The deformed configuration, as well as the free body diagram of the arch for the above six characteristic states are drawn in Fig. 8.

4.6 Dynamic analysis. Time steplength

In the dynamic analysis, a lumped mass is assigned to every free node of the truss model as shown in Fig. 4(b) with value:

$m = (\rho\ell bd)/2 = 7.85(\mathrm{t/m^3}) \cdot 0.05\mathrm{m} \cdot 0.12\mathrm{m} \cdot 0.005\mathrm{m}/2 = 0.00011775\mathrm{t}$

From the vibration modes described in section 3.3 and in Figs. 4(d), (e) the same upper bound for eigenfrequencies, in longitudinal and transverse direction of the arch is obtained because of the deliberate choose of a reduced transverse bars section with a value

$$\omega_0^2 = \frac{(2 \cdot E \cdot b \cdot d)}{(\ell \cdot m)} = \left[2 \cdot \left(2 \cdot 10^4 \frac{\mathrm{kN}}{\mathrm{cm}^2}\right) \cdot 12\mathrm{cm} \cdot 0.5\mathrm{cm}\right] / (0.05\mathrm{m} \cdot 0.00011775\mathrm{t}) = 4.076 \cdot 10^{10} \frac{\mathrm{rad}^2}{\mathrm{sec}^2}$$

Thus, $\omega_0 = 2.019 \cdot 10^5$rad/sec and the lower bound of eigenperiods results:

$T_0 = 2\pi/\omega_0 = 0.03112$msec.

The accuracy criterion of the algorithm is

$\omega_0 \Delta t < 0.5$rad that is $\Delta t < T0/4\pi = 0.03112$msec$/12.57 = 0.002476$msec

Finally, a time steplength $\Delta t = 0.0025$msec is chosen. By taking into account the critical load of forward snap-through found previously in static analysis, we design an appropriate time-history P-t of external load which is given in the input of dynamic analysis as shown in Fig. 9(a). In this time-history the load linearly increases up to 15kN, that is a value slightly larger than the critical load for $t = 0.5$msec and then a linear unloading is performed up to zero load for $t = 1.0$msec. So, a total number of time steps $n_s = 1.0\mathrm{msec}/\Delta t = 1.0\mathrm{msec}/0.0025\mathrm{msec} = 400$ are required.

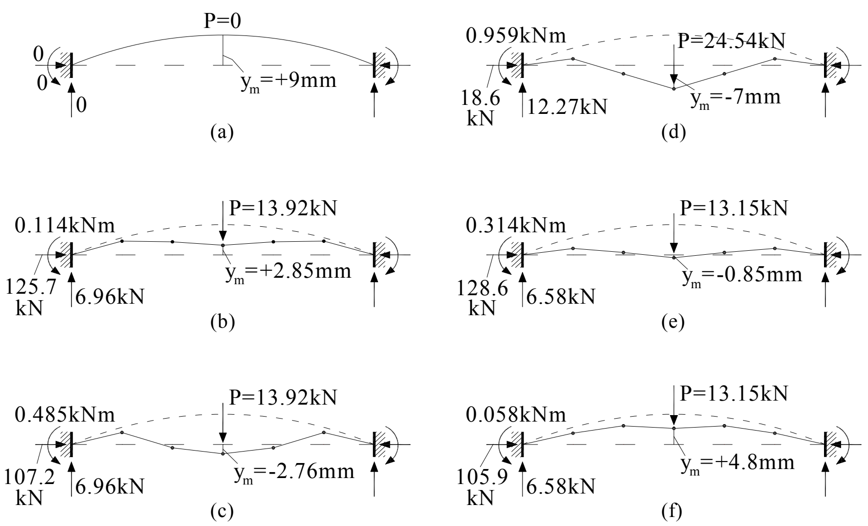

Fig. 8 Characteristic states of static analysis, through the loading-unloading procedure

4.7 Time-gistory of mid-point ordinate ym-t

Based on the output of dynamic analysis, the time-history of arch midpoint ordinate y_m-t is obtained as shown in Fig. 9(b). We observe in this diagram, after unloading an particularly after snap-back, continuous small vibrations. We also observe a minimum eigenperiod $T_{\min} \approx$ 0.035msec, which is close to the predicted value by its lower bound $T_0 = 0.03112$msec, in previous section 4.6.

4.8 Dynamic load-displacement curve

By combining the diagrams P-t and y_m-t of Figs. 9(a), (b) respectively, we obtain the dynamic load-displacement diagram P-y_m as shown in Fig. 9(c) which is different from the static P-y_m diagram of Fig. 7. In this dynamic P-y_m diagram of Fig. 9(c) we observe again, as in previous diagram y_m-t of Fig. 9(b), continuous small vibrations after unloading and particularly after snap-back.

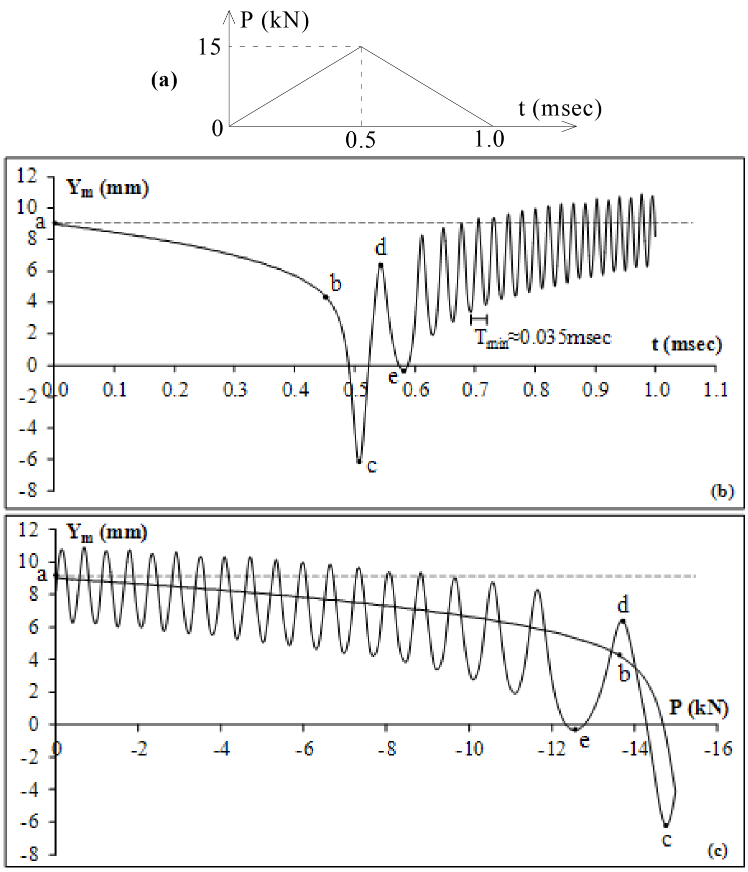

Fig. 9(a) Given time-history of external load, (b) time-history of arch midpoint ordinate y_m-t as a result of dynamic analysis, (c) dynamic load-displacement diagram P-y_m obtained from combination of diagrams P-t, y_m-t

4.9 Characteristic states in successive time Instants

For five characteristic states of the arch, corresponding to successive time instants of dynamic analysis noted in the diagrams y_m-t, P-y_m of Figs. 9(b), (c) by the letters (a) up to (e), the deformed configurations of the arch have been drawn in Fig. 10 along with the corresponding time instants t and the values of the external load P.

By comparing the results of static analysis in Figs. 7 and 8 with the corresponding results of dynamic analysis in Figs. 9 and 10, we observe a reasonable and satisfactory approximation between them.

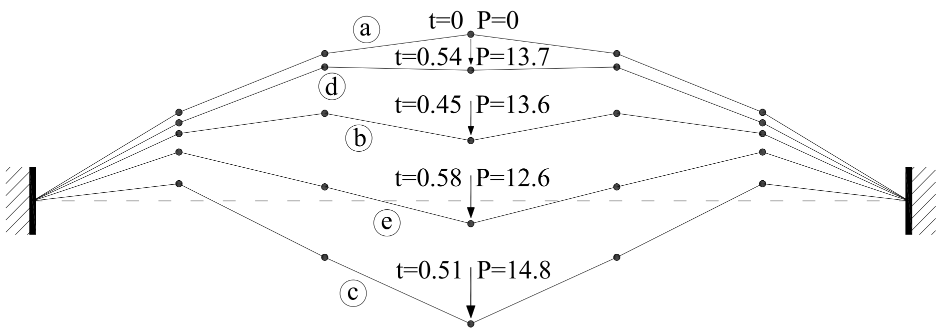

Fig. 10 Deformed configurations of the arch for characteristic states corresponding to successive time instants of dynamic analysis

5. Conclusions

Based on the assumptions and results of this study, the following conclusions can be drawn:

A truss model is used for the spatial discretization of a thin shallow arch subject to snap-through. Thanks to the very simple geometry of a truss, the equilibrium equations can be easily written and the global stiffness matrix can be easily updated with respect to the deformed truss within each step of static or dynamic analysis, so that to take into account the strong geometric nonlinearities.

A very coarse discretization is applied. So, the high frequency modes are suppressed in a very simple way from the beginning and there is no more need to develop afterwards a complicated reduced-order technique to suppress them.

One more technique has been devised to suppress high frequency modes. The theoretical cross-sections of the transverse bars of truss model (across the thickness of the arch) result very large, which means very high transverse stiffness and thus the appearance of very high frequency modes. However, numerical experiments show that by drastically reducing the cross-sections of transverse bars the resulting error is very small, whereas a great simplification is achieved by suppression of high frequency modes.

In static analysis an additional constraint is imposed at the application point of external load and an incremental prescribed displacement of this constraint is performed (displacement control), so that to obtain the nonlinear single-valued load-displacement curve.

Whereas in dynamic analysis, the step-by-step time integration algorithm of trapezoidal rule is used combined with a predictor-corrector technique with two corrections per step. So, no solution of algebraic system is needed within each time step.

Based on the proposed algorithms, two short, special purpose computer programs have been developed for the geometrically nonlinear static and dynamic analysis of a plane truss model of a structure with only approximately 150 and 100 Fortran90 instructions respectively. These two short fully documented computer programs compared to the often used very large general purpose computer programs exhibit the advantages of more transparency, simplicity and clarity of assumptions.

The proposed computer programs for the geometrically nonlinear static and dynamic analysis of a plane truss model of a structure are applied on a specific thin shallow arch with fixed ends and a concentrated load at midpoint subject to snap-through.

The static analysis gives as results, the nonlinear S-shaped generalized load displacement curve, which shows the snap-through, snap-back and their corresponding critical loads as well as the deformed configuration and the free body diagram of the arch for some characteristic states through the loading-unloading procedure.

Whereas, the dynamic analysis gives as results the time-history of the generalized displacement and the dynamic load-displacement curve different from the static one, where vibrations after unloading and snap-back can be observed as well as the deformed configuration of the arch for some characteristic states corresponding to successive time instants.

The results of the static analysis exhibit a reasonable and satisfactory approximation with the corresponding ones of dynamic analysis. So, the proposed method seems to prove useful in the geometrically nonlinear static and dynamic analysis of thin shallow arches subject to snap-through.

References

Argyris, J.H. (1978), International Conferences FENoMech (Finite Elements in Nonlinear Mechanics), Institute for Statics and Dynamics, Stuttgart, I. 1978, II. 1981, III. 1984, *Proceedings in the Journal of Computer Methods in Applied Mechanics and Engineering*.

Armero, F. and Romero, I. (2001a), "On the formulation of high frequency dissipative time-stepping algorithms for nonlinear dynamics. Part I: Low-order methods for two model problems in nonlinear elastodynamics", *Computer Methods in Applied Mechanics and Engineering*, **190**(20-21), 2603-2649.

Armero, F. and Romero, I. (2001b), "On the formulation of high frequency dissipative time-stepping algorithms for nonlinear dynamics.Part II: Second order methods", *Computer Methods in Applied Mechanics and Engineering*, **190**(50-51), 6783-6824.

Bathe, K.J. (2007), "Conserving energy and momentum in nonlinear dynamics: a simple implicit time-integration scheme", *Computer and Structures*, **85**, 437-445.

Bradford, M.A., Uy, B. and Pi, Y. L. (2002), "In plane elastic stability of arches under a central concentrated load", *Journal of Engineering Mechanics ASCE*, **128**(7), 710-719.

Chandra, Y. (2009), "Snap-through of curved beam", Master Thesis, University of Illinois at Urbana-Champaign.

Chen, J.S., Ro, W.C. and Lin, J.S. (2009), "Exact static and dynamic critical load of a sinusoidal arch under a point force at the midpoint", *International Journal of Non-Linear Mechanics*, **44**, 66-70.

Felippa, C.A. (2009), *Introduction to Finite Element Methods* (Class notes), http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html

Hollkamp, J. J. and Gordon, R.W. (2008), "Reduced-order models for nonlinear response prediction: Implicit condensation and expansion", *Journal of Sound and Vibration*, **318**, 1139-1153.

Papadopoulos, P.G., Arethas, I., Lazaridis, P., Mitsopoulou, E. and Tegos, J. (2008a), "A simple method using truss model for in-plane nonlinear static analysis of a cable-stayed bridge with a plate deck section", *Engineering Structures*, **30**(1), 42-53.

Papadopoulos, P.G., Papadopoulou, A.K. and Papaioannou, K.K. (2008b), "Simple nonlinear static analysis of steel portal frame with pitched roof exposed to fire", *Structural Engineering and Mechanics*, **29**(1), 37-53.

Papadopoulos, P.G., Lazaridis, P., Xenidis, H., Lambrou, P. and Diamantopoulos, A. (2012), "Modelling snap-through of thin shallow arches using coarse truss models", Paper 78 from CCP: 96, ISBN 978-1-905088-55-3, *Proceedings of 13th International Conference on Civil, Structural and Environmental Engineering Computing (CC2011),* Platanias, Chania, Crete, Greece, September.

Papadopoulos, P.G., Diamantopoulos, A., Xenidis, H. and Lazaridis, P. (2012), "Simple program to investigate hysteresis damping effect of cross-ties on cables vibration of cable-stayed bridges", *Hindawi Publishing Corporation, Advances in Civil Engineering*, Article ID463134, doi:10.1155/2012/463134.

Pi, Y.L., Bradford, M.A. and Uy, B. (2002), "In-plane stability of arches", *International Journal of Solids and Structures*, **39**, 105-125.

Pi, Y.L. and Bradfordm M.A. (2008a), "Dynamic buckling of shallow pin-ended arches under sudden central concentrated load", *Journal of Sound and Vibration*, **317**, 898-917.

Pi, Y.L., Bradford, M.A. and Tin-Loi, F. (2008b), "Non-linear in-plane buckling of rotationally restrained shallow arches under a central concentrated load", *International Journal of Non-Linear Mechanics*, **43**, 1-17.

Przekop, A. and Rizzi, S.A. (2006), "Nonlinear reduced-order response analysis of structures with shallow curvatures", *AIAA Journal*, **44**(8), 1767-1778.

Przekop, A. and Rizzi, S.A. (2007), "Dynamic snap-through of thin-walled structures by a reduced-order method", *AIAA Journal*, **45**(10), 2510-2519.

Spottswood, S.M., Hollkamp, J.J. and Eason, T.G. (2010), "Reduced-order models for a shallow curved beam under combined loading", *AIAA Journal*, **48**(1), 47-55.

Taylor, R.L. (2008), "FEAP, A Finite Element Analysis Program", Department of Civil and Environmental Engineering, University of California at Berkeley, version 8.2.

Appendix A: Listing of the static analysis program

TABLE A.1 Main program

```
Program STATIC_SNAP_THROUGH ! MAIN PROGRAM
   use DATA_MODULE
   implicit none
   Integer(4) K, I, K1, K2, K3, STEP, L, R
   Real(8) U0, SUXP, SUYP, YM, PM, STIF(30,30), P(30), U(30), PX(15), PY(15), FX(15),
FY(15)
   OPEN(100,FILE="G:\INPUT_STATIC.txt")  ! Open input data file for static analysis
   OPEN(200,FILE="G:\OUTPUT_STATIC.txt") ! Open output data file for static analysis
   ! READING INPUT
   READ(100, '(1x,I2,1x,I2,1x,F8.1,1x,I3,1x,I2)') NN,NB,ELAST0,NSTEP,NK
   Do K=1,NN
      READ(100, '(1x,I1,1x,I1,1x,F8.1,1x,F8.2,1x,F8.2,1x,F8.2,1x,F8.2,1x,F8.2)' ) &
      KX(K), KY(K), X0(K), Y0(K), PX(K), PY(K), UX0(K), UY0(K)
   END Do
   Do I=1,NB ; READ(100, '(1x,I2,1x,I2,1x,F6.2)') KL(I),KR(I),A(I) ; End Do
   Do I=1,NB
      L=KL(I) ; R=KR(I) ; LX=X0(R)-X0(L) ; LY=Y0(R)-Y0(L)
      L0(I)=SQRT((LX**(2.))+(LY**(2.))) ; ELAST(I) = ELAST0
   End Do
   Do K=1,NK ; READ(100, '(1x,I5,1x,F6.2)') STEPK(K), UK(K) ; End Do
   ! INITIAL CONTITIONS
   NN2=2*NN ; STEP = 0
   DO K=1,NN ; X(K)=X0(K) ; Y(K)=Y0(K) ; SUX(K)=0.0 ; SUY(K)=0.0 ; End Do
   Call NONL(PX,PY,FX,FY)
   ! ITERATION PROCEDURE
   WRITE (200,'(2x,A,7x,A,9x,A,A/)') "Step","PM","YM"
   Do STEP=1,NSTEP
      call UHIST(STEP, U0)
      Do K=1,NN
         If(KX(K).EQ.2) then
            SUXP=SUX(K)    ;    SUX(K)=UX0(K)*U0    ;    UX(K)=SUX(K)-SUXP    ;
X(K)=X0(K)+SUX(K)
         End If
         If(KY(K).EQ.2) then
            SUYP=SUY(K)    ;    SUY(K)=UY0(K)*U0    ;    UY(K)=SUY(K)-SUYP    ;
Y(K)=Y0(K)+SUY(K)
         End If
      End Do
      Call STIFF(STIF)
      Call NONL(PX,PY,FX,FY)
      Do K1=1,NN ; P(2*K1-1)=FX(K1) ; P(2*K1)=FY(K1) ; End Do
         Call GAUS(STIF,P,U)
```

```
        Do K2=1,NN
          If(KX(K2).EQ.0) then
            UX(K2)=U(2*K2-1) ; SUX(K2)=SUX(K2)+UX(K2) ; X(K2)=X(K2)+UX(K2)
          End If
          If(KY(K2).EQ.0) then
            UY(K2)=U(2*K2) ; SUY(K2)=SUY(K2)+UY(K2) ; Y(K2)=Y(K2)+UY(K2)
          End If
        End Do
        Do K3=1,NN ; FX(K3)=PX(K3) ; FY(K3)=PY(K3) ; End Do
        Call NONL(PX,PY,FX,FY)
        YM=(Y(7)+Y(8))/2.0 ; PM=FY(7)+FY(8)
        ! PRINTING RESULTS
        WRITE (200, '(1x,I4,1x,F10.2,1x,F10.2)') STEP, PM, YM
    End Do
    CLOSE(100) ; CLOSE(200)
End Program
```

TABLE A.2 Data Module

```
Module DATA_MODULE
implicit none
Integer(4) NK,NB,NSTEP,NN,NN2,KX(15),KY(15),KL(30),KR(30),STEPK(3)
Real(8)  A(30),L0(30),X0(15),Y0(15),UX0(15),UY0(15),E(30),S(30),LE(30),UK(3),  ELAST0,
LX, LY
Real(8) ELAST(30),X(15),Y(15),CX(30),CY(30),N(30),UX(15),UY(15),SUX(15),SUY(15)
End module
```

TABLE A.3 Subroutine NONL

```
Subroutine  NONL(PX,PY,FX,FY)  !  SUBROUTINE  FOR  NONLINEAR  EQUATIONS  OF
PROBLEM
    use DATA_MODULE
    implicit none
    Integer(4) K, I, L, R
    Real(8) DL, PX(15), PY(15), FX(15), FY(15)
    Do K=1,NN ; FX(K)=PX(K) ; FY(K)=PY(K) ; End Do
    Do I=1,NB
      L=KL(I) ; R=KR(I) ; LX=X(R)-X(L) ; LY=Y(R)-Y(L) ; LE(I)=SQRT(LX**2+LY**2)
      CX(I)=LX/LE(I) ; CY(I)=LY/LE(I) ; DL=LE(I)-L0(I)
      E(I)=DL/L0(I) ; S(I)=ELAST0*E(I)
      ELAST(I)=ELAST0 ; N(I)=S(I)*A(I) ; FX(L)=FX(L)+N(I)*CX(I)
      FY(L)=FY(L)+N(I)*CY(I) ; FX(R)=FX(R)-N(I)*CX(I) ; FY(R)=FY(R)-N(I)*CY(I)
    End Do
End Subroutine
```

TABLE A.4 Subroutine STIFF

```
Subroutine  STIFF(STIF)  !  SUBROUTINE  FOR  THE  FORMATION  OF  THE  STIFFNESS
MATRIX
```

```
   use DATA_MODULE
   implicit none
   Integer(4) I, K, L, R
   Real(8) STEL0, STELX, STELY, STELXY, STGE0, STGEX, STGEY, STGEXY,
STIF(30,30)
   Do I=1,NN2
      Do K=1,NN2 ; STIF(I,K)=0.0 ; End Do
   End Do
   Do I=1,NB
      STEL0=ELAST(I)*A(I)/L0(I) ; STELX=STEL0*CX(I)**(2.) ;
STELY=STEL0*CY(I)**(2.)
      STELXY=STEL0*CX(I)*CY(I) ; STGE0=N(I)/LE(I) ; STGEX=STGE0*CY(I)**(2.)
      STGEY=STGE0*CX(I)**(2.) ; STGEXY=-STGE0*CX(I)*CY(I) ; L=KL(I) ; R=KR(I)
      STIF(2*L-1,2*L-1)=STIF(2*L-1,2*L-1)+STELX  + STGEX
      STIF(2*L-1,2*L  )=STIF(2*L-1,2*L  )+STELXY + STGEXY
      STIF(2*L  ,2*L-1)=STIF(2*L  ,2*L-1)+STELXY + STGEXY
      STIF(2*L  ,2*L  )=STIF(2*L  ,2*L  )+STELY  + STGEY
      STIF(2*L-1,2*R-1)=STIF(2*L-1,2*R-1)-STELX  - STGEX
      STIF(2*L-1,2*R  )=STIF(2*L-1,2*R  )-STELXY - STGEXY
      STIF(2*L  ,2*R-1)=STIF(2*L  ,2*R-1)-STELXY - STGEXY
      STIF(2*L  ,2*R  )=STIF(2*L  ,2*R  )-STELY  - STGEY
      STIF(2*R-1,2*L-1)=STIF(2*R-1,2*L-1)-STELX  - STGEX
      STIF(2*R-1,2*L  )=STIF(2*R-1,2*L  )-STELXY - STGEXY
      STIF(2*R  ,2*L-1)=STIF(2*R  ,2*L-1)-STELXY - STGEXY
      STIF(2*R  ,2*L  )=STIF(2*R  ,2*L  )-STELY  - STGEY
      STIF(2*R-1,2*R-1)=STIF(2*R-1,2*R-1)+STELX  + STGEX
      STIF(2*R-1,2*R  )=STIF(2*R-1,2*R  )+STELXY + STGEXY
      STIF(2*R  ,2*R-1)=STIF(2*R  ,2*R-1)+STELXY + STGEXY
      STIF(2*R  ,2*R  )=STIF(2*R  ,2*R  )+STELY  + STGEY
   End Do
   Do K=1,NN
      IF(KX(K).NE.0) STIF(2*K-1,2*K-1)=1.E+10
      IF(KY(K).NE.0) STIF(2*K ,2*K )=1.E+10
   End Do
End Subroutine
```

TABLE A.5 Subroutine UHIST

```
Subroutine UHIST(STEP, U0)   ! SUBROUTINE FOR HISTORY OF PRESCRIBED
DISPLACEMENTS
   use DATA_MODULE
   implicit none
   Integer(4) K, STEP
   Real(8) U0
   Do K=1, NK-1
      IF((STEP - STEPK(K))*(STEP-STEPK(K+1)).GT.0) EXIT
      U0=UK(K)+(UK(K+1)-UK(K))/(STEPK(K+1)-STEPK(K))*(STEP-STEPK(K))
```

```
   End Do
End Subroutine
```

TABLE A.6 Subroutine GAUS

```
Subroutine GAUS(A1,B1,X1) ! SUBROUTINE TO SOLVE THE ALGEBRAIC SYSTEM
   use DATA_MODULE
   implicit none
   Integer(4) NM1, I, I1, J1, K, IL, J, N1
   Real(8) COEF, A1(30,30), B1(30), X1(30)
   N1=NN2 ; NM1=N1-1
   Do I=1,NM1
      I1=I+1
      Do J=I1,N1
         COEF=-A1(J,I)/A1(I,I) ; B1(J)=B1(J)+B1(I)*COEF
         Do K=1,N1 ; A1(J,K)=A1(J,K)+A1(I,K)*COEF ; End Do
      End Do
   End Do
   X1(N1)=B1(N1)/A1(N1,N1)
   Do I=1, NM1
      IL=N1-I ; I1=IL+1 ; X1(IL)=B1(IL)
      Do J=I1,N1 ; X1(IL)=X1(IL)-A1(IL,J)*X1(J) ; End Do
      X1(IL)=X1(IL)/A1(IL,IL)
   End Do
End Subroutine
```

Description of the basic parameters

Input parameters

NN = The number of nodes of the arch.
NB = The number of bar elements of the arch.
NK = The number of points of static Load-Displacement curve.
ELAST0 = Initial Modulus of Elasticity
NSTEP = The number of steps of the iteration procedure.
[KX(NN)], [KY(NN)] = Matrices with elements which indicate the type of restraint of nodes along X and Y axes: KX(or KY)=0 for unrestrained nodes, KX(or KY)=1 for restrained nodes, KX(or KY)=2 for nodes with forced displacement.
[X0(NN)], [Y0(NN)] = Initial coordinates of the nodes.
[PX(NN)], [PY(NN)] = External forces of nodes.
[UX0(NN)], [UY0(NN)] = Matrices which include the factors of the forced displacements: UX0 (or UY0)=1.00 for nodes with forces displacement. UX0 (or UY0)=0.00 for the other nodes.
[KL(NB)] = The index number of the node of the left edge of each bar.
[KR(NB)] = The index number of the node of the right edge of each bar.
[A(NB)] = Cross-section area of each bar.
[STEPK(NK)] = The number of the step which corresponds to each point of Load-Displacement curve.
[UK(NK)] = The value of forced displacement which corresponds to each point of Load-

Displacement curve.

Output parameters
PM= The value of the concentrated force in the middle of the arch.
YM= The value of the forced displacement in the middle of the arch.

Appendix B: Listing of the dynamic analysis program

TABLE B.1 Main program

```
Program DYNAMIC_SNAP_THROUGH ! MAIN PROGRAM
   use DATA_MODULE
   implicit none
   Integer(4) K,I,K1,K2,K3,K4
   Real(8)
GX(15),GY(15),X(15),Y(15),T,P0,XP(15),YP(15),GXP(15),GYP(15),X1(15),Y1(15),GX1(15),
GY1(15)
   Real(8) YM,PM,VX(15),VY(15),VXP(15),VYP(15),VX1(15),VY1(15)
   OPEN(100,FILE="G:\INPUT_DYNAMIC.txt")  ! Open input data file for dynamic analysis
   OPEN(200,FILE="G:\OUTPUT_DYNAMIC.txt")  ! Open output data file for dynamic
analysis
   ! READING INPUT
   READ(100, '(1x,I2,1x,I2,1x,F8.1,1x,F6.3,1x,F6.1,1x,I2)') NN,NB,ELAST0,DT,TMAX,NK
   Do K=1,NN
      READ(100, '(1x,I1,1x,I1,1x,F8.1,1x,F8.2,1x,F8.2,1x,F8.2,1x,F8.2,1x,F8.2,1x,F8.2)' ) &
      KX(K),KY(K),X(K),Y(K),PX(K),PY(K),PX0(K),PY0(K),M(K)
   END Do
   Do I=1,NB ; READ(100, '(1x,I2,1x,I2,1x,F6.2)') KL(I),KR(I),A(I) ; End Do
   Do K1=1,NK ; READ(100, '(1x,F6.2,1x,F8.2)') TK(K1),PK(K1) ; End Do
   ! INITIAL CONTITIONS
   STEP=0 ; T=0.0
   Do I=1,NB ; L=KL(I) ; R=KR(I) ; LX=X(R)-X(L) ; LY=Y(R)-Y(L) ;
L0(I)=SQRT(LX**2+LY**2) ; End Do
   Do K=1,NN ; VX(K)=0.0 ; VY(K)=0.0 ; End Do
   CALL EVAL(X,Y,GX,GY)
   ! ITERATION PROCEDURE
   WRITE (200,'(4x,A,14x,A,10x,A,A/)') "T","PT","YM"
   Do T=0.002,TMAX,DT
      CALL PHIST(T,P0)
      Do K=1,NN
         If(KX(K).EQ.2) then ; PX(K)=PX0(K)*P0 ; End If
         If(KY(K).EQ.2) then ; PY(K)=PY0(K)*P0 ; End If
      End Do
         Do K2=1,NN ! PREDICTION
            XP(K2)=X(K2) ; YP(K2)=Y(K2) ; VXP(K2)=0.0 ; VYP(K2)=0.0
            If(KX(K2).NE.1)        then        ;        XP(K2)=X(K2)+VX(K2)*DT        ;
```

```
VXP(K2)=VX(K2)+GX(K2)*DT ; End If
            If(KY(K2).NE.1)           then            ;            YP(K2)=Y(K2)+VY(K2)*DT           ;
VYP(K2)=VY(K2)+GY(K2)*DT ; End If
         End Do
         CALL EVAL(XP,YP,GXP,GYP)
      Do K3=1,NN ! FIRST CORRECTION
         X1(K3)=X(K3) ; Y1(K3)=Y(K3) ; VX1(K3)=0.0 ; VY1(K3)=0.0
         If(KX(K3).NE.1) then
            X1(K3)=X(K3)+(VX(K3)+VXP(K3))/2.*DT                                    ;
VX1(K3)=VX(K3)+(GX(K3)+GXP(K3))/2.*DT
         End If
         If(KY(K3).NE.1) then
            Y1(K3)=Y(K3)+(VY(K3)+VYP(K3))/2.*DT                                    ;
VY1(K3)=VY(K3)+(GY(K3)+GYP(K3))/2.*DT
         End If
      End Do
      CALL EVAL(X1,Y1,GX1,GY1)
      Do K4=1,NN ! SECOND AND LAST CORRECTION
         If(KX(K4).NE.1) then
            X(K4)=X(K4)+(VX(K4)+VX1(K4))/2.*DT                                     ;
VX(K4)=VX(K4)+(GX(K4)+GX1(K4))/2.*DT
         End If
         IF(KY(K4).NE.1) then
            Y(K4)=Y(K4)+(VY(K4)+VY1(K4))/2.*DT                                     ;
VY(K4)=VY(K4)+(GY(K4)+GY1(K4))/2.*DT
         End If
      End Do
      CALL EVAL(X,Y,GX,GY)
      YM=(Y(7)+Y(8))/2.0 ; PT=PY(7)+PY(8)
      ! PRINTING RESULTS
      WRITE (200, '(1x,F6.3,1x,F15.5,1x,F10.4)') T, PT, YM
      STEP = STEP + 1
   End Do
      CLOSE(100) ; CLOSE(200)
End Program
```

TABLE B.2 Data Module

```
Module DATA_MODULE
implicit none
Integer(4) NK,NN,NB,STEP,L,R,KL(30),KR(30),KX(15),KY(15)
Real(8) ELAST0,DT,TMAX,LX,LY,L0(30),FX(15),FY(15),E(30),ELAST(30),S(30),N(30),PT
Real(8) A(30),PX(15),PY(15),PX0(15),PY0(15),M(15),TK(5),PK(5)
End module
```

TABLE B.3 Subroutine PHIST

```
Subroutine PHIST(T,P0) ! SUBROUTINE FOR LOADING HISTORY
```

```
   use DATA_MODULE
   implicit none
   Integer(4) I
   Real(8) T, P0
   Do I=1,NK-1
      If((T-TK(I))*(T-TK(I+1)).GT.0.) CYCLE
      P0=PK(I)+(PK(I+1)-PK(I))/(TK(I+1)-TK(I))*(T-TK(I))
   End Do
End Subroutine
```

TABLE B.4 Subroutine EVAL

```
Subroutine EVAL(X,Y,GX,GY) ! SUBROUTINE TO EVALUATE THE PRESENT STATE
OF THE STRUCTURE
   use DATA_MODULE
   implicit none
   Integer(4) K, I
   Real(8) GX(15),GY(15),X(15),Y(15),LE,CX,CY,DL
   Do K=1,NN
      FX(K)=PX(K) ; FY(K)=PY(K)
   End Do
   Do I=1,NB
     L=KL(I) ; R=KR(I) ; LX=X(R)-X(L) ; LY=Y(R)-Y(L)
     LE=SQRT(LX**2+LY**2) ; DL=LE-L0(I) ; E(I)=DL/L0(I)
     ELAST(I)=ELAST0 ; S(I)=ELAST(I)*E(I) ; N(I)=S(I)*A(I)
     CX=LX/LE ; CY=LY/LE
     FX(L)=FX(L)+N(I)*CX ; FY(L)=FY(L)+N(I)*CY
     FX(R)=FX(R)-N(I)*CX ; FY(R)=FY(R)-N(I)*CY
   End Do
   DO K=1,NN
      GX(K)=0.0 ; GY(K)=0.0
      If(KX(K).NE.1) then
         GX(K)=FX(K)/M(K)
      End If
      If(KY(K).NE.1) then
         GY(K)=FY(K)/M(K)
      End If
   End Do
End Subroutine
```

Description of the basic parameters

Input parameters

NN, NB, NK, ELAST0, [KX(NN)], [KY(NN)], [X0(NN)], [Y0(NN)], [PX(NN)], [PY(NN)], [KL(NB)], [KR(NB)], [A(NB)], see the description of parameters of the program of static analysis.
[PX0(NN)], [PY0(NN)] = Matrices which include the factors of the external forces: PX0 (or PY0)=1.00 for nodes with external forces. PX0 (or PY0)=0.00 for the other nodes.

DT= The time step.
TMAX= Total time of dynamic analysis.
[M(NN)]= Matrix of masses of nodes.
[TK(NK)]= Time which corresponds to each point of Load-Displacement curve.
[PK(NK)]= The value of force which corresponds to each point of dynamic Load-Displacement curve.

Output parameters

PT= The value of the concentrated force in the middle of the arch.
YM= The value of the displacement in the middle of the arch.