www.lajss.org
Latin American Journal of Solids and Structures 1 (2004) 401–422

Two-dimensional analysis of slender structures using high order boundary elements

Luiz Eduardo T. Ferreira

Escola Politécnica, Universidade de São Paulo, Brazil.
Rua João de Sousa Dias, 520, a 123, C. Belo, São Paulo, 04318-006, Brazil.

Abstract

This paper addresses the use of higher order elements in 2D boundary element analysis of slender components subjected to bending. Firstly, the technology associated to an efficient use of high order boundary element is discussed. Subsequently, effective schemes for outside and inside integration of the kernel tensor functions over these macro elements, involving element subdivision, are provided, so that they can be efficiently employed to solve bending problems involving high order variation of displacements. Through numerical experiments, it is finally shown that the BE method can be successfully applied to analyze slender structures in bending with higher order elements, if specialized techniques are employed and the structural behavior is considered *a priori*, so that appropriate elements can be chosen to solve a particular problem.

Keywords: high order elements, inside integration, element subdivision, slender structures.

1 Introduction

The use of high order elements in boundary element analysis of elasticity problems is usually restricted to a few researchers working with numerical methods. Perhaps, this is due to the fact that specialized tools like specific programs or particular routines to deal with these elements, in addition to the difficulties associated with the traditional implementation of the method, many times do not reach the practicing engineers and the graduate students as it should.

On the other hand, available codes usually restrict the annalist to the use of linear and quadratic elements. These elements, for their very nature, are not away appropriated to the stress analysis of structures in bending, especially those involving slender components and complex displacements fields. To accomplish accuracy in these cases, higher order elements are required. Herein, by higher order elements, or simply HO elements, one will presume boundary elements with an unlimited number of nodes.

However, when high order elements are concerned, four main flexibility premises need to be satisfied while implementing the method. Firstly, the implementation will have to permit

[*] Corresponding author Email: leferrei@uol.com.br Received 30 August 2004; In revised form 22 October 2004

boundary element meshes to be organized with elements of different order. Secondly, no restrictions regarding the order of the element to be used must exist. In third place, the nodes of the element won't have to be equally spaced, so that a single HO element may be used, for example, to graduate the mesh in a given position of interest. Finally, an efficient scheme of numerical integration of the kernel tensor functions over the elements must be considered, so that accuracy of the result can be attained.

To fulfill these principles, in what follows, specialized procedures to deal with the differential geometry of the element are referred to and specific routines to deal with these elements efficiently, are provided. Subsequently, efficient schemes to treat inside and outside integration of the kernel tensor function are presented and discussed.

Finally, two numerical studies of slender structures subjected to bending were carried out. Through these studies it is shown that accuracy of results can be accomplished if the displacements responses are considered *a priori*, so that the appropriate element can be adequately chosen to solve a given problem. In this paper, only isoparametric elements will be considered, and for conciseness, only elasticity problems are addressed.

2 Interpolation with HO elements

In order to satisfy the above mentioned flexibility with HO elements, a versatile way to deal with the differential geometry of two-dimensional boundary elements, in its generalized form, will have to be considered, so that the traditional sets of equations for the shape functions and their derivatives will give place to more generalist procedures. Thus, the shape functions $\boldsymbol{N}_i$ of a HO element will be computed by expanding the general expression for the Lagrange polynomials [8, 12]:

$$N_i\left(\boldsymbol{\xi}\right) = \prod_{j=1, j\neq i}^{k} \frac{\boldsymbol{\xi} - \boldsymbol{\xi}_j}{\boldsymbol{\xi}_i - \boldsymbol{\xi}_j} \tag{1}$$

where $N_i(\boldsymbol{\xi})$ is the value of the shape function associated with node $\boldsymbol{i}$ of the element, $\boldsymbol{\xi}_i$ is the value of the intrinsic or normalized coordinate at this node and $\boldsymbol{\xi}_j$ are the values of the intrinsic coordinates at other nodes. The symbols $\prod$ stand for the product operator.

Analogously, the shape functions derivatives for an element with $\boldsymbol{k}$ nodes, will be given by:

$$\frac{\partial N_i\left(\xi\right)}{\partial \xi} = \frac{1}{\prod\limits_{\substack{j=1\\(j\neq i)}}^{k} \left(\xi_i - \xi_j\right)} \left(\sum_{n=i}^{k-1} \left[\prod_{\substack{j=1\\(j\neq i,\ j\neq n+1)}}^{k} \left(\xi - \xi_j\right) \right] + \sum_{n=1}^{i-1} \left[\prod_{\substack{j=i\\(j\neq i;\ j\neq n)}}^{k} \left(\xi - \xi_j\right) \right] \right) \tag{2}$$

where $\partial \boldsymbol{N_i}\ (\boldsymbol{\xi})/\ \partial\boldsymbol{\xi}$ is the value of the shape function derivative associated with node $\boldsymbol{i}$ of the element, $\boldsymbol{\xi_i}$ is the value of the intrinsic or normalized coordinate $\boldsymbol{\xi}$ at this node and $\boldsymbol{\xi}_j$

are the values of the intrinsic coordinates at other nodes. The symbol $\mathbf{\Sigma}$ stands for the summation operators.

Two subroutines to deal with equation (1) and (2) were presented by the author in a companion paper [6], however, to deal with elements formed by equally spaced nodes. To deal with elements having its nodes placed in a graduated fashion, two complete subroutines to compute the shape functions and their derivatives, written in FORTRAN95 language, are presented in Appendix A and B of this paper.

The main entries to be used with these routines are the extremes **a** and **b**, defining the intrinsic space (e.g. -1 and 1, 0 and 1, etc), the number of nodes in the element, **k**, and $\boldsymbol{R_t}$, the ratio between two adjacent nodes of the element. If equally spaced nodes are to be considered, the parameter $\boldsymbol{R_t}$ will be unit.

3 Integration of the kernel tensor functions

As a very central part of the boundary element method, the integration process of the kernel tensor functions, particularly when HO elements are employed, will have to ensure a certain consistency with regard to the length of each HO element and the number of Gauss points employed in the quadrature processes. For this purpose, in what follows equivalence between the integration of these functions along a single quadratic element and along all the quadratic elements that could be formed using the nodes of the HO element, will be evoked as a first approach.

3.1 Outside integration

It occurs when the source point of the fundamental solution doesn't coincide with one of the nodes of the element that is being integrated, so that none of the kernel tensor functions for displacements will presents singular behavior. Consequently, also when solving the internal point responses, full outside integration are considered. Within these cases, if a standard Gauss quadrature that integrates the whole intrinsic space without special considerations is employed, the contribution of a given boundary element, $\boldsymbol{U^e}$, can be written in a general form, for a given load point **P**, as:

$$U^e_{KLmj} \approx \sum_{n=1}^{ng} N_m\left(x_n\right)\ U_{KL}\left(P_j, x_n\right)\ J\left(x_n\right)\omega_n \tag{3}$$

where $\boldsymbol{x_n}$ and $\boldsymbol{\omega_n}$ are the Gauss Points coordinates and their corresponding weighting, $\boldsymbol{N_m(x_n)}$ is the shape function associated with node $\boldsymbol{m, J(x_n)}$, the Jacobian of the transformation, $\boldsymbol{P_j}$ is the unit load applied at **j** direction and **ng**, the number of Gauss points used.

Note that in the above equation the directions of the unit load and the displacements, as well as node number and collocation point order have to be observed for each element.

To consider all the collocation points, **ncp**, in the model, as well as the number of degrees of freedom per node, **d**, equation (3) can be rewritten in an expanded form:

$$U_{kl}^{e} \approx \sum_{r=1}^{ncp} \sum_{n=1}^{ng} \sum_{m=1}^{k} \sum_{j=1}^{d} N_m \left(x_{(n)} \right) \; U_{kl} \left(P_j \left(r \right), x_{(n)} \right) \; J \left(x_{(n)} \right) \; \omega_n \; J \tag{4}$$

On the other hand, the performance expected while using higher order elements will require a reasonable equilibrium between the number of elements used in the discretization process and the order of the quadrature employed. For example, cubic elements can be used instead of quadratic ones, to cover larger spaces with better performance, without necessarily increasing the number of unknowns in the problem. This can be accomplished just regrouping adjacent nodes in order to form higher order elements.

However, in many circumstances, higher order quadratures will be required in conjunction, since not only the response, but also the geometrical quantities need to be computed accurately.

To circumvent the inconvenient of using elevated quadrature orders, the integration process can be worked out in the inverse sense, i. e., by reducing the integration space, within the so called element subdivision process.

An attractive way to implement this technique is to break the element with **k** nodes, into $\boldsymbol{k-1}$ parts, so that each subinterval of integration, $\boldsymbol{S_i}$, will be the space confined between two consecutive nodes. Figure 1 shows a partial BE mesh of an arbitrary plane object, where elements of different orders have been employed in the discretization process. In the same figure, element E6 is also shown on the intrinsic space in (a) and subintervals of integration, $\boldsymbol{S_i}$, for this element in (b). For clarity, only the outer boundary was meshed.

To apply the above mentioned strategy, equation (3) and (4) can be rewritten to consider the subdivision process, as follows:

$$U_{kl}^{e} \approx \sum_{i=1}^{k-1} \sum_{r=1}^{ncp} \sum_{n=1}^{ng} \sum_{m=1}^{k} \sum_{j=1}^{d} N_m \left(x_{T(n)} \right) \; U_{kl} \left(P_j \left(r \right), x_{T(n)} \right) \; J \left(x_{T(n)} \right) \; \omega_n \; J_T \tag{5}$$

where $\boldsymbol{x_{T(n)}}$ are the transformed Gauss coordinates to be used in the integration process, computed with:

$$x_{T(n)} = \frac{\left(\xi_{i+1} + \xi_i \right) + \left(\xi_{i+1} - \xi_i \right) x_n}{2} \tag{6}$$

The Jacobian of the primary transformation occurring from the intrinsic space to the subspace of the confined interval will be given by:

$$J_T = \frac{\partial x_{T(n)}}{\partial x} = \frac{\xi_{i+1} - \xi_i}{2} \tag{7}$$

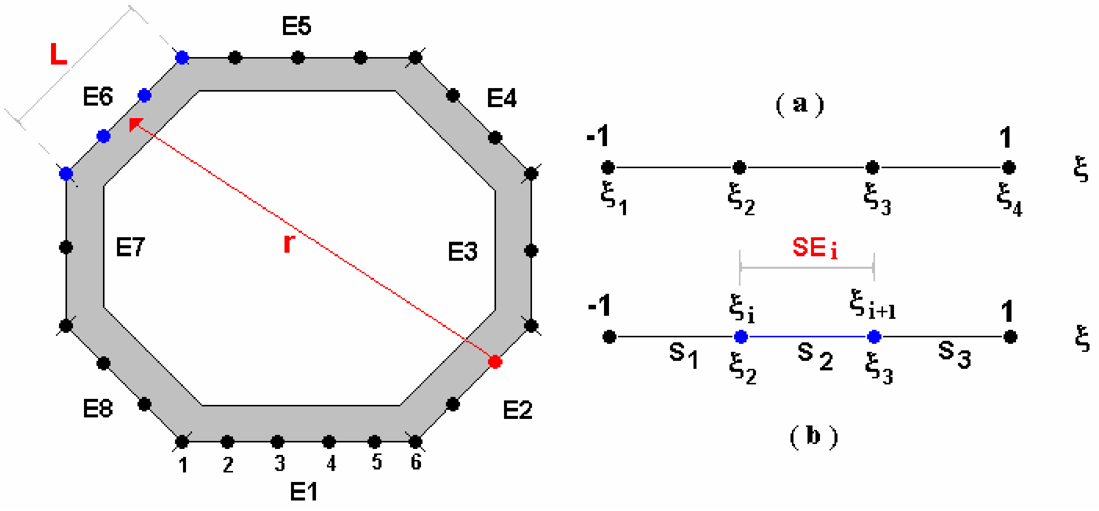

Figure 1: External boundary of a slender component discretized with different HO elements

This rule equally applies to computing the solution for the tractions:

$$T^e_{kl} \approx \sum_{i=1}^{k-1} \sum_{r=1}^{ncp} \sum_{n=1}^{ng} \sum_{m=1}^{k} \sum_{j=1}^{d} N_m\left(x_{T(n)}\right)\; T_{kl}\left(P_j\left(r\right), x_{T(n)}\right)\; J\left(x_{T(n)}\right)\; \omega_n\; J_T \tag{8}$$

The Jacobian, $\mathbf{J}$, that makes the transformation between the intrinsic space $\boldsymbol{\xi}$, where the element is defined, and the boundary path, $\boldsymbol{\Gamma}$, is calculated in the usual form:

$$J = \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2} = \sqrt{\left(\sum_{n=1}^{ng} \frac{\partial N_m\left(\xi_n\right)}{\partial \xi} x_m\right)^2 + \left(\sum_{i=1}^{ng} \frac{\partial N_m\left(\xi_n\right)}{\partial \xi} y_m\right)^2} \tag{9}$$

Note that, to apply equation (5), (8) and (9), equations (1) and (2) will be used in conjunction.

It is interesting to observe that the transformation given by equation (6) usually decreases the numerical value of the coordinates, with more or less significance, what will depend on the number of sub interval employed in the process.

As a consequence and considering also the high number of standard operation involved in the quadrature process, the numerical results may suffer some loss in accuracy, what will be particularly true if single precision arithmetic is employed.

To circumvent this, subroutines dealing with equation (1) and (2) can be directly used to perform the transformations given by equation (6) in a more accurate way, just performing a single operation on the limits **a** and **b**, defining the intrinsic space of the element, so that the intrinsic space is increased. For an element with **k** nodes to be subdivided into $\boldsymbol{k-1}$ sub elements, the extremes of the element, to integrate sub element $\boldsymbol{SE_i}$, will be set to (Fig. 1):

$$a_i = -SE_i \qquad \text{and} \qquad b_i = k + a_i \tag{10}$$

The variable $\boldsymbol{SE_i}$ found in equation (10), is the number of the sub element that is being integrated. The above expedient will provide the amplification of the intrinsic space by a factor of $\boldsymbol{k/2}$.

3.2 Outside integration rule

In general, the kernel tensor functions are integrated using standard Gauss quadrature. To achieve accuracy in the integration process, the error inherent to the numerical approximation will have to be restricted to some constant level.

Consequently, the number of sample points to be used is an important variable to be defined along the process. An attractive and simple way to define it is considering the ratio of the distance **r** that separates the source point from the element that is being integrated, to the element length, **L** [2].

In this case, the maximum error in the integration process is expected to occur when the distance from the source point to the nearest node of the element approaches zero, what suggests the use of higher order quadrature. On the other hand, the minimum errors will take place far away from the element, obviously requiring a smaller number of sample points.

An integration rule for outside integration, varying from 4 to 36 points and based on both, the principle outlined above and on the formula for the error of the Gauss-Legendre quadrature process [2], has been implemented in reference [5]. This rule makes use of an expression that relates the number of sample points to be used, **ngp**, with regard to the minimum relative distance, $\boldsymbol{r/L}$, separating the source point and the nearest node of the element being integrated and an expected error $\boldsymbol{\varepsilon}$ (Fig. 1). Based on the Reciprocal Logarithm model, this rule is of the form:

$$ngp = \frac{1}{a + b.\ln\left(\frac{r}{L}\right)} \tag{11}$$

with coefficients:

$\boldsymbol{a} = 0.12540225$ and $\boldsymbol{b} = 0.090458605$ (for $\boldsymbol{\varepsilon} < 1.0^{-9}$ and $3.95 < r/L < 0.3398$)
or
$\boldsymbol{a} = 0.18238562$ and $\boldsymbol{b} = 0.13156348$. (for $\boldsymbol{\varepsilon} < 1.0^{-6}$ and $1.67 < r/L < 0.3088$)

3.3 Inside Integration

This particular case, considered when the source point of the fundamental solution coincides with one of the nodes of the element being integrated, takes into account the singularity of order $\ln(1/r)$ existing in the displacements kernels $\boldsymbol{U_{kk}}$. In this case, the contribution $\boldsymbol{U_{kk}}$ of a given element to the global influence matrix can be written in a general form as:

$$U_{KK}^{E} = \frac{(3-4\nu)}{8\pi G\,(1-\nu)} \int\limits_{\Gamma} \ln\left(\frac{1}{r}\right) d\Gamma + \frac{1}{8\pi G\,(1-\nu)} \int\limits_{\Gamma} \left(\frac{r_K}{r}\right)^2 d\Gamma \tag{12}$$

It isn't totally evident, but the first term on the right-hand side of equation (12) must be separated into singular and non singular parts, due to the fact that intrinsic spaces of different natures involved, the first one, $\boldsymbol{\xi}$, related to the boundary element definition (usually running from -1 to 1), and the second, $\boldsymbol{\eta}$, related to the logarithmic integration space (running from 0 to 1), are to be considered simultaneously.

This separation, which implicates in several spatial transformations to account for the position of the source point inside the element, has been considered by other [1–3].

Alternative methodologies, as the self-adaptive one proposed by Telles [9], that makes use of standard Gauss-Legendre quadrature within a cubic transformation, are quite popular. However, for most two-dimensional elasticity applications, accuracy is accomplished (say, to the 4^{th} decimal place), only if quadratures of very high order (of about 36 Gauss stations) are used.

However, to treat the inside integration process when long elements are to be subdivided, some additional care must be taken, since existing singular quadrature rules many times permits direct reflections, translations but not scaling [4]. To circumvent this, Lin-Log quadrature [7] can be considered directly on the boundary, i.e., in terms of element length, **L**, by manipulating the intrinsic space where the element is defined. To accomplish this, the flexibility given by subroutines presented in this paper is used.

To better explain the proposed methodology, a simple example where the source point of the fundamental solution is placed on node 3 of the 6-noded HO element E1 of Fig. 1, will be considered. In this case, the subintervals of integration will be those confined between two consecutive nodes, as previously. From the definition of the Jacobian in the 2D space, the element length is preliminarily computed, using a minimum quadrature rule, as follows:

$$L = \sum_{i=1}^{npg} J\left(\xi\right)_i \omega_i \tag{13}$$

where $\boldsymbol{npg}$ in the number of Gauss points with coordinates and weights ($\boldsymbol{\xi_i}$;$\boldsymbol{\omega_i}$). To place the integration region in different situations, with regard to the position of the source point, a two-step analysis where the element is integrated by both sides of the node can be performed, as shown in Fig. 2.

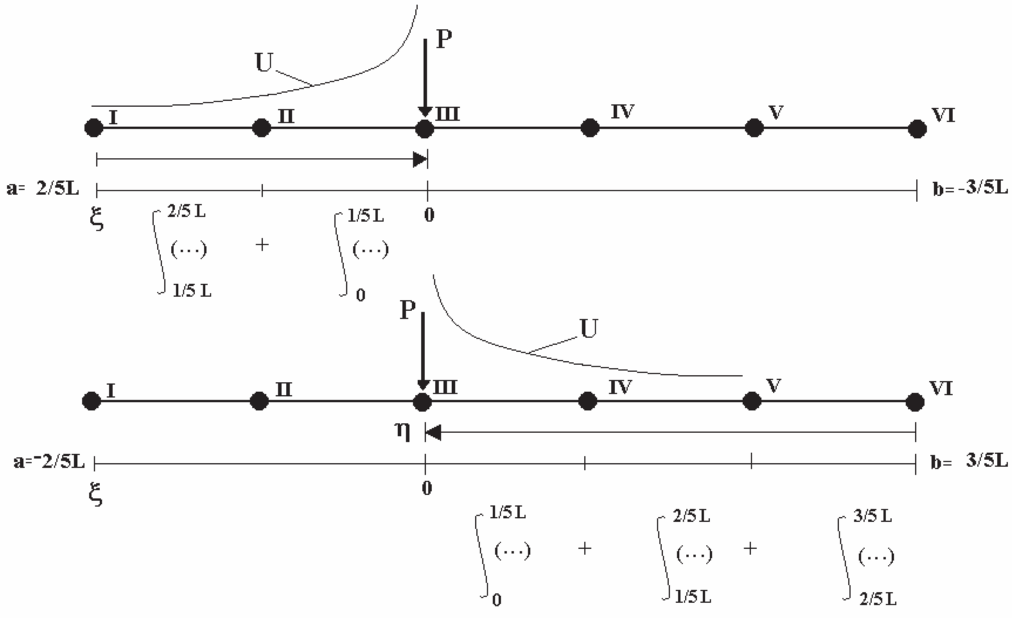

Figure 2: Inside logarithmic integration of a six noded HO element – Source point at node 3

To apply this concept to elements of any order, a simple rule to define the extremes **a** and **b** of the intrinsic space can be written in terms of **L**, to integrate the left and the right parts of the element, as follows:

By the left: (for n>1)

$$a = L.\left(\frac{n-1}{k-1}\right) \qquad \text{and} \qquad b = -L.\left(\frac{k-n}{k-1}\right) \tag{14}$$

By the right: (for n<k)

$$a = -L.\left(\frac{n-1}{k-1}\right) \qquad \text{and} \qquad b = L.\left(\frac{k-n}{k-1}\right) \tag{15}$$

In the above equations, **k** is the number of nodes of the element, **n** is the node where the source point is considered and **L** is the element length. The integration of the subinterval, whose limits are **c** and **d**, is performed using equation (16) and (17):

$$\int_c^d \ln\left(\frac{1}{r}\right)_\eta \left(\frac{\partial\xi}{\partial\eta}\right)_{c,d} J d\eta = -\int_c^d \ln\left(r\right)_\eta \left(\frac{\partial\xi}{\partial\eta}\right)_{c,d} J d\eta \approx -\sum_{i=1}^{np}\left(\ln\left(\xi_i\right)_{\eta_i}\omega_i\left(\frac{\partial\xi}{\partial\eta}\right)_{c,d} J\right), \tag{16}$$

$$\xi_i = \eta_i\,(d-c) + c \qquad \text{and} \qquad \frac{\partial \xi}{\partial \eta} = d - c. \tag{17}$$

In the above, $\boldsymbol{\eta_i}$ and $\boldsymbol{\omega_i}$ are the coordinates and weights of the Lin-Log quadrature, **np** is the number of points used in integration process and **J**, the Jacobian that makes the transformation from the boundary path, $\boldsymbol{\Gamma}$, to the intrinsic coordinate. The shape functions and their derivatives, needed to calculate the nodal influences are computed using equation (1) and (2) that remains valid in this case. Note that the above strategy can be applied to internally integrate elements of any order.

In general, the variable **r** appearing in equation (16) needs to be written in terms of $\boldsymbol{\eta}$, so that that the curvature of the element (especially of long elements) can be taken into account. This can be accomplished through a simple mapping of the coordinate $\boldsymbol{\eta_i}$ on the actual element, followed by a geometrical computation of **r**.

However, as the integration process in being performed over subintervals of the element, the eventual error associated with this detail would be of secondary importance, even for curved elements

Note that a one point Lin-Log rule, integrates exactly the function **ln** (1/x) on the interval running from 0 to 1. To integrate the kernels of equation (12), a four point rules showed to be precise enough, within numerical experiments carried out with curved elements.

Table 1 gives the coordinates and weights computed for a six-point rule using Lin Log quadrature. If higher order quadratures are desired, other coordinates and weights can be deduced based on the moments **ln**(x),1; x **ln**(x), x; x^2.**ln**(x), x^2, etc.

Table 1: Coordinates and weights for a six-point rule – Lin Log quadrature

xi	**wi**
0.003025802137546	0.011351338817273
0.040978254155951	0.075241069954917
0.170863295526877	0.188790041615416
0.413255708844793	0.285820721827227
0.709095146790628	0.284486427891408
0.938239590377167	0.154310399893758

4 Numerical experiments

To better evaluate the performance of HO elements of different order, two different problems involving slender structures have been analyzed using the program ELASCON [5], developed by the author. Based on the methodology previously outlined, this program can be applied to analyze meshes of mixed elements of any order.

First problem

Within this study, a circular ring subjected to bending is considered. Due to symmetry, only one quarter of the structure has been modeled. To numerically analyze the problem, a fixed number of 32 functional nodes have been used to develop four different BE meshes. To represent the main edges (internal and external), 13 nodes forming elements of different orders were employed for a plane stress elasticity solution.

On the other hand, two quadratic elements were used in each of the symmetry edges (in all cases). In this fashion, the different meshes were prepared just by regrouping nodes. Excluding the quadratic elements used on the symmetry edges, the four meshes were formed with six quadratic elements, four cubic elements, three 5-noded elements and two 7-noded elements per edge, respectively.

For completeness, 11 points, numbered from 2 to 12, were also placed inside the solution domain, along the structural axis, in alignment with the boundary nodes. Physical and geometrical data for the bending problem, as well as the BE meshes with same number of degrees of freedom are presented in Figure 3.

In addition, a study using only quadratic elements was performed. Within this study, the number of elements representing the main edges was successively increased, in order to check for convergence.

Results

The values of vertical displacement for point A (Fig. 3), obtained with the different meshes are depicted in Fig. 4. For comparison, this figure also shows the theoretical value (including the effects of compression and shear), given by beam theory [10]. The results of the convergence study performed exclusively with quadratic elements, as well as the predicted value, are shown in Fig. 5.

The normal (hoop) stress distributions, numerically and analytically computed at the outer edge nodes of the model, are shown in Fig. 6.

Finally, Fig. 7 shows the radial displacements determined from internal points results. This figure also depicts the predicted values of these displacements, computed along the structural axis.

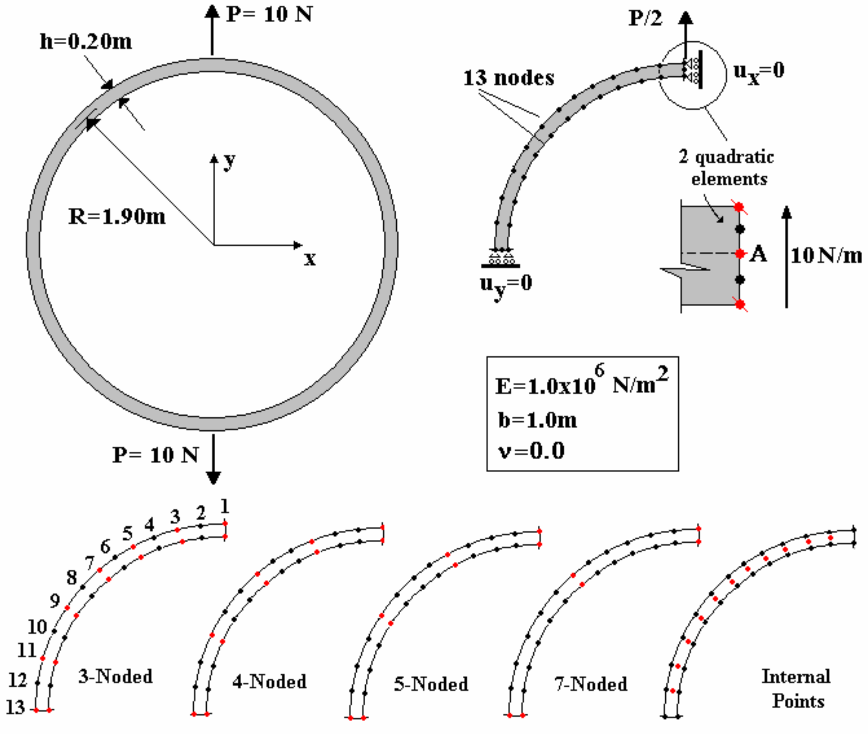

Figure 3: Circular ring subjected to bending – Physical model, BE discretization and material data.

Second problem

The main objective of this analysis was to check, in a qualitative sense, the performance of the different elements in situations where high gradients of the responses are present. To this end, a frame with built-in ends has been considered. It was subjected to a horizontal loading uniformly distributed with respect to the vertical axis, as examined by Timoshenko and Young [11], within a plane stress elasticity solution.

Figure 8 brings the physical and geometrical data for the bending problem as well as a general boundary element mesh composed of 92 functional nodes.

As in the previous problem, three different BE meshes were constructed using the same number of nodes, in order to keep the number of unknowns unchanged. The different meshes were formed by 42 quadratic elements, 28 cubic elements and 21 quartic elements, respectively.

To represent each of the built-in ends, 2 extra quadratic elements have been employed in all cases. Also within this analysis, a study using only quadratic elements has been performed in order to check for convergence.

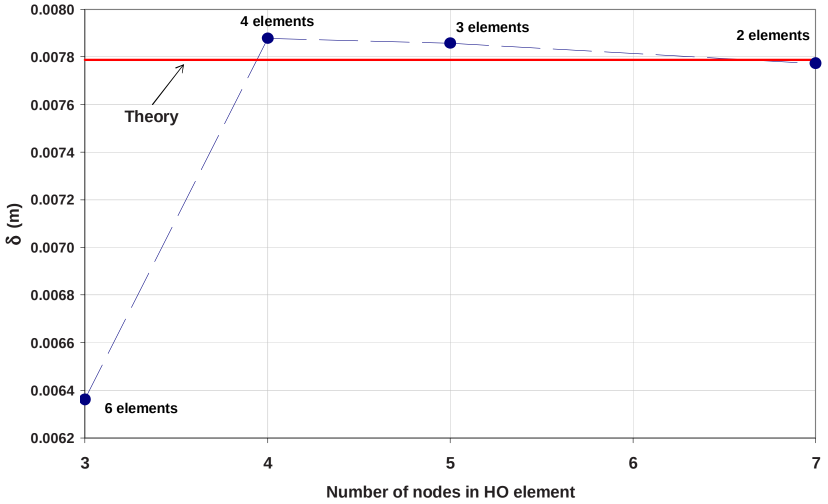

Figure 4: Displacements at point A, computed with different HO elements and same number of degrees of freedom

Results

The results of the first part of the analysis are graphically presented in Fig. 9, 10 and 11. These figures depict the deformed shape and the distribution of normal stresses, $\boldsymbol{\sigma_n}$, computed at the functional nodes placed on the inner and outer boundaries of the structure, respectively.

The results of the convergence study using quadratic elements only are presented in Fig. 12 and 13. In this case, the values of normal stresses, $\boldsymbol{\sigma_n}$, are depicted relatively to the nodal positions, with regard to the point "H" shown in Fig. 8.

The reference values adopted for comparison were those computed within the first part of this study, obtained with the quartic elements.

5 Analysis of results

First problem

In this case, all elements with order higher than quadratic showed the ability to deal satisfactorily where quadratic elements presented bad results. This demonstrates that the parabolic nature of the quadratic element is not sufficient to satisfactorily describe the variation of the displacements in the bending problem, even within a much finer discretization, as pointed out in the convergence study.

The best result for this displacement was accomplished with the 7-noded element, whereas the cubic and the quartic elements returned moderately higher results, if com-

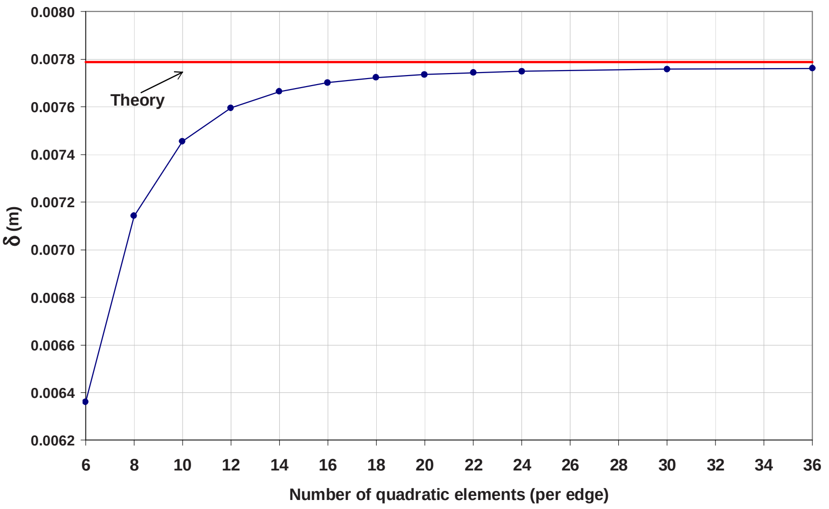

Figure 5: Displacement responses at point A, computed with meshes (gradually refined) of quadratic boundary elements.

pared with the predicted value given by beam theory. Similarly, the responses computed for the normal (hoop) stress, showed to be in good agreement with the predicted values, even considering that they had been obtained from coarse meshes, in terms of number of elements. Also in this case, the performance of the quadratic elements was very poor.

It is interesting to note that the small divergences occurring at the extremes of the predicted and computed curves, would be higher if a finer mesh of HO elements where used. This is due to the fact that the theory applied to predict this response, even accounting for compression and shear, is based on small deflections hypothesis, which in turn, does not perfectly apply to the slender structure analyzed herein.

In general, the internal results are very accurate in BEA, since the governing differential equations are satisfied in the solution domain. In the present case, however, the displacement field is described by a composition of sinoidal functions and cubic powers (related to position and curvature, respectively), so that the results for the radial displacements obtained with the most populous mesh (in terms of number of elements), demonstrated from a different angle the inadequacy of the quadratic elements to solve problems involving relatively high gradients of the responses. On the other hand, the 7-noded element, for being capable to interpolate with accuracy the high variations of the displacements at the extremes of the structure, presented better results (Fig. 4 and 7).

Second problem

The results obtained from this analysis confirmed those found in the previous one, regarding the performance of quadratic elements. From Fig. 10 and 11, it is possible to

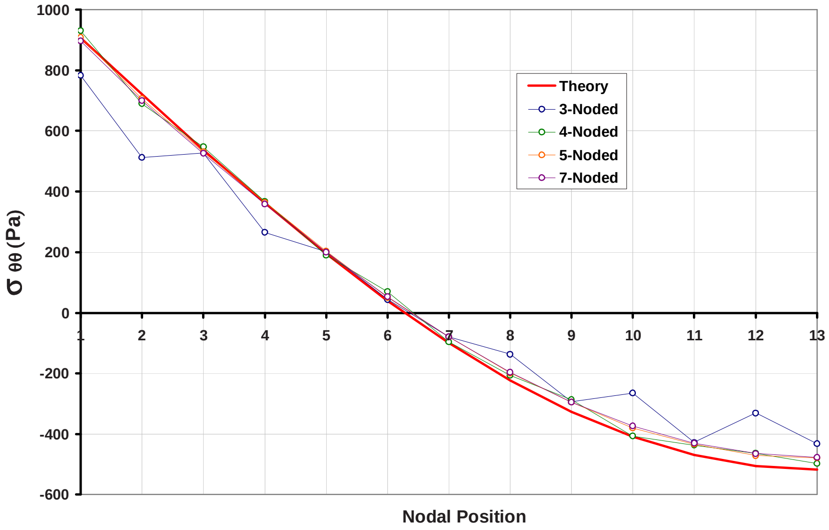

Figure 6: Nodal normal (hoop) stress distribution – outer edge.

note that the responses computed with these elements presented a highlighted fluctuation in regions where the responses present rapid variation.

It is interesting to observe that responses of this kind were obtained in both, straight and curved regions, what some how confirms that the low performance of the quadratic elements, comparatively to cubic and quartic elements, at least in this case is not due only to an eventual inability presented by the element to accurately interpolate the geometry, as intuition suggests.

In this regard, a counter example is given in Fig. 11 and 12. Within the regions enclosed by cycles “1” in these figures, which are placed on the curved boundary, the structural responses are relatively smooth so that, even in the case were very few quadratic elements have been used, the responses computed were more stable and accurate. On the other hand, within the region enclosed by cycle “2” in Fig. 11 (and 13), which is placed on a straight boundary, again the responses computed with quadratic elements become unstable and less accurate, since the displacement field is more complicated in this region (Fig. 9).

Also, the convergence study showed that a mesh of quadratic elements, about three times greater (in terms of degrees of freedom), was necessary to obtain satisfactory results in these regions (Fig. 13).

In fact, several numerical experiments carried out by the author [6] showed that, as the order of the variation of the displacements is increased (for example, when the structure is subjected to distributed loads of higher order, as the linear and the quadratic ones), the element whose order matches the expected order of the response, seems to perform better,

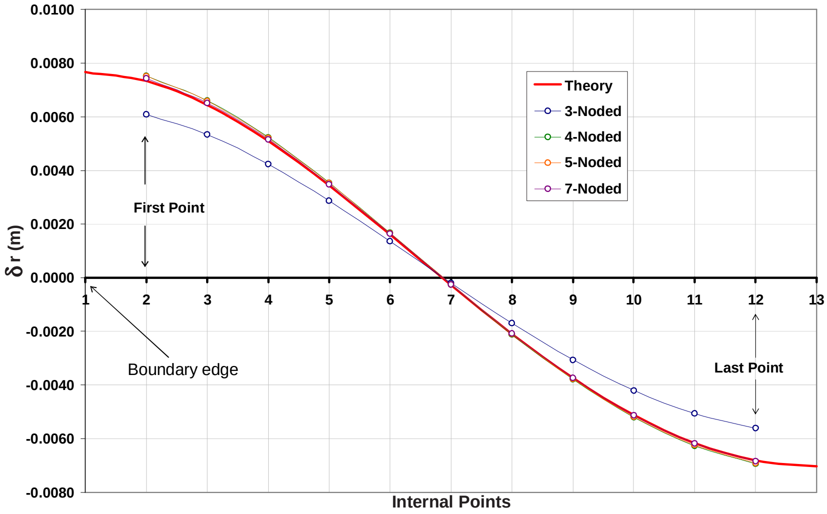

Figure 7: Radial displacements along structural axis computed with internal points results.

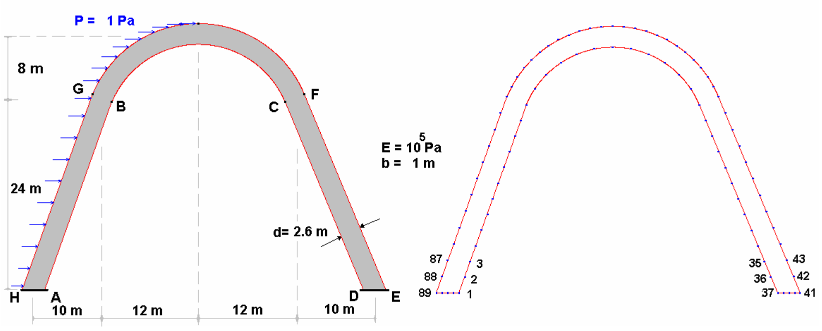

Figure 8: Frame subjected to horizontal load – Actual problem and general BE mesh.

if computational effort and number of degrees of freedom involved are taken into account simultaneously. In this case, both, a fine first approximation and a faster convergence will be accomplished.

6 Conclusions

To model the problems presented in this paper, a small number of degrees of freedom have been deliberately employed, so that the performance of different HO elements could be checked and compared in different ways. The results obtained in the previous studies show that the boundary element method is suitable to analyze structures in bending, even in extreme circumstance where slender components are considered.

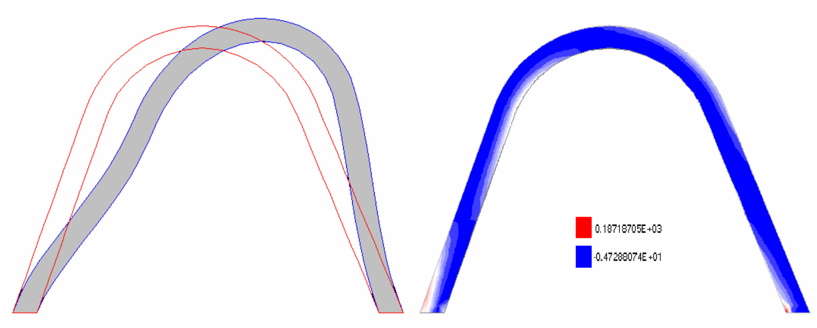

Figure 9: Deformed shape and contour of principal stress σ_1 (Pa) - quartic elements.

To this, the order of the element to be used must be compatible with the expected behavior of the component that is being analyzed. In this regard, a careful choose of the element order is necessary.

To accomplish this, an adaptive process can be conveniently implemented to check for convergence using elements of different order, just regrouping adjacent nodes, thus preserving the number of unknowns in the problem. On the other hand, these elements will perform adequately if appropriate schemes of integration, as those suggested in this paper, are primarily considered and implemented.

Acknowledgements: This research has not been sponsored. However, the continued support provided by FAPESP and CNPQ (Brazilian Founding Agencies) to researchers working in different areas, is gratefully recognized and appreciated.

References

[1] A. A. Becker. The boundary element method in engineering A complete course. McGraw-Hill Book Company, UK, 1992.

[2] G. Beer. Programming the boundary element method An introduction for engineers. John Wiley & Sons, Ltd, England, 2001.

[3] C. A. Brebbia and J. Domingues. Boundary elements an introductory course. Computational Mechanics Publications, 1992.

[4] C. A Brebbia, J. C. F. Telles., and L. C. Wrobel. Boundary element techniques Theory and applications in engineering. Springer-Verlag Berlin, Heidelberg, Germany.

[5] L. E. T. Ferreira. Elascon a code to solve 2d linear elastic problems using BEM and HO elements (software available from http://leteixei.vilabol.uol.com.br).

[6] L. E. T. Ferreira. Differential geometry management of higher order 2d boundary elements. Electronic Journal of Boundary Elements, http://ejbe.libraries.rutgers.edu/, 2(1), 2004.

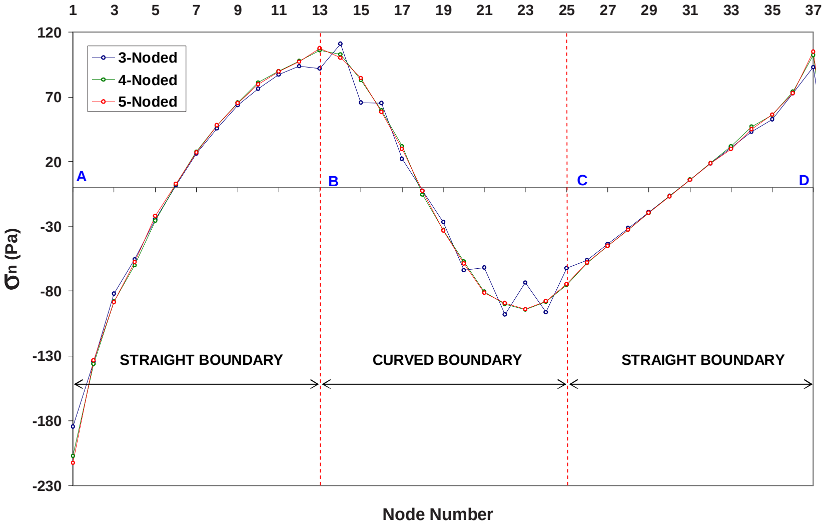

Figure 10: Normal stresses computed at boundary nodes using elements of different order– Inner boundary

[7] J. Ma, V. Rokhlin, and S. Wandzura. Generalized gaussian quadrature rules for systems of arbitrary functions. SIAM Journal on Numerical Analysis, 33(3):971–996, 1996.

[8] J. C. Miranda-Vanenzuela and K. H. Muci-Küchler. Adaptive meshing with boundary elements. In editors In Brebbia CA, editor, Topics in Engineering, volume 41, UK, 2002. WIT Press Southampton.

[9] J. C. F. Telles. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. International Journal for Numerical Methods in Engineering, 24:959–973, 1987.

[10] S. Timoshenko. Resistencia de materiales - teoría y problemas más complejos, volume 2. Espasa-Calpe S.A., Madrid, 1975.

[11] S. Timoshenko and D. H. Young. Theory of structures, second edition. McGraw-Hill Book Company, Japan, 1983.

[12] O. C Zienkiewicz and R. L. Taylor. The finite element method - Basic formulation and linear problems, volume 1 (4^{th} ed.). McGraw-Hill Book Company, England, 1994.

Appendices

A Subroutine to compute shape functions values

```
SUBROUTINE SHAPE_FUNCTION1(a, b, K, Rt, QSI, N)
!---------------------------------------------------------------------
! To compute the values of the shape functions for a given one-dimensional
! element
```

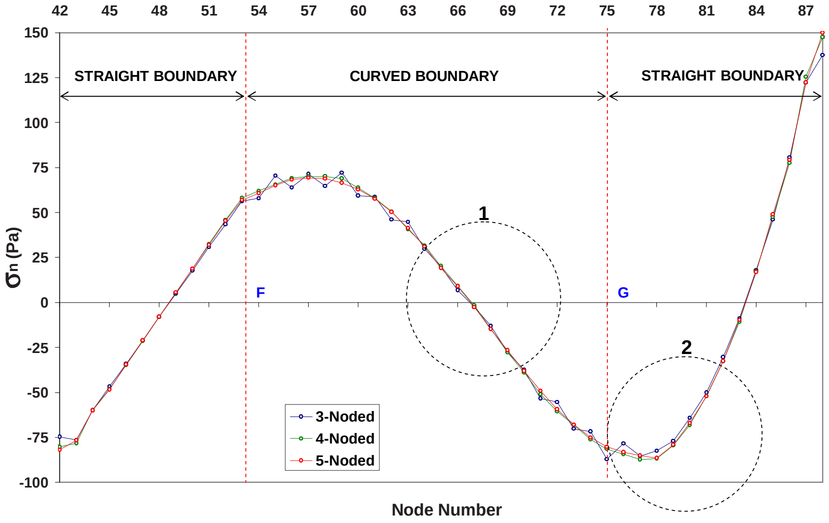

Figure 11: Normal stresses computed at boundary nodes using elements of different order – Outer boundary

```
! Written by: Luiz Eduardo T. Ferreira (leferrei@uol.com.br)
! Last modified: 08/30/04
! NOTE: ALL REAL VARIABLES ARE DOUBLE PRECISION
!-------------------------------------------------------------------------

    IMPLICIT NONE
!   Entries:
    REAL(8), INTENT(IN) :: a, b !Lower and upper limits of intrinsic elem.
    INTEGER, INTENT(IN) :: K !Number of nodes of element
    REAL(8), INTENT(IN) :: Rt !Ratio between two adjacent elements
    REAL(8), INTENT(IN) :: QSI !Position in which the S.F are to be evaluated
    REAL(8), INTENT(OUT),DIMENSION(K) :: N !Array with results

!   Local variables:
    REAL(8) :: DELTA_QSI, NUMERATOR, DENOMINADOR, SUMM
    REAL(8),DIMENSION(K) :: QSI_N
    INTEGER :: I, J
!   Process begins here, initialize variables:
    I=0; J=0
    N            =  0.0D0  ! Array to store results \\
    QSI_N        =  0.0D0  ! Positions of the nodes \\
    NUMERATOR    =  1.0D0  ! Numerator of equation1 \\
    DENOMINATOR  =  1.0D0  ! Denominator of equation1 \\
!-------------------
    SUMM=0.0D0
```

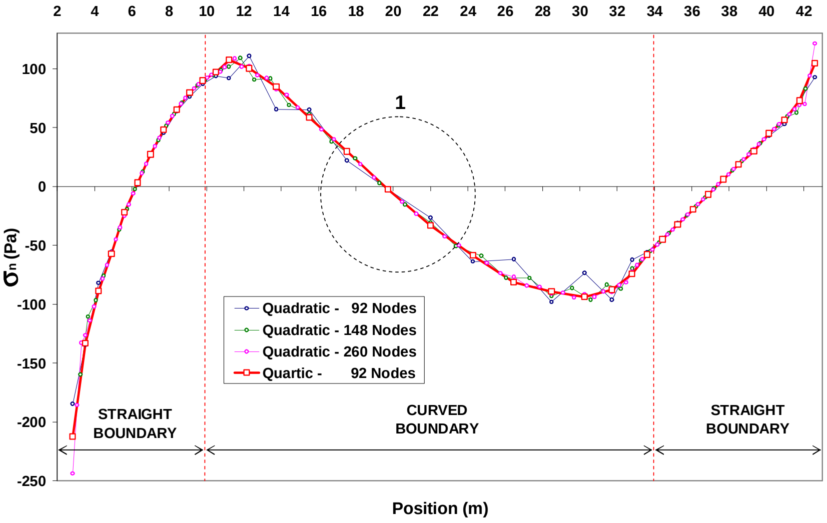

Figure 12: Normal stresses computed at boundary nodes with quadratic elements – Inner boundary

```
      RATIO: &
      DO I=0,K-2
          SUMM=SUMM+Rt**I
      END DO &
      RATIO
      DELTA_QSI = (b-a)/ SUMM ! Increment for the norm. coordinates
      QSI_N(1) =a
      QSI_N(2) =a + DELT_QSI
      NODES_OF_ELEM: &
          DO I=2, K-1
              SUMM=0.0D0
              ADJ_NODE: &
                  DO J= 0,I-1
                      SUMM = SUMM + Rt**(J)
                  END DO &
              ADJ_NODE
              QSI_N(I+1) =a + DELTA_QSI * SUMM
          END DO &
      NODES_OF_ELEM
!     Compute shape functions values, Ni, for all the 'K' nodes of element:
      SHAPE_FUN_Ni: &
      DO I=1,K
!     For all 'J' positions over the intrinsic element, compute equation1:
          PRODUCTORY: &
              DO J=1,K
```

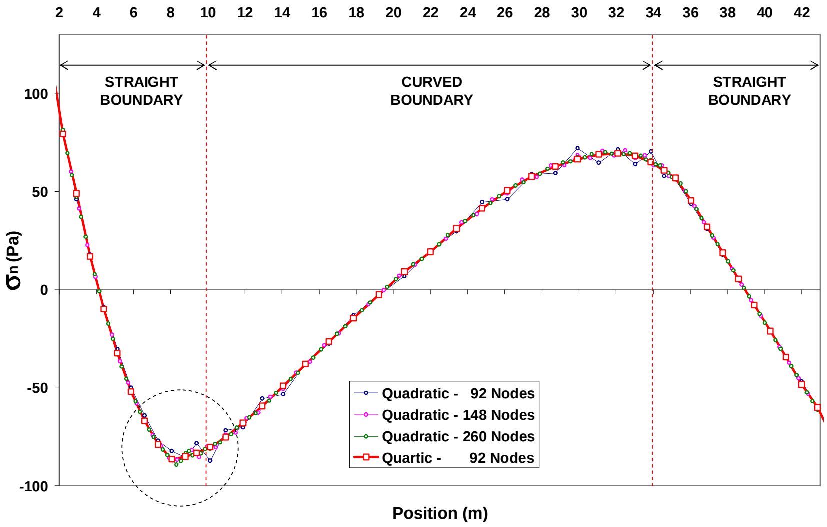

Figure 13: Normal stresses computed at boundary nodes with quadratic elements – Outer boundary

```
                IF(J==I) CYCLE PRODUCTORY
                NUMERATOR = NUMERATOR *(QSI-QSI_N(J))
                DENOMINATOR = DENOMINATOR*(QSI_N(I)-QSI{\_}N(J))
            END DO &
        PRODUCTORY
        N(I)=NUMERATOR/DENOMINATOR
        NUMERATOR = 1.0D0
        DENOMINATOR = 1.0D0
    END DO &
    SHAPE_FUN_Ni
END SUBROUTINE SHAPE_FUNCTION1
```

B Subroutine to compute shape function's derivatives

```
SUBROUTINE SHAPE_FUNCTION_DERIV1 (a, b, K, Rt, QSI, dN)}
!-----------------------------------------------------------------------
! To compute the values of the deriv. of shape functions for a given
! one-dimensional element
! Written by : Luiz Eduardo T. Ferreira (leferrei@uol.com.br)
! Last modified: 08/30/04
! NOTE: ALL REAL VARIABLES ARE DOUBLE PRECISION
!-----------------------------------------------------------------------
```

```
    IMPLICIT NONE
!   Entries:
    REAL(8), INTENT(IN) :: a, b ! Lower and upper limits of the
                                ! intrinsic elem.
    INTEGER, INTENT(IN) :: K    ! Number of nodes of element
    REAL(8), INTENT(IN) :: Rt   ! Ratio between two adjacent elements
    REAL(8), INTENT(IN) :: QSI  ! Position in which the deriv. is to
                                ! be evaluated
    REAL(8), INTENT(OUT), DIMENSION(K) :: dN ! Array with results
!   Local variables:
    REAL(8) :: DELTA_QSI, DENOMINATOR, SUM1, SUM2, PROD1, PROD2, SUMM
    REAL(8),DIMENSION(K) :: QSI_N
    INTEGER :: I, J, N
!   Process begins here, initialize variables:
    I=0; J=0; N=0
    dN = 0.0D0; QSI_N = 0.0D0
!   Fill the array of the nodal positions:
    SUMM=0.0D0
    RATIO: &
    DO I=0,K-2
        SUMM= SUMM + Rt**I
    END DO &
    RATIO
    DELTA_QSI = (b-a)/ SUMM ! Increment for the norm. coordinates
    QSI_N(1)  = a
    QSI_N(2)  = a + DELTA_QSI
    NODES_OF_ELEM: &
        DO I=2, K-1
            SUMM=0.0D0
            ADJ_NODE: &
                DO J= 0,I-1
                    SUMM = SUMM + Rt**(J)
                END DO &
            ADJ_NODE
            QSI_N(I+1) =a + DELTA_QSI * SUMM
        END DO &
    NODES_OF_ELEM
!   Compute derivatives of shape functions, dNi, for all the 'K' nodes of element:
    D_SHAPE_FUNCTION: &
    DO I=1,K
    DENOMINATOR = 1.0D0
!   For all 'J' positions on the normalized element, compute first productory:
        PRODUCTORY1: &
            DO J=1, K
            IF(J==I) CYCLE PRODUCTORY1
            DENOMINATOR = DENOMINATOR*(QSI_N(I)-QSI_N(J))
        END DO &
    PRODUCTORY1
SUM1 =0.0D0; SUM2 =0.0D0
!   Compute first summation:
    SUMMATION1: &
    DO N=I,K-1
    PROD1=1.0D0
!   Now, the inner productory:
    PRODUCTORY2: &
        DO J=1,K
            IF(J == I.OR.J == N+1) CYCLE PRODUCTORY2
            PROD1 = PROD1*(QSI-QSI_N(J))
        END DO &
    PRODUCTORY2
    SUM1=SUM1+PROD1
    END DO &
SUMMATION1
!   Compute second summation:
    SUMMATION2: &
        DO N=1, I-1
        PROD2=1.0D0
!   Now, the inner productory :
        PRODUCTORY3: &
            DO J=1,K
                IF (J == I.OR.J == N) CYCLE PRODUCTORY3
                PROD2 = PROD2*(QSI-QSI{\_}N(J))
            END DO &
        PRODUCTORY3
        SUM2=SUM2+PROD2
        END DO &
    SUMMATION2
!   Store the value computed at this node:
dN(I)=(SUM1+SUM2)/DENOMINATOR
END DO &
D_SHAPE_FUNCTION}
!
END SUBROUTINE SHAPE_FUNCTION_DERIV1
```