

p-ADAPTIVE BOUNDARY ELEMENTS

E. ALARCON AND A. REVERTER

INTRODUCTION

It is generally accepted that the numerical treatment of singular integral equations giving rise to the so-called boundary integral equation method (BIEM) began in 1963, with a paper by Jaswon¹ on potential theory.

Probably due to the finite element method (FEM) boom, the method was not given much attention, until Rizzo² and Cruse^{3,16} applied it to elastostatics, elastodynamics, etc.

The main step in the development of the method was the incorporation of the FEM isoparametric discretization philosophy, done by Lachat⁴ and Lachat and Watson,⁵ producing what they called the 'second generation of BIEM', or the boundary element method (BEM), as it is known today.

Since then, the method has been applied to many different fields and, as proof of its vitality, we can cite the Annual Conference, well established since 1978,⁶ or the various books⁷⁻¹⁵ that are being published at an increasing pace since the pioneer Cruse's class notes.¹⁶

At this moment, the method is well established as a powerful tool for problem solving and there are several codes developed and maintained on an industrial basis.

It is well known that one of the more attractive features of the method is the reduction of the discretization of the boundary of the domain under study. As drawbacks, we found the non-bandedness of the final matrix, that is, a full and asymmetric one, and the computational difficulties related to the obtention of the integrals which appear in the influence coefficients.

From our point of view, the last point is crucial when one is trying to reduce computational costs. Our experience^{17,18} with three-dimensional and axisymmetric elastoplastic problems brought us face to face with the dilemma of choosing closed form solutions, the reliability of which is always in doubt, due to the number of steps necessary to attain them, or the numerical approach that, with the exception of some cases, has to be done substantially in the same fashion as that used by Lachat and Watson,⁵ i.e. time-consuming processes.

What is needed in order to alleviate this situation are numerical formulae capable of dealing with the singular integrals which arise in three-dimensional elastostatics. If these are found, the influence of the method in the engineering community will undoubtedly increase.

On the other hand, it is clear that most of the examples shown in the literature are over-integrated, in the sense that the same accuracy could be obtained using fewer elements or an adaptive rule adjusting the number of integration points. While this latter strategy is again time consuming, the first has been tackled only from the point of view of the so-called *h*-convergence.¹⁹⁻²¹

What we purpose in this paper is, on the contrary, to adopt the *p*-convergence approach that has given such good results in the finite element context,²²⁻²⁴ as has been demonstrated in the recent Conference.²⁵ In this way and while better integration formulae are being developed, it is possible to diminish the computational effort by reducing to a minimum the number of integrals that have to be calculated.

REAPPRAISAL OF DIRECT BEM

In order to see how to introduce the *p*-adaptive ideas in BEM, let us recall some of the essential features of the method when applied to potential problems.

The problem to be solved is then

$$\nabla^2 \phi = 0 \quad (1)$$

inside a domain Ω with boundary $\partial\Omega$ and mixed boundary conditions

$$\begin{aligned} \phi(\mathbf{x}) &= \phi_0 \quad \forall \mathbf{x} \in \partial\Omega_1 \\ q(\mathbf{x}) &= \frac{\partial\phi}{\partial n} = q_0 \quad \forall \mathbf{x} \in \partial\Omega_2 \end{aligned} \quad (2)$$

$$\partial\Omega_1 \cup \partial\Omega_2 = \partial\Omega$$

It is well known that a representation formula on the boundary is

$$c\phi(\mathbf{x}) + \int_{\partial\Omega} \phi q^* = \int_{\partial\Omega} \phi^* q \quad \mathbf{x} \in \partial\Omega \quad (3)$$

where *c* collects the local geometrical properties of the boundary at point \mathbf{x} .

Equation (3) is the integral relationship used by BEM as the basic tool to be discretized as a parallel to the weak formulation used with FEM.

The classical BEM approach has been to interpolate using the polynomial locally based splines in order to take advantage of the physical meaning of the unknown coefficients. In that way

$$\begin{aligned} \phi &= \sum \phi_i N_i(\mathbf{x}) \\ q &= \sum q_i N_i(\mathbf{x}) \\ \mathbf{x} &= \sum \mathbf{x}_i N_i(\mathbf{x}) \end{aligned} \quad (4)$$

which, having been introduced in (2), produce a system

$$\mathbf{A}\phi = \mathbf{B}q \quad (5)$$

which, by imposing boundary conditions, can be reduced to a classical system of equations

$$\mathbf{K}\mathbf{x} = \mathbf{F} \quad (6)$$

The user has to choose only: (1) a set of boundary points; (2) the degree of polynomial interpolation. In that way, the same set of points is used: (a) to define the boundary, (b) to define the boundary

conditions, (c) to define the collocation points, and (d) to define the support of interpolating functions.

Let us now consider two instances of BEM application that should be used to introduce other ideas below. The first one (Figure 1a) is motivated, for instance, by the Saint-Venant torsion of an elliptic cylinder.

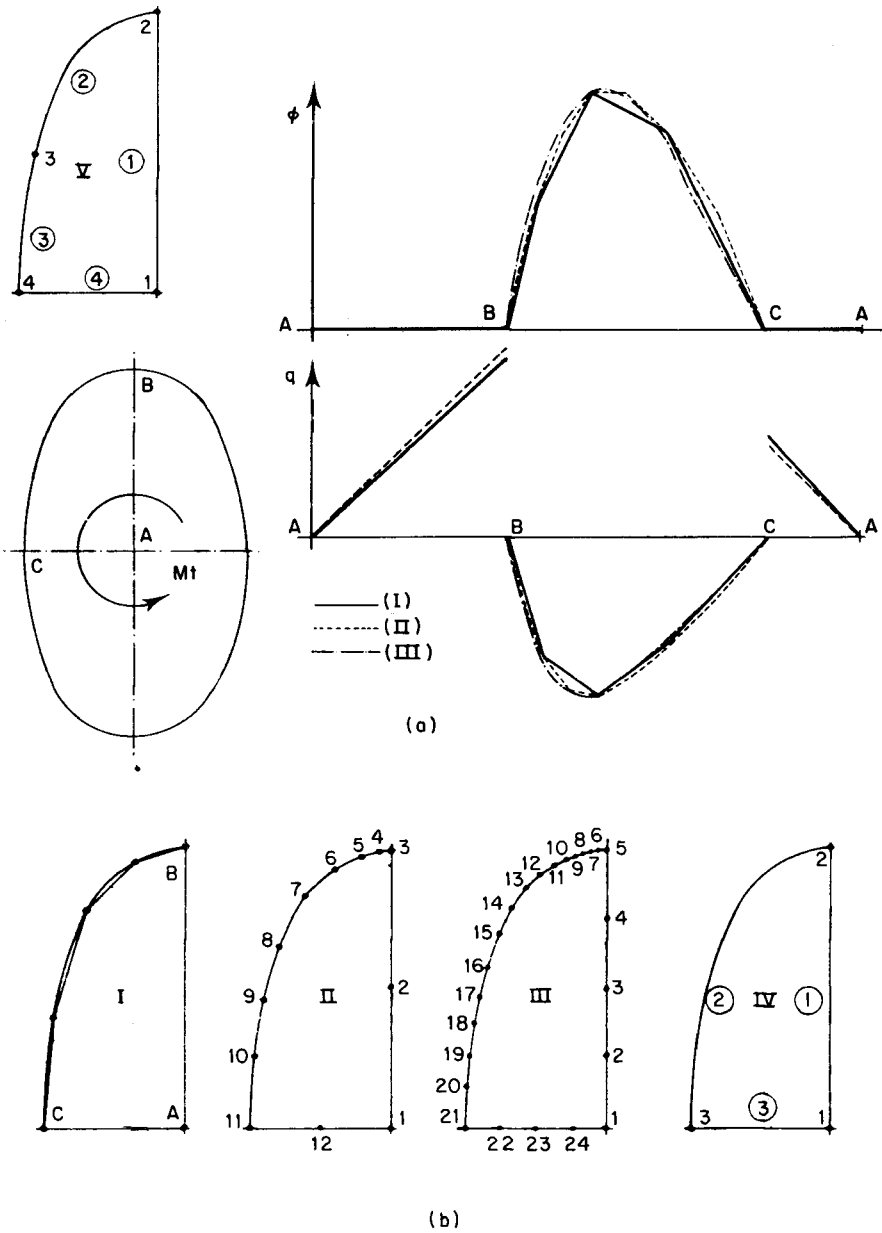


Figure 1. Torsion of an elliptic shaft. I, II and III linear elements (isoparametric representation of geometry and field variables); IV, V adaptive elements (geometry reproduced independently of the field variables). Evolution of potential ϕ and flux q along the boundary.

As interpolating functions, we have chosen linear ones that can be defined by the values at two neighbouring nodes. The meshes I, II and III converge to the correct solution due to the increased number of elements as well as to the increasing accuracy in the representation of the boundary geometry. In cases IV and V, the geometry has been imposed from scratch, independently of the degree of interpolation for field variables, as will be shown in the next section. The results will be commented on in Figure 10(a).

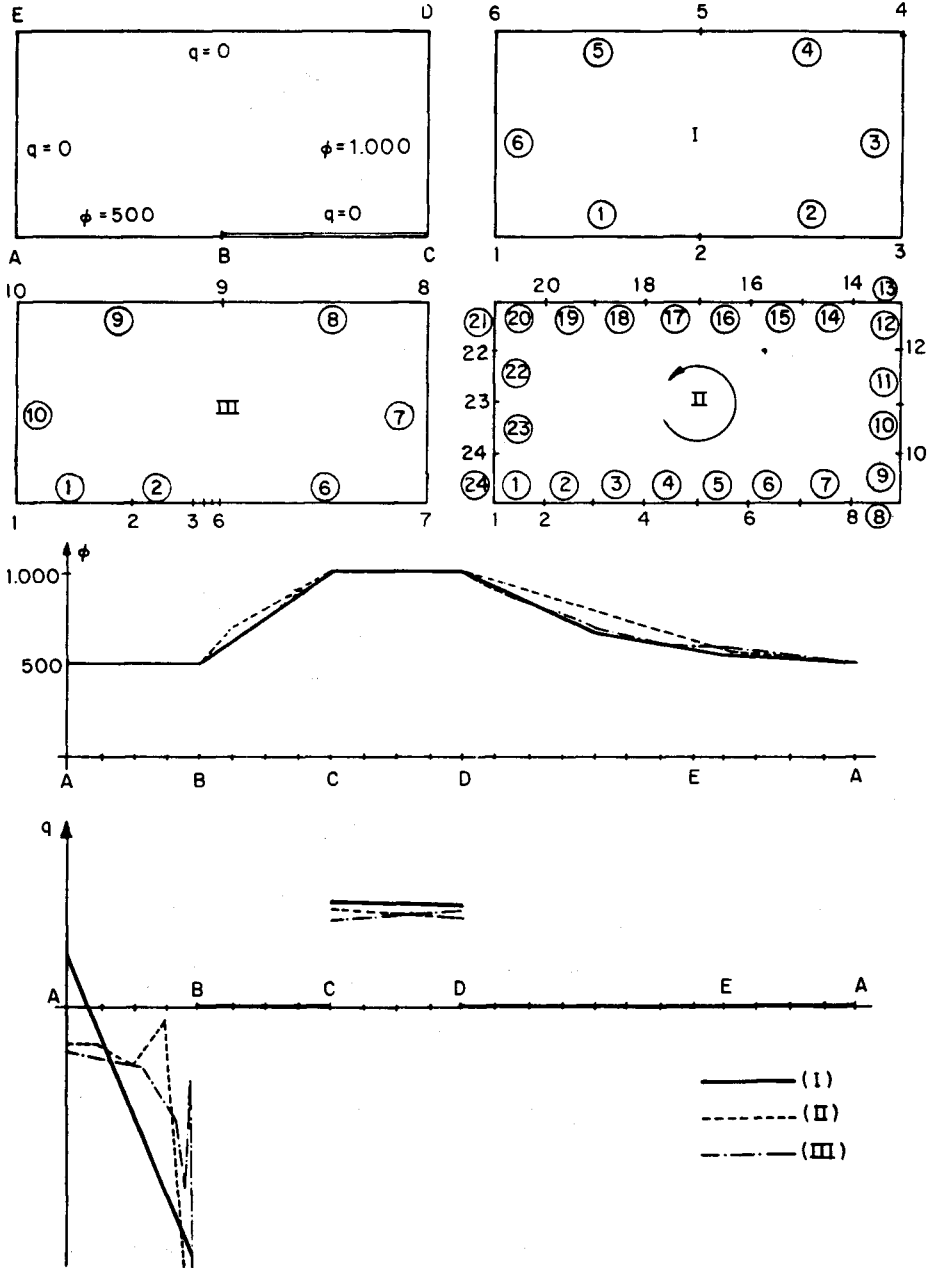


Figure 2. Problem with a singular point. I, II, III linear elements. Evolution of potential ϕ and flux q along the boundary

The second example (Figure 2) is interesting. Due to the straight sides, the geometry imposes no restriction on the number of elements but the sudden change of boundary conditions induces a singular point at B, producing high rates of variation in q that have to be treated by increasing the number of degrees-of-freedom near the singularity.

It is instructive to see the results for two-kind of meshes: one uniform, I and II, and the other denser, III, near the singularity. Near B, the flux values are polluted by the influence of the values in spite of the good results in other parts of the boundary.

Those examples are representative of the accuracy that can be obtained and the simplicity of the input data, but also of the stiffness of the approach as soon as the degree of the interpolation and the boundary points are chosen.

In the second example, for instance, the simplicity of the geometry cannot be used as soon as an irregular mesh is planned and, in the first ones, the linearity of the results in the straight sides is only known *a posteriori*.

It is evident that, in both cases, it is possible to obtain better results by refining the original mesh in what is known as an 'h-convergence' approach. But then it is necessary to start all the discretization process again from the very beginning and the previous computations are of no use for the successive refinements. Although it is possible to prepare an automatic mesh refinement,²¹ it would be desirable to implement a method capable of solving those inconsistencies and, moreover, to try to incorporate the Peano guidelines:²⁶ '... the new systems should provide error indicators for automated assessment of the accuracy of the solution as well as the capability for efficient reanalysis with improved accuracy without additional data preparation.'

p-ADAPTIVE BOUNDARY ELEMENTS

The idea of refining the approach by using a hierarchy of interpolating functions was first proposed by Zienkiewicz et al.,²² in 1970, when trying to produce transition elements. Peano,²³ in 1975, presented a philosophy which was developed on *p*-adaptive finite elements, and the mathematical foundation and establishment of *a posteriori* estimates was due to Babuska,²⁷ Szabo.²⁸ etc. An up-to-date account of what is happening in that area can be seen in Reference 29.

In boundary elements, only recently^{30,31} have the advantages of the new procedure been incorporated in a systematic fashion, although in 1977, Tüerteltaub & Paluszyn³⁶ proposed the use of extended interpolation functions in conjunction with a boundary collocation method. The method was implemented for elastostatics and straight elements only, but was not considered as a general adaptive tool.

The basic idea is the following: Due to the global character of the 'weighting functions' (the fundamental solution of the operator), the influence matrices of equation (5) are full and non-symmetric just as they are in equation (6). The typical bandedness of FEM is lost in spite of the using locally based interpolating functions. It is useful, then, to consider the possibility of globally based functions, at least over macro-elements, and to analyse the consequences that such a decision can have to the whole computational scheme, especially with reference to points (a) to (d) in the previous section. Our solution is given in the following paragraphs.

Representation of the geometry and boundary conditions

The first point to be solved is how to choose the macro-elements over which the refinement is to be done, and then, how to define them. Concerning the first part, we have decided to use the natural elements that are immediately apparent in each problem. In Figure 1, for instance,

it is clear that the corners impose at least three elements, i.e. AB, BC and CA, each of which can be defined separately. On the other hand, the boundary conditions are related to the same subdivision, and it seems logical to establish the broadest discretization possible, to take full advantage of the method. Figure 2 exemplifies how the boundary conditions interact with the discretization in spite of the simplicity of the geometry. In principle, there should be only four natural elements: AC, CD, DE, EA, because the corners impose discontinuous equations to the boundary as well as to the fluxes. But now, in the middle of AC, there is a sudden change of boundary conditions that imposes the addition of a node B in order to interpolate clearly fluxes or potentials. In this way, the broadest or natural mesh is AB, BC, CD, DE, EA.

The same line of reasoning can be applied to every problem. The rationale behind this is that, with the exception of corners and changes in boundary conditions, the data and results on the remaining elements must behave as smooth functions and a hierarchical approach, in the same sense as a Fourier development, will catch the behaviour with only a limited number of degrees-of-freedom. (The reasoning, of course, can be extended easily to three-dimensional problems.)

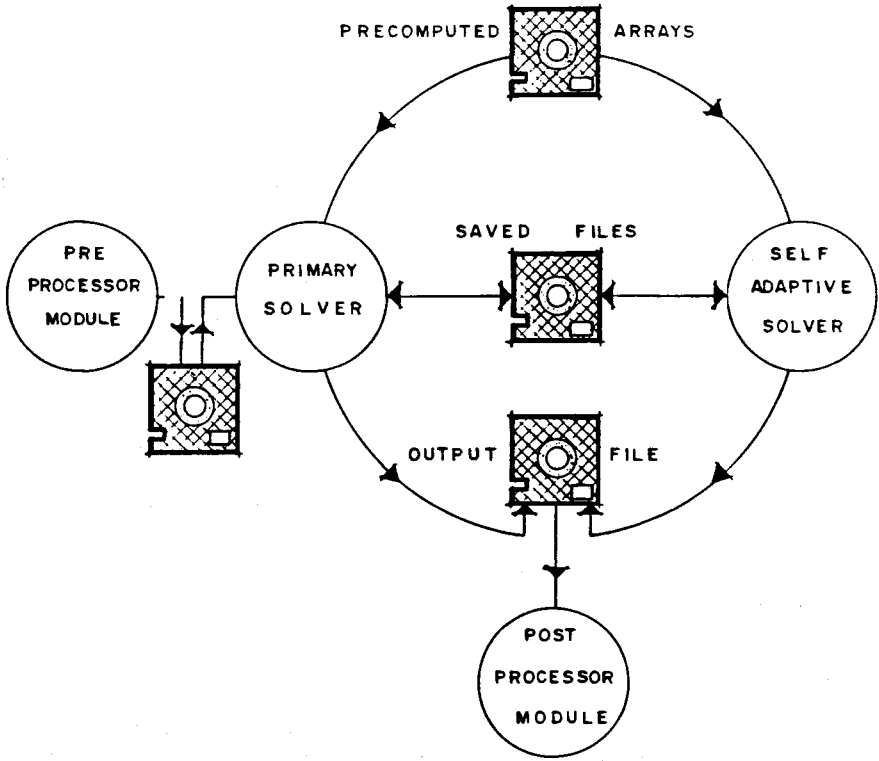
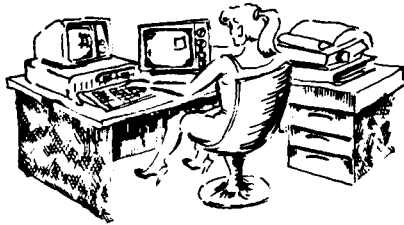


Figure 3. General scheme of adaptive procedure

It is a matter of convenience to use a finer mesh than the natural one (in the case of Figure 1, it could be interesting, for instance, to use a point in the middle of BC, as shown by mesh V), but that choice will be influenced by reasons which depend on the problem at hand and must be judged by the individual user.

As soon as the supports for the data and the solutions have been selected, it is necessary to represent the geometry and the boundary conditions inside each piece of boundary.

The freedom in choosing the procedure for this step is typical of adaptive methods. Due to the independence between geometry and interpolation, the solution to this part of the problem has no effect on the rest of the process, except on the accuracy of the representation.

In general, it is possible to use a pre-established preprocessor. On the contrary, in the program QUEIMADA (which we have written to run the various examples presented below), we have chosen to use a 'quasi'-isoparametric representation in the sense that the hierarchy of functions used to represent the geometry is the same as will be used to interpolate boundary data and boundary unknowns, although, of course, the number of members of the family used to represent each piece of information varies with the complexity. The isoparametric idea is not used, then, and equations (4) are no longer applied.

The interpolation is interactive (see Figure 3). The input is done according to whether or not the boundary is straight or curved. In the first case, only the vertex nodes are given, while in the second, it is necessary to introduce the interior points as well. What we do systematically is to define a 'centre' node, a 'quarter' one, etc., allowing the definition of linear, parabolic, cubic, quartic, etc., elements of the hierarchy. The same is repeated for the boundary conditions, where there is also an identification of Dirichlet-type corners that are immediately eliminated following the strategy described elsewhere.^{32,33}

Each choice is accompanied by a graphical test, comparing the input data and the interpolated curve in order to accept or to correct the approach. This is important because the curve interpolated through the hierarchical family described below is very sensitive to the correct position of the interior points.

Interpolating hierarchy

Once it has been decided how to define the geometry and the boundary conditions, the first decision for an adaptive procedure is the selection of a hierarchical family of interpolating functions.

In two-dimensional cases, due to the reduction to the boundary, the family is monodimensional. There are several possibilities, as can be seen, for instance, in Reference 34.

Once again, it is important to notice that, due to the independence of interpolating functions, it is of no use to try to look for orthogonal functions. This is why we have decided to work with the Peano family.

The first two members are the linear shape functions:

$$\begin{aligned} N_0 &= \frac{1}{2}(1 + \xi) \\ N_1 &= \frac{1}{2}(1 - \xi) \\ -1 &\leq \xi \leq 1 \end{aligned} \tag{7}$$

while the following are

$$N_p = \frac{1}{p!}(\xi^p - b)$$

where

$$\begin{aligned} p &> 2 \\ b &= 1 \quad \text{if } p \text{ odd} \\ b &= \xi \quad \text{if } p \text{ even} \end{aligned} \quad (8)$$

so that the development of a function $f(\xi)$ follows the pattern

$$f(\xi) = \frac{a_0}{2}(1 - \xi) + \frac{a_1}{2}(1 + \xi) + \frac{a_2}{2}(\xi^2 - 1) + \frac{a_2}{3!}(\xi^3 - \xi) + \dots \quad (9)$$

and the derivatives are

$$f''(0) = a_2; \quad f'''(0) = a_3; \quad f^{IV}(0) = a_4; \dots \quad (10)$$

giving a physical meaning to the parameters.

In this way, the first two members of the family produce the adjustment of the values at the ends of the macro-elements and the other ‘bubble’ functions correct the shape according to the derivatives in the middle of the interval.

Continuity of the geometry is respected, as is that of the potential. The derivatives along the boundary are not compatible in general, but this is an unimportant feature when working with the natural elements that have been selected in between corner places where derivatives present discontinuities for sure. Of course, the fluxes around corners are assumed to be discontinuous from the very beginning, following the scheme presented elsewhere.³² In this case, it is not necessary to establish any code to specify the type of boundary conditions because they are automatically identified during the input process. In addition, corners where boundary conditions are specified are locally solved without taking recourse to integral equations.

Establishment of influence matrices

Following the strategy indicated above, we see that the input data is minimum and we have a certain degree-of-freedom to choose the representation. In addition, we proceed by independent and autonomous refinement of the geometry, the potential and the fluxes, adjusting them as necessary.

The objective of the hierarchy is finally accomplished by interpolating the unknowns potential and fluxes, again in an independent fashion, where they are needed.

The criterion to add new functions to a previously approached solution is called an ‘indicator’ (and will be described below), and the sequential process is stopped when an ‘estimator’ measures a predetermined degree of error (see Figure 3).

The important thing is to realize that when the sequence of computation is finished, the number of integrals that has been computed is the minimum, according to the pre-established degree of accuracy. In this way, the most important part of the computational effort is reduced to an optimum and the computer time is minimized in a very rational manner. In addition, the procedure reduces computer time and produces better conditioning of matrices and weak couplings due to the ‘relative displacement’ approach inherent to successive refinements. Also, the influence matrices are nested so that all the previous work is used at each step and only the new elements of matrices need to be computed. Those effects are the same as observed when working with adaptive FEM and the pertinent conclusion is the same as well: that they present a promising field of application for iterative methods.

In order to see the nesting of influence matrices, let us imagine that the equations corresponding

to a certain discretization are as follows:

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{B}q \tag{11}$$

Each row corresponds to a collocation point, while each column reflects the integration over a shape function. Element a_{ij} is, for instance:

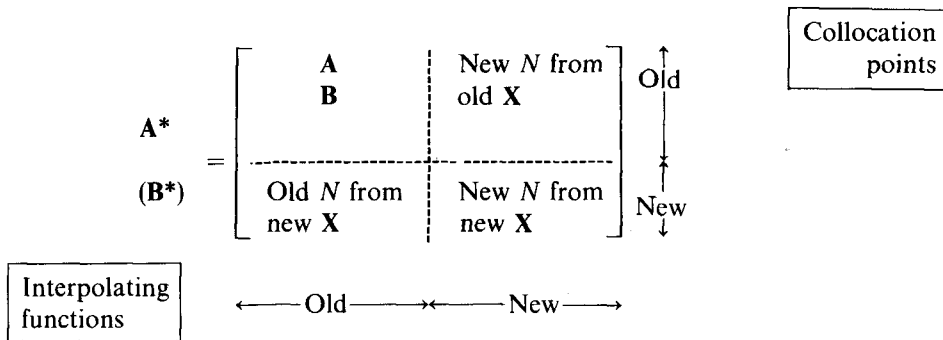
$$a_{ij} = \int_{\partial\Omega_j} N_j(\xi) q^*(\mathbf{x}_i, \xi) d\xi \tag{12}$$

where the integral is extended to the natural element supporting function N_j ; q^* is the weighting flux function produced when the fundamental solution is located at point \mathbf{x}_i ; \mathbf{x}_i is the collocation point used to establish the equation; and ξ is the co-ordinate along $\partial\Omega_j$.

If it is decided to introduce new interpolating functions, it is necessary to compute three blocks of integrals, i.e. (1) new interpolating functions viewed from old collocation points, (2) old interpolating functions viewed from new collocation points and (3) new interpolating functions viewed from new collocation points, producing the new system:

$$\mathbf{A}^*\boldsymbol{\phi} = \mathbf{B}^*q \tag{13}$$

where the new matrices have the following pattern:



As soon as the boundary conditions are imposed that nesting must be transmitted to the matrix to be solved, in such a way that, if the first system was

$$\mathbf{K}_{ii}\boldsymbol{\delta}_i = \mathbf{P}_i \tag{14}$$

the new one is

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ih} \\ \mathbf{K}_{hi} & \mathbf{K}_{hh} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_i \\ \boldsymbol{\delta}_h \end{bmatrix} = \begin{bmatrix} \mathbf{P}_i \\ \mathbf{P}_h \end{bmatrix} \tag{15}$$

As can be seen, \mathbf{K}_{ii} and \mathbf{P}_i are common in both formulations. Unfortunately,

$$\mathbf{K}_{hi} \neq \mathbf{K}_{ih}^T \tag{16}$$

and, then, to solve equation (15), it is necessary to compute all the new integrals. The procedure is to limit that computation to a minimum by using the 'estimator', which will be described below.

The decision about how to select the new collocation points which give rise to the new equations is also important. Once again, the freedom to choose is enormous and it would surely be worth while to study the influence of different choices on the accuracy of the results. In order to systematize the computations again, the collocation points are selected following the same strategy used to represent the geometry. In this way, the linear shape functions are associated

with the vertex, the parabolic with points in the middle of the element, the cubic with the 1/4th point, etc.

Estimators and indicators

As was suggested above, the objective of an adaptive method is to analyse the potentially most important contributions of higher order terms and to incorporate them to the solution in a sequential fashion until the desired level of accuracy has been reached.

The criterion developed to fulfil the first part of the problem is called an 'indicator', while the second is the stop criterion, or 'estimator'. Of course, mathematically speaking, that is the most interesting and difficult part of any adaptive procedure. Our solution here is only a first step based on previous results from FEM or on heuristic reasoning. From a computational point of view, the indicator has to be a reliable index, allowing clear decision making, but it must also be cheaply computable, i.e. the price of computing the indicator has to be less than that of adding a new degree-of-freedom without testing need for it.

The first idea for an indicator is that proposed by Peano²³ and based on the iterative solutions of the new system. The second set of equations in (15) can be written as

$$\mathbf{K}_{hi}\delta_i + \mathbf{K}_{hh}\delta_h = \mathbf{P}_h \quad (17)$$

so that if \mathbf{K}_{hh} is assumed to be diagonal, it is possible to estimate

$$\delta_j = \frac{P_j - \mathbf{K}_{ji}\delta_i}{K_{jj}} \quad (18)$$

Peano's criterion is based on

$$Q_j = \frac{\sum_{i \neq j} K_{ji}\delta_i - P_j}{\sqrt{K_{jj}}} \quad (19)$$

where δ_i is assumed to be equal to that of the previous step.

As shown by Gago,³⁴ equation (19) can be related to the refinement in energy while working with FEM (look at the dimensional units of Q_j).

In a collocation method like BEM, that line of reasoning is not so clear, but the value of (19) is maintained as an indicator of the convergence properties of system (15). The usual procedure is to test simultaneously (19) for all elements, assembling only those at which (19) are larger than a fixed bound, related in general to the relative different magnitudes Q_j .

In the case of FEM, $\mathbf{K}_{hi} = \mathbf{K}_{ih}^T$, which is not the case in BEM. Here, we found it easier to prepare matrix \mathbf{K}_{ih} during the previous computational steps, due to the fact that the \mathbf{K}_{ih} submatrix corresponds to integrations related to collocation points that are used for other integrals. The additional effort necessary to compute those integrals is not very large and this is why we substituted \mathbf{K}_{hi} with \mathbf{K}_{ih} .

In this way, for every step of the sequence, only vector \mathbf{P}_h and diagonal K_{jj} have to be computed.

Another indicator that we tried, called the ETSIIM indicator, is based on the hierarchical approach. If the previous variable is called u_1 and the refined one u_2 , we have approximately

$$\begin{aligned} u_2 - u_1 &\sim \delta_h N_h \\ u_1 &= \sum_0^{h-1} \delta_k N_k \end{aligned} \quad (20)$$

A measure of the relative importance in the total area that the introduction of a new function N_h

has with respect to the previous one is

$$\varepsilon_j^2 = \frac{\int_{\delta\Omega_j} (u_2 - u_1)^2}{\int_{\delta\Omega_j} u_1^2} \quad (21)$$

where the integral is done over the element j , where the refinement is being studied. Substituting (20) in (22), we obtain

$$\varepsilon_j^2 = \delta_h^2 \frac{\int_{\partial\Omega_j} N_h^2}{\int_{\partial\Omega_j} \left| \sum_{i=1}^{h-1} \delta_i N_i \right|^2} \quad (22)$$

To apply this criterion, it is necessary to estimate δ_h , which can be done using (18) or even by simply establishing

$$\delta_h \sim P_h / K_{hh} \quad (23)$$

The general trend of both indicators is to produce good results, although in some cases the second seems to produce more reasonable guidelines.

With regard to the estimator, the basis of existing codes in FEM is the *a posteriori* error estimates developed by Babuska *et al.*²⁴ Unfortunately, in our case, that mathematical support is still lacking, and we have been forced to use what we think can be a parallel to the equilibrium of residuals that is behind that idea.

In potential problems corresponding to the steady-state situation, the net flux around the boundary must be zero (i.e. globally 'equilibrated'):

$$\int_{\partial\Omega} q = 0 \quad (24)$$

In Neumann problems, where the boundary conditions are specified by giving the flux along the boundary, equation (24) is automatically fulfilled, but for mixed problems or the Dirichlet one, that condition can be used as a measure of the global error.

Another alternative is to use the equivalent to $E = \int_{\Omega} (\nabla\phi)^2$ on the boundary. That is, by using the relationship $\nabla(\phi\nabla\phi) = \nabla\phi \cdot \nabla\phi + \phi\nabla^2\phi$, it is possible to establish that, for the problem $\nabla^2\phi = 0$

$$E = \int_{\Omega} \nabla\phi \cdot \nabla\phi = - \int_{\Omega} \phi \nabla^2\phi + \int_{\Omega} \nabla(\phi\nabla\phi) = \int_{\partial\Omega} \phi q$$

COMPUTER PROGRAM

The previous ideas have been implemented in a simple program called QUEIMADA. It was written in BASIC for the IBM Personal Computer, using two screens, one to enter data and the other, a colour monitor, to deal with graphics. With the colour monitor, it is possible for the user to obtain a plot of the boundary, of the boundary conditions on every element, of the solution along the boundary and a spatial representation of the solution surface after having computed selected internal values.

The whole program is made up of four different subprograms (see Figure 3). The first is prepared for input data; the second, or primary solver, solves the problem by using a linear interpolation.

The third, or self-adaptive solver, does the adaptive refinement and the fourth subprogram computes internal points, plots results, etc.

Each program calls the following so that, from the user's point of view, it works as a unique block but has the advantage of using less storage space. Figures 4-6 present the main steps, while in Appendix I there is a listing of the primary solver.

PREPROCESSOR MODULE

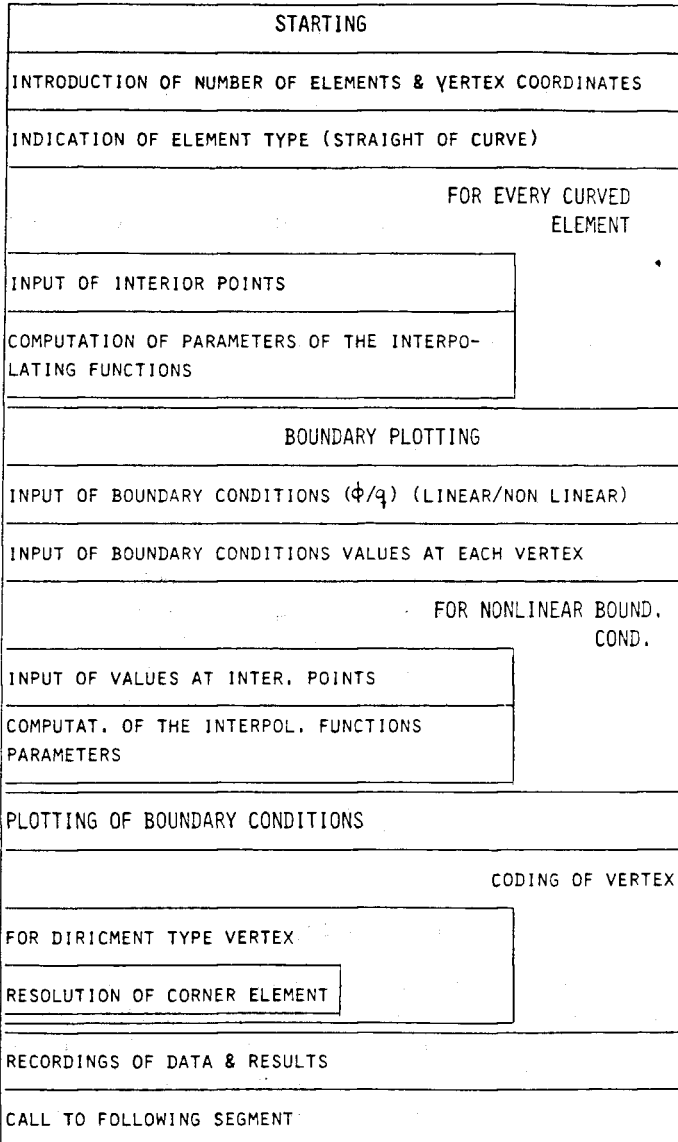


Figure 4. Nassi-Schneidermand block of preprocessor module

PRIMARY SOLVER

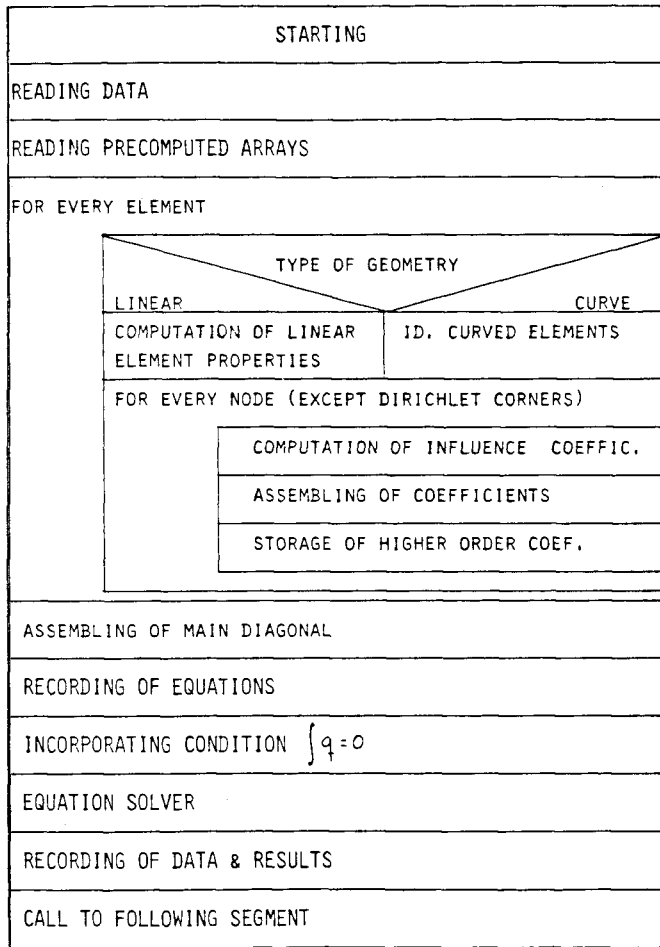


Figure 5. Nassi-Schneiderman block of primary solver

SELFADAPTIVE SOLVER

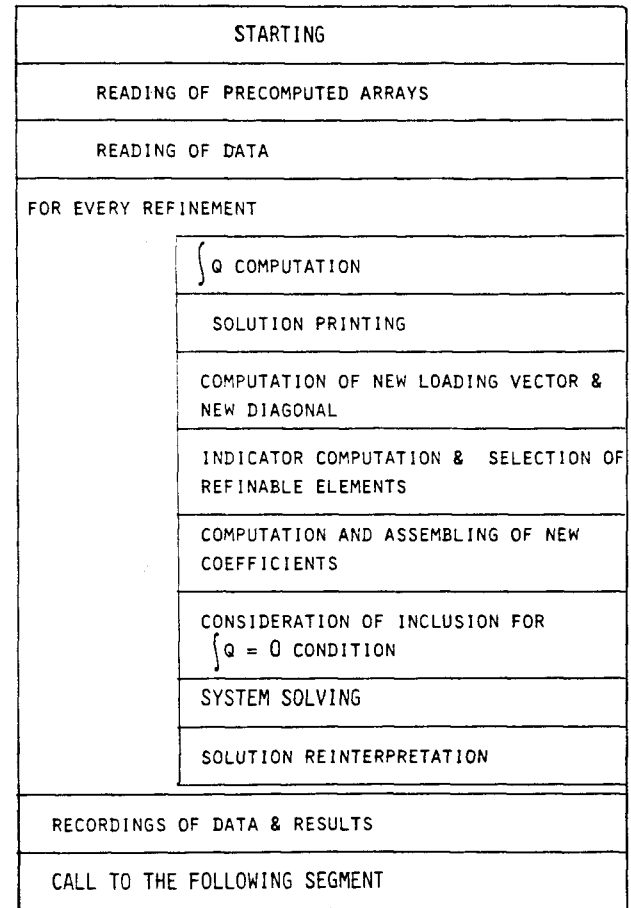


Figure 6. Nassi-Schneiderman block of self-adaptive solver

EXAMPLES

Consider the harmonic function (Figure 7):

$$\phi = x^3 - 3xy^2 \quad (25)$$

on the domain $-1 < x < 1$; $-1 < y < 1$. The natural elements are the four sides along which

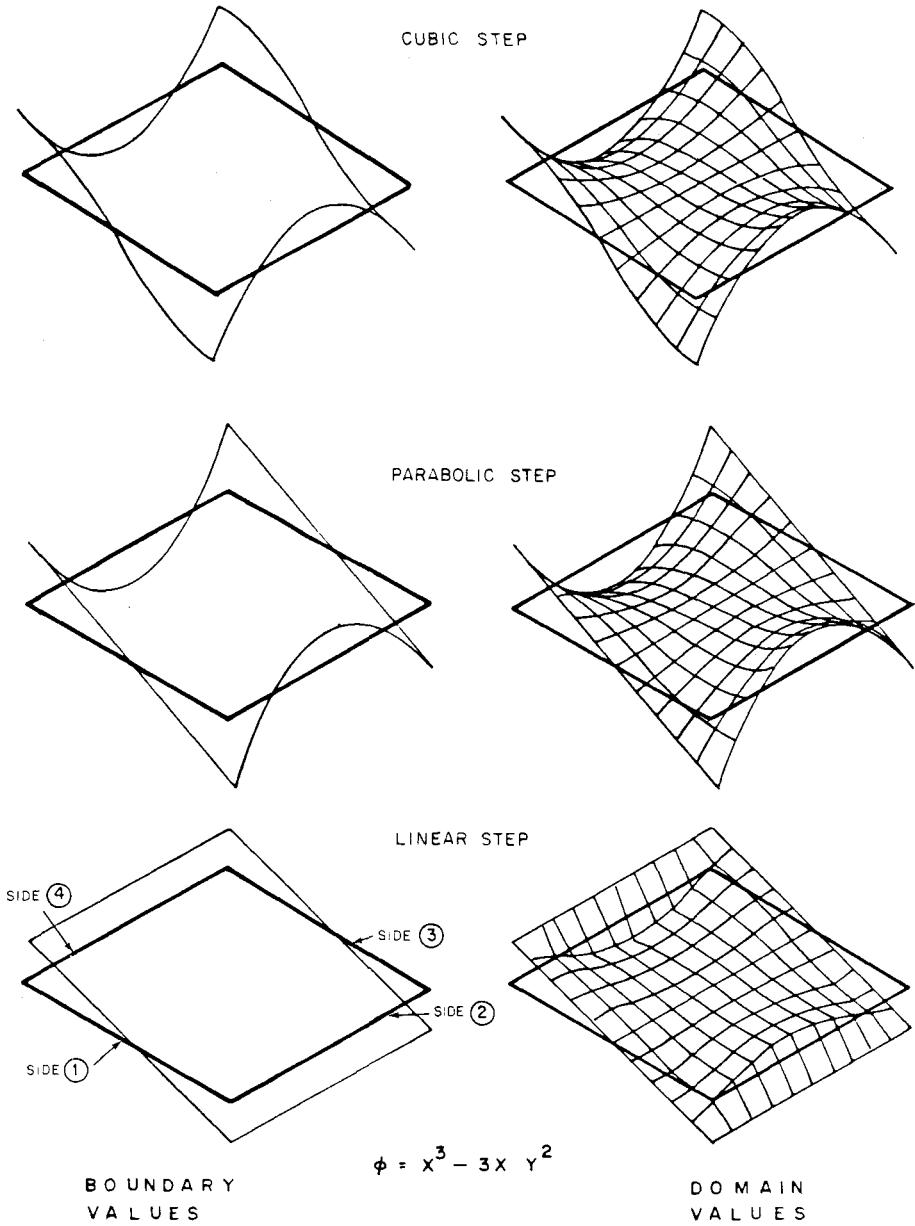


Figure 7. Adaptive solution of a Neumann problem in a square. $-1 < x < 1$; $-1 < y < 1$; $\phi = x^3 - 3xy^2$

potentials or fluxes are interpolated with the Peano series, according to the following:

$$\begin{aligned}
 \phi_1 &= 2N_0 - 2N_1 + 6N_3; & q_1 &= 6N_0 - 6N_1 \\
 \phi_2 &= -2N_0 - 2N_1 - 6N_2; & q_2 &= -6N_2 \\
 \phi_3 &= -2N_0 + 2N_1 - 6N_3; & q_3 &= -6N_0 + 6N_1 \\
 \phi_4 &= 2N_0 + 2N_1 + 6N_2; & q_4 &= 6N_2
 \end{aligned} \tag{26}$$

First, consider the Dirichlet problem; that is, the unknowns are fluxes along the boundary. In this case, it is not necessary to solve any system for the linear case because we are faced with four 'corner elements' that directly give the 8 values of the fluxes around the four corners. The results are:

$$\begin{aligned}
 q_1 &= 6N_0 - 6N_1 \\
 q_2 &= 0 \\
 q_3 &= -6N_0 + 6N_1 \\
 q_4 &= 0
 \end{aligned}$$

exactly reproducing the coefficients of (30) in elements 1 and 3. Once the linear approximation is obtained, the linear indicator values are as follows:

Element number	Q_j	ETSIIM
1	0.1363	0.015
2	25.96	1.7×10^{38}
3	0.1363	0.015
4	19.41	1.7×10^{38}

Both Q_j and ETSIIM indicate the convenience of refining sides 2 and 4. The 1.7×10^{38} values are really infinite because the area in the linear approach is zero, as mentioned above. When the second degree interpolating functions N_2 are added, the solutions are:

$$\begin{aligned}
 Q_2 &= -5.996N_2 \\
 Q_4 &= 5.996N_2
 \end{aligned}$$

which gives an idea of the accuracy obtained.

Due to the symmetries, the total flux is zero at all stages of refinement, i.e. the stopping criterion based on non-equilibrated flux will fail in this case.

Let us now study the Neumann problem. The solution is undetermined for a constant, but a point can be fixed in order to obtain 'relative' values. The results of the iteration as well as the indicator values are shown in Table I.

This problem has an interesting feature that is worth attention. On side 1 and 3, the potential variation is of the third degree, but the coefficient of the parabolic function is zero. On those sides, it is necessary to reach the third degree and, then, to pass through the second degree, which implies that the indicator cannot make any distinction as it happens.

In Figure 7, we show the results of the different iterations, not only for the boundary but also for interior points. The values corresponding to the linear approach have been displaced in order to get comparable figures. It is interesting to see the smoothing properties of the representation

Table I. Results for the Neumann problem in a square—first three steps

No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3
1	Φ	$-0.8000E+00$	$-0.3367E+01$	$+0.0000E+00$	$+0.0000E+00$
	q	$+0.6000E+01$	$-0.6000E+01$	$+0.0000E+00$	$+0.0000E+00$
2	Φ	$-0.3367E+01$	$-0.3367E+01$	$+0.0000E+00$	$+0.0000E+00$
	q	$+0.6000E+01$	$+0.0000E+00$	$-0.6000E+01$	$+0.0000E+00$
3	Φ	$-0.3367E+01$	$-0.8000E+00$	$+0.0000E+00$	$+0.0000E+00$
	q	$-0.6000E+01$	$+0.6000E+01$	$+0.0000E+00$	$+0.0000E+00$
4	Φ	$-0.8000E+00$	$-0.8000E+00$	$+0.0000E+00$	$+0.0000E+00$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.6000E+01$	$+0.0000E+00$

Linear interpolation

	No.	Q_j	No.	EPS	
Peano's criterion	1	$+0.1535E+01$	1	$+0.2023E+00$	ETSII criterion
	2	$+0.6747E+01$	2	$+0.5838E+00$	
	3	$+0.1367E+01$	3	$+0.1801E+00$	
	4	$+0.1226E+01$	4	$+0.4463E+00$	

No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3
1	Φ	$+0.1998E+01$	$-0.1999E+01$	$-0.8954E-06$	$+0.0000E+00$
	q	$+0.6000E+01$	$-0.6000E+01$	$+0.0000E+00$	$+0.0000E+00$
2	Φ	$-0.1999E+01$	$-0.1999E+01$	$-0.6133E+01$	$+0.0000E-00$
	q	$+0.0000E+00$	$+0.0000E+00$	$-0.6000E+01$	$+0.0000E+00$
3	Φ	$-0.1999E+01$	$+0.1998E+01$	$+0.1885E+06$	$+0.0000E+00$
	q	$-0.6000E+01$	$+0.6000E+01$	$-0.0000E+00$	$+0.0000E+00$
4	Φ	$+0.1998E+01$	$+0.1998E+01$	$+0.6133E+01$	$+0.0000E+00$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.6000E+01$	$+0.0000E+00$

Iteration number 1

	No.	Q_j	No.	EPS	
Peano's criterion	1	$+0.3565E+02$	1	$+0.3208E+01$	ETSII criterion
	2	$+0.3070E+00$	2	$+0.3482E-01$	
	3	$+0.2506E+02$	3	$+0.2255E+01$	
	4	$+0.5058E+01$	4	$+0.5736E+00$	

No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3
1	Φ	$+0.2001E+01$	$-0.2001E+01$	$-0.2961E+06$	$+0.6156E+01$
	q	$+0.6000E+01$	$-0.6000E+01$	$+0.0000E+00$	$+0.0000E+00$
2	Φ	$-0.2001E+01$	$-0.2001E+01$	$-0.6001E+01$	$+0.0000E+00$
	q	$+0.0000E+00$	$+0.0000E+00$	$-0.6000E+01$	$+0.0000E+00$
3	Φ	$-0.2001E+01$	$+0.2001E+01$	$+0.0000E+00$	$-0.6156E+01$
	q	$-0.6000E+01$	$+0.6000E+01$	$+0.0000E+01$	$+0.0000E+00$
4	Φ	$+0.0000E+01$	$+0.2001E+01$	$+0.6001E+01$	$+0.0000E+00$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.6000E+01$	$+0.0000E+00$

Iteration number 2

Table II. Results for the problem with a singular point—first three steps

Linear Interpolation						Iteration number 1	
$\Sigma q = 28.62521$						Peano's criterion	
No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3	No.	Q/j
1	Φ	$+0.5000E+03$	$+0.5000E+03$	$+0.0000E+00$	$+0.0000E+00$	1	$+0.9559E+04$
	q	$+0.3139E+02$	$-0.1423E+03$	$+0.0000E+00$	$+0.0000E+00$	2	$+0.3408E+04$
2	Φ	$+0.5000E+03$	$+0.1000E+04$	$+0.0000E+00$	$+0.0000E+00$	3	$+0.7760E+04$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	4	$+0.2925E+04$
3	Φ	$+0.1000E+04$	$+0.1000E+04$	$+0.0000E+00$	$+0.0000E+00$	5	$+0.3819E+04$
	q	$+0.5377E+02$	$+0.6531E+02$	$+0.0000E+00$	$+0.0000E+00$	$\Sigma q =$	0.7882691
						ETSII criterion	
						No.	EPS
4	Φ	$+0.1000E+04$	$+0.5181E+03$	$+0.0000E+00$	$+0.0000E+00$	1	$+0.5775E+02$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	2	$+0.6949E+00$
						3	$+0.5857E+02$
5	Φ	$+0.5181E+03$	$+0.5000E+03$	$+0.0000E+00$	$+0.0000E+00$	4	$+0.4174E+00$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	5	$+0.1168E+01$
						Iteration Number 2	
						Peano's criterion	
No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3	No.	Q/j
1	Φ	$+0.5000E+03$	$+0.5000E+03$	$+0.0000E+00$	$+0.0000E+00$	1	$+0.8786E+04$
	q	$-0.4074E+02$	$-0.2049E+03$	$-0.2140E+03$	$+0.0000E+00$	2	$+0.7245E+04$
						3	$+0.5069E+04$
2	Φ	$+0.5000E+03$	$+0.1000E+04$	$-0.1086E+03$	$+0.0000E+00$	4	$+0.6381E+04$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	5	$+0.1062E+05$
					$\Sigma q =$	1.091766	
3	Φ	$+0.1000E+04$	$+0.1000E+04$	$+0.0000E+00$	$+0.0000E+00$	ETSII criterion	
	q	$+0.5644E+02$	$+0.4794E+02$	$+0.1854E+01$	$+0.0000E+00$		
						No.	
4	Φ	$+0.1000E+04$	$+0.5840E+03$	$+0.1973E+03$	$+0.0000E+00$	EPS	
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$		
5	Φ	$+0.5840E+03$	$+0.5000E+03$	$-0.3951E+02$	$+0.0000E+00$	1	$+0.5159E+01$
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	2	$+0.5029E+00$
					3	$+0.4432E+01$	
					4	$+0.3399E+00$	
					5	$+0.1061E+01$	
No.	Φ/q	Coeff. No.	Coeff. No. 1	Coeff. No. 2	Coeff. No. 3		
1	Φ	$+0.5000E+03$	$+0.5000E+03$	$+0.0000E+00$	$+0.0000E+00$		
	q	$-0.1339E+02$	$-0.2338E+03$	$-0.2174E+03$	$-0.5055E+03$		
2	Φ	$+0.5000E+03$	$+0.1000E+04$	$-0.1140E+03$	$+0.4393E+03$		
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$		
3	Φ	$+0.1000E+04$	$+0.1000E+04$	$+0.0000E+00$	$+0.0000E+00$		
	q	$+0.5087E+02$	$+0.5165E+02$	$+0.3320E+01$	$+0.6384E+02$		
4	Φ	$+0.1000E+04$	$+0.5839E+03$	$+0.1933E+03$	$+0.1703E+03$		
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$		
5	Φ	$+0.5839E+03$	$+0.5000E+03$	$-0.3396E+02$	$+0.1724E+03$		
	q	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$	$+0.0000E+00$		

formula that produces better results inside the domain, in spite of the rough approximations of the boundary.

The second example has been treated repeatedly in technical literature and corresponds to Figure 2. Here, the interesting thing is whether or not the singular point is assimilated by successive refinements.

The values taken in the first three steps are given in Table II. Q_j indicators detect that it is necessary to specially refine elements 1 and 3 (see Figure 2, mesh I), where the unknowns in flux are located ETSIIM also does so, but more clearly. For the cubic interpolation, Q_j is once again a bit misleading, while ETSIIM insists clearly on 1 and 3.

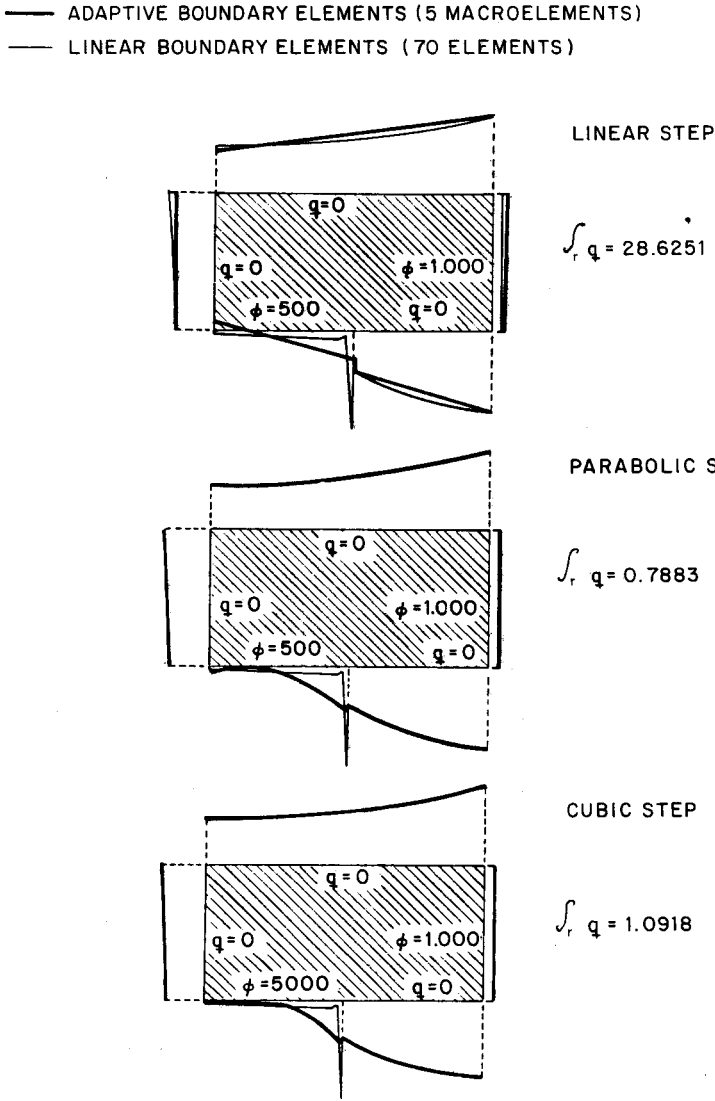


Figure 8. Adaptive solution of a mixed problem with a singular point—boundary values. Thin line: solution obtained with a mesh of 70 linear elements; thick line: approximation obtained by applying the adaptive technique to 5 macro-elements

In Figure 8, we see how the evolution of the solution tries to accommodate the singularity. The estimator $\int q$ follows the first refinement well, while the second produces a saturation that seems to be logical because the global trends of the solutions have already been caught. This saturation is also seen in other examples and can be used to stop the sequence.

In Figure 9, we represent how the surface converges and, once again, how rough results in the boundary convey refined ones inside the domain. Although the mesh near the singularity is very broad, it is possible to see the trend of the slopes near it.

As an example of the above-mentioned saturation, Figure 10 presents the convergence in energy for the problem in Figure 1. It is interesting to see that the rate of convergence is higher with P

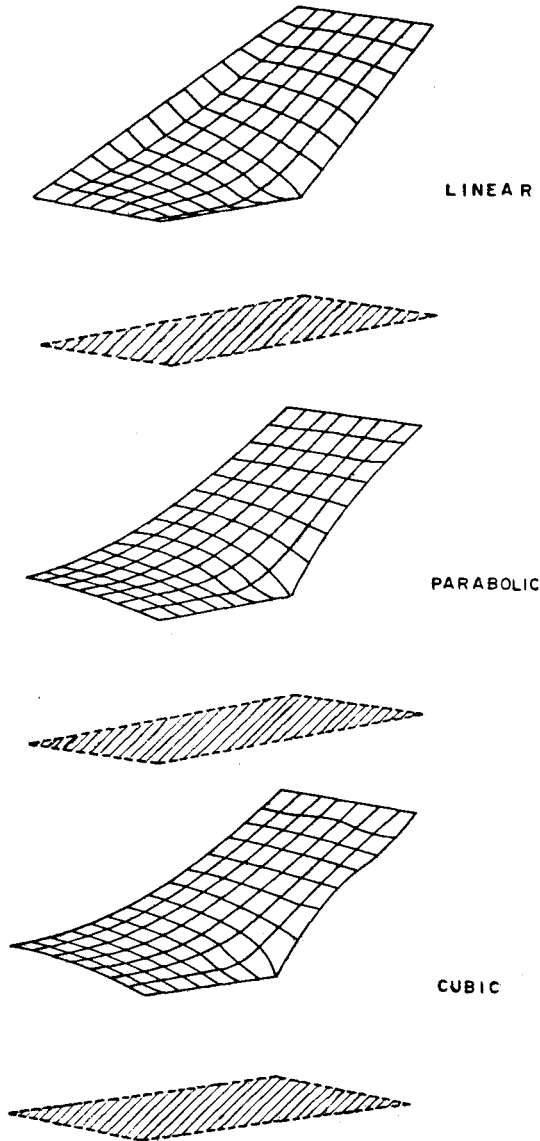
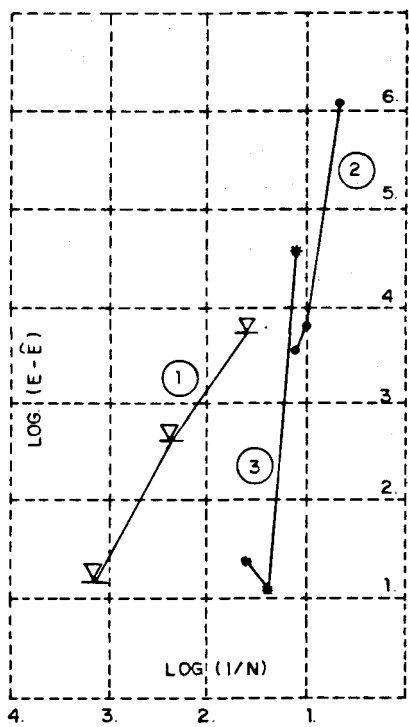
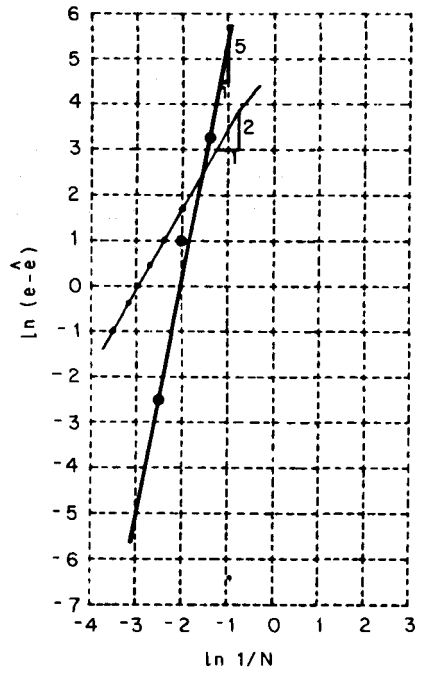


Figure 9. Adaptive solution of mixed problem with a singular point—domain values



▽ LINEAR
 * ADAPTIVE

(a)



— LINEAR
 — ADAPTIVE

(b)

Figure 10. (a) Error evolutions for torsion of elliptic shaft: (1) linear elements; (2) ● three boundary elements; (3) ★ four boundary elements. (b) Error evolution for the Neumann problem in the square; E , theoretical value; \hat{E} , computed value

adaptive elements, especially when the curved side is discretized by two arches. It is also interesting to see the saturation showed by p -adaptive elements, indicating that it is impossible to progress without further subdivisions. On the other hand, Figure 10(b) shows that the evolution of the E estimator on the square is monotonic, at least as far as the degree we have tested. The trend of higher rates of convergence against a classical linear BEM approach is again confirmed.

CONCLUSIONS

The p -adaptive approach seems to be a promising area in BEM research. As we have seen, the method works well even if only approximate mathematical criteria for refinement and convergence are available.

In general, and due to the correct behaviour of solutions in elliptic problems, the convergence rate is high enough, and only the minimum number of elements compatible with geometric or boundary condition discontinuities have to be used. The advantages of that are twofold: the effort to prepare input data is greatly reduced and the storage required is less than in the classical

approach. The computational effort is also reduced to a minimum, so that micros can be used very effectively to solve problems.

The graphic possibilities of micros have also been exploited since the very beginning. For instance, the user masters the degree of approximation. Since the refinement is selective (either automatic or interactive), it acts there where it is strictly necessary.

In addition, a single set of data allows the obtention of different solutions in opposition to the classical approach, where every refinement can only be done after completely redoing a new data set.

The extension to elasticity problems presents no difficulty, except for the vectorial character of the basic equation that increases the number of variables to be interpolated.

APPENDIX

```

10 'SAVE"LABEM' 'version de grado 9
20 '*****
30 '* P R O G R A M Q U E I M A D A *
40 '* ( primary solver ) *
50 '*****
60 '
70 OPTION BASE 1: MMM=9
80 CLS : LPRINT CHR$(15) : WIDTH"LFT1:",100
81 DIM PGAUSS(8),WI(8)
82 PGAUSS(1)=-.96028986# :PGAUSS(2)=-PGAUSS(1):PGAUSS(3)=.79666648#:PGAUSS(4)=-PGAUSS(3):PGAUSS(5)=.52553241#:PGAUSS(6)=-PGAUSS(5):PGAUSS(7)=.18343464#:PGAUSS(8)=-PGAUSS(7)
84 WI(1)=-.10122854#:WI(2)=WI(1):WI(3)=.22381034#:WI(4)=WI(3):WI(5)=-.31370635#:WI(6)=WI(5):WI(7)=-.36268378#:WI(8)=WI(7)
90 '
100 ' READING OF THE DATA
110 ' -----
120 '
130 OPEN "FABEM1" FOR INPUT AS #1
140 INPUT #1,NVER,NLCURV,NCOND4,NCOND1
150 NEC=NVER-NCOND4 : NECM1=NEC+1
160 DIM GEOMET(NVER,9),CARGA(NVER,10,2),ICODE(NVER)
170 IF NEC=0 THEN 200
180 DIM WEIGAUSS(4),STROUD(4),DSETA(4),ADJAG(NEC),COEF(NEC,NECM1)
190 DIM CHIC(4,18),N(18),A(9),B(9),PCOEF(NEC,7*NVER)
200 IF NLCURV=0 THEN 220
210 DIM CARCURV(NLCURV,21)
220 FOR I=1 TO NVER
230 FOR J=1 TO 9
240 INPUT #1,GEOMET(I,J)
250 NEXT J
260 NEXT I
270 FOR I=1 TO NVER
280 FOR J=1 TO 10
290 INPUT #1,CARGA(I,J,1),CARGA(I,J,2)
300 NEXT J
310 NEXT I
320 FOR I=1 TO NVER
330 INPUT #1,ICODE(I)
340 NEXT I
350 CLOSE #1
360 IF NEC=0 THEN 1390
370 '
380 ' READING OF PRECOMPUTED ARRAYS
390 ' -----
400 '
410 OPEN "FABEM3" FOR INPUT AS #1
420 FOR I=1 TO 4
430 INPUT #1,WEIGAUSS(I),STROUD(I),DSETA(I)

```

```

440     FOR J=1 TO 18
450         INPUT #1,CHIC(I,J)
460     NEXT J
470 NEXT I
480 CLOSE #1
490 *
500 *
510 *             COMPUTATION OF INFLUENCE MATRIX
520 * -----
530 *
540 CLS : PRINT "COEFFICIENTS OF THE SYSTEM USING LINEAR INTERPOLATION";
550 LPRINT " LINEAR INTERPOLATION "
560 PRINT " ARE BEING COMPUTED"
570 NCOL1=0
580 CV=0
590 FOR I=1 TO NVER
600     IF I=NVER THEN II=1 ELSE II=I+1
610     IF I=1 AND ICODE(I)<>4 THEN NCOL1=1
620     IF I>1 THEN NCOL1=NCOL2
630     IF ICODE(II)<>4 THEN NCOL2=NCOL1+1 ELSE NCOL2=NCOL1
640     IF II=1 THEN NCOL2=1
650 *
660     IF GEOMET(I,1)=4 THEN B30
670 *
680     X1=GEOMET(I,2)
690     Y1=GEOMET(I,3)
700     X2=GEOMET(II,2)
710     Y2=GEOMET(II,3)
720     DX=X2-X1
730     DY=Y2-Y1
740     LONG=DX*DX+DY*DY
750     LONG=SQR(LONG)
760     UN1=DY/LONG
770     UN2=-DX/LONG
780     B1=(.75 - .5 *LOG(LONG))*LONG
790     B2=(.25 - .5 *LOG(LONG))*LONG
800     B3=(-.2777778 + .3333333 *LOG(LONG))*LONG
810     B4=2.77778E-02 *LONG
815     B5=(-13/450+LOG(LONG)/30)*LONG
816     B6=-.24444444*LONG/120
817     B7=LONG*(-.7605442-.85714285*LOG(1/LONG))/720
818     B8=.29047619*LONG/5040
819     B9=LONG*(-.8041093# -.88888888# *LOG(1/LONG))/40320
820     B0TO 1010
830
840     CV=CV+1
850     FOR M=1 TO 4
860         DXCHI=0
870         DYCHI=0
880         FOR N=1 TO 4
890             CARCURV(CV,1)=I
900             CARCURV(CV,5*M-3)=CARCURV(CV,5*M-3)+GEOMET(I,2*N)*CHIC(M,N)
910             CARCURV(CV,5*M-2)=CARCURV(CV,5*M-2)+GEOMET(I,2*N+1)*CHIC(M,N)
920             DXCHI=DXCHI+GEOMET(I,2*N)*CHIC(M,N+9)
930             DYCHI=DYCHI+GEOMET(I,2*N+1)*CHIC(M,N+9)
940         NEXT N
950         DSCHI=SQR(DXCHI^2+DYCHI^2)
960         CARCURV(CV,5*M-1)=DSCHI
970         CARCURV(CV,5*M)=DYCHI/DSCHI
980         CARCURV(CV,5*M+1)=-DXCHI/DSCHI
990     NEXT M
1000 *
1010     NF1=0
1020     FOR J=1 TO NVER
1030         IF ICODE(J)=4 THEN 1190
1040         FOR K=1 TO 9 : A(K)=0 : B(K)=0 : NEXT K
1050         NF1=NF1+1
1060         IF GEOMET(I,1)=4 THEN 1150

```

```

1070 *
1080     IF J=I THEN B(1)=B1:B(2)=B2:B(3)=B3:B(4)=B4:B(5)=B5:B(6)=B6:B(7)=B7:B
(8)=B8:B(9)=B9: GOTO 1170
1090     IF J=II THEN B(1)=B2:B(2)=B1:B(3)=B3:B(4)=-B4:B(5)=B5:B(6)=-B6:B(7)=B
7:B(8)=-B8:B(9)=B9:- GOTO 1170
1100     XP=GOMET(J,2) : YP=GOMET(J,3)
1110     RADN=(X1-XP)*UN1+(Y1-YP)*UN2
1120     GOSUB 1750
1130     GOTO 1170
1140 *
1150     GOSUB 1960
1160 *
1170     ADIAG(NFI)=ADIAG(NFI)-A(1)-A(2)
1180     GOSUB 2560
1190     NEXT J
1200     NEXT I
1210 *
1220     GOSUB 3090
1230     OPEN "COEFI" FOR OUTPUT AS #1
1240     FOR I=1 TO NEC
1250         FOR J=1 TO NECM1
1260             PRINT #1,COEF(I,J)
1270         NEXT J
1280     NEXT I
1290     CLOSE #1 : BEEP :CLS
1291     IF NCOND1=NVER THEN 1340
1300     PRINT " DO YOU WANT TO CHANGE ANY EQUATION OR ADD THE CONDITION  $\Sigma q=0$  ";
1310     PRINT " ( c/a/n ) ? "
1320     A$=INKEY$ : IF A$="" THEN 1320
1330     IF A$<>"n" AND A$<>"N" THEN GOSUB 4020
1340     GOSUB 3570
1350     GOSUB 3270
1360 *
1370 *
1380 *           RECORDING OF THE DATA
1390 * -----
1400 *
1410 *
1420     OPEN "FABEM2" FOR OUTPUT AS #1
1430     PRINT #1, NVER,NLCURV,NEC,NCOND1
1440     FOR I=1 TO NEC
1450         PRINT #1,VEC(I)
1460     NEXT I
1470     FOR I=1 TO NVER
1480         FOR J=1 TO 10
1490             PRINT #1,CARGA(I,J,1),CARGA(I,J,2)
1500         NEXT J
1510     NEXT I
1520     FOR I=1 TO NVER
1530         FOR J=1 TO 9
1540             PRINT #1,GOMET(I,J)
1550         NEXT J
1560     NEXT I
1570     FOR I=1 TO NVER
1580         PRINT #1,ICODE(I)
1590     NEXT I
1600     FOR I=1 TO NEC
1610         FOR J=1 TO 7*NVER
1620             PRINT #1,PCOEF(I,J)
1630         NEXT J
1640     NEXT I
1650     IF NLCURV=0 THEN 1710
1660     FOR I=1 TO NLCURV
1670         FOR J=1 TO 21
1680             PRINT #1,CARCURV(I,J)
1690         NEXT J
1700     NEXT I
1710     CLOSE #1

```

```

1720 RUN"ABEM
1730 *
1740 *
1750 *          INTEGRAL COMPUTATIONS IN STRAIGHT ELEMENTS
1760 *
1770 *

```

```

1775 GOSUB 8410
1776 GOTO 1930
1780   FOR K=1 TO 4
1790     XL0C=X1*CHIC(K,1)+X2*CHIC(K,2)
1800     YL0C=Y1*CHIC(K,1)+Y2*CHIC(K,2)
1810     RAD2=(XP-XL0C)^2+(YP-YL0C)^2
1820     RAD=SQR(RAD2)
1830     WK=WEIGAUSS(K)
1840     FOR N=1 TO 9
1850       A(N)=A(N)-(CHIC(K,N)/RAD2)*WK
1860       B(N)=B(N)-CHIC(K,N)*LOG(RAD)*WK
1870     NEXT N
1880   NEXT K
1890   FOR K=1 TO 9
1900     A(K)=A(K)*RADN*LONG/2
1910     B(K)=B(K)*LONG/2
1920   NEXT K
1930   RETURN

```

```

1940 *
1950 *
1960 *          INTEGRAL COMPUTATIONS IN CURVED ELEMENTS
1970 *
1980 *

```

```

1990 XD=GEOMET(J,2)
2000 YD=GEOMET(J,3)
2010 NCURV=CV
2020 FOR K=1 TO 4
2030   X1=CARCURV(NCURV,5*K-3)
2040   Y1=CARCURV(NCURV,5*K-2)
2050   JAC=CARCURV(NCURV,5*K-1)
2060   N1=CARCURV(NCURV,5*K)
2070   N2=CARCURV(NCURV,5*K+1)
2080   WK=WEIGAUSS(K)
2090   RAD2=(X1-XD)*(X1-XD)+(Y1-YD)*(Y1-YD)
2100   RAD=SQR(RAD2)
2110   F1=((X0-X1)*N1+(Y0-Y1)*N2)*(JAC/RAD2)*WK
2120   FOR N=1 TO 9
2130     A(N)=A(N)+F1*CHIC(K,N)
2140   NEXT N
2150   IF I<>J AND II<>J THEN F2=1/RAD : GOTO 2170
2160   IF I=J THEN F2=CHIC(K,2)/RAD ELSE F2=CHIC(K,1)/RAD
2170   F2=LOG(F2)
2180   FOR N=1 TO 9
2190     B(N)=B(N)+F2*JAC*WK*CHIC(K,N)
2200   NEXT N
2210 NEXT K
2220 IF I<>J AND II<>J THEN 2530
2230 *
2240 *   computation of singular part of coefficient B
2250 *
2260 FOR K=1 TO 4
2270   IF I=J THEN CSI=2*DSETA(K)-1 ELSE CSI=1-2*DSETA(K)
2280   N(1)=.5*(1-CSI)
2290   N(2)=.5*(1+CSI)
2300   N(3)=.5*(CSI*CSI-1)
2310   N(4)=CSI*N(3)/3
2315   N(5)=(CSI^4-1)/24
2316   N(6)=(CSI^5-CSI)/120
2317   N(7)=(CSI^6-1)/720
2318   N(8)=(CSI^7-CSI)/5040
2319   N(9)=(CSI^8-1)/40320
2320   N(10)=-.5

```



```

2330 N(11)=.5
2340 N(12)=CSI
2350 N(13)=.166666*(3*CSI*CSI-1)
2355 N(14)=CSI^3/6
2356 N(15)=(5*CSI^4-1)/120
2357 N(16)=(CSI^5)/120
2358 N(17)=(7*CSI^6-1)/5040
2359 N(18)=CSI^7/5040
2360 X1=0
2370 Y1=0
2380 DDX=0
2390 DDY=0
2400 FOR L=1 TO 4
2410 X1=X1+N(L)*GEOMET(I,2*L)
2420 Y1=Y1+N(L)*GEOMET(I,2*L+1)
2430 DDX=DDX+N(9+L)*GEOMET(I,2*L)
2440 DDY=DDY+N(9+L)*GEOMET(I,2*L+1)
2450 NEXT L
2460 JAC=DDX*DDX+DDY*DDY
2470 JAC=SGR(JAC)
2480 WKK=STROUD(K)
2490 FOR L=1 TO 9
2500 B(L)=B(L)+JAC*WKK*N(L)*2
2510 NEXT L
2520 NEXT K
2530 RETURN
2540 *
2550 *
2560 * SYSTEM ASSEMBLAGE
2570 *
2580 *
2590 *LPRINT "ON SIDE ";I;"-";I1,"AT THE NODE ";J
2600 *LPRINT "A1=";A(1);" A2=";A(2);" A3=";A(3);" A4=";A(4);" A5=";A(5)
2610 *LPRINT "B1=";B(1);" B2=";B(2);" B3=";B(3);" B4=";B(4);" B5=";B(5)
2620 ON ICODE(I) GOTO 2640,2680,2710,2680,2710
2630 *
2640 COEF(NF1,NECM1)=COEF(NF1,NECM1)+B(1)*CARGA(I,2,2)
2650 COEF(NF1,NCOL1)=COEF(NF1,NCOL1)+A(1)
2660 GOTO 2740
2670 *
2680 COEF(NF1,NECM1)=COEF(NF1,NECM1)-A(1)*CARGA(I,2,1)+B(1)*CARGA(I,2,2)
2690 GOTO 2740
2700 *
2710 COEF(NF1,NECM1)=COEF(NF1,NECM1)-A(1)*CARGA(I,2,1)
2720 COEF(NF1,NCOL1)=COEF(NF1,NCOL1)-B(1)
2730 *
2740 ON ICODE(I1) GOTO 2760,2830,2800,2800,2830
2750 *
2760 COEF(NF1,NECM1)=COEF(NF1,NECM1)+B(2)*CARGA(I,3,2)
2770 COEF(NF1,NCOL2)=COEF(NF1,NCOL2)+A(2)
2780 GOTO 2860
2790 *
2800 COEF(NF1,NECM1)=COEF(NF1,NECM1)-A(2)*CARGA(I,3,1)+B(2)*CARGA(I,3,2)
2810 GOTO 2860
2820 *
2830 COEF(NF1,NECM1)=COEF(NF1,NECM1)-A(2)*CARGA(I,3,1)
2840 COEF(NF1,NCOL2)=COEF(NF1,NCOL2)-B(2)
2850 *
2860 IF CARGA(I,1,1) <> 4 THEN 2890
2870 COEF(NF1,NECM1)=COEF(NF1,NECM1)-A(3)*CARGA(I,4,1)-A(4)*CARGA(I,5,1)
2880 *
2890 IF CARGA(I,1,2) <> 4 THEN 2920
2900 COEF(NF1,NECM1)=COEF(NF1,NECM1)+B(3)*CARGA(I,4,2)+B(4)*CARGA(I,5,2)
2910 *
2920 IF CARGA(I,1,2)=0 THEN 2970
2930 PCOEF(NF1,7*I-6)=A(3)
2935 PCOEF(NF1,7*I-5)=A(4)
2940 PCOEF(NF1,7*I-4)=A(5)

```

```

2945 PCOEF (NFI, 7*I-3)=A(6)
2946 PCOEF (NFI, 7*I-2)=A(7)
2947 PCOEF (NFI, 7*I-1)=A(8)
2948 PCOEF (NFI, 7*I)=A(9)
2950 GOTO 3000
2960 *
2970 PCOEF (NFI, 7*I-6)=-B(3)
2975 PCOEF (NFI, 7*I-5)=-B(4)
2980 PCOEF (NFI, 7*I-4)=-B(5)
2985 PCOEF (NFI, 7*I-3)=-B(6)
2986 PCOEF (NFI, 7*I-2)=-B(7)
2987 PCOEF (NFI, 7*I-1)=-B(8)
2988 PCOEF (NFI, 7*I)=-B(9)
2990 *LPRINT :LPRINT "FUTURE COEFFICIENTS"
3000 *FOR P=1 TO NEC
3010 *   FOR I=1 TO 3*NVER
3020 *     LPRINT PCOEF (P, I);
3030 *     NEXT I
3040 *   LPRINT
3050 *NEXT P
3060 RETURN
3070 *
3080 *
3090 *           DIAGONAL COEFFICIENTS ASSEMBLAGE
3100 *-----
3110 *
3120 NFI=0
3130 FOR I=1 TO NVER
3140   ON ICODE (I) GOTO 3160, 3200, 3200, 3230, 3200
3150 *
3160   NFI=NFI+1
3170   COEF (NFI, NFI)=COEF (NFI, NFI)+ADIAG (NFI)
3180   GOTO 3230
3190 *
3200   NFI=NFI+1
3210   COEF (NFI, NECM1)=COEF (NFI, NECM1)-ADIAG (NFI) *CARGA (I, 2, 1)
3220 *
3230 NEXT I
3240 RETURN
3250 *
3260 *
3270 *           BACKSUSTITUTION
3280 *-----
3290 *
3300 CLS :PRINT " SOLUTION HAS BEEN OBTAINED AND IT'S BEING DECODED"
3310 NFI=0 : I=NVER
3320 FOR II=1 TO NVER
3330   ON ICODE (II) GOTO 3350, 3400, 3440, 3520, 3480
3340 *
3350   NFI=NFI+1
3360   CARGA (I, 3, 1)=VEC (NFI)
3370   CARGA (II, 2, 1)=VEC (NFI)
3380   GOTO 3520
3390 *
3400   NFI=NFI+1
3410   CARGA (I, 3, 2)=VEC (NFI)
3420   GOTO 3520
3430 *
3440   NFI=NFI+1
3450   CARGA (II, 2, 2)=VEC (NFI)
3460   GOTO 3520
3470 *
3480   NFI=NFI+1
3490   CARGA (I, 3, 2)=VEC (NFI)
3500   CARGA (II, 2, 2)=VEC (NFI)
3510 *
3520   I=II
3530 NEXT II

```

```

3540 RETURN
3550 '
3560 '
3570 '      SOLUTION OF THE LINEAR EQUATIONS SYSTEM
3580 '
3590 '-----
3600 CLS : PRINT " SYSTEM IS BEING SOLVED"
3610 DIM VEC(NEC),AL(NEC+1)
3620 'PRINT "SISTEMA DE ECUACIONES":FOR I=1 TO NEC
3630 '  FOR J=1 TO NEC+1
3640 '      PRINT COEF(I,J);
3650 '      LPRINT COEF(I,J);
3660 '  NEXT J
3670 'PRINT:LPRINT
3680 'NEXT I
3690 FOR I=1 TO NEC-1
3700   IN=0
3710   ME=0
3720   FOR J=1 TO NEC
3730     IF (ABS(COEF(J,I))>ME) THEN ME=ABS(COEF(J,I)):IN=J
3740   NEXT J
3750   IF IN=0 THEN PRINT "singular matrix":STOP
3760   FOR J=1 TO NEC+1
3770     VI=COEF(I,J)
3780     COEF(I,J)=COEF(IN,J)
3790     COEF(IN,J)=VI
3800   NEXT J
3810   FOR K=1 TO NEC+1
3820     AL(K)=COEF(I,K)/COEF(I,I)
3830   NEXT K
3840   FOR K=I+1 TO NEC
3850     MA=COEF(K,I)
3860     FOR J=1 TO NEC+1
3870       COEF(K,J)=COEF(K,J)-AL(J)*MA
3880     NEXT J
3890   NEXT K
3900 NEXT I
3910 VEC(NEC)=COEF(NEC,NEC+1)/COEF(NEC,NEC)
3920 FOR K=NEC-1 TO 1 STEP -1
3930   SUMA=0
3940   FOR KK=K+1 TO NEC
3950     SUMA=SUMA-VEC(KK)*COEF(K,KK)
3960   NEXT KK
3970   VEC(K)=(COEF(K,NEC+1)+SUMA)/COEF(K,K)
3980 NEXT K
3990 RETURN
4000 '
4010 '
4020 '      CHANGE OF ANY EQUATION FOR THE FLUX CONDITION
4030 '
4040 '-----
4050 IF A$="a" OR A$="A" THEN NECU=NEC:LPRINT" ADDED  $\Sigma q=0$  CONDITION":GOTO 4130
4060 INPUT " INPUT THE EQUATION NUMBER ";NECU
4070 LPRINT : LPRINT " EQUATION NUMBER";NECU;" IS CHANGED FOR  $\Sigma q=0$ "
4080 IF NECU < 1 OR NECU > NEC THEN PRINT: BEEP:GOTO 4060
4090 '
4100 FOR I=1 TO NEC+1
4110   COEF(NECU,I)=0      , ' the equation is put to zero
4120 NEXT I
4130 NCOL1=0 : NCOL2=0
4140 FOR I=1 TO NVER
4150   IF I=NVER THEN II=1 ELSE II=I+1
4160   IF I=1 AND ICODE(I)<>4 THEN NCOL1=1
4170   IF I>1 THEN NCOL1=NCOL2
4180   IF ICODE(II)<>4 THEN NCOL2=NCOL1+1 ELSE NCOL2=NCOL1
4190 '
4200   IF GEOMET(I,1)=4 THEN 4260
4210   DX=(GEOMET(I,2)-GEOMET(I,4))^2

```

```

4220   DY=(GEOMET(I,3)-GEOMET(I,5))^2      ? computations of the length
4230   LONG=SQR(DX+DY)                    ? for straight elements
4240   GOTO 4340
4250 ?
4260   LONG=0
4270   FOR K=1 TO NLCURV
4280       IF CARCURV(K,1)=1 THEN NC=K : GOTO 4300      ? computations of
4290   NEXT K                                           ? the length for
4300   FOR J=1 TO 4                                       ? curved elements
4310       LONG=LONG+CARCURV(NC,5*J-1)*WE16AUSS(J)
4320   NEXT J
4330 ?
4340   JAC=LONG/2
4350   IF CARGA(1,1,1)<>0 THEN 4400
4360   IN=(CARGA(1,2,2)+CARGA(1,3,2)-.6666667*CARGA(1,4,2))*JAC
4370   COEF(NECU,NECM1)=COEF(NECU,NECM1)-IN
4380   GOTO 4500
4390 ?
4400   ON ICODE(I) GOTO 4500,4500,4410,4430,4410
4410   COEF(NECU,NCOL1)=COEF(NECU,NCOL1)+JAC
4420   GOTO 4450
4430   COEF(NECU,NECM1)=COEF(NECU,NECM1)-JAC*CARGA(1,2,2)
4440 ?
4450   ON ICODE(I1) GOTO 4500,4460,4500,4480,4460
4460   COEF(NECU,NCOL2)=COEF(NECU,NCOL2)+JAC
4470   GOTO 4500
4480   COEF(NECU,NECM1)=COEF(NECU,NECM1)-JAC*CARGA(1,3,2)
4490 ?
4500 NEXT I
4510 RETURN
6300 ?
6310 ? -----
6320 N(1)=.5*(1-CHI)
6330 N(2)=.5*(1+CHI)
6340 N(3)=.5*(CHI^2-1)
6350 N(4)=CHI*N(3)/3
6360 N(5)=(CHI^4-1)/24
6370 N(6)=(CHI^5-CHI)/120
6380 N(7)=(CHI^6-1)/720
6390 N(8)=(CHI^7-CHI)/5040
6400 N(9)=(CHI^8-1)/(5040*8)
6410 RETURN

```