

A DYNAMIC PROGRAMMING APPROACH TO THE FORMULATION AND SOLUTION OF FINITE ELEMENT EQUATIONS

Nestor DISTEFANO

and

Avelino SAMARTIN

A method for formulating and algorithmically solving the equations of finite element problems is presented. The method starts with a parametric partition of the domain in juxtaposed strips that permits sweeping the whole region by a sequential addition (or removal) of adjacent strips. The solution of the difference equations constructed over that grid proceeds along with the addition removal of strips in a manner resembling the transfer matrix approach, except that different rules of composition that lead to numerically stable algorithms are used for the stiffness matrices of the strips. Dynamic programming and invariant imbedding ideas underlie the construction of such rules of composition. Among other features of interest, the present methodology provides to some extent the analyst's control over the type and quantity of data to be computed. In particular, the one-sweep method presented in Section 9, with no apparent counterpart in standard methods, appears to be very efficient insofar as time and storage is concerned. The paper ends with the presentation of a numerical example.

1. Introduction

The finite element approach, initially proposed by J.H. Argyris in the early fifties [1a], has had an enormous success in many modern engineering applications. It seems apparent however that most of the research effort in this area has been devoted to the construction of the elements themselves [1b] and, more recently, to several related problems in approximation theory [1c]. Less attention seems to have been paid to the development of a methodology for the formulation and solution of finite element equations which can efficiently deal with some of the natural constraints such as the topological properties of the domain, limitations on the rapid access storage, type and quantity of information required from the solution – etc. which are seldom recognized in available finite element procedures. This paper intends to be a step in that direction.

Once the stiffness matrices of the elements have been determined, standard finite element procedures usually rely on more or less empirical rules for the numeration of the nodes, plus a number of efficient numerical methods to solve the resulting band matrix equations. A departure from that trend appears to be the *frontal method* by B. Irons [2], which uses the discretizing mesh as the underlying framework for the construction of algorithmic solutions of the equations. Unfortunately, lack of analytic explicitness of that method makes difficult the study of the solutions with respect to stability and other aspects of interest. Along the same lines, but cast under a broader and deeper

conceptual framework, is the excellent article by E. Schrem [3]. Bearing little analytical relation with the method presented here, although sharing similar motivational principles, is the *tearing method* initially proposed by G. Kron [4] to solve linear engineering networks.

Here we start with a parametric partition of the domain in juxtaposed strips (or super-elements) that permits sweeping the whole region by a sequential addition (or removal) of adjacent strips. The solution of the equations proceeds along with the associated growing process of strip adding in a manner that resembles the transfer matrix approach except that different rules of composition that lead to numerically stable algorithms are used for the stiffness matrices of the strips. Dynamic programming and invariant imbedding underlie the construction of such rules of composition. Previous use of ideas of dynamic programming and invariant imbedding for the solution of elliptic partial differential equations, may be found in [5] and [6] where the difference equations are obtained by application of classical finite difference techniques.

Several advantages stem from the combined application of finite elements and invariant imbedding ideas. We shall discuss this in Section 12. Here we only wish to stress the most remarkable features. Firstly, we note that, in addition to the numerical stability mentioned above, the algorithms derived from this method furnish meaningful information at each step of the computation. This is particularly relevant when the strip configuration reflects a physical attribute of the process to be studied, such as the assemblage or construction stages in a structural mechanics context, etc. Secondly, the ability of the present method to compute only a selected type and amount of information, as illustrated in Section 9, seems to be a feature not shared by existing finite element procedures.

We finally point out that, although the methods presented here are self-contained in whatever dynamic programming and invariant imbedding ideas are concerned, the reader wishing to expand those notions may consult [5], [6], [9].

2. The Poisson equation

We shall introduce the method in the context of a simple, namely the solution of Poisson's equation in a given domain of the plane x, y , subject to appropriate boundary conditions. Clearly, the method is applicable to general n -th order elliptic equations, not necessarily of the self-adjoint type.

Let $u(x, y)$ be a function with piecewise continuous first derivatives in the domain Ω with boundary $\partial\Omega$, subject to prescribed values $u_0(x, y)$ over the portion $\partial\Omega_1 \subset \partial\Omega$ of the boundary, i.e.

$$u(x, y) = u_0(x, y), \quad \text{on } \partial\Omega_1 \quad (2.1)$$

We consider the quadratic functional

$$J(u) = \iint_{\Omega} (u_x^2 + u_y^2 + 2pu) \, dx \, dy + \int_{\partial\Omega_2} (2qu + \alpha u^2) \, ds, \quad (2.2)$$

where u_c stands for $\partial u / \partial c$ ($c = x, y$) p is a function defined on Ω , and q and $\alpha \geq 0$ are functions defined on $\partial\Omega_2 = \partial\Omega - \partial\Omega_1$, i.e. on the remaining portion of the boundary. Under the usual smoothness requirement the function u that satisfies boundary conditions (2.1) and minimizes (2.2), is the (unique) solution of the Poisson equation

$$u_{xx} + u_{yy} = p, \quad (2.3)$$

subject to

$$\frac{\partial u}{\partial n} + q + \alpha u = 0, \quad \text{on } \partial\Omega_2, \quad (2.4)$$

where n refers to the direction normal to the boundary $\partial\Omega_2$.

It is often convenient to replace the constrained minimization problem associated with (2.1) and (2.2) by an unconstrained one. This may be done by incorporating the constraint (2.1) into the functional (2.2) by using a Courant parameter λ [7], i.e. by constructing the functional

$$J(u) = \iint_{\Omega} (u_x^2 + u_y^2 + 2pu) \, dx \, dy + \int_{\partial\Omega_2} (2qu + \alpha u^2) \, ds + \lambda \int_{\partial\Omega_1} (u - u_0)^2 \, ds. \quad (2.5)$$

In general, we expect that $u(x, y, \lambda)$ which minimizes (2.5) will approach the extremal of (2.2) subject to (2.1), as $\lambda \rightarrow \infty$.

3. Finite element discretization

The functional given by (2.5) may be conveniently discretized using finite element techniques. Since procedures of this type are widely known in the literature [1], we shall only present here those steps in the derivation which are particularly relevant to the development of our method.

We consider the domain Ω subdivided in finite elements Ω_l . We introduce a local numeration to denote the K^l nodes $i = 1, 2, \dots, K^l$ of the element K^l . In addition, a global coordinate system will be employed to attach to each node the pair (x_i^l, y_i^l) . In the following, a superindex l denotes the element. We now construct a family of test functions with compact support relative to the finite element partition $\{\Omega_l\}$:

$$u(x, y) = \begin{cases} \sum_{i=1}^{K^l} N_i^l(x, y) u_i^l & \text{if } (x, y) \in \Omega_l + \partial\Omega_l, \\ 0 & \text{otherwise,} \end{cases} \quad (3.1)$$

where u_i^l is the value of $u^l(x, y)$ at the node i of element l , K^l is the number of nodal points in element l , and $N_i^l(x, y)$ are suitable interpolation functions such that

$$N_i^l(x_j^l, y_j^l) = \delta_{ij}, \quad (3.2)$$

where δ_{ij} is the Kronecker delta. Additionally, N_i^l in (3.1) must satisfy appropriate conditions of completeness and integrability.

Introducing the test function $u^l(x, y)$ given by (3.1) in (2.5), we obtain

$$J(u_i^l) = \sum_l E^l, \quad (3.3)$$

where E^l , the contribution of element l to the functional J , is given by

$$E^l = \sum_{i,j=1}^{K^l} u_i^l k_{ij}^l u_j^l + 2 \sum_{i=1}^{K^l} p_i^l u_i^l + \lambda \int_{\partial\Omega_1^l} u_0^2 ds, \quad (3.4)$$

where

$$k_{ij}^l = \iint_{\Omega_l} \left(\frac{\partial N_i^l}{\partial x} \frac{\partial N_j^l}{\partial x} + \frac{\partial N_i^l}{\partial y} \frac{\partial N_j^l}{\partial y} \right) dx dy + \int_{\partial\Omega_1^l} N_i^l \alpha N_j^l ds + \lambda \int_{\partial\Omega_1^l} N_i^l N_j^l ds \quad (3.5)$$

and

$$p_i^l = \iint_{\Omega_l} N_i^l p dx dy + \int_{\partial\Omega_2^l} N_i^l q ds - \lambda \int_{\partial\Omega_1^l} N_i^l u_0 ds. \quad (3.6)$$

In (3.6) the line integrals are defined over $\partial\Omega_1^l$ and $\partial\Omega_2^l$ given by

$$\begin{aligned} \partial\Omega_1^l &= \partial\Omega_l \cap \partial\Omega_1, \\ \partial\Omega_2^l &= \partial\Omega_l \cap \partial\Omega_2, \end{aligned} \quad (3.7)$$

where $\partial\Omega_l$ is the boundary of element l . It is clear that the matrix of the influence coefficients $k_{ij}^l (i, j = 1, 2, \dots, K^l)$ (to be referred to as the *stiffness matrix* of the element) is a symmetric, positive definite matrix.

4. Standard finite element method

Once the stiffness matrices of the elements have been computed, a standard finite element procedure consists of the coupling of the elements by requiring continuity of u at the nodal points. This is usually accomplished by dropping the local numeration of the nodes in favor of a global one. We shall use capital subindexes for the global numeration in order to avoid confusion with the local one. In this fashion, the quantities k_{ij}^l and p_i^l given by (3.5) and (3.6), respectively, are replaced by the corresponding ones in the global numeration system, say k_{IJ}^l and p_I^l . With the introduction of the vector \mathbf{u} of nodal values u_I , i.e.

$$\mathbf{u} = \{u_1 u_2 \dots u_I \dots u_R\}, \quad (4.1)$$

where R is the total number of nodes, eqs. (3.3) and (3.4) may be written as the inner product

$$J(\mathbf{u}) = (\mathbf{u}, \mathbf{K}\mathbf{u}) + 2(\mathbf{p}, \mathbf{u}), \quad (4.2)$$

where the generic element K_{IJ} of matrix \mathbf{K} is given by

$$K_{IJ} = \sum_l k_{IJ}^l, \quad I, J = 1, 2, \dots, R, \quad (4.3)$$

and the I -th component p_I of vector \mathbf{p} is given by

$$p_I = \sum_I p_I^I, \quad I = 1, 2, \dots, R. \quad (4.4)$$

In (4.3) the sum is extended over those elements having nodes I and J in common, and in (4.4) over those elements having node I in common.

In the construction of (4.2) we have implicitly considered the continuity of the function $u(x, y)$ at the nodes. The continuity of u along the common edges of contiguous elements is not however insured by this operation. Such continuity, if required, needs additional restrictions on the interpolation functions. In the example we are presently discussing we require C^0 continuity along the interface of the elements, a condition very easy to satisfy.

Finally, by minimizing (4.2) we obtain

$$Ku + p = o, \quad (4.5)$$

a linear system of equations in the R unknown nodal values u_1, u_2, \dots, u_R .

We recall here that boundary conditions of the type given by (2.1) were incorporated into the variational functional (2.5) by using a Courant parameter λ . Thus, by the choice of a sufficiently large λ , the solution of (4.5) will satisfy the boundary conditions to any prescribed degree of accuracy. In this sense, the Courant parameter technique resembles the trick advocated in [1b] of affecting the diagonal term of the influence matrix by a large number and appropriately modifying the corresponding term in the loading vector in order to take care of boundary conditions in terms of displacements. We could go even further and prove that the procedure of [1b] is a special case of the Courant parameter technique where the "matching" of the boundary conditions is made pointwise on the nodal boundary points only instead of in a continuous fashion along the boundary curve.

5. Imbedding

There are several difficulties associated with the formulation of the finite element equations as outlined in the previous sections. Well known for example is the intimate connection between the structure of matrix K and the order in which the nodes are numbered. The usual rule is to number the nodes such as to obtain the influence matrix with the smallest band-width. In some simple cases this is easy to do, but in general this proposition leads to very involved combinatorial-optimization problems. On the other hand, although band-width is an important and meaningful consideration, it should hardly be the only one for the construction of the finite element equations. Other aspects such as the topological properties of the domain, or bounds on the rapid access storage, or the type and quantity of information required out of the solution, etc. also should be determining factors for the construction of the mathematical model.

It is therefore interesting to develop a unified methodology to deal with all these problems in a consistent fashion. This paper intends to be a step in that direction. The notion of *imbedding* will play the leading role in the method. In fact, our objective will be the introduction of appropriate imbeddings to efficiently deal with a number of the major questions. The way to introduce an imbedding will be by making explicit the intrinsic *difference* nature of the finite element equations. The underlying algorithms to be used for the solution of the resulting equations will be those of dynamic programming and invariant imbedding.

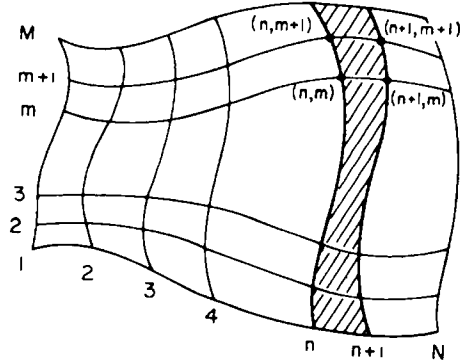


Fig. 1. Decomposition of region in adjacent strips.

It is clear that there is not a unique way to write the finite element equations in difference form. In order to illustrate the methodology, we present here one based on the decomposition of the region in adjacent strips as in the illustrative example presented in fig. 1. In order to introduce the quantities of interest, it is convenient to denote the nodes by the pair (n, m) indicating the column and row, respectively, of the node under consideration. Thus the value of $u(x, y)$ at nodal point (n, m) will be denoted $u_{n,m}$. With this new notation we introduce \mathbf{u}_n , the vector of all the nodal values along the coordinate n , i.e.

$$\mathbf{u}_n = \{u_{n,1} u_{n,2} \dots u_{n,M_n}\}, \quad (5.1)$$

where M_n is the number of nodes along the coordinate n .

A strip will now be formed by all the elements lying between the coordinate lines n and $n + 1$. Hence, the energy stored in the strip $(n, n + 1)$ will depend on the vectors \mathbf{u}_n and \mathbf{u}_{n+1} , and the total energy of the $N - 1$ strips will be given by

$$J(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N) = \sum_{n=1}^{N-1} E_n(\mathbf{u}_n, \mathbf{u}_{n+1}), \quad (5.2)$$

where E_n , the energy of the strip, is given by

$$\begin{aligned} E_n(\mathbf{u}_n, \mathbf{u}_{n+1}) = & (\mathbf{u}_n, \mathbf{K}_{11}(n)\mathbf{u}_n) + (\mathbf{u}_{n+1}, \mathbf{K}_{22}(n)\mathbf{u}_{n+1}) \\ & + 2(\mathbf{u}_n, \mathbf{K}_{12}(n)\mathbf{u}_{n+1}) + 2(\mathbf{u}_n, \mathbf{p}_1(n)) + 2(\mathbf{u}_{n+1}, \mathbf{p}_2(n)), \end{aligned} \quad (5.3)$$

where the submatrices $\mathbf{K}_{ij}(n)$, $i, j = 1, 2$, and the loading vectors $\mathbf{p}_1(n)$ and $\mathbf{p}_2(n)$ of the strip $(n, n + 1)$ are constructed from the local stiffness (influence) matrices of the single elements composing the strip by the usual rules of composition of stiffness matrices as briefly described in Section 4. For example, in the case where the strip is composed of $M - 1$ quadrilateral elements such as indicated in fig. 2, we shall have

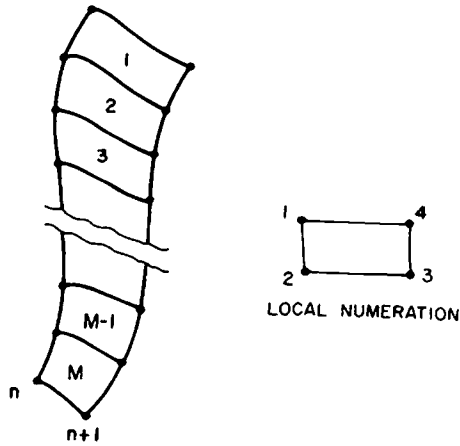


Fig. 2. A strip with quadrilateral elements.

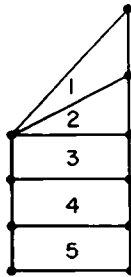


Fig. 3. A strip with quadrilateral and triangular elements.

It is convenient to bear in mind that in eqs. (5.4) to (5.6) a subindex 1 refers to the side n of the strip while 2 is associated with the side $n + 1$.

6. Dynamic programming

To solve the minimization problem associated with (5.2), we shall employ dynamic programming. Introducing

$$f_n(u_n) = \min_{u_i} \sum_{i=n}^{N-1} E_i(u_i, u_{i+1}), \quad (6.1)$$

application of standard arguments in dynamic programming leads to

$$f_n(u_n) = \min_{u_{n+1}} [E_n(u_n, u_{n+1}) + f_{n+1}(u_{n+1})], \quad (6.2)$$

a functional equation for the value function $f_n(u_n)$. The optimality condition is obtained by differentiating in (6.2), namely

$$\frac{\partial E_n}{\partial \mathbf{u}_{n+1}} + \frac{df_{n+1}}{d\mathbf{u}_{n+1}} = 0, \quad (6.3)$$

an expression of Castigliano's theorem. It may be easily shown that $f_n(\mathbf{u}_n)$ is quadratic in \mathbf{u}_n , i.e.

$$f_n(\mathbf{u}_n) = (\mathbf{R}_n \mathbf{u}_n, \mathbf{u}_n) + 2(\mathbf{r}_n, \mathbf{u}_n) + s_n, \quad (6.4)$$

where the matrices \mathbf{R}_n , vectors \mathbf{r}_n and scalar s_n are quantities to be determined. Combining (5.3) and (6.4), the optimality condition (6.3) reduces to

$$\mathbf{K}_{22}(n) \mathbf{u}_{n+1} + \mathbf{K}_{21}(n) \mathbf{u}_n + \mathbf{p}_2(n) + \mathbf{R}_{n+1} \mathbf{u}_{n+1} + \mathbf{r}_{n+1} = \mathbf{o}, \quad (6.5)$$

whence

$$\mathbf{u}_{n+1} = -(\mathbf{R}_{n+1} + \mathbf{K}_{22}(n))^{-1} (\mathbf{K}_{21}(n) \mathbf{u}_n + \mathbf{r}_{n+1} + \mathbf{p}_2(n)). \quad (6.6)$$

Now, substitution of (6.4) and (6.6) in (6.2) yields, after collecting terms in equal powers of \mathbf{u}_n ,

$$\begin{aligned} \mathbf{R}_n &= \mathbf{K}_{11}(n) - \mathbf{K}_{12}(n) \mathbf{A}(n) \mathbf{K}_{21}(n), \\ \mathbf{r}_n &= \mathbf{p}_1(n) - \mathbf{K}_{12}(n) \mathbf{A}(n) \mathbf{p}_2(n) - \mathbf{K}_{12}(n) \mathbf{A}(n) \mathbf{r}_{n+1}, \\ s_n &= -(\mathbf{p}_2(n) + \mathbf{r}_{n+1}, \mathbf{A}(n)(\mathbf{p}_2(n) + \mathbf{r}_{n+1})) + s_{n+1}, \end{aligned} \quad (6.7)$$

where $\mathbf{A}(n)$ is given by

$$\mathbf{A}(n) = (\mathbf{K}_{22}(n) + \mathbf{R}_{n+1})^{-1}. \quad (6.8)$$

In order to determine initial conditions for (6.7) we observe that $n = N$ is a free edge since the boundary conditions have been introduced in the variational formulation. Thus we must pick u_N such as to minimize the value function

$$f_{N-1}(\mathbf{u}_{N-1}) = E_{N-1}(\mathbf{u}_{N-1}, \mathbf{u}_N), \quad (6.9)$$

i.e.

$$\begin{aligned} \mathbf{u}_N &= \arg \min_{\mathbf{u}} E_{N-1}(\mathbf{u}_{N-1}, \mathbf{u}) \\ &= -\mathbf{K}_{22}^{-1}(N-1) \mathbf{K}_{21}(N-1) \mathbf{u}_{N-1} - \mathbf{K}_{22}^{-1}(N-1) \mathbf{p}_2(N-1), \end{aligned} \quad (6.10)$$

Substituting \mathbf{u}_N given by (6.10) in (6.9) and taking into account (5.3) and (6.4), we obtain

$$\begin{aligned}
R_{N-1} &= K_{11}(N-1) - K_{12}(N-1)K_{22}^{-1}(N-1)K_{21}(N-1), \\
r_{N-1} &= p_1(N-1) - K_{12}(N-1)K_{22}^{-1}(N-1)p_2(N-1), \\
s_{N-1} &= -(p_2(N-1), K_{22}^{-1}(N-1)p_2(N-1)),
\end{aligned} \tag{6.11}$$

a set of initial conditions for eqs. (6.7). Comparison of (6.11) and (6.7) shows that (6.11) may be obtained by making

$$\begin{aligned}
R_N &= O, \\
r_N &= O, \\
s_N &= o.
\end{aligned} \tag{6.12}$$

an expected, very convenient set of initial conditions. Once the quantities R_N and r_N have been computed recursively in the backward direction using (6.7) and (6.12), we may compute u_n , $n = 1$ to N , by using eq. (6.6) subject to the initial condition

$$u_1 = \arg \min_{u_1} f_1(u_1) = -R_1^{-1}r_1 \tag{6.13}$$

that follows from the condition of free boundary at $n = 1$. It is interesting to note that instead of (6.13) we may define

$$K_{21}(0) = K_{22}(0) = O, \quad p_2(0) = u_0 = o, \tag{6.14}$$

quantities that substituted in (6.6) yield the initial condition (6.13).

7. Invariant imbedding

In this section we present an alternative treatment of the minimization problem formulated in previous sections that results in a novel solution of the finite element equations. We first derive the Euler equations associated with the minimization of J given by (5.2). To this end we form the first variation of J

$$\delta J = \frac{\partial E_1}{\partial u_1} \delta u_1 + \sum_{n=2}^{N-1} \frac{\partial}{\partial u_n} (E_{n-1} + E_n) \delta u_n + \frac{\partial E_{N-1}}{\partial u_N} \delta u_N, \tag{7.1}$$

which immediately leads to the (equilibrium) equations

$$K_{21}(n-1)u_{n-1} + (K_{22}(n-1) + K_{11}(n))u_n + K_{12}(n)u_{n+1} = -p_1(n) - p_2(n-1). \tag{7.2}$$

The natural boundary conditions associated with (7.2) are given by

$$\frac{\partial E_1}{\partial u_1} = K_{11}(1)u_1 + K_{12}(1)u_2 + p_1(1) = 0 \quad (7.3)$$

and

$$\frac{\partial E_{N-1}}{\partial u_N} = K_{21}(N-1)u_{N-1} + K_{22}(N)u_N + p_2(N-1) = 0 \quad (7.4)$$

To solve this two point boundary value problem, we employ invariant imbedding ideas and seek for solutions of the form

$$u_{n+1} = Q_n u_n + q_n . \quad (7.5)$$

Eliminating u_n and u_{n+1} in (7.2) by using (7.5), we obtain the following recurrence relations for matrix Q_n and vector q_n ,

$$Q_{n-1} = -(K_{22}(n-1) + K_{11}(n) + K_{12}(n)Q_n)^{-1} K_{21}(n-1) , \quad (7.6)$$

$$q_{n-1} = -(K_{22}(n-1) + K_{11}(n) + K_{12}(n)Q_n)^{-1} (K_{12}(n)q_n + p_1(n) + p_2(n-1)) , \quad (7.7)$$

subject to the initial conditions

$$Q_{N-1} = -K_{22}^{-1}(N) K_{21}(N-1) , \quad (7.8)$$

$$q_{N-1} = -K_{22}^{-1}(N) p_2(N-1) , \quad (7.9)$$

respectively, obtained from substitution of (7.5) in (7.4). Once the quantities Q_n and q_n have been recursively computed using (7.6) to (7.9), we may compute u_n using (7.5) subject to the initial condition

$$u_1 = -(K_{11}(1) + K_{12}(1)Q_1)^{-1} (K_{12}(1)q_1 + p_1(1)) , \quad (7.10)$$

derived from substitution of u_2 given by (7.5) in (7.3).

8. Connection with dynamic programming

Clearly, the two methods presented in Sections 6 and 7 are equivalent. In fact, it is easy to derive the relationship between the corresponding quantities, namely

$$R_n = K_{11}(n) + K_{12}(n)Q_n , \quad (8.1)$$

$$r_n = p_1(n) + K_{12}(n)q_n .$$

9. One-sweep method

The method presented in Section 7 requires the storage of the quantities Q_n and q_n and two sweeps to solve the problem. In this sense it resembles the upper triangularization and back substitution operations used in the standard finite element method when direct procedures are employed to solve the equations. Further, we may easily prove the equivalence of the invariant imbedding procedure described in Section 7 and the algorithm resulting from application of the Gauss elimination method to eqs. (7.2), (7.3) and (7.4).

In this section we present a method that requires much less storage and only one computational sweep. In addition, it will permit the selective computation of quantities of interest at specified locations only. This, a feature not shared by any of the available standard methods, leads to additional computational efficiency. Again, the underlying ideas in the development of the method will be those of invariant imbedding.

Suppose that our aim is to determine the nodal values of u along a line i . By superposition we can write

$$u_i = S(i, n) u_n + s(i, n) , \quad (9.1)$$

i.e. there exists a linear relationship between the displacements along two arbitrary lines i and n . We assume now that $n \leq i$ and, while keeping i constant, we consider the domain of length $N - (n + 1)$, i.e.

$$u_i = S(i, n - 1) u_{n-1} + s(i, n - 1) . \quad (9.2)$$

Substituting $u_n = Q_{n-1} u_{n-1} + q_{n-1}$ in (9.1), subtracting (9.1) from (9.2), and collecting terms in u_{n-1} , we find

$$S(i, n - 1) = S(i, n) Q_{n-1} , \quad (9.3)$$

$$s(i, n - 1) = s(i, n) + S(i, n) q_{n-1} , \quad (9.4)$$

subject to the initial conditions

$$S(i, i) = I(M_i) , \quad (9.5)$$

$$s(i, i) = 0 , \quad (9.6)$$

obtained from comparison of both sides of (9.1) when $n = i$. In (9.5) $I(M_i)$ denotes the identity matrix of order M_i . It is clear that the quantities $S(i, n)$ and $s(i, n)$ are not defined for $i < n$. The computational process consists in the recursive, backward computation of Q_n and q_n using (7.6) to (7.9). At $n = i$ we adjoin eqs. (9.3) to (9.6) and continue the recursive computation of matrices Q_n , $S(i, n)$ and vectors q_n and $s(i, n)$. We finally obtain u_i by employing (9.1) at $n = 1$, i.e.

$$u_i = S(i, 1) u_1 + s(i, 1), \quad (9.7)$$

where u_1 is given by (7.10).

We finally note that one sweep methods were first introduced in connection to the integration of ordinary differential equations subject to two-point value conditions.

10. Example of imbeddings

In the previous sections we used dynamic programming and invariant imbedding ideas to construct algorithmic solutions of the finite element equations conveniently written in difference form. We shall now illustrate the method by presenting some examples of possible configurations. This is shown in fig. 4a to 4f, where the various imbedding strips are indicated with heavy solid lines. In 4a and 4c the domain was discretized such as to have strips with the same dimensionality. In 4b and 4d the strips are of different dimensionality. Figs. 4e and 4f are examples of radial and annular imbeddings, respectively.

The six examples presented in fig. 4 are of course illustrative, not exhaustive. Many alternative imbedding configurations are possible for any given domain. It is interesting to note that the

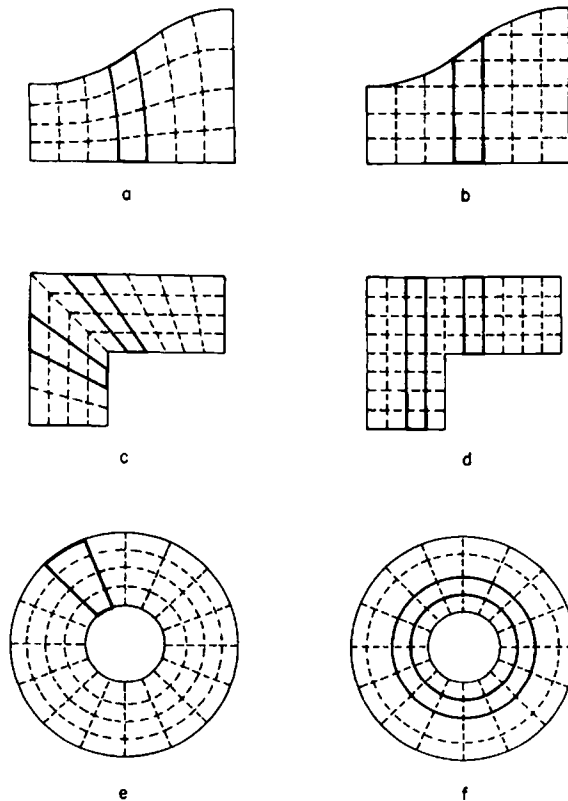


Fig. 4. Imbedding configurations.

method presented in previous sections handles in a uniform fashion any of the cases presented here or any other constructed along similar lines. In particular, there is no operational distinction between constant and variable strip dimensionality. In this respect the present method by-passes some of the difficulties appearing in the solution of problems with irregular boundaries when conventional finite differences are employed for the discretization of the equations [5]. Additional advantages will be listed at the end.

11. Numerical example

As an example involving a higher dimensional equation, we consider a slab under a plane state of stresses with geometry and boundary conditions as indicated in fig. 5, subject to uniform body forces $X = 1$ and $Y = 1$. The functional to be minimized is the potential energy of the slab in terms of displacements

$$J(u, v) = G \iint_{\Omega} \left[\frac{1}{1-\nu} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)^2 - 2 \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{1}{2} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 \right] dx dy + \iint_{\Omega} (Xu + Yv) dx dy, \quad (11.1)$$

where u and v are the displacements in the x and y directions, respectively, subject to appropriate conditions along the clamped and free edges of the plate. The elastic constants were given values $G = 500$ and $\nu = 0$. The partition in finite elements is that indicated in the figure. The elements considered are isoparametric with quadratic shape functions varying linearly along the edges. The stiffness matrices were determined using the SAP routine [11], and the resulting equations, as-

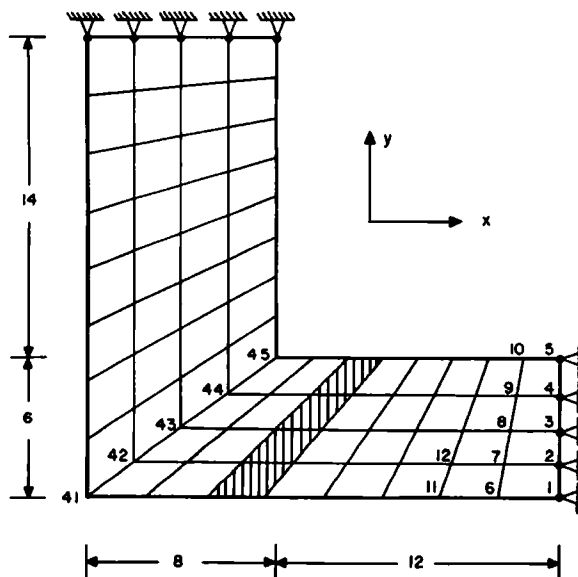


Fig. 5. A slab under a plane state of stress.

Table 1
Nodel displacements

Node	u_i		v_i	
41	2.749	E-01	2.293	E-01
42	2.696	E-01	2.315	E-01
43	2.620	E-01	2.331	E-01
44	2.542	E-01	2.334	E-01
45	2.494	E-01	2.299	E-01

sembled using the nodal numeration indicated in fig. 5, were solved by Gauss elimination in the classical two-sweep process. The nodal values of u and v at nodes 41 to 45 obtained by this method were compared with those obtained in a one-sweep method using the algorithm described in Section 9. The shadowed area in fig. 5 indicates a generic strip employed in the one-sweep method. As expected, the results using both methods were found identical. The numerical results are given in table 1.

12. Discussion

The imbedding strip method presented in this paper, combining finite element and dynamic programming ideas, accentuates the advantages of both approaches. In fact, to the versatility of the finite element method that handles in a uniform manner arbitrary geometries, general boundary conditions etc. we must add the nonetheless important advantages derived from the invariant imbedding approach. We summarize here some of them.

In the first place we observe that each step in the numerical calculations has a clear physical meaning since by adding strips we are solving a family of problems in a sequential fashion. Hence, by properly choosing the imbedding strip configuration, we can naturally model a number of growing processes such as those associated with construction sequences etc. at no additional cost in the computational effort. Secondly, the numerical processes associated with dynamic programming are stable [9], a desirable property when dealing with high dimensional problems. Furthermore, the Riccati-like algorithms employed in this paper may be shown to be exponentially stable, a property stemming from the positive definite nature of stiffness matrices and the intrinsic stability of the Riccati equation [10].

Finally we note that the one-sweep method presented in Section 9, with no counterpart in the standard finite element methodology, permits to some extent the analyst's control over the type and amount of data to be computed. Procedures of this type, in addition to offering advantages derived from reduced storage requirements, might prove of interest in connection with modern parallel computational approaches.

Acknowledgements

The authors wish to thank Professor John H. Argyris for encouragement and valuable comments for the preparation of the manuscript.

References

- [1a] J.H. Argyris, Energy theorems and structural analysis, Aircraft Engineering 1954/55, also in book form (Butterworth's Scientific Publications, London, 1960) rth reprint (1973).
- [1b] O.C. Zienkiewicz, The finite element method in engineering science (McGraw Hill, 1971).
- [1c] A.K. Aziz (ed.), The mathematical foundations of the finite element method with applications to partial differential equations (Academic Press, 1972).
- [2] B. Irons, A frontal solution program for finite element analysis, Int. J. Num. Meth. Eng. 2 (1970) pp. 5–32.
- [3] E. Schrem, Computer implementation of the finite element procedure (Paper presented at the ONR Symposium, University Illinois, Urbana, September 8–10, 1971), in: S.J. Fenves et al., (eds.) Numerical and Computer Methods in Structural Mechanics (Academic Press, 1973).
- [4] H.H. Happ, The theory of network diakoptics (Academic Press, 1971).
- [5] E. Angel and R. Bellman, Dynamic programming and partial differential equation (Academic Press, 1972).
- [6] N. Distefano, Nonlinear processes in engineering (Academic Press, 1974).
- [7] T. Butler and A.V. Martin, On a method of Courant for minimizing functionals, J. Math. and Physics 41 (1962) pp. 291–299.
- [8] R. Bellman, H. Kagiwada and R. Kalaba, Invariant Imbedding and the Numerical Integration of Boundary-Value Problems for Unstable Linear Systems of Ordinary Differential Equations, Comm. A.C.M. 10 (1967) pp. 100–102.
- [9] R. Bellman, Dynamic programming (Princeton University Press, Princeton, New Jersey, 1957).
- [10] D. Rappaport and L.M. Silverman, Structure and Stability of Discrete-Time Optimal Systems, IEEE Transactions on Automatic Control, AC-16, No. 3 (June 1971) pp. 227–233.
- [11] E. Wilson, SOLID SAP – A Static Analysis Program for Three Dimensional Solid Structures, UC SESM 71–19, Structural Engineering Laboratory, University of California, Berkeley, 1971.