Numerical experiments in finite element grid optimization by direct energy search

Carlos. A. Felippa

Structures Laboratory, Lockheed Research Laboratory, Palo Alto, California 94304, USA
(Received 30 September 1976)

A series of numerical experiments is conducted to assess the feasibility and practical value of finite element grid optimization based on direct minimization of the total potential energy of the discrete model with respect to the node locations. An implementation relying upon non-linear programming techniques is found to be numerically reliable and to lead to improved grids in accord with engineering intuition. This rigorous approach is hampered, however, by the excessive computational effort required by the energy minimization process. A combination of related techniques is therefore proposed to make dynamic node distribution a useful tool within the framework of large-scale finite element analysis. The combined strategy involves use of substructuring methods, application of a local 'energy-balancing' optimality criterion for fast node distribution, and automatic refinement of previously-improved coarse grids.

Introduction

This paper is a sequel to a previous article,[1] in which general principles for problem-controlled optimization of finite element grids were formulated. The nomenclature used in that publication will be followed throughout.

The practical realization of automated grid improvement rests primarily on two decisions: the selection of a grid optimization principle, and the choice of an algorithm for implementing the resulting node-distribution process. As a first step in the study of the subject, the total-energy minimization criterion was selected as the basis for optimizing energy-bounding finite element grids. A non-linear programming approach was then implemented to minimize the discrete energy functional E_v^* defined by equation (15) of reference 1. Particular attention was paid to the method-selection guidelines discussed in the last two sections of that paper.

Minimization algorithms

Three minimization algorithms were tried in the course of this study; the conjugate-gradient method of Fletcher and Reeves[2] with numerical estimation of derivatives of the total-energy function E_v^*. Secondly the derivative-free method of Rosenbrock,[3] closely following the implementation reported in Kuester and Mize[4]. And third the derivative-free method of Powell[5]. The implementation of this method follows the code presented in Himmelblau[6] along general lines, but with substantial problem-related enhancements.

The Fletcher–Reeves algorithm was afflicted by repeated premature terminations caused by the sensitivity of the computed gradient to the selection of numerical differentiation points. Rosenbrock's method proved to be intolerably slow if the number of optimization variables (N_X) exceeded 5. On the other hand, Powell's method performed reliably on all test problems, and was the least affected by dimensionality

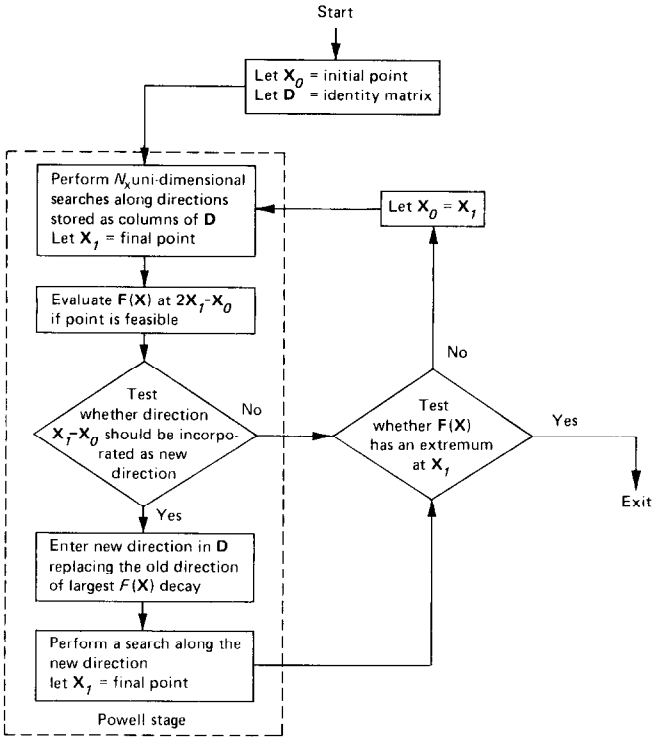

Figure 1 Basic steps of Powell's function minimization method

effects, which corroborates the findings of Parkinson and Hutchinson[7]. On the basis of its superior reliability, Powell's method was therefore selected as the standard energy-minimization algorithm for the test problems reported in the sequel.

Powell's method

Given an initial feasible point $\boldsymbol{X}_0$, Powell's method[5] attempts to locate a minimum of an N_X-dimensional object function $F(\boldsymbol{X})$ by performing successive uni-dimensional searches along conjugate directions generated by the procedure. A schematic flow chart of the method is shown in *Figure 1*. Computer implementation details are discussed by Himmelblau[6]. Only an informal outline is given here, which will be useful in the discussion of numerical experiments.

The method maintains a direction matrix $\boldsymbol{D}$ of order N_X. The columns of $\boldsymbol{D}$ represent normalized search directions in $\boldsymbol{X}$-space. Starting from the initial point $\boldsymbol{X}_0$, $\boldsymbol{D}$ is set to the identity matrix $\boldsymbol{I}$, i.e., the initial search directions are parallel to the X_j axes.

The object function $F(\boldsymbol{X})$ is numerically minimized along each of the search directions in $\boldsymbol{D}$. The value of $\boldsymbol{X}$ on exit from the previous search becomes the initial point for the next one. If a search penetrates an infeasible region[1], the last feasible point is returned. After all directions in $\boldsymbol{D}$ have been processed, the method investigates whether a new search direction obtained by joining the starting point $\boldsymbol{X}_0$ and the final point $\boldsymbol{X}_1$ should be incorporated in $\boldsymbol{D}$. If so, another uni-dimensional search is carried out along the new direction, which is adjoined to $\boldsymbol{D}$, and $\boldsymbol{X}_1$ is updated. The full sweep across $\boldsymbol{D}$ will be called a 'Powell stage'.

Numerical tests for acceptance of an extremum of $F(\boldsymbol{X})$ at the last point $\boldsymbol{X}_1$ are then performed. If an extremum is not detected, another stage is initiated using $\boldsymbol{X}_1$ as new starting point $\boldsymbol{X}_0$. The process continues until a numerical extremum is accepted, or some run-termination condition is met.

Test problems

Several one- and two-dimensional boundary value problems which arise in structural mechanics were used to acquire computational experience with the automated grid improvement technique.

The one-dimensional sample problems involved the lateral deflection of non-uniform beams under various types of loading and support conditions. The beams were modelled by the usual Hermitian finite elements. These problems were primarily used for debugging purposes and for 'tuning up' the energy-function minimizer. The results were not particularly interesting, because the energy improvement is very small on account of the high accuracy with which smooth solutions of one-dimensional problems are approximated by cubic polynomials. Consequently, the one-dimensional examples will be omitted for brevity.

The two-dimensional test problems displayed more interesting behaviour. The examples presented in the sequel involve the linear plane-stress analysis of thin plates loaded in their own plane. The potential energy functional for such problems has the form:

$$E(\boldsymbol{u}) = \frac{1}{2}\int_D \boldsymbol{\varepsilon}^T \boldsymbol{C}\,\boldsymbol{\varepsilon}\, h\,\mathrm{d}A - \int_B \boldsymbol{p}^T\boldsymbol{u}\,\mathrm{d}s \qquad (1)$$

where $\boldsymbol{u}$ is a two-vector of displacement components (u_x, u_y); $\boldsymbol{\varepsilon}$ is a 3-vector of infinitesimal mechanical strains associated with $\boldsymbol{u}$; $\boldsymbol{C}$ is a 3×3 material-property matrix; $\boldsymbol{p}$ is a 2-vector of in-plane traction forces (p_x, p_y) which acts on the plate boundary B; h is the plate thickness, and $\mathrm{d}A$, $\mathrm{d}s$ are elements of plate mid-surface area and boundary arc length, respectively. The energy functional equation (1) contains only first derivatives of the state vector $\boldsymbol{u}$; consequently, essential boundary conditions involve only prescribed values of $\boldsymbol{u}$ on B.

The test plane-stress problems were modelled by energy-bounding quadrilateral finite elements. Each quadrilateral is assembled by four triangles, defined by joining the four quadrilateral corners to an internal point located at the arithmetic mean of the corner locations. The variation of the displacement components over each sub-triangle is linear.

Cantilever beam

The first two-dimensional example, defined in *Figure 2a*, involves a uniform cantilever beam of thin rectangular section and aspect ratio (span: height) of 4. The material is linearly elastic and isotropic, Young's modulus, $E = 100$, and Poisson's ratio $\nu = 0.0$. Two load conditions are considered: (S) end shear, and (U) uniform lateral load.

The two regular grids shown in *Figure 2b* were used as initial meshes for the grid improvement process, the results of which are displayed in *Figures 3–6*.

While optimizing the coarser (3×2) mesh, it was noticed that the transverse motion (y motion) of the nodes had a negligible effect on the numerical results, affecting at most the fourth decimal place of E_r^*. It was therefore decided to permit the variable nodes to move only along the longitudinal direction (x axis).

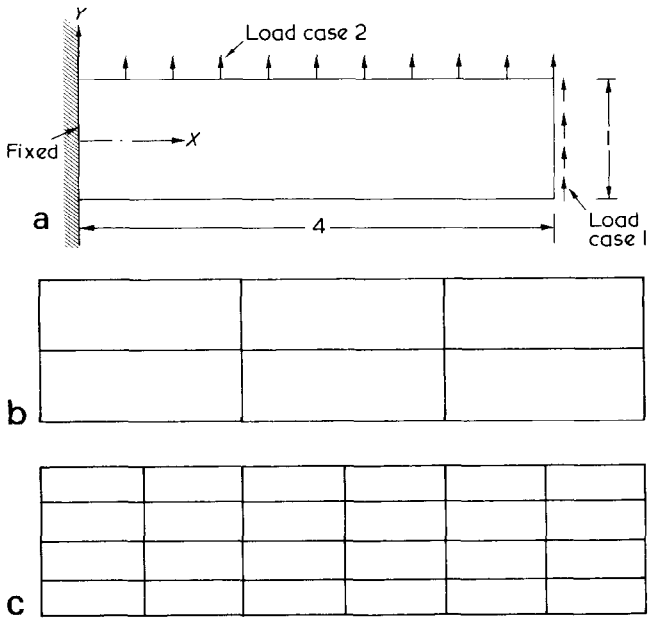

Figure 2 Cantilever beam test problem. (a), problem definition; material data: $E = 100$; $\nu = 0$. Initial grids: (b), 3 × 2 mesh; (c), 6 × 4 mesh

These kinematic constraints have the beneficial effect of substantially reducing the number of optimization variables (N_X is cut from 10 to 6 in the 3 × 2 mesh, and from 41 to 20 in the 6 × 4 mesh).

As might be expected, nodes tend to 'cluster' near the cantilever root, which is the region of highest energy density, Perhaps the most significant result is the impressive improvement in the total energy obtained with the coarse grid (32% and 44% for load cases (*S*) and (*U*), respectively). Such energy variations directly translate into similar improvements in the average magnitude of the transverse deflections and of the computed stress field.

Figures 3 and *5*, which show the variation of E_v^* as a function of the number of energy evaluations, illustrate some of the performance characteristics of Powell's minimization algorithm. The jerky variation of E_v^* near the end of the Powell stages, and the relatively fast decay of E_v^* in the case of the coarser grid, as opposed to the much-slower progress in the finer grid are particularly noticeable. In the former, the

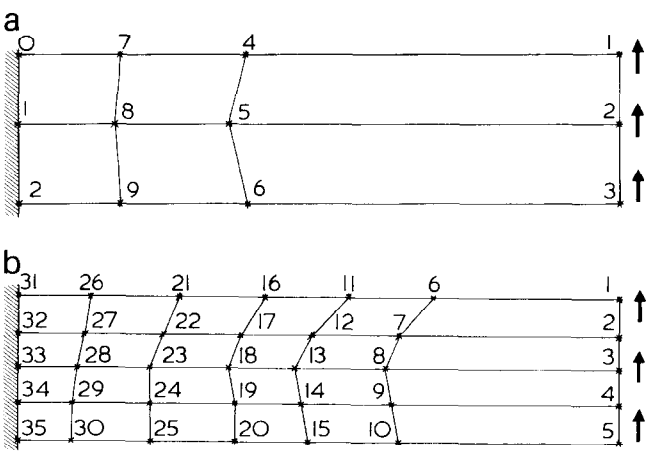

Figure 3 Energy-improved grids for cantilever beam under end shear load.

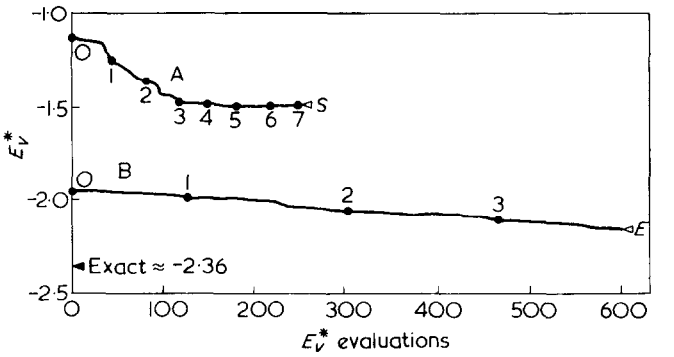

Figure 4 Energy reduction history in cantilever beam under end shear load. Key: ●, Powell stage number; ◁*S*, satisfied minimum acceptance criteria; ◁*E*, exceeded minimum evaluations; *N*, number of optimization variables; *I*, improvement in E_v^*, A, 3 × 2 mesh ($N = 6$, $I = 32\%$); B, 6 × 4 mesh, ($N = 20$, $I = 8\%$)

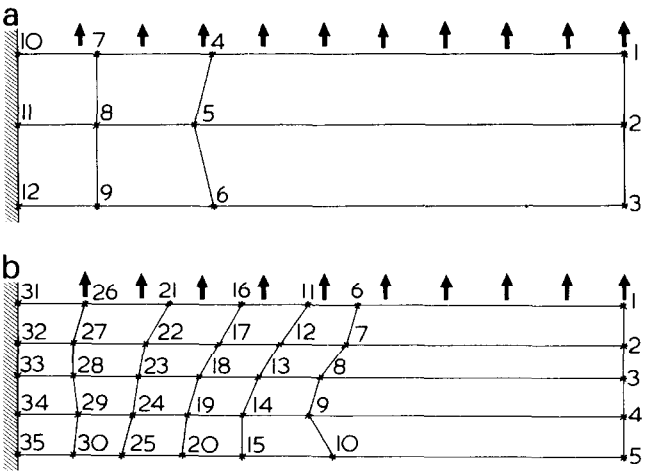

Figure 5 Energy-improved grids for cantilever beam under uniform lateral load.

energy minimization process was terminated at 251 and 289 energy evaluations for load cases (*S*) and (*U*), respectively, after all numerical tolerances for acceptance of a local minimum were met. On the other hand, no minimum was detected in the finer grid runs before the maximum number of energy evaluations (set to 600) was exceeded. The approximate CPU times required per energy evaluation on the UNIVAC 1108

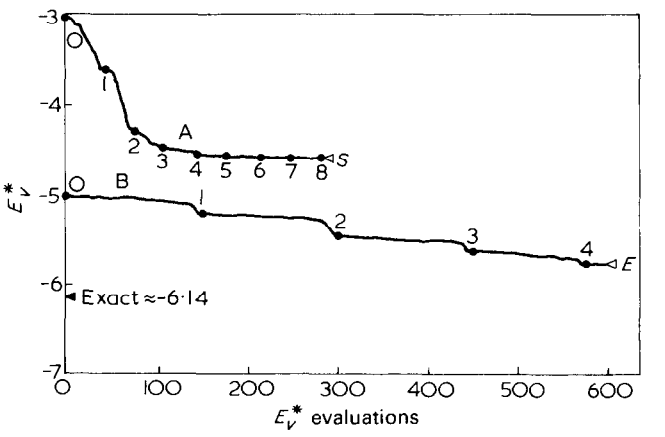

Figure 6 Energy reduction history in cantilever beam under uniform lateral load. (See *Figure 4* for key). A, 3 × 2 mesh ($N = 6$, $I = 44\%$). B, 6 × 4 mesh ($N = 20$, $I = 14\%$)

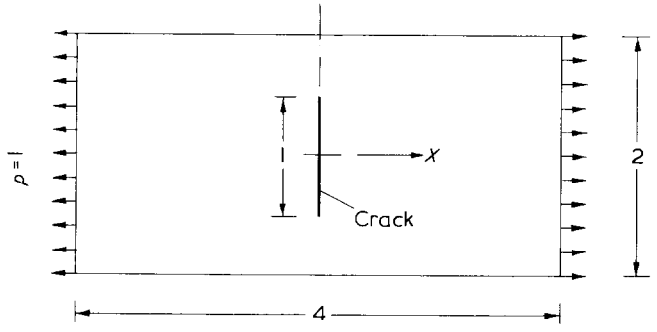

Figure 7 Cracked plate test problem. Material data: $E = 100$, $\nu = 0.25$

computer were 52 and 216 msec for the 3×2 and 6×4 grids, respectively.

Cracked plate in tension field

The second example is defined in *Figure 7*. A thin, rectangular plate is subjected to uniform x-tractions. The plate contains a hair-line crack extending across one-half of the plate x-width. Symmetry conditions allow consideration of one-quarter ($x \geqslant 0$, $y \geqslant 0$) of the plate.

A 4×4 regular mesh is generated over the first quadrant as initial input to the grid optimizer. The node motion constraints and the improved grid obtained after 500 energy evaluations are displayed in *Figure 8*. The number of optimization variables is 26. Each energy evaluation required 148 msec on a UNIVAC 1108 computer. *Figure 9* shows the initial and improved grid layouts over the entire plate.

In this problem the region of high energy density occurs in the vicinity of the crack tip, toward which the nodes tend to 'drift'. *Figure 10* shows the variations in the total energy E_v^* plotted against number of energy evaluations. The rapid decay of E_v^* near the end of the first Powell stage is caused by the motion of the two sliding nodes located on the y-axis (which were numbered last) towards the crack tip. The improvement in E_v^* is rather small as compared to the cantilever beam example. This is not surprising, since most of the potential energy is associated with the almost-uniform extension of the plate in the x-direction, and the magnitude of the edge displacements is only slightly affected by the presence of the crack. A similar energy–decay pattern can be expected from most of the problems which involve stress concentrations that do not appreciably affect the displacements of the applied loads. The improvement in stress definition near the crack tip is, however, significant.

Computational effort assessment

The running time of the grid optimization program is largely controlled by the number of object-function calls, i.e., evaluations of the energy functional E_v^*. (The overhead spent in the optimization driver is negligible for most problems.) Each energy evaluation corresponds to a full linear analysis of the discrete model with a different coefficient matrix $\boldsymbol{A}$ (see equation (5) of reference 1). When Powell's method is used, the number of energy evaluations N_E required to attain a satisfactory grid can be estimated to be:

$$N_E \cong N_S N_E^S \cong N_S N_E^U (N_X + 1) \tag{2}$$

where N_S = number of Powell stages required; N_E^S = average number of energy evaluations per stage; N_E^U = average number of energy evaluations per uni-dimensional search, and N_X = number of optimization variables.

The numerical experiments indicate that $N_E^U \approx 6$ whereas N_S varies from 3 to 7. Using an average value $N_S \cong 5$ and assuming that $N_X \cong N_X + 1$, Equation (2) gives:

$$N_E \approx 30 N_X \tag{3}$$

In finite element grids commonly utilized for production-level analysis of complex engineering systems, N_X would typically be in the range 10^3 to 10^4. The estimated N_E would then fall in the range 30 000 to 300 000, which is clearly prohibitive. In finite element analysis, the cost per energy evaluation generally grows as N_X^m, where the exponent m varies from 2 to 3 dependent on whether the assembly of $\boldsymbol{A}$ or the solution of the linear system dominates the computational effort, respectively. The total grid optimization cost, C, can therefore be expected to increase as:

$$C \propto N_X^3 \text{ to } N_X^4 \tag{4}$$

Reduction of computational effort

Evidently the chief obstacle in the exploitation of the direct energy minimization method as a practical node placement scheme is the substantial computational effort associated with the non-linear programming approach. The implementation of more refined non-linear programming algorithms is unlikely to affect the computational cost other than marginally. Instead, a reduction to acceptable levels will most likely require a concerted plan of attack which involves a combination of the following techniques:

Substructuring: the node distribution process is applied only within design-critical regions identified by the analyst.

Fast node distribution: a scheme to displace nodes to high-energy density regions based on a 'local'—rather than global—optimality criterion.

Automatic grid refinement: an optimized 'coarse' grid is refined by suitable injection of new nodes in regions which exhibit high energy gradients.

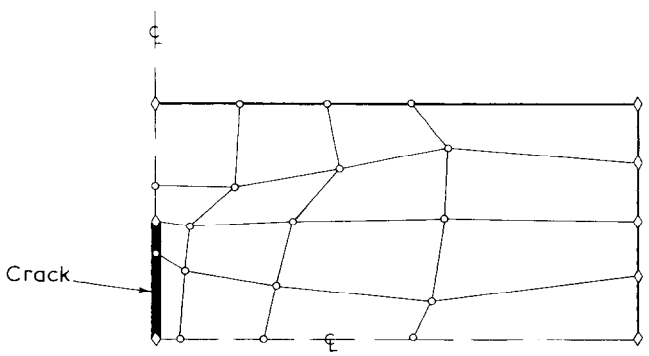

Figure 8 Node motion constraints in cracked plate problem. ◇, fixed nodes; ○, moving nodes

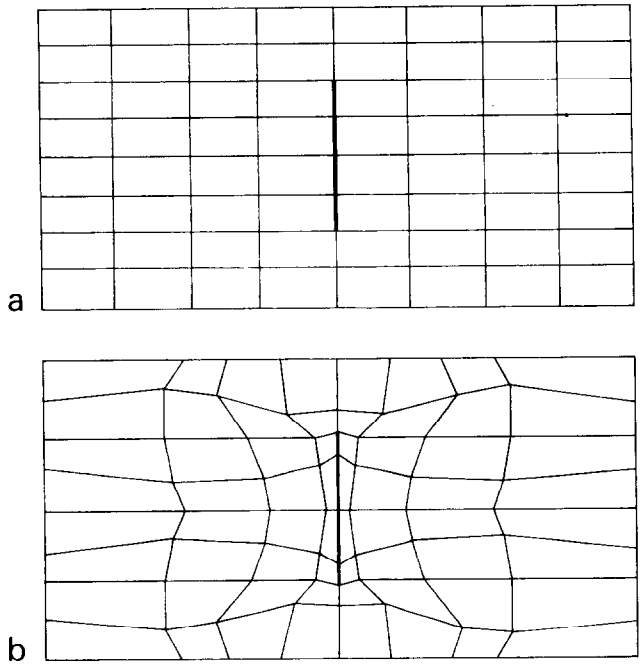

Figure 9 Results of grid optimization process in cracked plate problem. (a), initial grid; (b), improved grid

Substructuring

The design of a complex three-dimensional structural system is generally controlled by the behaviour of *critical* components, which represent the weakest links as regards some failure criterion. The wing–fuselage intersection of an aircraft would be a typical example. These components can sometimes be identified *a priori* by experienced analysts, but most often emerge during the course of the analysis–design cycle.

Using the system-dissection approach known as substructure or 'superelement' partitioning[8], critical regions can be embedded within subdomains $D_1, D_2, \ldots D_k$ called substructures, which contain an integral number of finite elements. Outside of these regions, the setting up of the finite element model is relatively unimportant and can be routinely handled by experienced analysts.

The key idea is to apply the automated grid refinement scheme only to the critical substructures D_j. Nodes located on the boundary of D_j will usually be considered fixed for compatibility with the grid layout external to D_j. Typically the initial grid over each substructure will follow regular patterns such as the example grids, since regular grids are easier to generate.

An initial analysis of the complete structure is performed. This analysis supplies stress–displacement boundary conditions over the D_j interfaces. Substructure grids are then improved following a fast node-distribution procedure, such as the one proposed below. If several load cases are active, a 'compromise' grid will have to be selected.

The attractiveness of the substructuring approach—as opposed to attempting to optimize the grid over the entire structure—lies in the substantial savings achieved in computational effort (cf. equation 4) without sacrificing the quality of design-relevant information. Furthermore, this approach is readily integrated with a computer-aided interactive design system, in which sections of the design data base are periodically examined and updated by the analysts. Note that the computational savings would be even more impressive if it turned out that locally optimized grid layouts could be reused throughout the model; for example, the cracked plate of *Figure 7* might represent many identical substructures.

Energy-balancing node distribution

The application of a rigorous optimality criterion based on the minimization of the total discrete energy E_v^* requires the performance of many complete structural analyses over the spatial domain D of interest. The same would be true of any grid-optimality condition that involves a discretization error integral over D, such as Equation (6) of reference 1. An order-of-magnitude speed-up of the node distribution process can only be attained by use of a less restrictive optimality criterion based on a *local* (element-level) condition rather than on a global one.

A node-distribution scheme presently under consideration is based on the use of the internal energy–density function $d_I(u)$ given by equation (9) of reference 1, as a *nodal density function*[9] in the following sense: each finite element which pertains to a substructure domain D is to store the same amount of energy. The term 'energy-balancing' is therefore an appropriate method identifier. The resultant distribution algorithm consists of three basic stages.

1 *Initially energy density calculation*: the energy density function $d_I(\boldsymbol{X}_0)$ is evaluated at the node location vector $\boldsymbol{X}_0$ of the initial grid from a preliminary analysis of the discrete model.

2 *Node sweep*: a loop is performed over all movable nodes. For each node K the following sequence of calculations is carried out: compute the total

(a) energy absorbed by elements that surround node K (a numerical quadrature scheme would normally be employed to that effect);

(b) compute the energy unbalance U^K at K resulting from adjacent elements, and the partial derivatives U_x^K, U_y^K, and U_z^K pertaining to virtual motions of the node K along the global coordinate directions x, y, and z (all other nodes being assumed fixed);

(c) displace node K to a new location that approximately solves the local energy-balancing equation $U^K = 0$ by applying one or two Newton-iteration cycles. Partial constraints on the motion

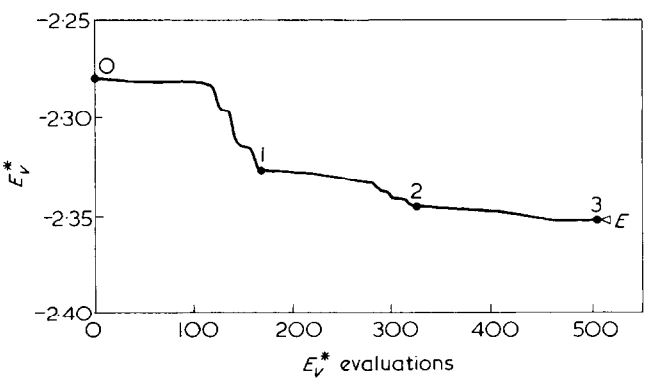

Figure 10 Energy reduction history in cracked plate problem (see *Figure 4* for key)

(e.g., node K must remain on a surface or curve) can easily be taken into account;

(d) recompute the energy density $d_I(x_K, y_K, z_K)$ at the new location of node K by appropriate interpolation.

Energy density recalculation: repeated node sweeps will displace nodes toward regions of higher energy density. After a number N_{SW} of node 2 sweeps have been performed, the energy–density function which corresponds to the new grid should be recalculated by solving the boundary value problem over the substructure. (N_{SW} could be automatically adjusted by monitoring the sum of the distances travelled by the nodes since the last complete analysis.) The second and third steps are repeated as necessary.

If the actual boundary value problem is non-linear or time-dependent, it is expected that optimized grids can be reused through many incremental solution cycles as long as solution patterns remain sensibly constant over the pertinent substructures.

Some algorithmic similarities between the energy-balancing node-distribution process and structural optimization techniques based on strain energy distribution[10] are worth mentioning.

Automated grid refinement

Finite element grids optimized by node distribution retain the number of degrees of freedom and the topological configuration of the initial grid. It often happens that the analyst, after looking at the results of the discrete analysis, wishes to refine the grid by the insertion of new nodes. For example, the analyst may decide that the four quadrilateral elements adjacent to the crack tips in *Figure 9* ought to be further subdivided so as to capture the stress singularity better. This type of 'node injection' may be amenable to automatic treatment according to the following element-splitting technique:

1 The user specifies an energy-density gradient threshold (variation of internal energy density over an element span) that will trigger node injection over a given substructure region.

2 The program examines each finite element which pertains to the last optimized grid. If the energy–density threshold is exceeded over an element, that element is subdivided into an appropriate number of subelements (e.g., a quadrilateral is partitioned into four quadrilaterals, a hexahedron into eight hexahedra, etc.), and nodes placed at the new corner positions. The partition is carried out in such a way that the total energy of the original element is approximately apportioned in equal amounts to the resulting subelements.

3 The node-augmented grid may be resubmitted to the grid optimizer for a few improvement cycles if so desired.

Conclusions

Numerical experiments indicate that node distribution by direct minimization of the total potential energy is a technically feasible procedure for optimizing energy-bounding finite element grids. The chief advantage of this technique is that estimates of the local discretization error are not required. The main disadvantage of the realization as a non-linear programming problem is the significant amount of computational effort involved, which can be expected to grow as the third to fourth power of the number of degrees of freedom of the discrete model. This rapid growth restricts the practical application range of direct total-energy minimization to one-dimensional and coarse two-dimensional grids.

The principal factors to be considered in subsequent developments of dynamic grid optimization are: the implementation and assessment of fast node distribution schemes based on local energy-balancing, and the integration of grid-optimization software with model partitioning (substructuring) processors designed to operate in an interactive mode.

Acknowledgment

This research has been sponsored by the Air Force Office of Scientific Research/AFSC, United States Air Force, under Contract No. F44620-71-C-0109.

References

1 Felippa, C. A. *Appl. Math. Modelling* 1976, **1,** 93
2 Fletcher, R. and Reeves, C. M. *Computer J.* 1964, **7,** 149
3 Rosenbrock, H. H. *Comput. J.* 1960, **3,** 175
4 Kuester, J. L. and Mize, J. H. 'Optimization Techniques with Fortran', McGraw-Hill, New York, 1973
5 Powell, M. J. D. *Comput. J.* 1964, **7,** 155
6 Himmelblau, D. M. 'Applied nonlinear programming', McGraw-Hill, New York, 1972
7 Parkinson, J. M. and Hutchinson, D. in 'Numerical methods for nonlinear optimization' (Ed. D. Murray), Academic Press, London, 1973, 99–114
8 Egeland, O. and Araldsen, P. O. *J. Comp. Structures* 1974, **4,** 41
9 Jensen, P. S. 'Computer methods for partial differential equations', *Proc. AICA Int. Symp., Lehigh Univ.*, 1975, 80
10 Venkayya, V. B. *J. Comp. Structures* 1971, **1,** 265