

AN ITERATIVE SOLVER FOR BOUNDARY ELEMENT ANALYSIS

J O Watson

School of Mining Engineering
University of New South Wales
Sydney 2052, Australia
jo.watson@unsw.edu.au

Key Words: *Boundary Elements, Equation Solver*

ABSTRACT

In typical three dimensional elastostatic analyses encountered in engineering practice, the solution of simultaneous equations is by far the most time consuming computational task unless an alternative to Gaussian elimination on a fully populated matrix can be found. The purpose of the research outlined here is to develop an alternative that reliably and economically solves systems resulting from the use of relatively coarse meshes of higher order elements, especially for finite domains with mixed boundary conditions or minimal constraint on displacement. Currently, the goal is to solve systems of order 3000 to 60000 from around 5 to 50 times faster than a conventional solver, depending upon problem size. 60000 degrees of freedom corresponds to about 5000 nine-node quadrilateral elements, and it is envisaged that analyses will be run on a system with about 1Gb RAM and 120Mb hard disc.

In the proposed clustering scheme [1], advantage is taken of the decay to zero at infinity of kernels of the boundary integral equation. Where there are traction unknowns, hypersingular equations of nodal collocation [2] are taken, to achieve the same rate of decay as for the integral equations written elsewhere. An octree clustering scheme is constructed, with maximum cluster size currently 30 nodes. Two system matrices are assembled: the matrix \mathbf{A} which at least for smaller problems is fully populated, and a sparsely populated matrix \mathbf{A}^* in which submatrices corresponding to pairs of clusters sufficiently far apart in relation to their diameters are lumped into smaller numbers of coefficients. Solution of $\mathbf{A}\mathbf{u} = \mathbf{b}$ is by the iterative scheme

1. set $\mathbf{b}^{(0)} = \mathbf{b}$ and $\mathbf{u} = \mathbf{0}$;
2. for $i = 1, 2, 3, \dots$ compute $\mathbf{u}^{(i)} = (\mathbf{A}^*)^{-1} \mathbf{b}^{(i-1)}$, $\mathbf{u} = \mathbf{u} + \mathbf{u}^{(i)}$, $\mathbf{b}^{(i)} = \mathbf{b}^{(i-1)} - \mathbf{A}\mathbf{u}^{(i)}$, until $|\mathbf{u}^{(i)}|/|\mathbf{u}| < \epsilon$

so that the solution obtained is practically the same as that which would have been given by direct solution of $\mathbf{A}\mathbf{u} = \mathbf{b}$.

A submatrix of \mathbf{A}^* corresponding to a pair of clusters may be lumped into columns, into rows, or both rows and columns. Bearing in mind the significance of bending in many analyses of finite domains, lumping into columns is achieved by the adoption of an

approximate linear variation of displacement (or traction):

$$\mathbf{u}^*(\boldsymbol{\eta}) = \mathbf{a}_0 + \mathbf{a}_1 \eta_1 + \mathbf{a}_2 \eta_2$$

where η_j are cluster local cartesian coordinates. Let the numbers of rows and columns of the submatrix be $3m$ and $3l$ respectively. The additional degrees of freedom \mathbf{a}_0 , \mathbf{a}_1 and \mathbf{a}_2 are related to nodal displacements (or tractions) by the constraint equations

$$l \mathbf{a}_0 = \sum_{c=1}^l \mathbf{u}_c \quad \text{and} \quad I_j^\eta \mathbf{a}_j = \sum_{c=1}^l \eta_j^c \mathbf{u}_c \quad \text{where} \quad I_j^\eta = \sum_{c=1}^l (\eta_j^c)^2, \quad j = 1, 2$$

and l is the number of nodes of the cluster, and the subscript or superscript c denotes nodal value; these equations are added to the system of simultaneous equations to be solved. It can be shown that lumping of 3×3 submatrices is carried out as follows.

$$\sum_{c=1}^l \mathbf{A}_{rc} \mathbf{u}_c \approx \mathbf{R}_0^r \mathbf{a}_0 + \mathbf{R}_1^r \mathbf{a}_1 + \mathbf{R}_2^r \mathbf{a}_2 \quad \text{where} \quad \mathbf{R}_0^r = \sum_{c=1}^l \mathbf{A}_{rc}, \quad \mathbf{R}_j^r = \sum_{c=1}^l \eta_j^c \mathbf{A}_{rc}, \quad j = 1, 2$$

To lump a submatrix into rows, 3×3 submatrices of coefficients are taken to vary linearly with respect to local coordinates ξ_j of the cluster of collocation nodes:

$$\mathbf{A}^*_{rc} = \mathbf{C}_0^c + \mathbf{C}_1^c \xi_1 + \mathbf{C}_2^c \xi_2$$

\mathbf{C}_0^c , \mathbf{C}_1^c and \mathbf{C}_2^c are coefficients of the nine weighted average equations which replace those of nodal collocation. It can be shown that they are given by

$$m \mathbf{C}_0^c = \sum_{r=1}^m \mathbf{A}_{rc} \quad \text{and} \quad I_j^\xi \mathbf{C}_j^c = \sum_{r=1}^m \xi_j^r \mathbf{A}_{rc} \quad \text{where} \quad I_j^\xi = \sum_{r=1}^m (\xi_j^r)^2, \quad j = 1, 2$$

where the subscript or superscript r denotes collocation point number.

Column lumping has been operational for some time, whereas row lumping is in early stages of testing. For around 4000 degrees of freedom, row lumping alone typically accelerates solution by a factor of between two and three with $|\mathbf{u}^{(i)}|/|\mathbf{u}| < 10^{-5}$ after less than ten iterations, for domains of finite and infinite extent. With both row and column lumping there is a further improvement for infinite domains, but for finite domains some difficulties remain to be resolved. In the longer term, a hierarchical scheme in which the existing small clusters are grouped into larger ones may be developed. For typical coarse meshes of higher order elements, the surface patches corresponding to the larger clusters will include edges and corners, and linear approximate variations of displacement and 3×3 submatrices will not be applicable to them.

REFERENCES

- [1] W. Hackbusch and Z. P. Nowak, 'On the fast matrix multiplication in the boundary element method by panel clustering', *Numerische Mathematik*, Vol. **54**, pp. 463-491 (1989).
- [2] J. O. Watson, 'Boundary elements for cracks and notches in three dimensions', *Int. J. Numer. Meth. Engng*, Vol. **65**, pp. 1419-1443 (2006).