1992

Determination of Scroll Wrap Contact Stresses Usingthe Boundary Element Method

M. M. Marler
United Technologies Research Center

K. B. Kumar
United Technologies Research Center

DETERMINATION OF SCROLL WRAP CONTACT STRESSES USING THE BOUNDARY ELEMENT METHOD

M.E. Marler (Assistant Research Engineer)
K.B. Kumar (Research Assoc., Clarkson Univ.)

United Technologies Research Center
East Hartford, CT 06108

ABSTRACT

The Boundary Element Method (BEM) has been known to be extremely useful for the solution of elastic stress analysis problems involving high stress/strain gradients. In recent years the advent of massively parallel computing and boundary element computational procedures designed to effectively exploit these characteristics have paved the way for the static stress analysis of complex structures which were previously uneconomical to analyze due to CPU and memory limitations. These issues led to the development of BEACON-3D (Boundary Element Analysis on the CONnection Machine-3D) at the United Technologies Research Center and its use to predict stresses and displacements in the scroll orbiter. The finite element analysis of the stresses in the orbiting scroll due to pressure, thermal and centrifugal loading, predicts a high tensile stress concentration at the base of the wrap near the discharge port. However, the magnitude of the stress concentration can be significantly affected by the transfer of load due to contact between the fixed and orbiting scroll. Memory and CPU limitations made the finite element contact stress analysis impractical, whereas BEACON-3D did not pose these constraints. Hence, a contact stress analysis of the orbiting scroll was performed using BEACON-3D, which gave a more accurate prediction (significantly lower levels) of the wrap stress concentration. A brief description of the basic concepts of the BEM and the contact algorithm used in BEACON-3D are discussed. Numerical results which demonstrate the impact of contact on stress concentrations are also presented.

NOMENCLATURE

C_{ij}	Boundary discontinuity term
F_{ij}	Traction kernel
G_{ij}	Displacement kernel
u	Nodal displacement
t	Nodal traction
x_j	Coordinates of the load point
ξ_j	Coordinates of the sample point
S	Surface over which integration is being performed
μ	Modulus of rigidity
ν	Poisson's ratio
δ_{ij}	Kronecker delta
E	Young's modulus
[A]	Boundary element LHS matrix
$\{x\}$	Vector of unknown displacements and tractions
$\{b\}$	RHS vector

INTRODUCTION

The BEM is a computational technique for the solution of elastic stress analysis problems. In the BEM, only the surface of the structure to be analyzed is discretized by elements. Hence, it is much easier to generate boundary element models, when compared to other solution techniques such as the Finite Element Method (FEM) which requires modeling the volume of the domain, see Figures 1-2. It can also

give more accurate results in the region of rapid stress change or flux variation and is very well suited for shape optimization problems because only the boundary of the problem need be evolved through a series of changing shapes.

Computations based on boundary element analysis typically involve three major steps : (1) numerical calculation of integrals for each element/node pair, (2) assembling a matrix of coefficients on the basis of prescribed boundary conditions, and (3) numerical solution of the resulting equation system. The boundary element stress analysis of a compressor component such as the orbiting scroll may involve as many as 1000-1500 elements leading to the calculation of several million integrals. Experience in conducting 3D boundary element stress analyses of large models has shown that it is simply uneconomical to perform the analysis even when these codes are exercised on a CRAY system [1]. However, in recent years the advent of massively parallel computing and the implementation of boundary element procedures designed to effectively exploit these characteristics, have made the 3D boundary element analysis economical for very large problems. A 3D boundary element code BEACON-3D was developed using the massively parallel computing environment.

The particular machine on which the computations reported in this paper were made is the Connection Machine (CM-2) with a capacity of 16,384 physical processors. The Connection Machine is a SIMD (Single Instruction Multiple Data) machine in which each processor executes the same instruction set on a separate data packet. A front-end provides the programing environment for the system allowing program storage there. User input such as the data required for executing the programs, can be stored on the front-end. More details pertaining to the computing environment are provided in the subsequent section.

BEACON-3D supports linear elastostatic stress analysis with mechanical, thermal, centrifugal and rigid body contact loading. The elements used in modeling are an eight-noded quadrilateral (higher-order) element.

COMPUTATIONAL ENVIRONMENT

Figure 3 shows the computing environment which includes the front-end and the massively parallel processors, i.e. the CM-2 connected to its data vault. A front-end computer (VAX 6320) serves as a gateway to the Connection Machine system. It provides software development tools, software debugging tools, and a program execution environment familiar to the user. From the point of view of the user, the Connection Machine environment appears to be an extended version of the normal front-end computer environment. The front-end computer also contains specialized hardware called a Front-End Bus Interface which allows communication with the Connection Machine which has access to its own secondary storage area.

A brief description of the CM-2 is provided below and is drawn from Reference [2]. The interested reader is directed to Reference [2] and the references cited there in for more details. The connection machine model CM-2 is an integrated system of hardware and software for parallel computing. This type of computing associates one processor with each data element to exploit the natural computational parallelism inherent in many data-intensive problems. The system software is based upon the operating environment of the front-end computer.

The system hardware is able to directly implement parallel programming instructions. The hardware consists of 65,536 physical processors each with its own local memory. The machine architecture is classified as a SIMD type, i.e. all the processors perform the same instruction on the processors specific data at the same time and there exists more than one path to access information from the other processors. Parallel data structures are spread across the data processors, with a single element stored in each processor memory. When parallel data structures have more then 65,536 data elements (the normal case), the hardware operates in a virtual processor mode, presenting the user with a larger number of processors, each with a correspondingly smaller memory. The ratio of the number of virtual processors to the number of actual physical processors is known as the Virtual Processor ratio (VP ratio).

A CM-2 system can be configured as 64K, 32K, or 16K data processors (here "K" stands for 1024) representing a full, half, or quarter of a machine. Each data processor has 64K bits (2048 32-bit words)

of bit-addressable local memory and an arithmetic logic unit that can operate on variable-length operands. Each data processor can access its memory at a rate of at least 5 megabits per second. A fully configured CM-2 thus has 512 megabytes of memory that can be read or written at about 300 gigabits per second. When 64K processors are operating in parallel, each performing a 32-bit integer addition, the CM-2 parallel processing unit operates at about 2500 Mips. The CM-2 parallel processing unit also has an optional parallel floating point processor per 32 data processors that currently perform at 3500 MFlops (single precision) or 2500 MFlops (double precision) using the parallel machine language, PARIS. Even higher performance is possible by micro-coding these floating point processors (special cases have exceeded 20 GFlops).

FORMULATION

A boundary integral equation governing the response in a linear elastic solid mechanics problem (body forces are neglected for simplicity) can be derived using fundamental solutions and the reciprocal theorem. This equation is also known as Somigliana's identity and can be found in References [3,4].

$$C_{ij}(\xi)u_i(\xi) = \int_S \{t_i(x)G_{ij}(x,\xi) - F_{ij}(x,\xi)u_i(x)\}\, dS \tag{1}$$

Where $G_{ij}(x,\xi)$ is the i component of the displacement at a point x in an infinite elastic medium due to a unit load applied at ξ in the j direction. $F_{ij}(x,\xi)$ is the i component of the traction at a point x in an infinite elastic medium due to a unit load at ξ in the j coordinate direction. G_{ij} and F_{ij} are called fundamental solutions [3,4] and are shown to be :

$$G_{ij} = \frac{1}{16\pi\mu R(1-\nu)}\{R_{,i}R_{,j} + (3-4\nu)\delta_{ij}\} \tag{2}$$

$$F_{ij} = \frac{-1}{8\pi R^2(1-\nu)}\{R_{,k}N_k[(1-2\nu)\delta_{ij} + 3R_{,i}R_{,j}] + (1-2\nu)[R_{,i}N_j - R_{,j}N_i]\} \tag{3}$$

$$R = \left(\sqrt{R_1^2 + R_2^2 + R_3^2}\right) \tag{4}$$

$$R_{,j} = \frac{R_j}{R} \tag{5}$$

$$R_j = x_j - \xi_j \tag{6}$$

$$C_{ij}(\xi) = \frac{\delta_{ij}}{2} \textit{ on a smooth surface} \tag{7}$$

where x_j, ξ_j are the coordinates of the load point and sample point, respectively. S is the surface over which integration is being performed while u_i and t_i are the actual displacements and tractions on the surface S.

In order to solve the boundary integral equation (1) numerically, the boundary of the domain is discretized with elements over which displacements and tractions are written in terms of their values at a series of nodal points. Equation (1) is applied in discretized form to each nodal point on the boundary of the domain and integrals are computed by a numerical quadrature scheme. $C_{ij}(\xi)$ is computed from rigid body conditions [3,4]. Writing the discretized form of Equation (1) for every node point, a system of linear algebraic equations are obtained :

$$[F]\{u\} = [G]\{t\} \tag{8}$$

where [F] and [G] are known matrices that involve contributions from each element on the surface.

{u} is the column vector of nodal point displacements and {t} is the column vector of tractions. In a well posed problem either u or t at a node point is known.

In the above equation all the unknown tractions are transferred to the left hand side of the equation, and all the known displacements are transferred to the right hand side of the equation. In this process

the columns of [G] which correspond to the unknown tractions are negated and are transferred to the left hand side. Similarly the columns of [F] which correspond to the known displacements are transferred to the right hand side. Hence by applying the boundary conditions in the above fashion all the unknowns can be grouped, resulting in the equation :

$$[A]\{x\} = \{b\} \tag{9}$$

where {x} is the column vector of all the unknowns, {b} is the known right hand side vector and $[A]$ is known in terms of $[F]$ and $[G]$. The system can be solved for all the unknown values at the boundary nodes.

The integrals of Equation (1) extend over the entire boundary and are numerically computed over each element for all the node points. Thus in a typical sequential calculation a term such as $t_i(x)G_{ij}(x,\xi)$ is calculated for an element and all the node point pairs. This step is repeated for all the elements. Hence one can easily recognize that numerical computation of the integrals in Equation (1) is a computationally expensive step.

The power of massively parallel computation is evident by recognizing the fact that each processor calculates the integrals of Equation (1) so that the entire calculation for one element and all the node point pairs can be performed *simultaneously* and are either written to the datavault or copied onto the front-end for further use. In a massively parallel computing environment such as the CM-2 each processor of the CM-2 is associated with a node point and the front-end is associated with the element. Thus the computations for an element and all the node points are done in parallel. This step is repeated for all the elements. In fact, if there are sufficient processors (or virtual processors) all the element and node point pairs can be integrated in parallel. Thus the saving in time is clearly evident. It is to be noted that there is no inter-processor communication, while the integrals are being computed. The computed integrals are written in parallel onto the data vault for use in assembly.

Assembly of the boundary element system matrices $[A]$ and $\{b\}$ are carried out by taking advantage of the fact that multiple rows of the system matrix can be assembled in parallel. For problems with large numbers of nodes, such as the stress analysis of the scroll, system matrices cannot fit into CM-2's memory (in-core) and cannot be assembled in-core. Hence these matrices are assembled out-of-core and are saved onto the data vault for use in the solver. Direct matrix equation solution procedures such as Gaussian elimination (LU decomposition) is used in solving the matrix equation $[A]\{x\} = \{b\}$. Due to the large size of matrix $[A]$ the matrix is partitioned into a set of smaller submatrices (blocks), each of which fit into CM-2's memory. An out-of-core block-LU algorithm is used in solving these system matrices to obtain all the unknowns. Once the displacements and tractions at all the boundary nodes are known, stresses at these nodal points can be easily computed. Stresses at interior points can also be computed from values at the boundary nodes.

CONTACT ALGORITHM

When one body comes into contact with another body, the actual surface area where the two are touching is unknown in most problems. This introduces a nonlinearity into the equation $[F]u = [G]t$ since in the area of contact, the tractions t are determined by the displacements u , i.e. $t = f(u)$. Thus an iterative algorithm is introduced to obtain the solution.

A contact algorithm which considers external rigid body contact is implemented in BEACON-3D. A list of node numbers and element numbers expected to be in contact with the external rigid surface is provided by the user. Other details like the gap between the contact surface and the external rigid surface (see Figure 4), the direction of expected contact, the tolerance value to be used in the algorithm, etc. are to be provided by the user. The contact algorithm of BEACON-3D is as follows :

1. Analyse the structure with all the loads and boundary conditions but without assuming any external rigid surface.
2. Find the maximum displacement in the direction of contact from all the expected contact nodes. All the expected contact nodes having a displacement value greater than or equal to (maximum

displacement - a tolerance) are assumed to be in contact with the external rigid surface (tolerance value is provided by the user).

3. The system matrix equation and the right hand side is reconstructed and solved to reflect the nodes that have come into contact in step 2. From this new solution vector find the maximum displacement among the expected contact nodes in the direction of contact. All the expected contact nodes having a displacement greater than or equal to (maximum displacement-tolerance) are assumed to be in contact with the rigid surface. At this time a check is performed to see if any of the nodes fixed in step 2 have reversed its traction direction and if so these nodes are released from being in contact. It is to be noted that the tolerance used here is less than the tolerance used in step 2 and is provided by the user.

4. Repeat step 3 until all the nodes from the expected contact list are either in contact or have displacement values less than the gap.

5. Stresses at all the boundary nodes are obtained after step 4 is completed.

It is to be noted that there is an alternate form of contact algorithm described in Reference [5], which determines the contact nodes by increasing the load gradually. This algorithm was not implemented in BEACON-3D.

ANALYSIS

The boundary element model for a typical orbiting scroll was built using PATRAN. Since the BEM only requires the surface of the scroll to be defined, the model is much easier to build than a finite element model which requires the internal volume (Figure 5). Because the scroll wrap is curved and thin, a very large number of lower-order elements would be needed to model the complex stress states present. This makes both the modeling and analysis process impractical. Using higher-order elements significantly reduces the number of elements needed to model this complex geometry, consequently, an eight-noded quadrilateral (higher-order) boundary element was used.

This analysis considers the loads due to fluid pressure, centrifugal acceleration, thermal expansion and rigid body contact. The fluid pressures in the compression and back chambers were provided by a thermodynamic simulation code developed at UTRC. The centrifugal acceleration load was due to the constant translational rotation of the orbiter. The temperature profile was determined from experimental data. Displacement boundary conditions were applied as follows: The scroll bearing case was fixed in the radial direction, the Oldham keyway area was fixed to prevent rotation about the Z axis, and the thrust surface between the fixed and orbiting scroll was fixed against Z displacement (see Figure 6).

Further displacement constraints are imposed when the deformation of the orbiting scroll forces it into the fixed scroll (i.e. contact). The contact on the scroll wrap surfaces was categorized as tip contact and flank contact. The tip surface is the section of wrap that seals against the base plate of the fixed scroll while the flank surface seals between the fixed and orbiting scroll wraps (Figure 7). The tip Z displacement is restricted by the heavy cast iron base plate of the fixed scroll. To model this effect, the fixed scroll base plate was defined as a rigid body (Figure 6). Therefore, as the wrap tip deforms it is not allowed to penetrate this surface.

To properly model the contact between the flanks of the orbiting and fixed scroll, flexible body contact must be considered. This capability is currently being developed for BEACON-3D.

RESULTS

The initial analysis of the scroll considered fluid pressure, centrifugal, and thermal loads but neglected the added displacement restrictions imposed by contact between the fixed and orbiting scroll. In this analysis, predictions for displacement and stress at the base of the wrap near the discharge port yielded high values when extremes of the operating envelope were analyzed. For example, at the 30/155/50 condition, regions of the wrap are subjected to pressure differentials in excess of 175 psi. Deformed shape plots, Figures 9-12, have been scaled for visual purposes. A comparison of the undeformed (Figure 8) and

deformed wrap (Figure 9) demonstrates the bending induced by the pressure differentials in the compression chambers. The Z displacement of the wrap tip, shown in Figure 10, is due to wrap bending and thermal expansion. The individual contributions of the respective loads to the maximum stresses and displacements are listed in Table 1. The location of these stress and displacement values are shown in Figures 9-12. The bending stresses due to fluid pressure and centrifugal loading contribute to the majority of the tensile stress σ_t. The addition of thermal expansion increases the stress levels slightly but increases the displacements significantly.

Pressure, centrifugal and thermal loading was combined with the displacement restrictions imposed when the tip contacts the rigid base plate. The deformed wrap plot (Figure 11) in the X-Y plane does not seem to have changed significantly from the analysis neglecting contact (Figure 9) whereas, a comparison of the Z displacement, Figures 10&12, shows a significant decrease. The displacement restriction due to the rigid base plate introduces a compressive stress into the wrap which significantly decreases the stress concentration previously predicted without tip contact. For example at the 30/155/50 operating condition, the maximum tensile stress decreased over 10,000 psi. Likewise, the maximum compressive stress increased a similar amount. These results demonstrate the importance of considering the interaction between the orbiting and fixed scroll in the analysis.

CONCLUSION

The bending stresses due to fluid pressure and centrifugal loading contribute to the majority of the tensile stress σ_t. The addition of thermal expansion increases the stress levels slightly but increases the displacements significantly. The effect of the transfer of load due to the contact between the tip of the orbiting scroll wrap and the rigid base plate of the fixed scroll produces a more realistic prediction of the wrap tip stress concentration, and shows the maximum tensile stress to be significantly lower while the maximum compressive stress is significantly increased.

The effect of elastic contact between the fixed scroll wrap tip and the orbiting scroll base plate, as well as the flank contact between the fixed and orbiting scroll, will provide a better understanding of the stress distribution and displacements in the scroll compressor. The capability to perform these analyses is currently being developed for BEACON-3D.

ACKNOWLEDGEMENT

The authors wish to extend their thanks to R.B. Wilson of Pratt and Whitney for his technical guidance and encouragement in developing BEACON-3D. The authors are indebted to Mr. D.J. McFarlin, Dr. A.A. Peracchio and Dr. H.T. Shu of UTRC for their support and encouragement.

REFERENCES

1. Kumar, K., Kane, J.H., Wilson, R.B., and Srinivasan, A.V.; "The influence of Massively Parallel Processing in Boundary Element Computations," International Symposium on Boundary Element Methods 1989.

2. Wake, B.E., and Egolf, T.A.; Implementation of a Rotary-Wing Three-Dimensional Navier-Stokes Solver on a Massively Parallel Computer, AIAA Paper 89-1939-CP, AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY, 1989.

3. Banerjee, P.K. and Butterfield, R.; Boundary Element Methods in Engineering Science, McGraw-Hill Book Company, 1981.

4. Brebia, C.A.,Telles, J.C.F.,Wrobel, L.C.; Boundary Element Techniques, Springer-Verlag,1984.

5. Karami, G.;Boundary Element Analysis of Elasto-Plastic Contact Problems, Computers Structures, Vol. 41, pp. 927-935, 1991.

Table 1 BEM Analysis Results for Cast Iron Orbiter

P: Pressure & Centrifugal Loading,
T: Thermal Loading,
C: Contact Wrap Tip,
σ_t: Tensile Stress, σ_c: Compressive Stress,
δ_a: Axial Deflection, δ_r: Radial Deflection

Condition	Load	Max. Stress		Max. Deflection	
		σ_t	σ_c	δ_a	δ_r
45/130/65	P	2.90	1.00	1.00	2.94
	P+T	4.13	3.27	9.11	4.78
30/155/50	P	11.04	3.86	1.28	8.94
	P+T	12.27	5.18	9.56	11.17
	P+T+C	6.94	8.26	2.67	7.83

(Stress and defl. normalized to σ_r & δ_r of 45/130/65 condition, respecitvely)

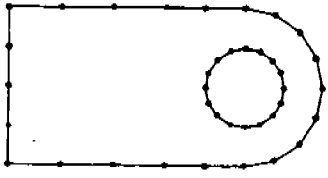

Figure 1 BEM mesh

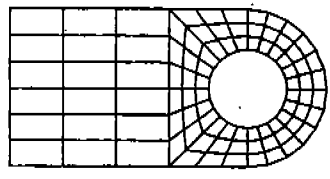

Figure 2 FEM mesh

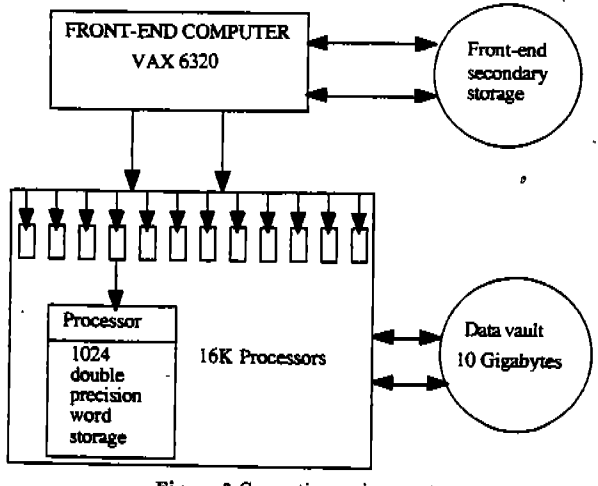

Figure 3 Computing environment

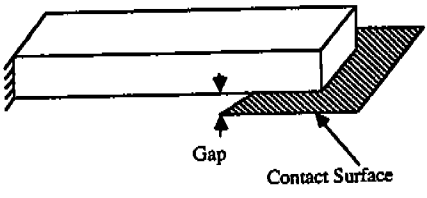

Figure 4 Contact gap example

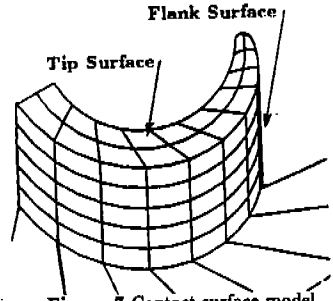

Figure 7 Contact surface model

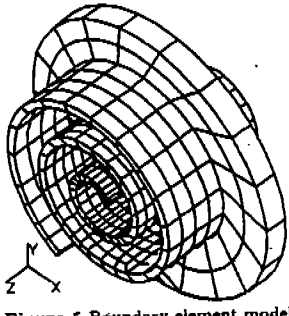

Figure 5 Boundary element model

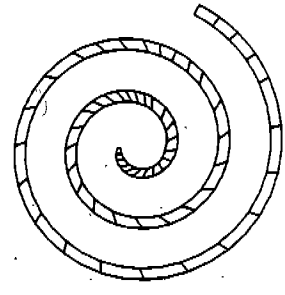

Figure 8 Undeformed wrap

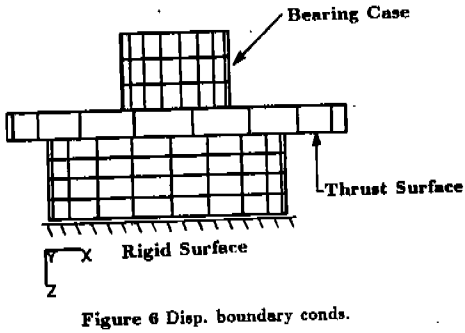

Figure 6 Disp. boundary conds.

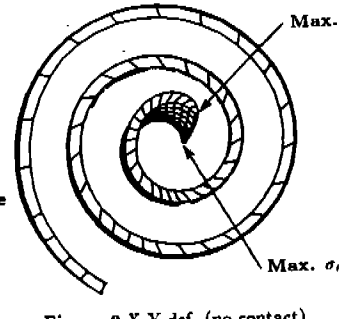

Figure 9 X-Y def. (no contact) 30/155/50.condition

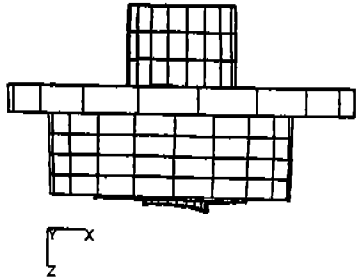

Figure 10 X-Z def. (no contact)
30/155/50 condition

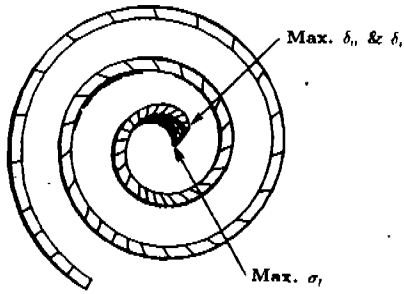

Figure 11 X-Y def. (with contact)
30/155/50 condition

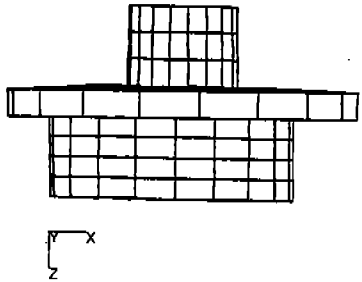

Figure 12 X-Z def. (with contact)
30/155/50 condition