A CONTACT SEARCHING ALGORITHM FOR CONTACT-IMPACT PROBLEMS*

Wang Fujun (王福军) Cheng Jiangang (程建钢) Yao Zhenhan (姚振汉)
(Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China)

ABSTRACT: A new contact searching algorithm for contact-impact systems is proposed in this paper. In terms of the cell structure and the linked-list, this algorithm solves the problem of sorting and searching contacts in three dimensions by transforming it to a retrieving process from two one-dimensional arrays, and binary searching is no longer required. Using this algorithm, the cost of contact searching is reduced to the order of $O(N)$ instead of $O(N\log_2 N)$ for traditional ones, where $N$ is the node number in the system. Moreover, this algorithm can handle contact systems with arbitrary mesh layouts. Due to the simplicity of this algorithm it can be easily implemented in a dynamic explicit finite element program. Our numerical experimental result shows that this algorithm is reliable and efficient for contact searching of three dimensional systems.

KEY WORDS: contact impact, contact search, finite element method

1 INTRODUCTION

Investigations of the effects of structural impacts are of great importance in many engineering fields. Considering the potential large deformations, a contact can occur anywhere within the structure. The ability to determine contact areas of a general structural system is a fundamental requirement of the contact-impact analysis[1]. Usually, the contact searching includes two different processes[1~3]. The first is the global search that roughly finds all possible candidate contact nodes of surface segments, the second is the local search that locates exactly contact positions after the global search. This paper is concentrated on the global search.

So far, a number of various algorithms have been used for the global search. Of them, a typical one is the bucket-sorting algorithm[4] that is embedded in the master-slave contact algorithm[5] and the single-surface contact algorithm[2] of DYNA3D by Hallquist. It performed sorting and searching in three dimensions in a nested manner. Later Oldenburg and Nilsson reduced the three-dimensional search to sort and search only in one dimension, that is the position code algorithm[1]. Probing appropriate position codes in a position code vector is done by the binary searching. Thus the complexity of the search is reduced substantially so that the cost of operations of the global search is $O(N\log_2 N)$. In paper [6],

Received 29 January 2000, revised 21 July 2000
* The project supported by the National Natural Science Foundation of China (59875045), and the State Key Laboratory of Automobile Safety and Energy Saving (K9705)
an NBS contact searching algorithm for large-scale discrete element simulations is proposed, with total search cost proportional to $N$. However, the NBS algorithm can be only applied to systems comprising spherical bodies of similar size\cite{6}.

In this paper, we propose a new algorithm for global search. It is derived by using the linked-list\cite{7} that is widely used as a data structure in computer programming, and transforms the problem of sorting and searching in three dimensions to a process of sorting and searching within a one-dimension array. It does not need the binary searching, and reduce the searching cost to the order of $O(N)$. Moreover, this algorithm can handle contact systems without limitation of their mesh layouts.

2 FINITE ELEMENT FORMULATIONS FOR CONTACT-IMPACT PROBLEMS

As described by Bathe\cite{8}, the equation of the system motion can be transformed into the following discrete equilibrium equation using the standard finite element procedure

$$M^t \ddot{u} = tQ - tF + tF_c$$  \hspace{1cm} (1)

In Eq.(1), $M$ is the mass matrix, $^t \ddot{u}$ is the acceleration vector, $^tQ$ is the external load vector, $^tF$ is the internal load vector and $^tF_c$ is the nodal contact force vector. $^tF_c$ can be calculated according to the result of contact searching using the penalty method\cite{2}, the Lagrange multiplier method\cite{9,10} or other methods.

Suppose that $M$ is a diagonal mass matrix, Eq.(1) can efficiently be integrated with central differences. The displacements at time $t + \Delta t$ are obtained from their values at time $t - \Delta t$ and $t$

$$t^+ \Delta t u = M^{-1} [\Delta t^2 (^tQ - ^tF + ^tF_c) + 2M^t u - M^t - \Delta t^2 u]$$  \hspace{1cm} (2)

where $\Delta t$ is the time increment.

In Eq.(2), the contact force $^tF_c$, at time $t$, is the only unknown variables. To calculate the contact force $^tF_c$, the contacting boundaries have to be determined using a contact searching algorithm.

3 NEW CONTACT SEARCHING ALGORITHM

The goal of the global search procedure is to find out those nodes that are candidates for contact from all pre-defined contact nodes in the structure. We propose a new algorithm through which the contact searching is accomplished by using a cell structure. The cell structure is fixed in space, and the cell sizes are considered to be close to or larger than the average element sizes. The basis of this searching procedure is to identify all nodes in a cell. The key part of this algorithm is the utilization of a special linked-list.

3.1 Cell Structure and Location of Nodes

To identify these nodes and elements that may contact, the concept of cell\cite{11} is introduced. A typical cell structure is shown in Fig.1 that the cells are of equal size in all directions and their edges are parallel to the $x, y$ and $z$ axis.

The cell domain is described as follows

$$(x_{min}, x_{max}, N_x) \hspace{1cm} (y_{min}, y_{max}, N_y) \hspace{1cm} (z_{min}, z_{max}, N_z)$$  \hspace{1cm} (3)$$