Engineering Analysis with Boundary Elements 58 (2015) 1–6

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Efficient visibility criterion for discontinuities discretised by triangular surface meshes

Nicholas Holgate, Grand Roman Joldes*, Karol Miller

Intelligent Systems for Medicine Laboratory, School of Mechanical and Chemical Engineering, The University of Western Australia, 35 Stirling Highway, Crawley/Perth, WA 6009, Australia

ARTICLE INFO

Article history:
Received 5 September 2014
Received in revised form
8 January 2015
Accepted 25 February 2015
Available online 20 March 2015

Keywords:
Meshless methods
Discontinuities
Surgical simulation
Biomechanics

ABSTRACT

This study proposes a computationally efficient algorithm for determining which pairs of points among many predetermined pairs in three dimensions will maintain straight line visibility between one another in the presence of an arbitrary surface mesh of triangles. This is carried out in the context of meshless numerical methods with the goal of implementing near-real-time discontinuity propagation simulation. A brief overview is given of existing discontinuity modelling techniques for meshless methods. Such techniques necessitate determination of which key pairs of points (nodes and quadrature points) lack straight line visibility due to the discontinuity, which is proposed to be modelled with a surface mesh of triangles. The efficiency of this algorithm is achieved by allocating all quadrature points and surface mesh triangles to the cells of an overlayed three-dimensional grid in order to rapidly identify for each triangle an approximately minimal set of quadrature points whose nodal connectivities may be interrupted due to the presence of the triangle, hence eliminating most redundant visibility checking computations. Triangles are automatically split such that any size of overlayed cubic grid cells can be employed, and the parameters governing triangle splitting and binning have been examined experimentally in order to optimise the visibility algorithm.

1. Introduction

This work seeks to make an addition to meshless numerical methods as applied to emerging fields such as computational biomechanics, in which one of the present challenges is near-real-time simulation of soft tissue cutting. In applications of biomechanical simulation, meshless methods have many practical advantages over FEM. Presently, the most important is that accurate model generation of patient-specific organ geometry from pre-operational images can be automated, while FEM models comprised of workable elements require days of adjustment by an analyst [1,2]. Removal of this workflow bottleneck makes meshless methods an ideal candidate for implementation in near-real-time intra-operational surgical simulations of tissue deformation, which for many procedures demands simulation of cutting.

Attempts to model discontinuities within meshless simulations have mostly been focused on cracks and their propagation. Rabczuk et al. [3,4] use nodes in the crack path to superimpose a discontinuous enrichment function to the nearby displacement field. Use of these so-called “cracked particles” applies only to finely cracking solids and not to deforming soft bodies with arbitrary discontinuities, and the existence of discontinuities only at particular particles limits the accuracy with which they can be modelled. Level-set functions proposed by Osher and Sethian [5] and applied to FEM crack growth modelling by Stolarska et al. [6], have also been applied to the problem of using meshless methods to model surgical cutting of brain tissue in two and three dimensions by Jin et al. [7]. In two dimensions, discontinuities are represented by a series of straight line segments, each of which uses the vector between the segment's beginning and end to define a level-set function with values of opposing signs on opposing sides of the segment, and another such vector and level-set function perpendicular to the end of the segment. This allows a natural division of the space into four subdomains. By calculating the two function values for each of any two points in space, it can be immediately determined whether the line segment under consideration will block straight line visibility between the points, allowing appropriate adjustment of support domains. This idea can be extended to three dimensions [8], and can also make use of level-set functions whose zeroes are not straight line segments or planes, allowing more complex discontinuities without additional segments, provided that an appropriate closed-form level-set function can be found. A potential drawback to this method is that it requires intricate piecewise function definitions to define jagged or curved shapes, which may entail speed and accuracy reductions. It is also difficult to update the level-sets for intersecting discontinuities or sharp changes in the direction of the cut.

Krysl and Belytschko [3,9] have previously proposed and implemented meshes of triangular elements for modelling of arbitrary crack growth in conjunction with the visibility criterion.

* Corresponding author.
E-mail address: grand.joldes@uwa.edu.au (G.R. Joldes).

http://dx.doi.org/10.1016/j.enganabound.2015.02.014

At a time step in which crack growth occurs, only the new portion of the discontinuity needs to be considered to alter the necessary support domains. For each new triangle in the propagating crack, its bounding box, inflated by the simulation's largest support size, is used as a maximum region in which shape functions could possibly need to be altered due to that discrete portion of the discontinuity. When a set of new triangles is added to extend the crack, for each quadrature point enclosed within the union of all the maximum regions, the rays between it and each associated node are checked for intersection with the new triangles. If the ray intersects the triangle, the "visibility criterion" between the two points of interest fails, and the node is removed from the quadrature point's list of neighbours which influence the local shape functions.

An efficient algorithm for finding which quadrature points are contained within a given maximum region is not proposed in their paper. For a particular crack growth time step there may be many such regions requiring identification of contained quadrature points, and it may be too slow to retrieve the appropriate points from lists of global quadrature point coordinates to permit real-time growth simulation. In furthering the strategy put forward by their paper, it is worth noting that it is not necessary to check all the quadrature points found in the union of the maximum regions against every contained triangle, but rather just the ones inside each individual inflated bounding box against its associated triangle. Additionally, an inflated bounding box will always include unnecessary quadrature points near the corners, which can be mostly eliminated by instead using an appropriately inflated bounding sphere.

In many applications, surface meshes of triangles may be the most desirable method for representing either static or propagating discontinuities. An analyst may easily manually place the vertices of a set of triangles interconnected such that they closely replicate a real discontinuity. Accurate automation of surface mesh creation from an existing three dimensional image featuring a clear discontinuity is also a simple task compared to that of automatically meshing a volume. For propagating discontinuities, addition of triangles to the outer edges of the surface is a natural operation which does not require adjustment to the existing discontinuity.

In Sections 2 and 3, this paper outlines an algorithm for efficiently conducting the visibility checks required to model meshless methods discontinuities with a surface mesh of triangles. Section 4 presents experimental findings pertaining to execution times which justify its use in near-real-time applications such as surgical simulation, even for meshes composed of very many triangles.

2. Algorithm overview

Quadrature points which are more distant than the simulation's largest nodal support size from all points on a triangular discontinuity portion cannot have visibility blocked from the nodes that influence their local shape functions, and so they do not need to have their neighbouring nodes checked for straight line visibility. By cycling through every triangle comprising the surface mesh, and identifying as small as possible a set of quadrature points which contains all of those sufficiently nearby to warrant nodal visibility checks, the number of visibility criterion checks required to appropriately adjust the nodal support domains due to the discontinuity will be minimised.

As a simple and computationally efficient way of enclosing all of the quadrature points within the maximum support size distance from all points of a triangle, bounding spheres are proposed for each triangle. The surface of each sphere is such that all points are at least the maximum support size away from all points on the triangle which it surrounds. It is not viable to find which of the simulation quadrature points lie within the sphere by checking all of their Euclidian distances from its centre. Rather, inspiration is drawn from the field of computational contact mechanics, in which the "bucket search" algorithm (proposed by Benson and Hallquist [10]) is commonly used to efficiently detect the occurrence of contact between two disjoint bodies in a simulation. The adaptation of this algorithm as applied to the present problem begins with superposition of a three-dimensional grid of cubic cells over the physical coordinates. Each surface mesh triangle and each quadrature point is allocated (or "binned") into a unique cell, with triangles being split until they each fit into a unique cell within certain overhang tolerances. With each triangle approximately confined to a single cubic cell, the bounding sphere can be approximated by a "sphere of cubes" consisting of a set of grid cells, centred upon the cell containing the triangle. This scenario is demonstrated in Fig. 1.

The set of cells required to fully include a bounding sphere of a particular radius given a particular cell length can be precomputed, such that all the quadrature points already allocated to that

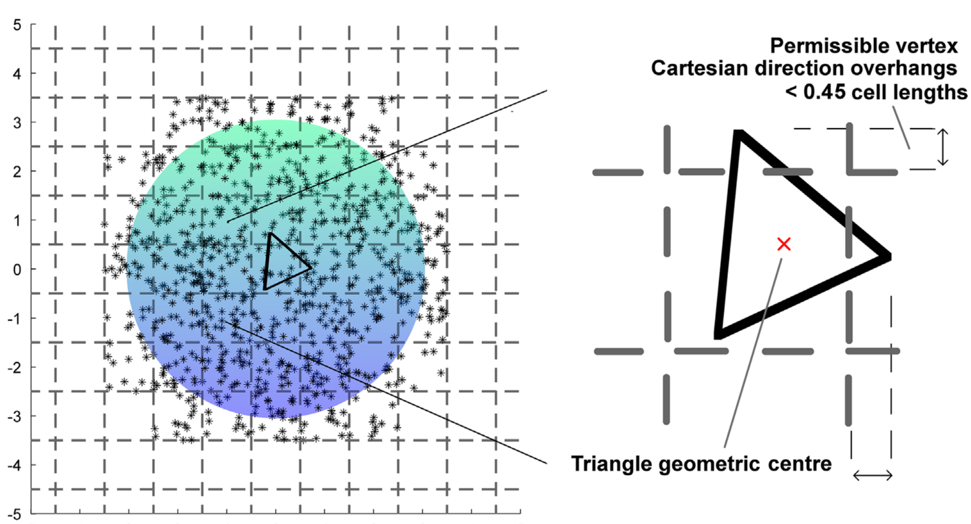

Fig. 1. A sphere of cubes viewed along the cell grid *x*-axis – with visible overestimations, the sphere of cubes includes the entire idealised bounding sphere of the triangle of concern, which in turn includes all quadrature points whose nodes may fail the visibility criterion due to the triangle. The idealised sphere is inflated slightly to allow for the 0.45 cell length maximum possible overhang of the triangle vertices into cells adjacent to the triangle's allocation cell.