



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

An Efficient Inclusion Test for A Massive Point Distribution

J. Yao

May 15, 2013

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

An Efficient Inclusion Test for A Massive Point Distribution

Yao, Jin

Lawrence Livermore National Laboratory, California, USA

Abstract

An efficient inclusion test for a massive distribution of spatial points with a close boundary is described for two-dimensions and three dimensions. Utilizing a background mesh and the wall-sharing relationship between cell walls, the cost of the proposed algorithm is a small fraction of a full inclusion test. Intersections of simple planar geometrical objects locally are the only mathematical operations required.

1 INTRODUCTION

Testing if a given point is enclosed in a closed boundary is a classical problem in computational geometry (figure 1), it has many applications in computational mathematics / physics. Various algorithms exist for solving this problem, some of them are elegant and easy to code ^{[4],[5]}. In general an algorithm to test inclusion involves intersection of rays and boundary faces, or projection to boundary faces, and calculation of angle / solid angle spanned by a point and boundary faces. Almost every algorithm requires to test through every boundary faces. If there are massive points to test against a boundary with a great many faces, the cost of testing is certainly a consideration to software developers.

In this paper, an efficient approach of inclusion test for a massive point distribution with an arbitrary closed boundary represented by faces is described. This method does not test every boundary face for every node, instead it only tests near boundary nodes and for each node in consideration, only the boundary portion local to this node is examined. Assume there are N points to test against a boundary with M faces, the cost for intersection / projection is reduces from $O(MN)$ to a small fraction of it. This fraction is determined by the ratio between the volume of the average area (length on 2D) of a boundary face, multiplied by a characteristic length, to the volume of the region in the problem. This fraction is considerably

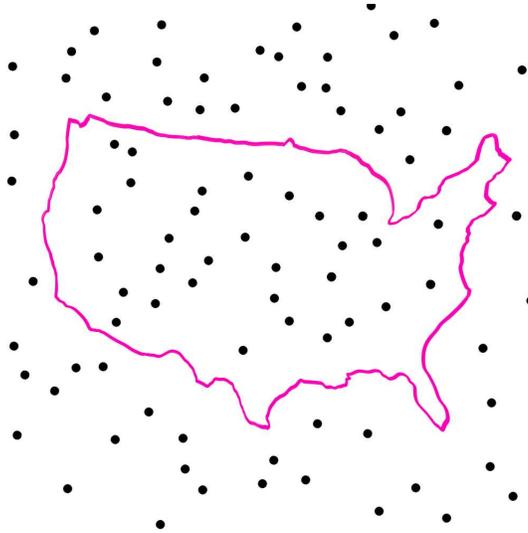


Figure 1: A two-dimensional inclusion problem.

small and mathematically can go to *zero*. The compensation is additional storage of an array in $d+1$ dimensions with d the dimension of space. The volume of the array is the volume of the region in concern divided by the volume of a single cube. Only local intersections between basic geometrical objects are required for the algorithm to work.

The solution approach in three-dimensions is a direct extension of the two-dimensional solution method proposed here, thus only the two-dimensional solution is demonstrated with figures in this article.

This method is natural to parallelize and should fit X-scale calculations well.

2 DESCRIPTION OF THE ALGORITHM

2.1 DEFINITION OF THE PROBLEM

A point set of N points is randomly distributed in space (two-dimensional or three-dimensional). A non self-intersecting closed boundary is represented by a set of planar faces defined by a set of nodes ordered counter-clock-wise, thus by a right-hand rule, the normal is pointing exterior in the three-dimensional case, and the the

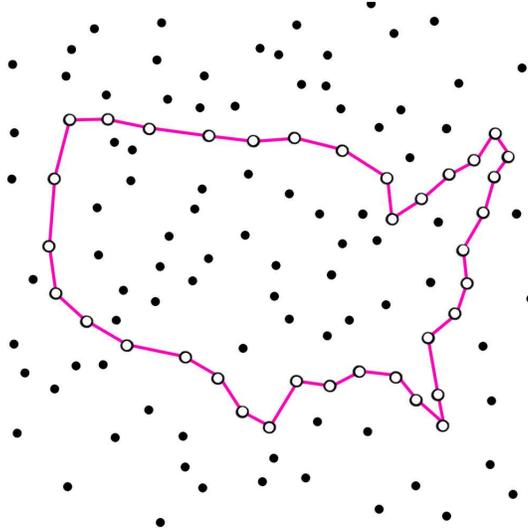


Figure 2: Boundary representation with planar faces.

interior in on the left in the two-dimensional case.

The questions is that how to identify for the whole point set that which of the points are interior, which are exterior.

2.2 THE CHARACTERISTIC LENGTH

The solution approach in three-dimensions is a direct extension of the two-dimensional solution method proposed here, thus only the two-dimensional solution is described in this article. Nevertheless the 2D algorithm can be directly extended to 3D.

It is assumed that the size of boundary faces are not ill-conditioned, i.e., the ratio between the longest face and the shortest face is not too big. However even this assumption is not satisfied, the proposed method would still work with only minor increment of effort. We take a length ℓ_c to scale the problem. This means, without loss of generality ℓ_c is scaled to 1 after the problem is nondimensionalized.

Furthermore for convenience the origin can be set at $(x_0, y_0) = (x_{Min}, y_{Min})$ where x_{Min} and y_{Min} are the lower bounds of the coordinates of the point set.

2.3 THE VIRTUAL CUBICAL COMPLEX

Assume the problem has been scaled by ℓ_c in space.

The entire computational therefore is contained in a Cartesian grid, or a virtual cubical complex (fig. 3). Imagine the unit square defined by diagonals $(i, j), (i + 1, j + 1)$ be virtual cubical (i, j) , each of the given point $p = (x, y)$ must be contained in such a cube. The cube ID that owns p is simply the integer parts of x and y (fig. 4).

It is evident that if the boundary presents a normal body, only a small number of these virtual cubes will intersect the boundary. To be quantitative, because only a single band of cubes contains the boundary (fig. 5), which makes the number of cubes intersect the boundary the order of the volume $A\ell_c$, where A is the area (length in 2D) of the boundary. Therefore about only $O(A\ell_c/V)$ (the volume of the narrowband that covers the boundary curve divided by the volume of the region of computational domain) of the cubes intersect the boundary. In the limit that the characteristic length $\ell_c \rightarrow 0$, the number of boundary cells is of higher order. However, the smaller the ℓ_c , the higher the storage required so there is a balance of computational efficiency and data storage. Nevertheless, with the proposed method the real work is done with the boundary cubes only, locally. This feature makes the proposed method efficient.

2.4 BOUNDARY, INTERIOR, AND EXTERIOR VIRTUAL CUBES

The virtual cubical complex can be "colored" such that one can identify the interior, the boundary, and the exterior cubes with their colors. Say each interior virtual cube is colored green, each boundary virtual cube is colored yellow, and each exterior virtual cube is colored blue (fig. 4). A point contained by a green cube must be interior, a point contained by a yellow cube maybe either interior or exterior, finally, a point contained in a blue cube must be exterior. We only need to deal with points contained in a yellow (boundary) cube. The next question is how to color the virtual cube complex. The approach here is to identify the boundary cubes first, then use the cube wall-sharing relation to find "pure" cubes (to be colored either blue or green) neighboring to "mixed" cubes (colored yellow).

A virtual cube class has three properties: its i coordinate, its j coordinate, and its *color*. An class-array with the length of the total number of virtual cubes can be used to hold the virtual cube complex.

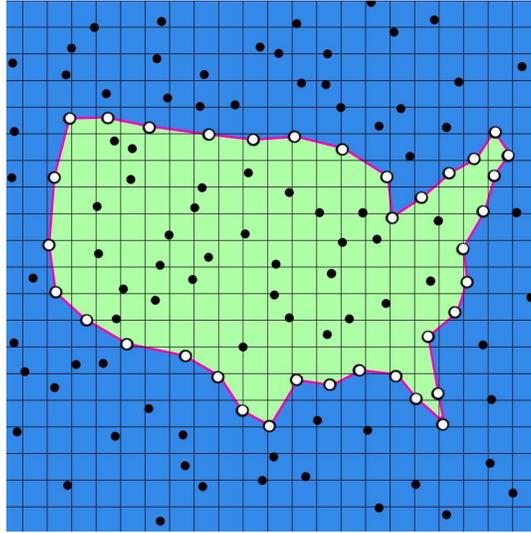


Figure 3: The virtual cubical complex.

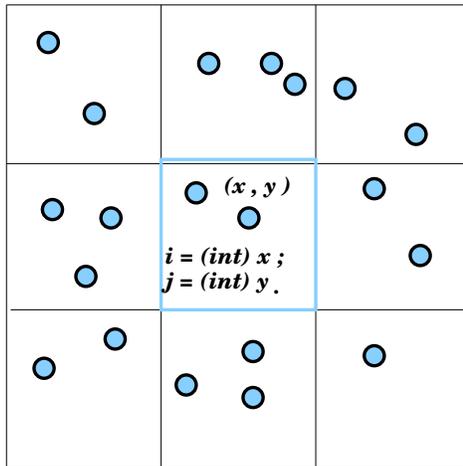


Figure 4: A point (x, y) is contained by virtual cube (i, j) with $i = (\text{int})x, j = (\text{int})y$.

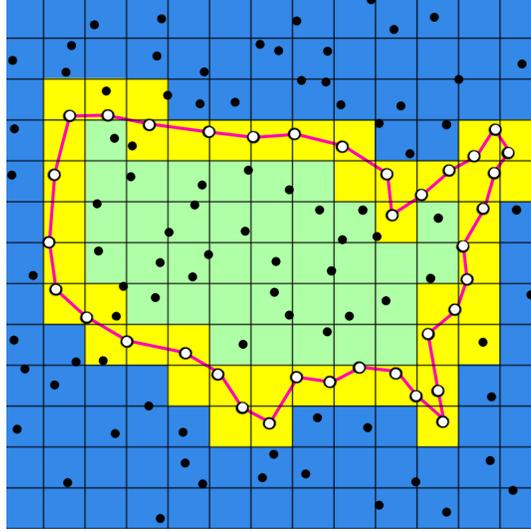


Figure 5: The *interior*, *boundary*, and *exterior* cubes.

2.5 IDENTIFY THE BOUNDARY CUBES

It is straightforward to identify boundary (mixed) nodes.

2.5.1 *The first step*

Pick an arbitrary boundary node n , the integer parts of its physical coordinate (i_n, j_n) shall provide the ID of a mixed cube that contains this node. One starts a directional walking ^[3] from this node along the boundary with right hand side toward the boundary normal until arrive at one of the four walls of the cube (i_n, j_n) . At this moment, paint the cube (i_n, j_n) yellow (fig. 6).

2.5.2 *The second step*

Continue the walk across the wall to enter next cube, then keep walking till arrive at another wall of this cube, mark this cube yellow.

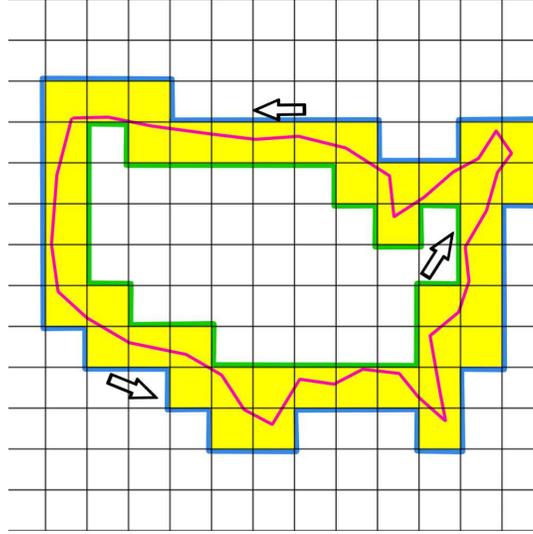


Figure 7: Identify boundary cubes with a directional walk in 2D.

2.5.3 *repeat the second step until starting node is revisited*

If there is only a single surface, because of continuity of the boundary, when the starting boundary node is revisited, all the cubes that contains a piece of the boundary is painted yellow. All cubes left that are not yellow, are “pure” cubes that are either interior or exterior. In three-dimensions, one also starts from a boundary node n and identify the cube ID (i_n, j_n, k_n) as the integer parts of its physical coordinates. Then find the first level of neighbor cubes that contain pieces of the boundary, then keep propagating this search to the second level of mixed neighbor cubes, repeating until the entire surface has been completely covered.

The case of multiple boundaries does not add complexity to the above procedure.

2.6 IDENTIFY THE PURE CUBES WITH WALL-SHARING

A point contained by an interior cube is interior, a point contained by an exterior cube is exterior, once the interior and exterior cubes are identified. The question is how to know if a pure cube is interior (or exterior)? We can achieve this using the narrowband of boundary cubes identified. We “paint” the cubes with a *wall-sharing* relationship.

It is evident that the inner boundary of the narrowband of “mixed” cubes are a collection of walls of pure green cubes, and the outer boundary are a collection of walls of pure blue cubes. We make a collection of all faces that are green which is a closed surface that enclose all the interior cubes.

We mark all the yellow cubes 0 for their logical distance to the boundary are *zeros*. With looping over the set of green faces and find the cubes that share green walls with the cubes marked 0, mark all the cubes found with their logical distance to the boundary 1. Then the boundary of these cubes with mark 1 encloses the rest of interior cubes. Repeating this collecting / marking operation with the logical distances to the boundary, utilizing the wall sharing relationship (the spatial continuity) until no cube with a larger logical distance to the boundary can be found. At this moment, all interior cubes are marked with a *positive integer* (fig. 8).

The above operation can be seen as a loop over the level of neighbors (logical distances to the boundary). Not a single interior cube will be missed because if so, the wall-sharing relation will be broken, this is against spatial continuity.

Exterior cubes can be ranked with their logical distance to the boundary (say, with negative integers) the same way. However it is unnecessary because one can take the default status of cubes as exterior initially (painted blue).

2.7 PUTTING POINTS INTO VIRTUAL CUBES

With a single loop over the given massively distributed point set, each point (x_n, y_n) will be contained in to the cube with its ID $[(int)x_n, (int)y_n]$. Any point that in owned by an interior cube is *interior*, any point that is owned by an exterior cube is *exterior*. The only thing left is that: how to determine if a point owned by a mixed cell is exterior or interior?

Nevertheless, the whole problem which is global is broken into a set of problems of a single point inside a mixed cube, for the case of a massive distribution of points. The method is a collection of *completely local* inclusion tests so is easy to parallelize.

2.8 BOUNDARY CONFIGURATIONS IN A MIXED CUBE

This is the last step and maybe the most challenging part with this algorithm. To make the problem here simpler to deal with, we go back to the phase of identifying the mixed cubes. It is clear that once a cube is found being mixed, the surface faces that intersect this cube are known, this information needs to be kept all the way

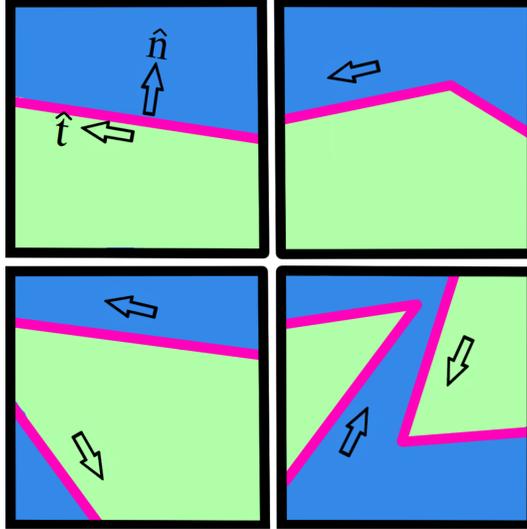


Figure 9: Several ways for a boundary to intersect a two-dimensional virtual cube.

Because a cube is convex this line-segment shall be completely inside the cube. Then one counts the number of intersection between this line-segment and the boundary faces that intersect the cube. If the count is odd, the point is interior, if even the point is exterior (fig. 10).

One can pick ℓ_c as the smallest dimension of a boundary face (assume no extremely short edge existing), this would simplify the searching-intersecting tests associated with this method because for each piece of boundary contained in a mix cube, there is at most a single boundary node on it.

Therefore by looping over all boundary cubes and identify for each cube the points contained in are interior / exterior, or on boundary, we have the solution of an inclusion test for a massive distribution of points in two-dimensions and three-dimensions.

2.10 THE COST OF THE PROPOSED ALGORITHM

The cost of this algorithm depends on the characteristic length ℓ_c that is used to scale the problem. Assume the point set \mathbf{P} is a random distribution of \mathbf{N} points in spatial region of volume \mathbf{V} , and the total volume of the narrowband boundary cubes regarding the boundary \mathbf{B} (defined by M planar faces, say) is $MA\ell_c$ with

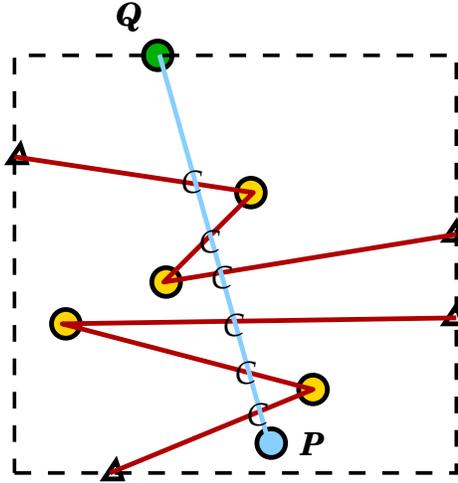


Figure 10: Point P contained in the virtual cube shown is linked with a point Q known to be interior (green) on the wall of the cube, by a line-segment. PQ crossed the boundary 6 times (crossing points are marked by letter C), and *six* is even, thus P is interior. A three-dimensional case is similar.

A being the average area of each face. The number of cubes that contain the boundary is then in the order of $p = O(MA/\ell_c^{d-1})$, where d is the dimensionality of the problem. The number of total cubes is V/ℓ_c^d , therefore in average a cube contains $q = N\ell_c^d/V$ points. The total local inclusion test with the boundary cubes are then $O(pq) = O(MNA\ell_c/V)$. Recall that performing N single inclusion tests requires $O(MN)$ tests, the cost with the new method is $A\ell_c/V$ of the former. Since the narrowband volume $MA\ell_c$ is usually small compared to the volume of the region of interest, and the fraction is M^{-1} of it, the saving on cost is significant. Be aware that the cost is directly proportional ℓ_c the thickness of the narrowband of boundary cubes, as $\ell_c \rightarrow 0$, the saving can be close to 100%.

Additional costs mainly come from the directional walking to find mixed cells (with a cost of MA/ℓ_c^{d-1} , not related to N , plus an insignificant overhead cost for painting the pure cells, and looping over all points to find the integer parts of their physical coordinates.

3 CONCLUSION

The inclusion test for a massively distributed point set is scaled with a characteristic length of choice, and broken into local inclusion tests in a narrowband of virtual cubes that contains the boundary. By effectively painting the virtual cubes with colors (yellow for boundary, green for interior, blue of exterior), the integer parts of the physical coordinate of a point determines if a point is interior for interior cubes, exterior for exterior cubes, or to be determined for boundary (mixed) cells. Simplified local inclusion tests are performed for only the points that are contained by the mixed cubes. The saving of cost compared to full test is significant. The mathematical technique required for the algorithm is finding intersections of a spatial polygon and a spatial unit square (in 3D) and a line-segment, or intersection of line segments (in 2D) for a boundary represented with planar faces. The wall-sharing relationship of cells is utilized to paint the boundary cubes, and the interior cubes with a loop over the logical distance to the boundary for eliminating inclusion tests of most points (that are contained in pure cubes).

ACKNOWLEDGMENTS

The author appreciates discussions with Jim Reus and Jeff Keasler of the ALE3D group regarding this problem. Their technical knowledge and expertise provided the author with interest and guidance.

REFERENCE

- [1]. Chat with Jim Reus.
- [2]. Chat with Jeff Keasler.
- [3]. Yao, Jin, Directional Walking: A Simple Way to Intersect Arbitrary Geometries, *LLNL-JRNL-400055*, 2007.
- [4]. Yehuda, E. Kalay, "Determining the Spatial Containment of a Point in General Polyhedra", *Computer Graphics and Image Processing* 19, 303-334, 1982.
- [5]. F. Feito, J. C., Torres and A. Urena, "Orientation, Simplicity, and Inclusion Test for Planar Polygons", *Computer & Graphics* vol. 19, No.4, 595-600, 1995.