

UN GENERADOR DE MALLAS PLANAS DE TRIÁNGULOS USANDO QUADTREES Y EL ALGORITMO DE BRESENHAM

A PLANAR TRIANGLE MESH GENERATOR USING QUADTREES AND BRESENHAM'S ALGORITHM

Claudio E. Jouglard y Juan Pablo Romaris

Departamento de Ingeniería Civil, Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Mozart 2300, C1407IVT Buenos Aires, Argentina, claudio.jouglard@frba.utn.edu.ar,

<http://www.frba.utn.edu.ar>

Palabras clave: Generación de mallas, triángulos, árboles cuaternarios, algoritmo de Bresenham.

Resumen. En este trabajo se presenta un generador de mallas planas de triángulos basado en árboles cuaternarios (quadtrees). Los generadores de malla quadtree tienen la ventaja de adaptarse a la densidad de los datos, refinando la malla sólo donde es necesario conduciendo a mallas muy eficientes. Se asume un contorno poligonal y se seleccionan aquellas celdas cuadradas de la grilla que representan al contorno y al interior del dominio. Luego se subdividen estas celdas por alguna de sus diagonales y queda conformada una malla de triángulos. Para seleccionar las celdas del contorno se utiliza el algoritmo de Bresenham. Este algoritmo se utiliza en computación gráfica para rasterizar líneas y existen variantes para diferentes tipos de curvas y también para quadtrees. Este algoritmo también permite seleccionar los nodos de la grilla que mejor aproximan cada recta de contorno.

Keywords: Mesh Generation, triangles, quadtrees, Bresenham's algorithm.

Abstract. In this work, we present a mesh generator for planar triangles based on quadtrees. Quadtree mesh generators have the advantage of adapting to data density, refining the mesh only where necessary and providing highly efficient meshes. A polygonal boundary is assumed, and only the cells representing the boundary and the interior of the domain are selected. These cells are subdivided along one of their diagonals, producing a mesh of triangles. To select the boundary cells, Bresenham's algorithm is used. This algorithm, originally developed for computer graphics to rasterize lines, has variants for different types of curves and also for quadtrees. It selects the nodes of the grid that best approximate the boundary.

1. INTRODUCCIÓN

La generación de mallas mediante *quadtrees* es una técnica jerárquica utilizada en análisis numéricos como el Método de los Elementos Finitos (FEM) (Castellazzi et al., 2015) y el Método de Volúmenes Finitos (FVM) (Bergmann et al., 2022; Gravenkamp y Duczek, 2017). La idea principal consiste en dividir recursivamente el dominio en celdas cuadradas (en 2D) o cúbicas (en 3D, denominados *octrees*) hasta alcanzar un criterio de refinamiento predefinido, como la geometría, gradientes de campo o proximidad a condiciones de contorno.

Un quadtree comienza con un cuadrado inicial el cual es recursivamente subdividido. Durante esta subdivisión pueden aparecer nodos colgantes (*hanging nodes*), que son nodos que están sobre un lado de una celda pero no pertenecen a esa celda. Decimos que la subdivisión es balanceada si los lados de cada celda tienen a lo sumo un nodo colgante (subdivisión 2:1), caso contrario el quadtree se dice *no balanceado*. En la figura 1 tenemos dos quadtrees, uno balanceado a la izquierda y otro no balanceado a la derecha, pues presenta lados con más de un nodo colgante.

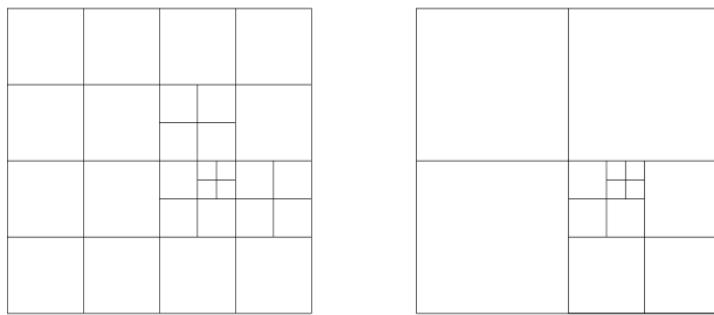


Figura 1: Quadtree balanceado y no balanceado

Para generar una malla de triángulos de una región planar con borde poligonal se puede usar directamente un quadtree cuyas celdas son divididas por sus diagonales y usando patrones específicos de subdivisión en el contorno para generar la malla (Yerry y Shephard, 1984; Shephard et al., 1986; Frey y George, 2008). La mayoría de los autores utilizan quadtrees balanceados, pues producen una transición suave de tamaños de los elementos finitos generados. También se suelen usar los quadtrees como un medio auxiliar para localizar puntos que serán mallados por otros métodos, como el método de Delaunay (Bern et al., 1994; Bern y Eppstein, 1995; Dari, 1994).

En una línea de trabajo similar a la aquí presentada podemos citar el artículo de (Gravenkamp y Duczek, 2017), aunque en ese trabajo se usaron elementos finitos poligonales para aproximar el contorno. En la figura 2 (extraída de ese trabajo) se pueden apreciar a la izquierda los nodos colgantes (*hanging nodes*), y a la derecha el procedimiento de subdivisión aplicado para eliminar los nodos colgantes.

La eliminación de los nodos colgantes es necesaria si queremos generar mallas conformes de elementos finitos, esto es, donde dos elementos adyacentes comparten todos los nodos de sus lados con sus vecinos.

En ocasiones cuando el contorno se extrae con poca definición, como en el caso de imágenes médicas o de escaneo, se lo suele aproximar directamente con las celdas completas (marcadas en gris en la figura 2). Aunque si se requiere aproximar al contorno con precisión se deben particionar todas las celdas intersectadas por la poligonal de contorno (celdas de contorno) y

en general se recurre a la utilización de patrones de subdivisión de celdas. Esta no es una tarea sencilla ya que implica detectar todas las celdas intersectadas (Frey y George, 2008).

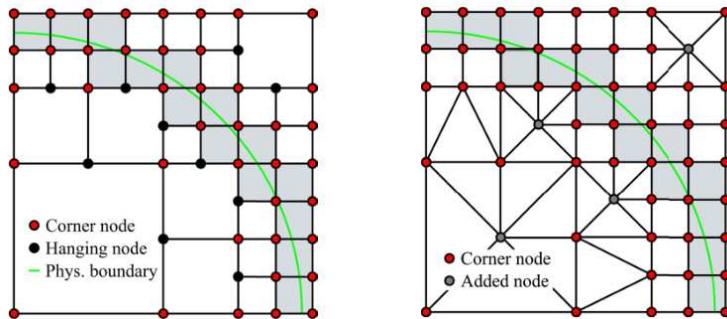


Figura 2: Quadtree aproximación del contorno (Gravenkamp y Duczek, 2017)

En este trabajo presentaremos un generador de mallas de elementos triangulares basado en la subdivisión por las diagonales de las celdas de un quadtree balanceado. Se asumirá que el contorno viene definido por una poligonal y la detección de celdas de contorno se hará utilizando el algoritmo de Bresenham (Bresenham, 1965) y se mostrará como llegar a la malla final conservando todos los detalles del contorno.

2. EL ALGORITMO DE BRESENHAM

Este algoritmo se utiliza para rasterizar un segmento de recta en una grilla de píxeles, esto es, se deben seleccionar los píxeles a iluminar que mejor representen al segmento. En la figura 3 tenemos una grilla de píxeles (centrados en las celdas), donde los píxeles marcados en verde son datos (los extremos del segmento) y los marcados en rojo los selecciona el algoritmo.

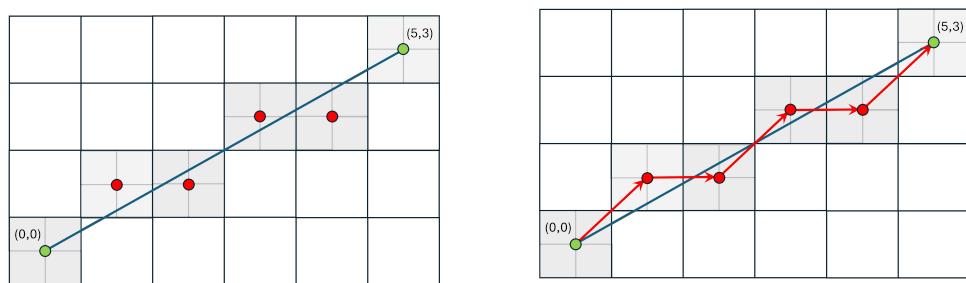


Figura 3: Algoritmo de Bresenham para grillas de celdas.

El funcionamiento es bastante sencillo, se asumen todos los datos enteros, esto es, los extremos del segmento a rasterizar están centrados en celdas y se recorren desde un extremo hacia el otro en la dirección de máximo crecimiento (x en este caso). Se incrementa en uno el índice x de cada celda y se selecciona la celda cuyo centro está más cercano al segmento en vertical. Una vez recorrido todo el segmento tendremos los puntos rojos indicados en la figura 3.

El algoritmo de Bresenham fue ideado por su autor para trazar segmentos de líneas en un plóter digital (Bresenham, 1965), donde la punta de impresión se centraba en celdas y sólo po-

día moverse mediante engranajes en vertical, horizontal o diagonal. A la derecha de la figura 3 tenemos la poligonal trazada que aproxima al segmento de recta. Notemos que además tenemos seleccionadas las celdas que representan el contorno, esto es importante para distinguir las celdas interiores de las exteriores.

2.1. Algoritmo de Bresenham en grillas de píxeles

Para la generación de mallas es mejor utilizar grillas con píxeles centrados en los puntos de la grilla (ya que luego se asociarán a vértices de la malla). Consideraremos ahora que el segmento puede tener extremos reales, esto es, no necesitan coincidir con un centro de celda. Estos extremos se deben asociar al punto de grilla más cercano que marcamos en verde.

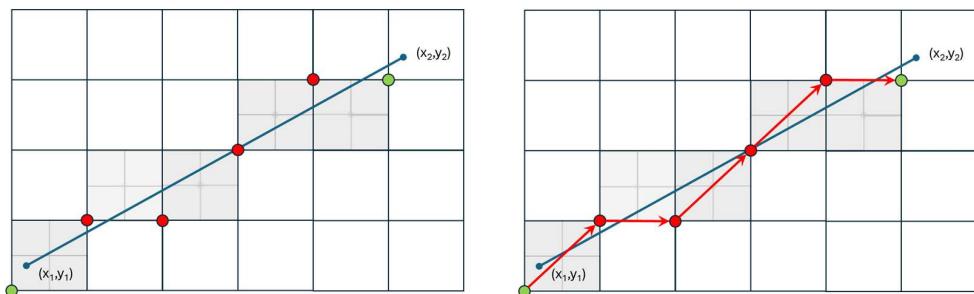


Figura 4: Algoritmo de Bresenham para grillas de puntos.

Ahora debemos encontrar los puntos de la grilla más representativos del segmento, para ello lo recorremos en la dirección de máximo crecimiento (horizontal en este caso) partiendo del punto asociado a un vértice. Luego, incrementamos en uno la coordenada horizontal del punto de grilla y seleccionamos al punto de grilla más cercano al segmento en vertical. Notemos que en este proceso también podemos seleccionar durante el trazado a las celdas más representativas que serán aquellas cuyos centros estén más cercanos a la recta en la vertical.

2.2. Algoritmo de Bresenham en Quadtrees

El algoritmo de Bresenham también se puede adaptar para el trazado de segmentos de recta en grillas de quadtrees ([Hwang y Cheng, 1988](#)), como se muestra en la figura 5.

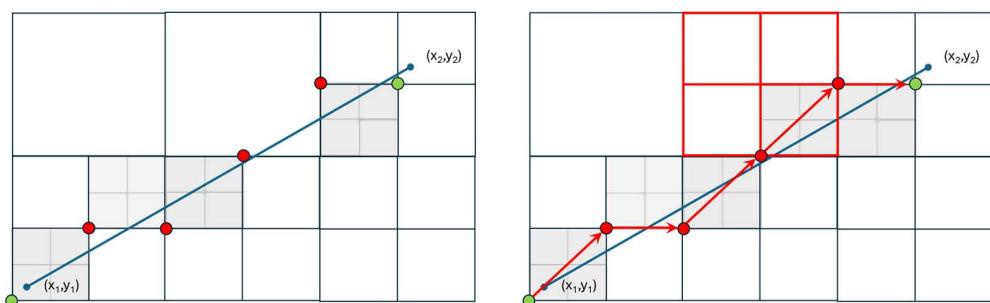


Figura 5: Algoritmo de Bresenham para grillas de quadtrees.

Notemos que en este caso debemos tener la precaución que los puntos de la grilla seleccionados no deben ser *puntos colgantes* porque esto afectará al algoritmo de generación de mallas. Luego si alguno de los puntos seleccionados es colgante debemos subdividir la celda a la que pertenece para sacarlo de esa condición.

3. GENERACIÓN DE MALLA

Para la generación de malla partiremos de un contorno cerrado poligonal encerrado en un quadtree. Asociaremos cada vértice de la poligonal a un punto regular del quadtree, esto es que no sea colgante. Si hay más de un vértice de la poligonal asociado a un mismo punto del quadtree, entonces subdividimos las celdas que contienen estos vértices. Finalmente debemos llegar a un quadtree subdividido donde cada sector de la poligonal de contorno tenga sus vértices asociados a un único punto del quadtree.

Luego, recorreremos la poligonal de contorno en sentido antihorario de manera que a la izquierda tendremos el interior y a la derecha el exterior y usando el algoritmo de Bresenham seleccionamos los puntos de la grilla que mejor representan al contorno. Notemos que dos puntos consecutivos obtenidos por Bresenham representan un sector que, o bien pertenece a un lado de una celda o a una diagonal. Para seleccionar las celdas de contorno usamos el siguiente criterio: 1) si el sector de Bresenham coincide con un lado, la celda de contorno queda a la izquierda del sector o 2) si el sector coincide con una diagonal, la celda se subdivide en dos triángulos siendo la celda de contorno triangular y ubicada a la izquierda del sector.

La idea es que la malla quede formada por las celdas del quadtree contenidas dentro de la poligonal de contorno. Además, en el contorno debemos desplazar las coordenadas de los puntos del quadtree para que las celdas intersectadas queden completamente afuera o adentro.

Finalmente llegaremos a un conjunto de celdas encerradas por la poligonal de contorno. Estas celdas tendrán forma cuadrilateral o triangular. Subdividiendo las celdas cuadrilaterales por una de sus diagonales tendremos un conjunto de celdas triangulares que forman la malla.

A continuación describiremos las principales operaciones necesarias para generar la malla.

3.1. Ajuste en el Contorno

En el contorno debemos desplazar los puntos de la grilla de manera que las celdas de contorno queden completamente adentro del dominio.

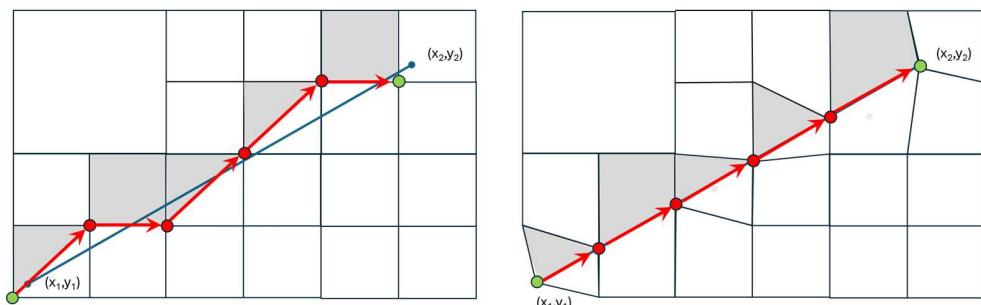


Figura 6: Ajuste de puntos en el contorno.

Una vez hecho el ajuste debemos pasar a la diagonalización de las celdas cuadrilaterales contenidas en el interior del dominio a mallar.

3.2. Diagonalización de las celdas cuadrilaterales

Para diagonalizar las celdas cuadrilaterales debemos elegir en cada una de estas celdas una de las diagonales para subdividirlas. Podemos elegir cualquiera de las diagonales pero adoptaremos la diagonal menor ya que, en general, conduce a triángulos menos distorsionados (Frey y George, 2008).

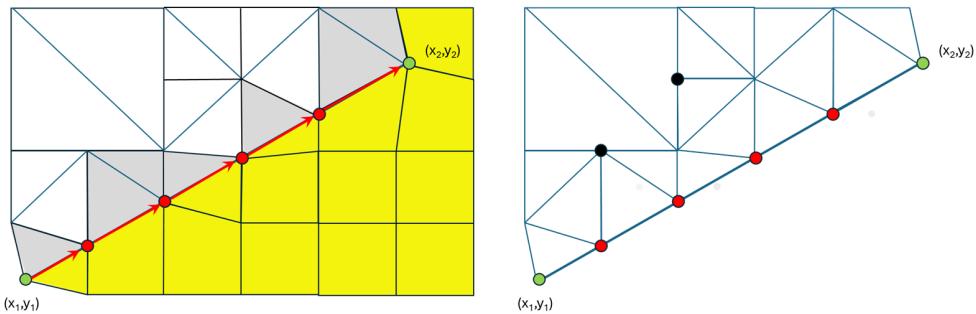


Figura 7: Diagonalización de celdas.

En la figura 7 podemos apreciar a la izquierda los cuadriláteros discretizados y en amarillo las celdas externas. A la derecha podemos ver la malla final con las celdas externas eliminadas.

3.3. Eliminación de nodos colgantes

Notemos que en la malla final mostrada en la figura 7 aparecen nodos colgantes marcados con círculos negros. Si fuese necesario eliminar estos puntos subdividiremos los triángulos que los contienen usando el algoritmo de partición por el lado más largo (Rivara, 1984, 1996).

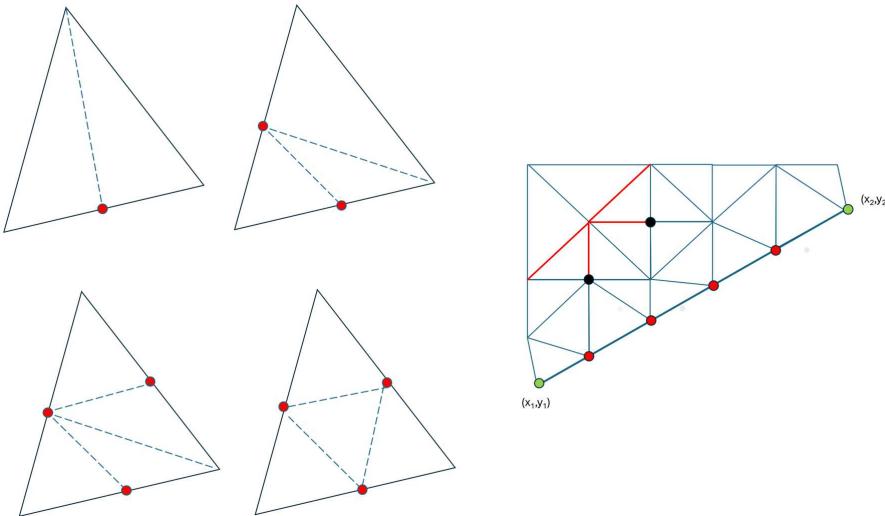


Figura 8: Partición por el lado más largo.

Básicamente en este algoritmo si el nodo colgante se encuentra sobre el lado más largo se une con el vértice opuesto, sino estuviera sobre el lado más largo entonces se partitiona ese lado generando un nuevo nodo colgante y se usa el patrón de subdivisión mostrado en la figura

8 donde también se muestran dos patrones posibles cuando hay tres nodos colgantes. El proceso se repite hasta eliminar todos los nodos colgantes. En la misma figura vemos la aplicación del proceso al ejemplo previo.

4. EJEMPLO DE MALLADO

A continuación mostraremos paso a paso el proceso para obtener una malla de triángulos partiendo de un contorno poligonal encerrado en un cuadrado, como se muestra en la figura 9. A la izquierda tenemos la poligonal de contorno con vértices azules inserta en un quadtree inicial. El siguiente paso es asignar a cada vértice de la poligonal un punto de la grilla que no sea un nodo colgante. Estos puntos son los más cercanos a cada vértice y se muestran en verde.

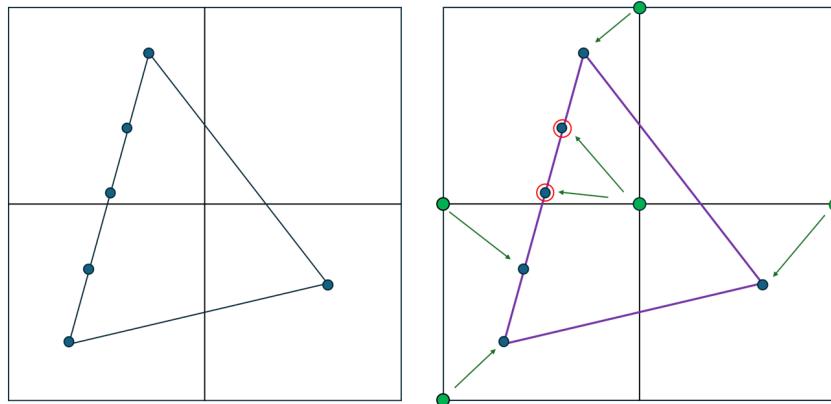


Figura 9: Asignación de puntos (paso 1).

Notemos que existe un conflicto, pues dos vértices comparten el mismo punto de la grilla. Luego se subdividen las celdas que contienen los vértices y se vuelven a asignar puntos de la grilla (paso 2). Se ha asignado un punto colgante a un vértice y se elimina la irregularidad subdividiendo la celda que contiene a dicho punto.

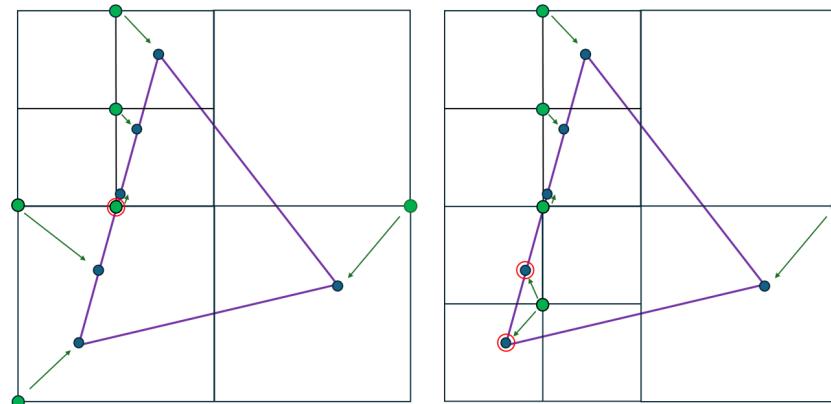


Figura 10: Asignación de puntos (pasos 2 y 3).

Al finalizar el paso 3 anterior se ha generado un conflicto entre dos vértices y deben subdividirse las celdas que los contienen (paso 4). Luego de la subdivisión un punto de asignación

se ha tornado colgante y debe subsanarse esta condición subdividiendo la celda que lo contiene (paso 5).

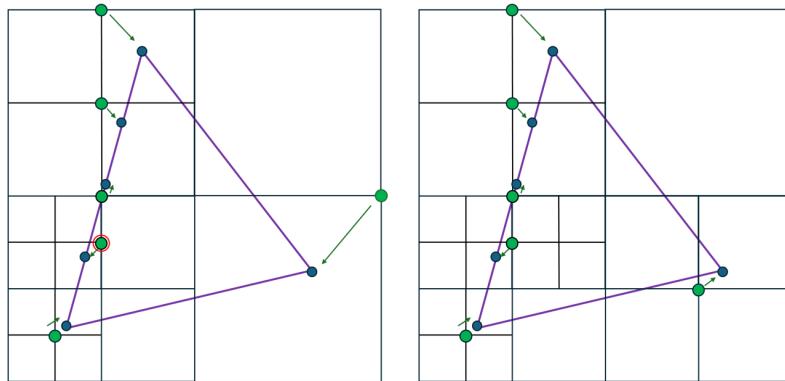


Figura 11: Asignación de puntos (pasos 4 y 5).

Ahora puede comenzar el trazado de los lados usando Bresenham eligiendo los puntos de la grilla más cercanos a la recta. Si algún punto seleccionado es colgante se subdivide la celda que lo contiene. En las figuras 12 y 13 se muestran los puntos obtenidos para cada lado en rojo.

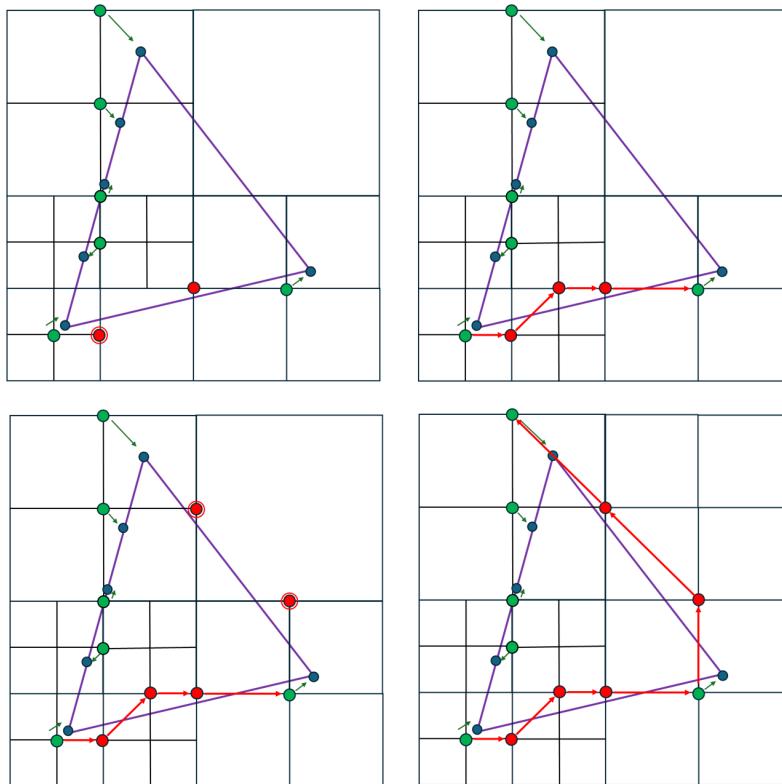


Figura 12: Trazado de primer y segundo lados (pasos 6, 7, 8 y 9).

Luego debe efectuarse el ajuste al contorno de la poligonal de Bresenham (paso 11).

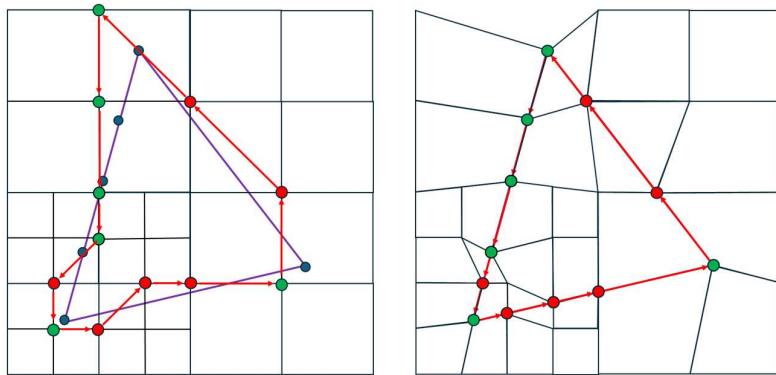


Figura 13: Trazado de tercer lado (paso 10) y ajuste al contorno (paso 11).

Finalmente se realiza la diagonalización de los cuadriláteros interiores (paso 12) y la eliminación de las celdas exteriores (paso 13) como se muestra en la figura 14.

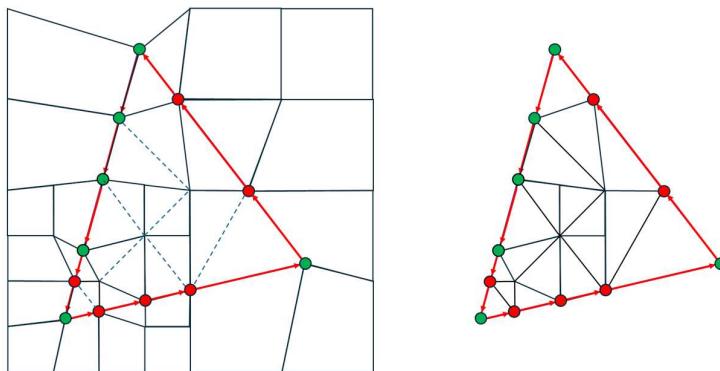


Figura 14: Diagonalización (paso 12) y eliminación de celdas exteriores (paso 13).

Al finalizar este proceso se deben eliminar los nodos colgantes (paso 14) para obtener la malla final.

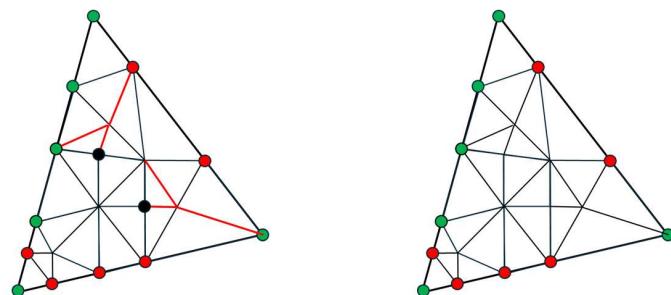


Figura 15: Eliminación de nodos colgantes (paso 14) y malla final.

5. CONCLUSIONES

Se ha presentado un generador de mallas triangulares usando quadtrees y el algoritmo de Bresenham. La metodología es más simple que otras presentadas previamente (Shephard et al., 1986; Frey y George, 2008), ya que no requiere patrones especiales de subdivisión de cuadriláteros en el contorno y sólo utiliza patrones para triángulos para eliminar puntos colgantes. La utilización del algoritmo de Bresenham permite la detección de las celdas de contorno de manera simple y efectiva. Si bien se han asumido contornos poligonales es posible incorporar información de la curvatura en cada segmento mediante NURBS o splines para mallar contornos curvos. Además, podría extenderse la metodología a 3 dimensiones para mallas de tetraedros adaptando las operaciones necesarias.

REFERENCIAS

- Bergmann M., Fondanèche A., y Iollo A. An Eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes. *Journal of Computational Physics*, 471:111647, 2022. <http://doi.org/10.1016/j.jcp.2022.111647>.
- Bern M. y Eppstein D. Mesh generation and optimal triangulation. En *Computing in Euclidean Geometry*, volumen 4 de *Lecture Notes Series on Computing*, páginas 47–123. World Scientific, 2 edición, 1995.
- Bern M., Eppstein D., y Gilbert J. Provably good mesh generation. *Journal of Computer and System Sciences*, 48(3):384–409, 1994. [http://doi.org/10.1016/S0022-0000\(05\)80059-5](http://doi.org/10.1016/S0022-0000(05)80059-5).
- Bresenham J.E. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1), 1965.
- Castellazzi G., D’Altri A.M., Bitelli G., Selvaggi I., y Lambertini A. From Laser Scanning to Finite Element Analysis of Complex Buildings by Using a Semi-Automatic Procedure. *Sensors*, 15(8):18360–18380, 2015. <http://doi.org/10.3390/s150818360>.
- Dari E. *Contribuciones a la Triangulación Automática de Dominios Tridimensionales*. Tesis Doctoral, Instituto Balseiro, Universidad de Cuyo, San Carlos de Bariloche, Argentina, 1994.
- Frey P. y George P.L. *Mesh Generation: Application to Finite Elements*. John Wiley & Sons, 2nd ed. edición, 2008.
- Gravenkamp H. y Duczek S. Automatic image-based analyses using a coupled quadtree-SBFEM/SCM approach. *Computational Mechanics*, 60(4), 2017. <http://doi.org/10.1007/s00466-017-1424-1>.
- Hwang J.P. y Cheng K.Y. Bresenham’s Algorithm in Quadcodes. Research Report TR-88-021, Academia Sinica, Taiwan, 1988.
- Rivara M.C. e Iribarren G. The 4-Triangles Longest-Side Partition of Triangles and Linear Refinement Algorithms. *Mathematics of Computation*, 65(216):1485–1502, 1996.
- Rivara M.C. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *International Journal for Numerical Methods in Engineering*, 20(4):745–756, 1984. <http://doi.org/10.1002/nme.1620200412>.
- Shephard M.S., Yerry M.A., y Baehmann P.L. Automatic mesh generation allowing for efficient a priori and a posteriori mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 55(1):161–180, 1986. [http://doi.org/10.1016/0045-7825\(86\)90090-3](http://doi.org/10.1016/0045-7825(86)90090-3).
- Yerry M.A. y Shephard M.S. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20(11):1965–1990, 1984. <http://doi.org/10.1002/nme.1620201103>.