

# ***Capítulo 4***

## ***Optimización topológica***

### ***4.1.-Introducción***

Un problema clásico de la ingeniería consiste en determinar la configuración geométrica óptima de un cuerpo que minimice o maximice una cierta función de costo al tiempo que satisface las restricciones o condiciones de contorno del problema. La solución de este problema puede ser planteada utilizando dos estrategias: como un problema de optimización de forma o de optimización de la topología.

La optimización de forma consiste en modificar la geometría del dominio preservando su topología, es decir sin crear huecos o cavidades en su interior. Este tipo de análisis es usualmente conocido como Análisis de Sensibilidad al Cambio de Forma y sus bases matemáticas se encuentran bien establecidas. La principal inconveniente del Análisis de Sensibilidad al Cambio de Forma es que sólo permite cambios en la frontera del dominio, lo que limita su campo de aplicación.

Una manera más general de controlar un dominio es mediante modificaciones de su topología, lo que permite obtener la configuración deseada partiendo de una morfología inicial distante de la óptima. Los métodos de homogeneización son posiblemente los más utilizados para la optimización topológica. Estos consisten en caracterizar la topología a través de su densidad, es decir, los huecos se identifican con regiones de densidad nula. De esta forma la solución del programa resulta en una distribución ficticia de material, siendo

necesario en muchos casos utilizar métodos de penalización o filtros para obtener un resultado de utilidad ingenieril.

Un método alternativo de optimización topológica son los basados en Análisis de Sensibilidad Topológica o Derivada Topológica. Esta familia de métodos apunta a resolver las limitaciones de los métodos basados en técnicas de homogeneización, y su idea principal es la evaluación de la sensibilidad de una dada función de costo ante la creación de una cavidad o hueco.

Se presenta en este trabajo una aplicación de la DT para la optimización de problemas de elasticidad en dos dimensiones utilizando el Método de los Elementos de Contorno (MEC). La formulación del problema está basada en resultados recientes reportados por Novotny et al que permiten calcular la DT a partir de resultados de potencial y flujo. El algoritmo es implementado utilizando una formulación directa del MEC. Dado que el MEC prescinde de la discretización del dominio del problema, se evitan los inconvenientes relacionados con variaciones en la densidad del material y los problemas numéricos asociados.

## **4.2.- Reseña histórica**

Hace aproximadamente cuatro décadas Schmidt (1960) propuso una idea revolucionaria que dio origen a una nueva disciplina: los ingenieros, en general, tratan de diseñar objetos o sistemas de coste mínimo que durante su vida útil deben ser capaces de resistir las sollicitaciones máximas que se puedan producir; por tanto, los problemas de diseño (óptimo) podrían plantearse de forma sistemática en términos de problemas de minimización con restricciones, y podrían resolverse mediante técnicas de programación no lineal utilizando ordenadores digitales de alta velocidad.

Desde entonces, la optimización de formas y dimensiones en ingeniería estructural se ha planteado habitualmente mediante formulaciones de mínimo peso, con restricciones no lineales impuestas con el fin de limitar los valores admisibles de los campos de desplazamientos y tensiones.

Sin embargo, desde que Bendsøe y Kikuchi desarrollaron los conceptos básicos en 1988, los problemas de optimización topológica se han planteado tradicionalmente mediante formulaciones de máxima rigidez. Con este tipo de planteamientos se pretende distribuir una cantidad predeterminada de material en un recinto de forma que se maximice la rigidez (se minimice la energía de deformación) de la pieza resultante para un determinado estado de carga .

De esta forma se evita tener que trabajar con numerosas restricciones altamente no lineales, habida cuenta del elevado número de variables de diseño que es consustancial a los problemas de optimización topológica. A cambio, no es posible contemplar múltiples estados de carga, y las formulaciones de máxima rigidez conducen —en principio— a problemas intrínsecamente mal planteados, cuyas soluciones oscilan indefinidamente al refinar la discretización.

### 4.3.- La derivada topológica para problemas elásticos

El concepto original de la DT está relacionado con la sensibilidad de una dada función costo  $\psi(\Omega)$  cuando la topología del dominio  $\Omega$  es alterado. El caso más simple consiste en crear un hueco de radio  $\varepsilon$  en el dominio del problema. Sin embargo este concepto es difícil de trabajar y evaluar porque no es posible establecer un isomorfismo entre los dos dominios con distintas topologías (con y sin hueco).

Feijóo et al (2002) resolvieron la dificultad anterior al modificar la definición de la DT utilizando la idea de que la creación de un hueco puede ser pensada como la perturbación de un hueco preexistente cuyo radio  $\varepsilon$  tiende a cero (ver Figura 1). De esta forma ambas topologías son similares y es posible establecer un mapeo entre ellas. De acuerdo a la nueva definición la expresión de la DT resulta:

$$D_T(x) = \lim_{\substack{\varepsilon \rightarrow 0 \\ \delta\varepsilon \rightarrow 0}} \frac{\psi(\Omega_{\varepsilon+\delta\varepsilon}) - \psi(\Omega_\varepsilon)}{f(\varepsilon + \delta\varepsilon) - f(\varepsilon)} \quad (1)$$

donde  $\delta\varepsilon$  es una pequeña perturbación del radio del agujero y  $f$  una función regularizadora negativa, tal que  $f(\varepsilon) > 0$  cuando  $\varepsilon > 0$ . La ventaja de la expresión (1) es que todas las herramientas matemáticas desarrolladas para el Análisis de Sensibilidad al Cambio de Forma pueden ser utilizadas para el cálculo de la DT.

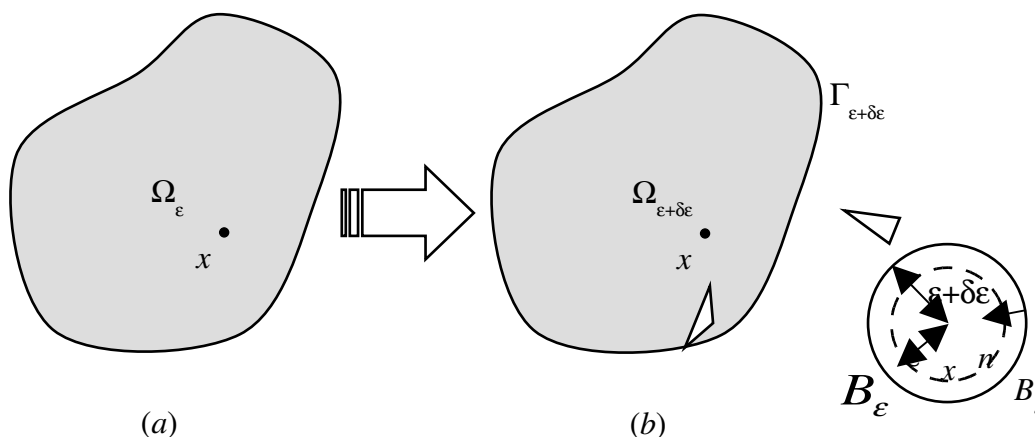


Figura 4.1: Versión modificada de la DT:  
 (a) Dominio original, (b) Dominio perturbado.

El interés de nuestro trabajo es la evaluación de problemas elásticos planos por lo que el problema consiste 0 sobre fundamentalmente en encontrar una solución  $\mathbf{u}_\varepsilon \mid \text{div} \boldsymbol{\sigma}_\varepsilon = \mathbf{b}$  en el dominio  $\Omega_\varepsilon$  sujeta a condiciones de contorno de desplazamientos,  $\mathbf{u}_\varepsilon = \bar{\mathbf{u}}$  sobre la porción  $\Gamma_u$ ,  $\boldsymbol{\sigma}_\varepsilon \cdot \mathbf{n} = \bar{\mathbf{u}}$  sobre la porción  $\Gamma_t$  y  $\boldsymbol{\sigma}_\varepsilon \cdot \mathbf{n} = 0$  sobre  $\Gamma_\varepsilon$ .

Dada la función de costo como la función de energía total de deformación, podríamos escribirla de la siguiente forma:

$$\Psi(\mathbf{u}_\tau) = \frac{1}{2} \int_{\Omega} \mathbf{C} \nabla_{\tau} \mathbf{u}_\tau \cdot \nabla_{\tau} \mathbf{u}_\tau d\Omega - \int_{\Omega} \mathbf{b} \cdot \mathbf{u}_\tau d\Omega - \int_{\Gamma} \mathbf{q} \cdot \mathbf{u}_\tau d\Gamma_{\tau} = \quad (2)$$

$$\frac{1}{2} \mathbf{a}_{\tau}(\mathbf{u}_{\tau}, \mathbf{u}_{\tau}) - \mathbf{l}_{\tau}(\mathbf{u}_{\tau})$$

donde  $\tau$  es el parámetro de perturbación asociado a la velocidad de cambio de forma. La sensibilidad de la función de costo con respecto a  $\tau$  puede ser obtenida da la derivada de Gateaux de la configuración perturbada dada por

$$\frac{d}{d\tau} \Psi(\Omega_{\tau})_{\tau=0} = \lim_{\tau \rightarrow 0} \frac{\Psi(\Omega_{\tau}) - \Psi(\Omega_{\tau=0})}{\tau} = 0 \quad (3)$$

Tras un intensivo trabajo de análisis, la derivada topológica resulta, en ausencia de caras internas:

$$D_T(\hat{\mathbf{x}}) = - \lim_{\varepsilon \rightarrow 0} \frac{1}{f'(\varepsilon)} \int_{\Gamma_\varepsilon} \frac{1}{2\rho E} \boldsymbol{\sigma}_\varepsilon^{tt} d\Gamma_\varepsilon \quad (4)$$

Utilizando un análisis asintótico de la solución  $\mathbf{u}_\varepsilon$ , se llega a la siguiente expresión:

$$D_T(\hat{\mathbf{x}}) = \frac{2}{1+\nu} \boldsymbol{\sigma} \cdot \boldsymbol{\varepsilon} + \frac{3\nu-1}{2(1-\nu^2)} \text{tr} \boldsymbol{\sigma} \text{tr} \boldsymbol{\varepsilon} \quad (5)$$

que puede ser particularizada para el problema de deformación plana como:

$$D_T(\hat{\mathbf{x}}) = \frac{2}{(1+\nu)(1-2\nu)} \boldsymbol{\sigma} \cdot \boldsymbol{\varepsilon} + \frac{(1-\nu)(4\nu-1)}{2(1-2\nu)} \text{tr} \boldsymbol{\sigma} \text{tr} \boldsymbol{\varepsilon} \quad (6)$$

**4.4.-**

## Implementación de la DT

Una vez obtenido un programa de BEM de gran capacidad de calculo y velocidad, procedemos a darle una aplicación. Se utilizará el método de optimización topológica antes mencionado para conseguir estructuras lo mas livianas posibles eliminando iterativamente el material menos importante del dominio. Si es un punto de la frontera se quitará y se modificará la frontera mientras que si es un punto interno en su lugar aparecerá un agujero nuevo. Podemos observar estas posibilidades en la figura 4.2:

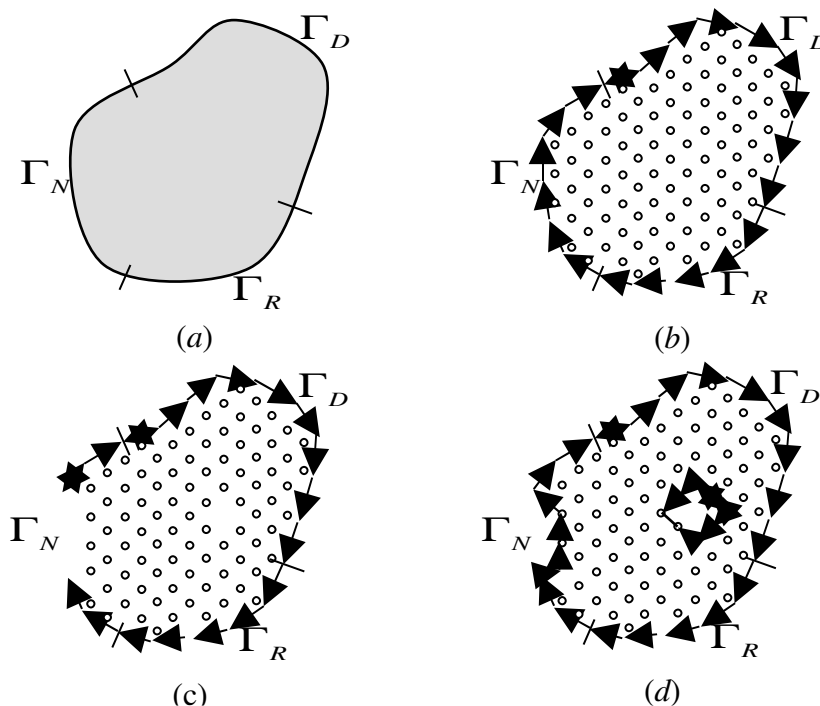


Figura 4.2: Implementación de la DT: (a) Dominio y condiciones de contorno,

Con el programa "DETOPO"(Anexo I), diseñado en Octave, leemos el archivo de entrada, mandamos hacer los cálculos al subprograma paralelo SERBAPA, y de la salida de este elegimos los puntos prescindibles cuyo valor de derivada topológica sea menor. El número de puntos a eliminar por iteración es una variable que en nuestro caso dependerá del área a eliminar por paso y del área total a eliminar o número de iteraciones máximas que queremos realizar. Por lo general un valor del 1% del área total por iteración parece una cantidad lógica por lo que decidimos quitar esa cantidad.

Una vez elegidos los puntos a eliminar se crea un archivo de datos con todas las coordenadas de puntos del contorno y puntos internos. Esta “nube” de puntos es leída por el programa Meshsuite, que hace una malla de triángulos con dichas coordenadas y diferencia los puntos que quedan en el contorno de los internos. Dibujamos dicha geometría resultante y pueden aparecer dos tipos de problemas:

- Que dos contornos compartan un nodo, para lo cual se implementó un algoritmo que eliminaba el punto común y volvía a calcular la nube sin dicho punto. De nuevo llamábamos al Meshsuite y comprobábamos si había errores. Un ejemplo de este algoritmo podemos verlo en la figura 4.3 donde se eliminan dos puntos comunes y se dibuja de nuevo el contorno.

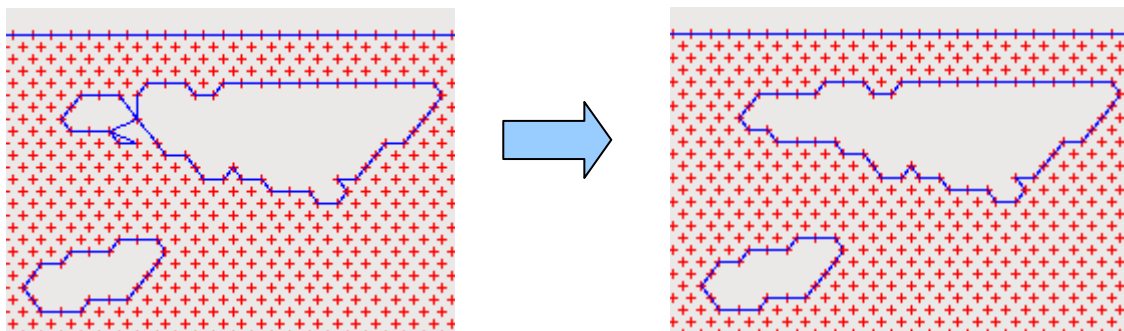


Figura 4.3

- Que queden contornos no conectados con el contorno principal para lo cual se creó otro algoritmo para resolverlo. Este caso tenía el problema de que existían agujeros en el contorno permitidos con la única diferencia de estos “trozos sueltos” en el sentido de las conectividades de su frontera. El mecanismo de resolución es el siguiente: Se supone que es un agujero por lo que se busca un punto interno a través de la normal del primer elemento. A partir de este punto se calculan los ángulos que forman todos los elementos con él y si la suma es de  $360^\circ$ , es que nos encontramos en el interior. En caso contrario, se trata de un trozo suelto, por lo que habrá que eliminarlo. Como en principio no sabemos si habrá puntos internos y el calcular si los hay es muy costoso computacionalmente, la alternativa elegida es eliminar los puntos que pertenecen a dicha frontera, y recalcularse de nuevo la nube de puntos. Con esta técnica, podemos decir que se “liman” los trozos sueltos

hasta que desaparecen por completo. Un ejemplo de eliminación de estos trozos sueltos sería el caso representado en la figura 4.4.

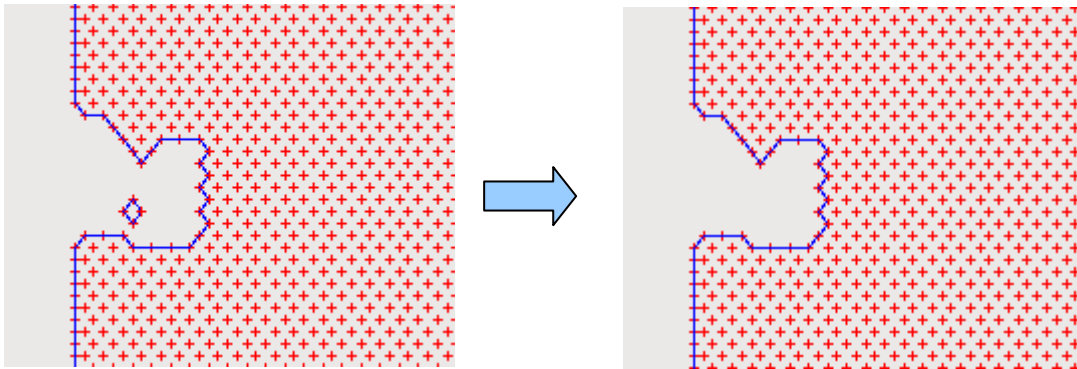


Figura 4.4

En el caso de que se den las dos patologías simultáneamente, tendrá preferencia el algoritmo de resolución de puntos dobles sobre el de eliminación de trozos sueltos. La secuencia resultante sería como la de la figura 4.5

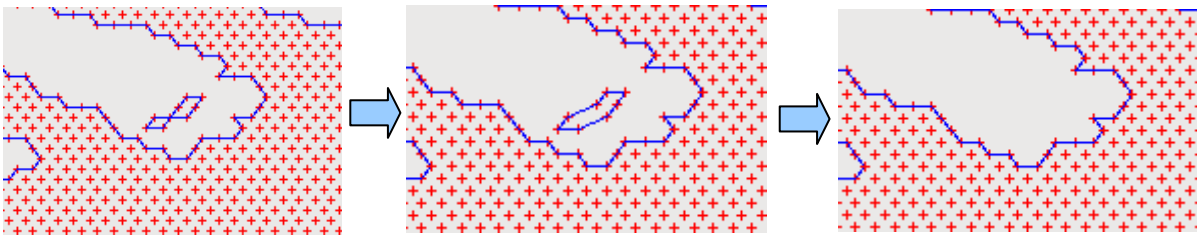


Figura 4.5

Una vez que hemos solucionado cualquiera de estas patologías posibles, procedemos a calcular el área total quitada para ver si se cumple la restricción de área mínima del problema o seguimos con una nueva iteración. En caso de continuar, creamos un archivo de entrada para la próxima iteración con las condiciones de contorno que almacenamos al principio del programa. Estas condiciones están almacenadas por coordenadas por lo que no importa que cambien las numeraciones de los puntos del contorno. Continuamos de esta forma hasta llegar al punto deseado pudiendo comprobar en cada iteración el dibujo de la estructura resultante.

Existe la posibilidad de dibujar la estructura inicial y la deformada pero por tardar mucho tiempo el ploteador de Octave recomendamos que se haga luego en un reprocesamiento de datos.

## 4.6.- Ejemplos de problemas de derivada topológica

- Ejemplo de una placa de 200x100 m apoyado en los dos vértices inferiores y sometido a una fuerza distribuida a lo largo del eje horizontal inferior. (Problema del puente). En la figura 4.6 puede observarse un diagrama del problema inicial y a continuación se presentan algunos pasos intermedios y el resultado final. Se programó con 20000 puntos y quitando una cantidad de 0,5% por iteración.

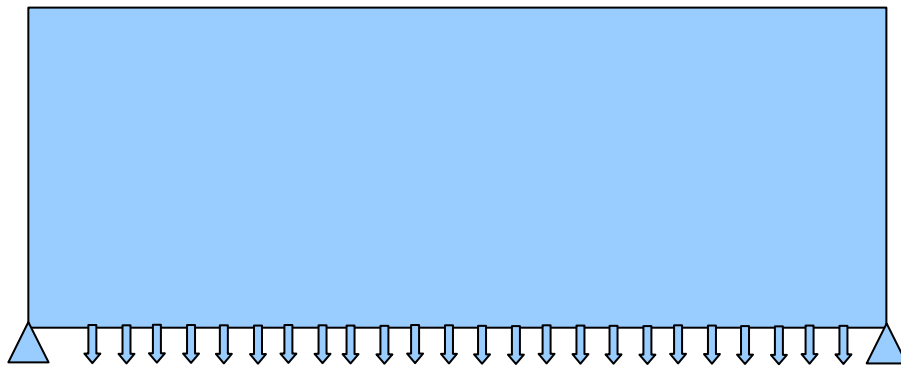


Figura 4.6

A continuación se presentan gráficamente algunas de las etapas del proceso según el número de iteraciones realizadas:

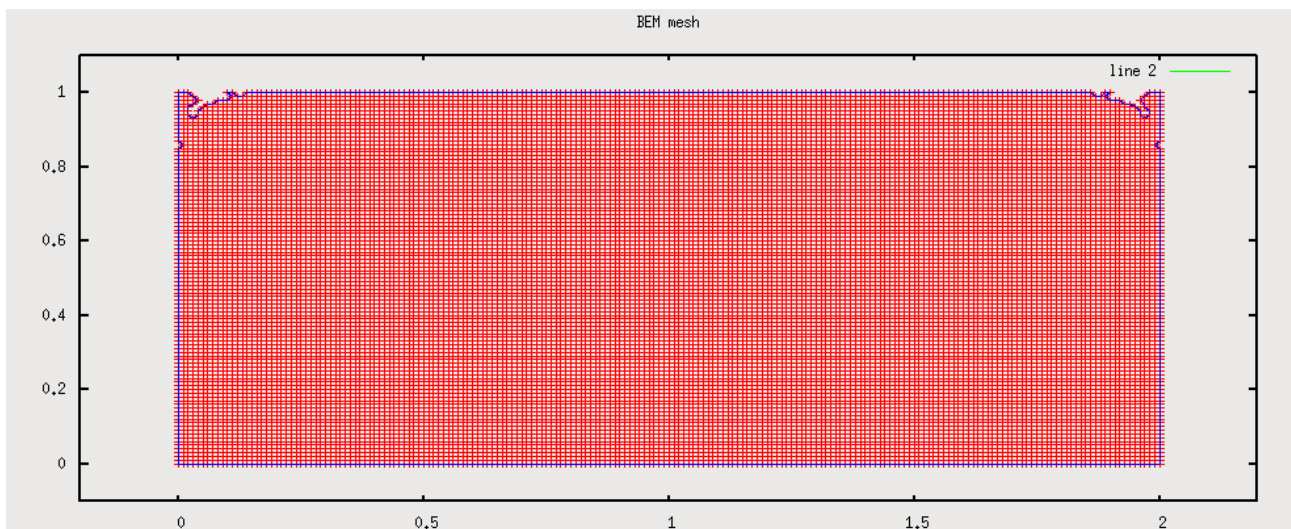


Figura 4.7: Iteración nº 1

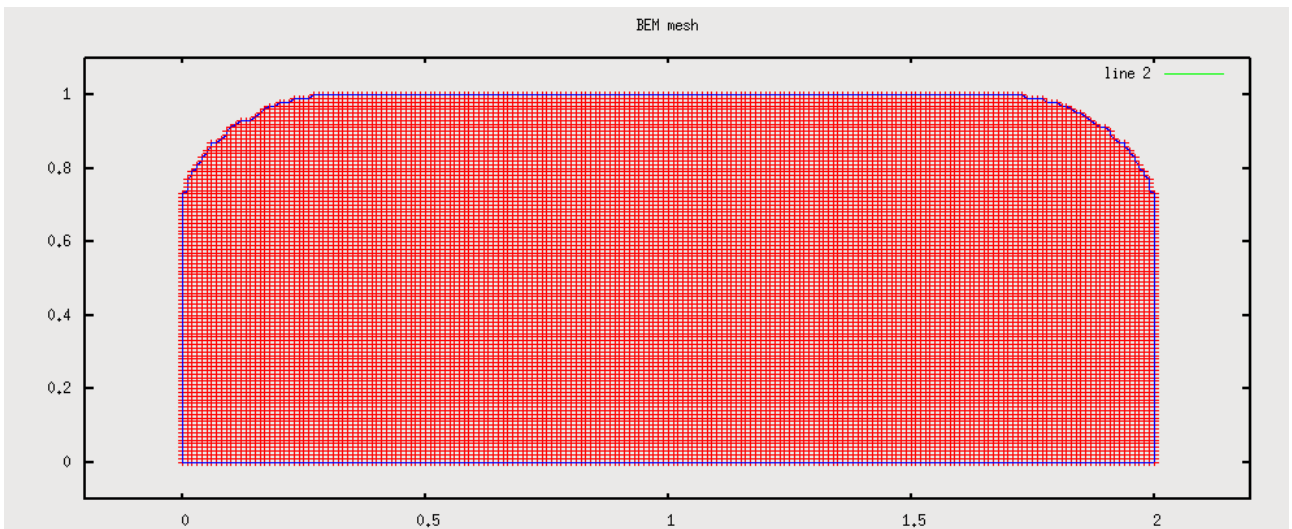


Figura 4.8: Iteración nº 20

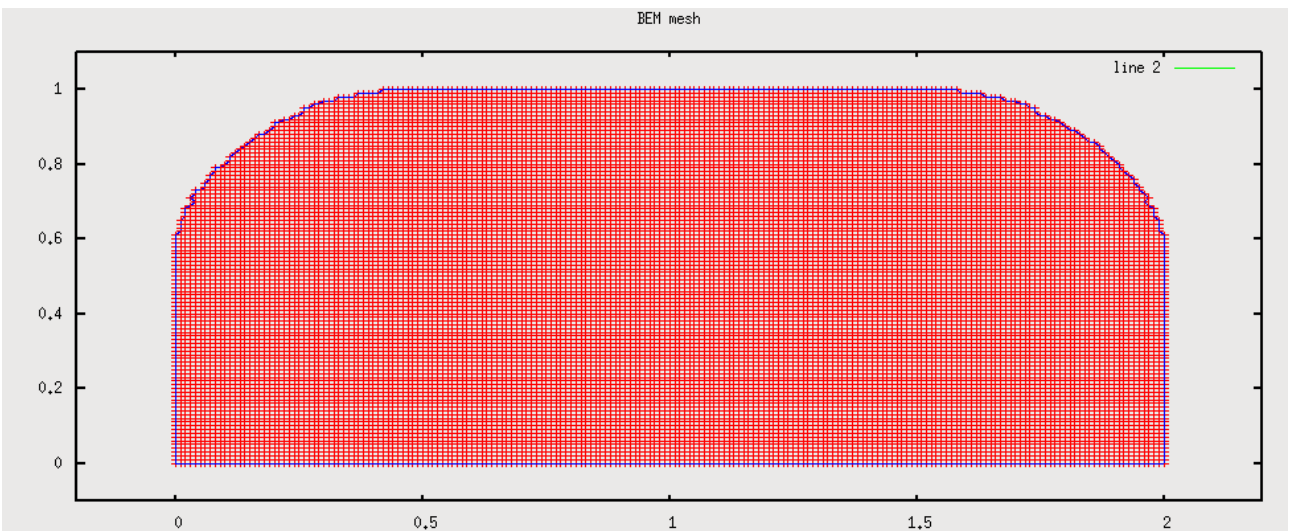


Figura 4.9: Iteración nº 40

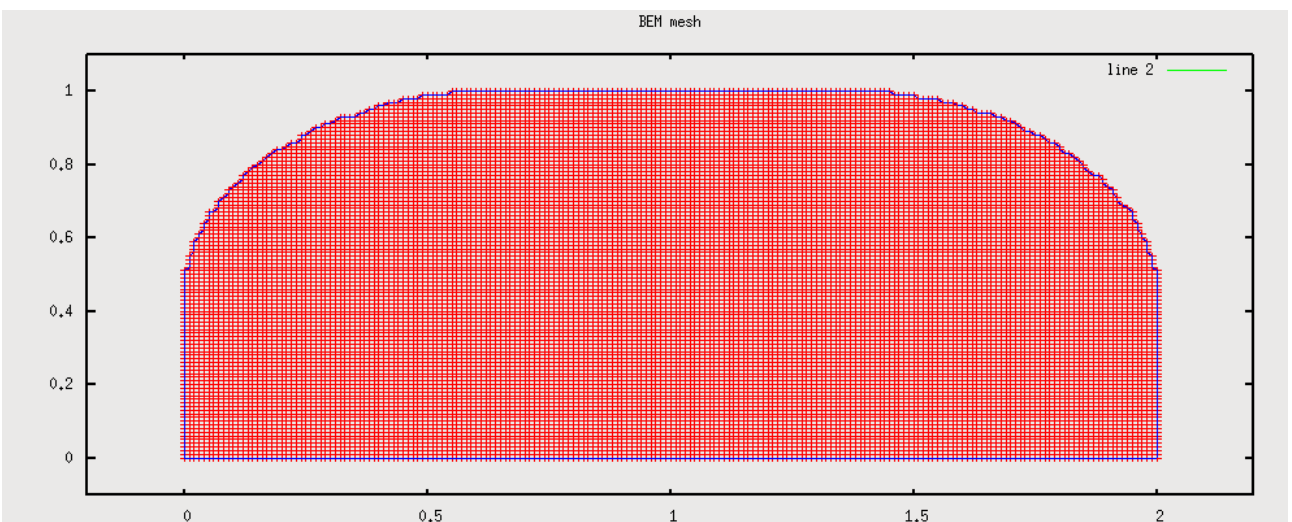


Figura 4.10: Iteración nº 60

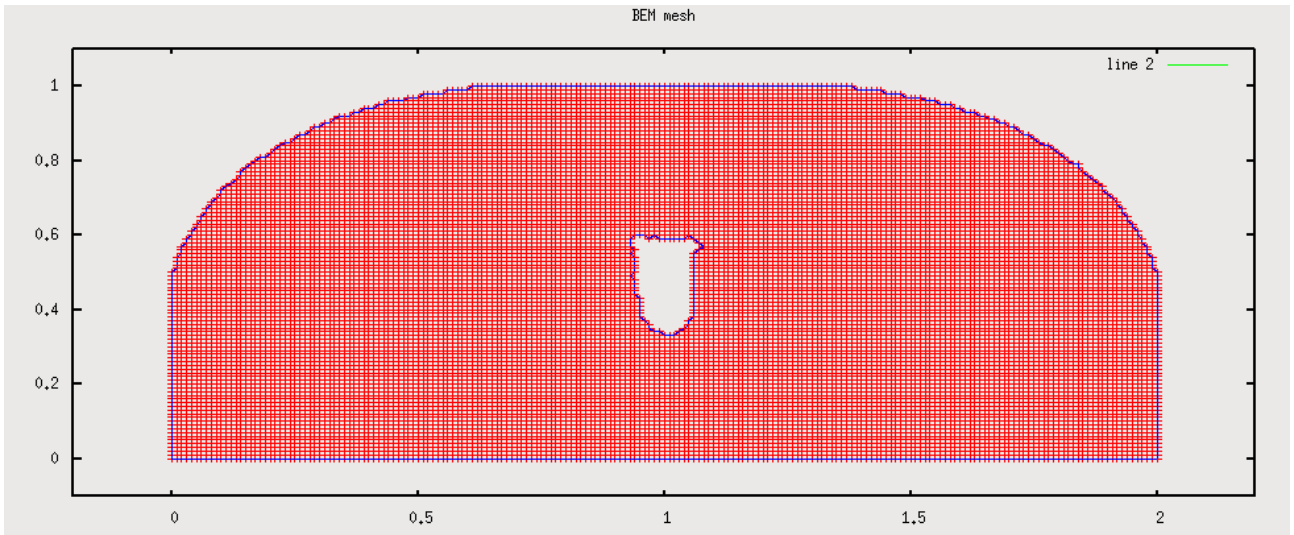


Figura 4.11: Iteración nº 80

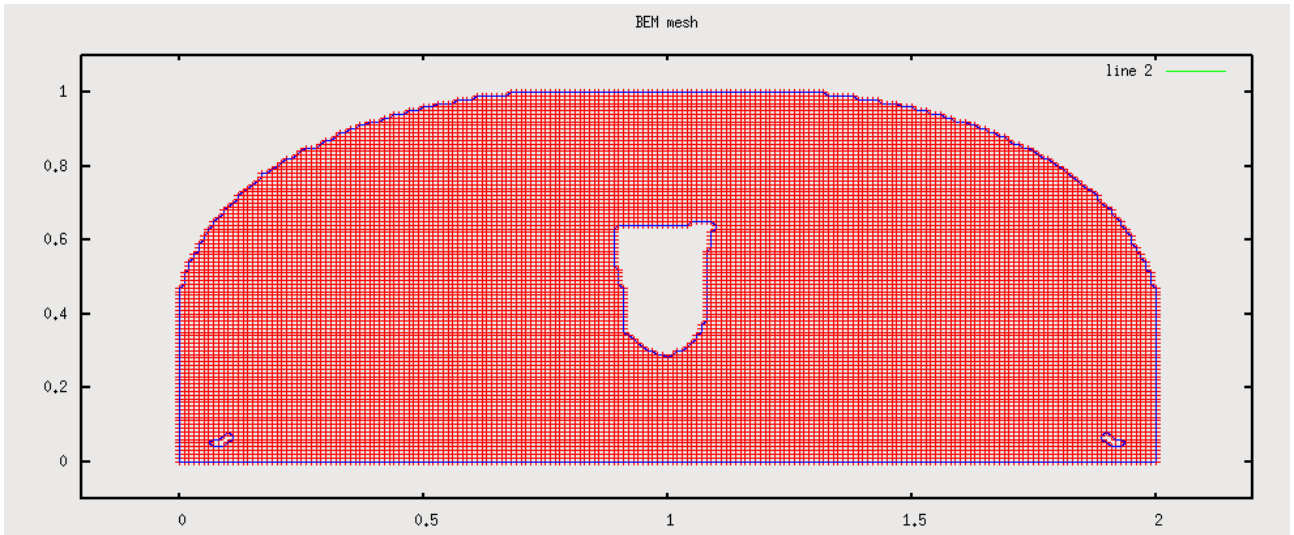


Figura 4.12: Iteración nº 100

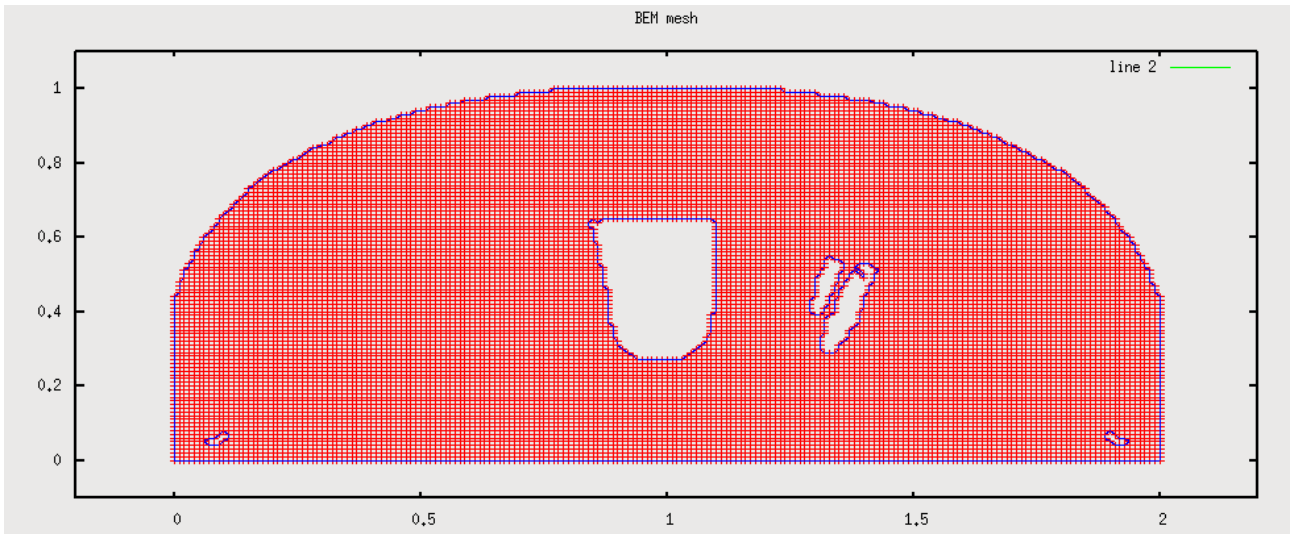


Figura 4.13: Iteración nº 125

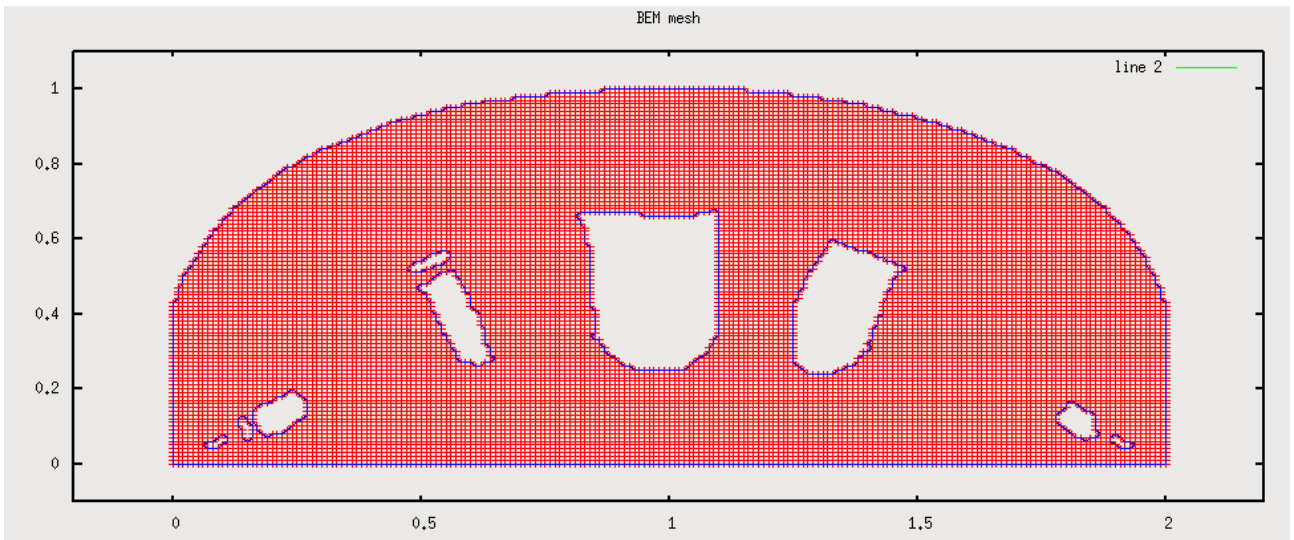


Figura 4.14: Iteración nº 150

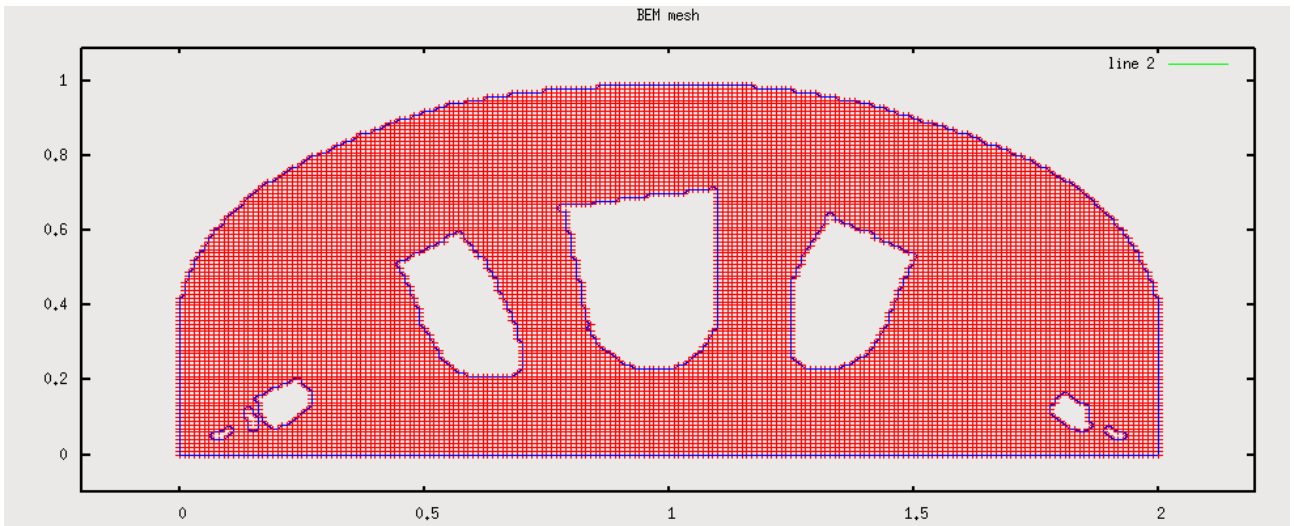


Figura 4.15: Iteración nº 175

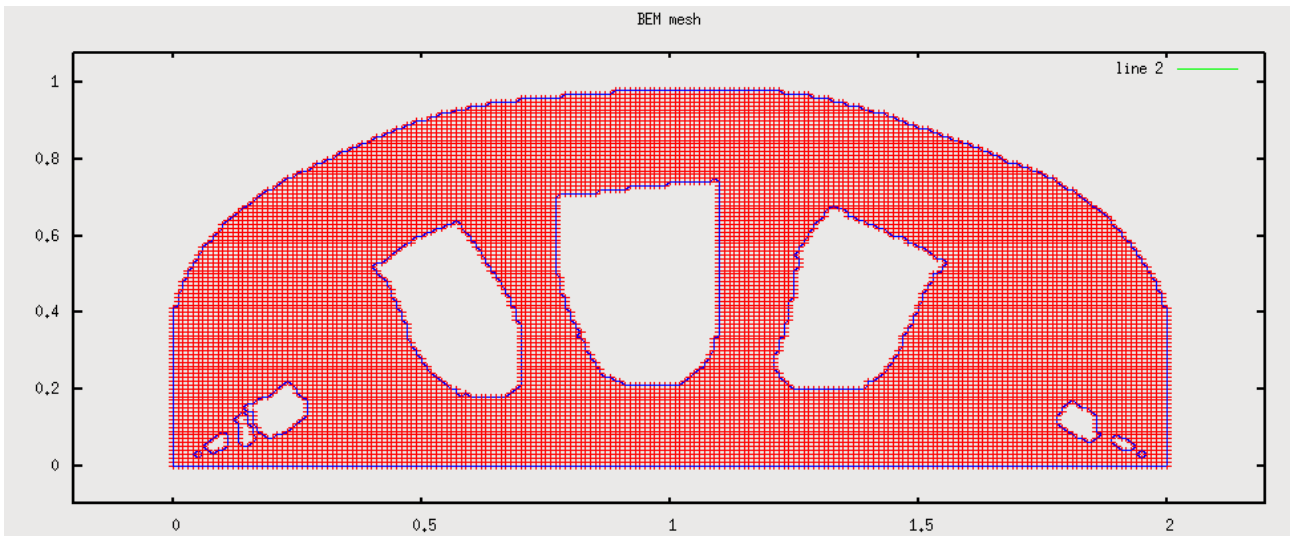


Figura 4.16: Iteración nº 200

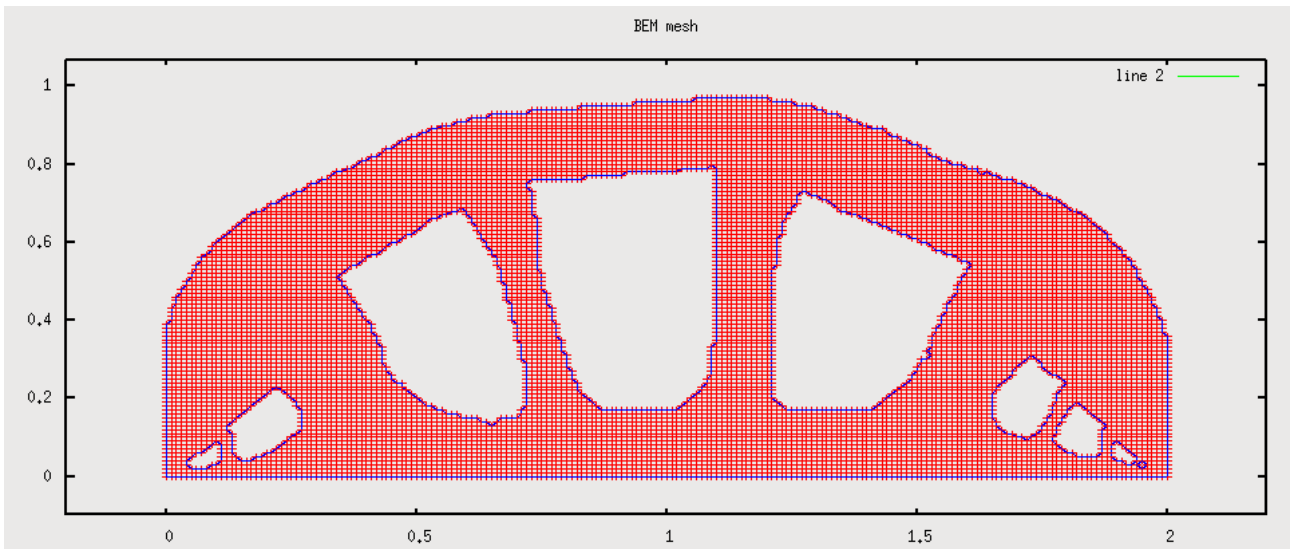


Figura 4.17: Iteración nº 250

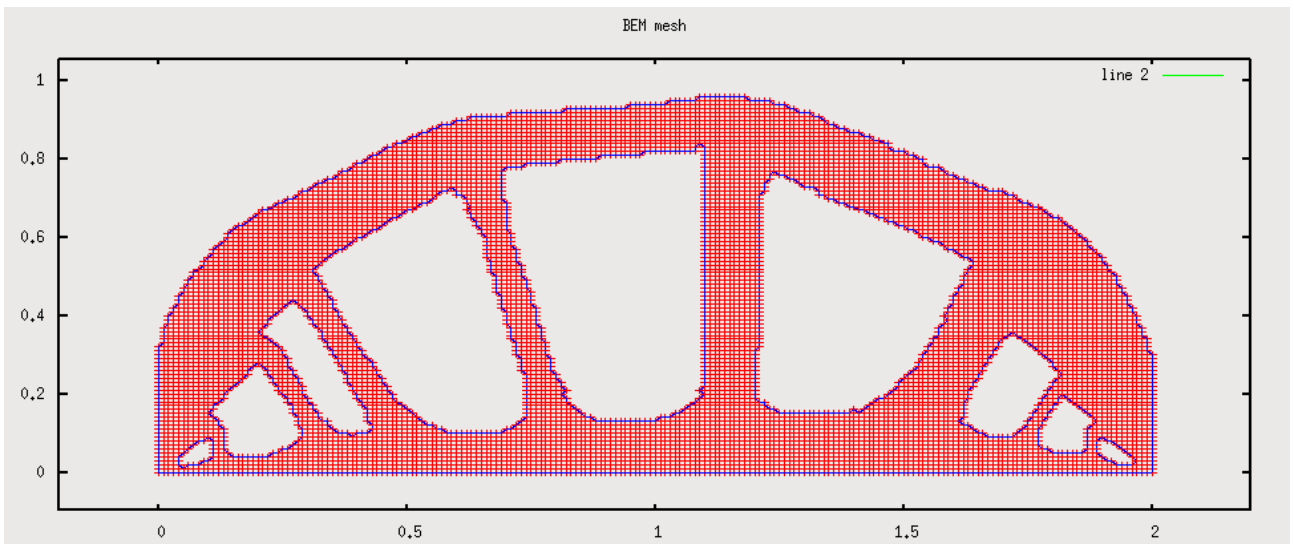


Figura 4.18: Iteración nº 300

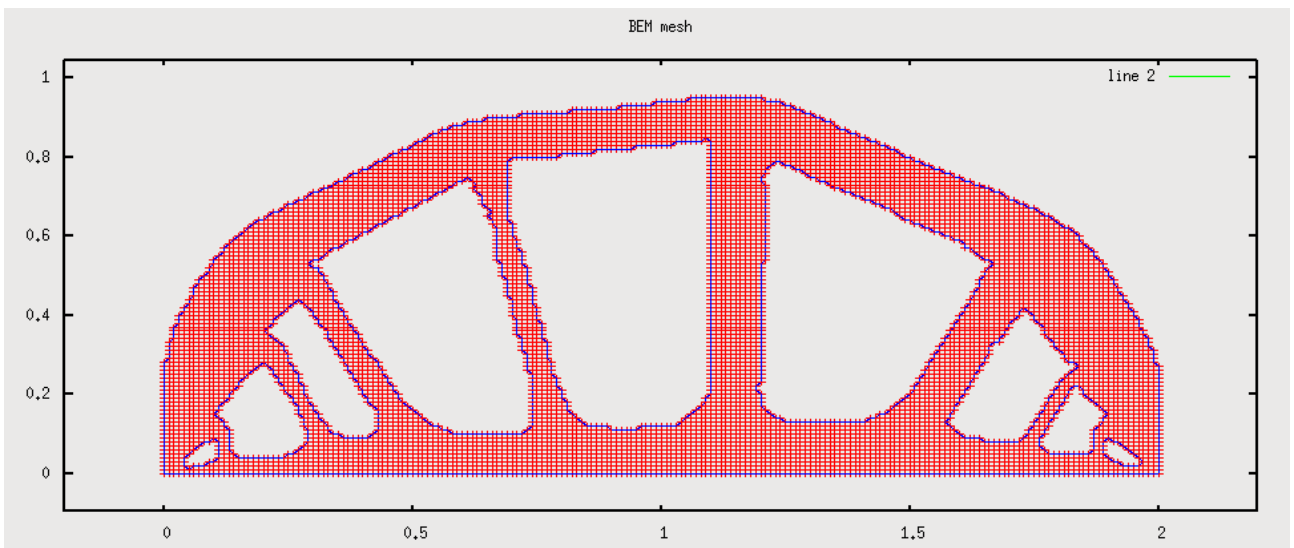


Figura 4.19: Iteración nº 325

Podemos observar como el algoritmo comienza eliminando las esquinas superiores ya que son los puntos con menos tensión debido a que son salientes descargados. Una vez que la geometría toma una forma redondeada en dichas esquinas, se abre un agujero en la parte central de la placa, aproximadamente en la iteración número 70, aumenta dicho agujero hasta que el algoritmo encuentra otro lugar para hacer otro agujero. Con el tiempo se va formando una estructura consistente en un semianillo que está conectado con la base a través de tirantes que soportan el “peso” de la estructura.

Este problema es un ejemplo clásico dentro de la optimización topológica y podemos observar la gran similitud con resoluciones de otros autores como M.P.Bendsoe y O.Sigmund en “Topology Optimization, Theory, Methods and Applications”.

- Ejemplo de una placa de 100x100 sujeta en su lado izquierdo y sometida a una fuerza cortante en el punto medio de su lado derecho.

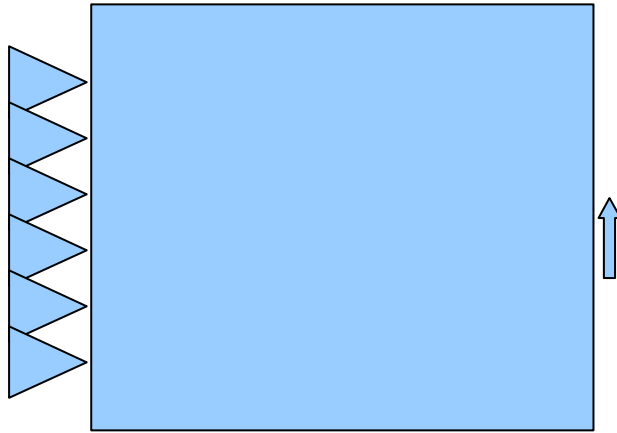


Figura 4.20

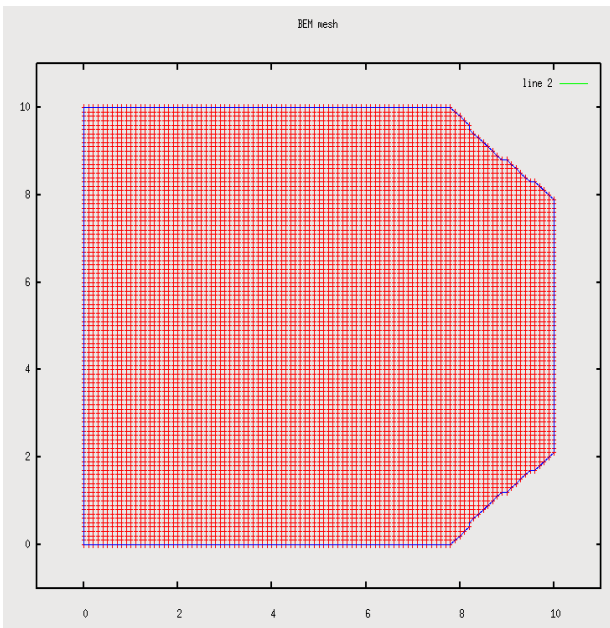


Figura 4.21: Iteración nº 1

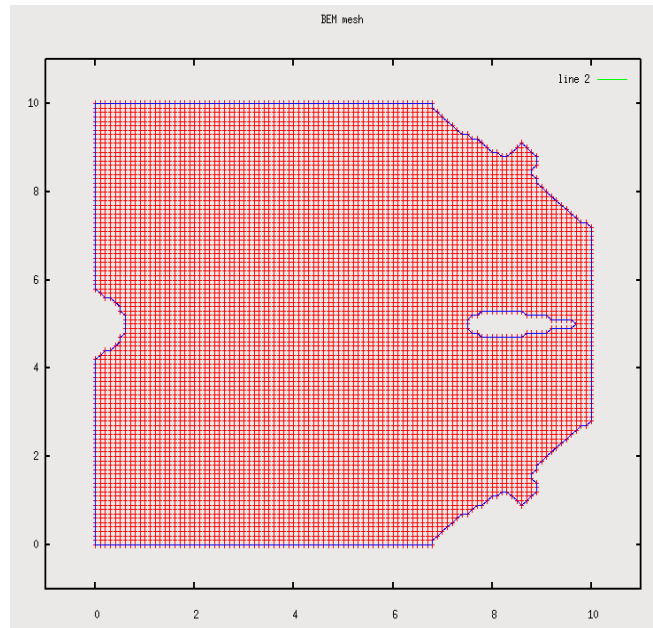


Figura 4.22: Iteración nº 2

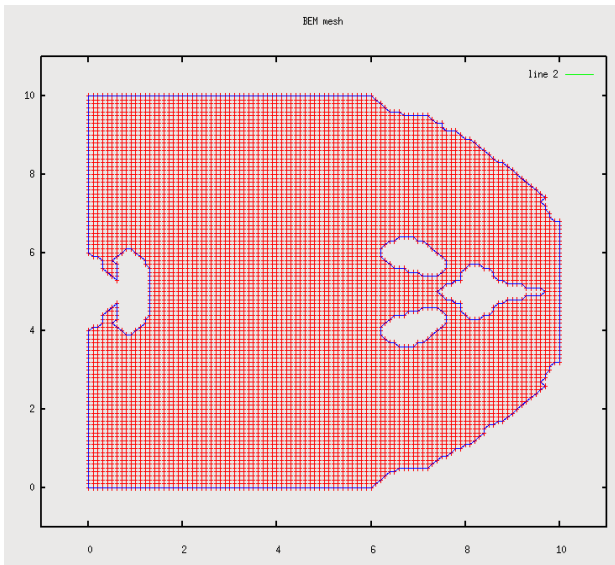


Figura 4.23: Iteración nº 3

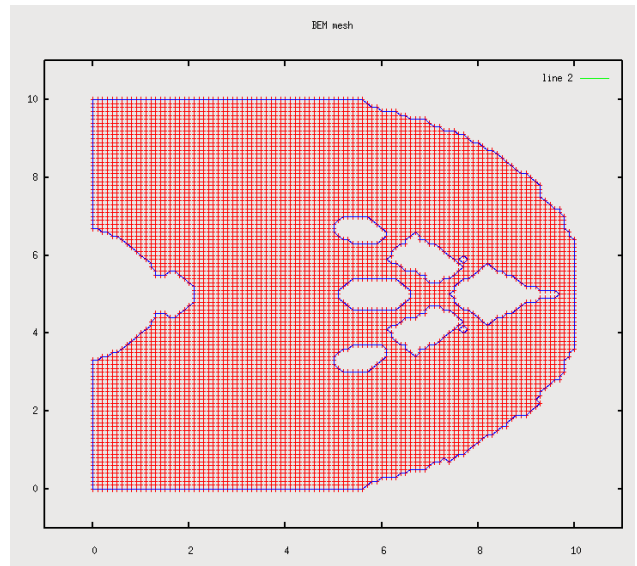


Figura 4.24: Iteración nº 4

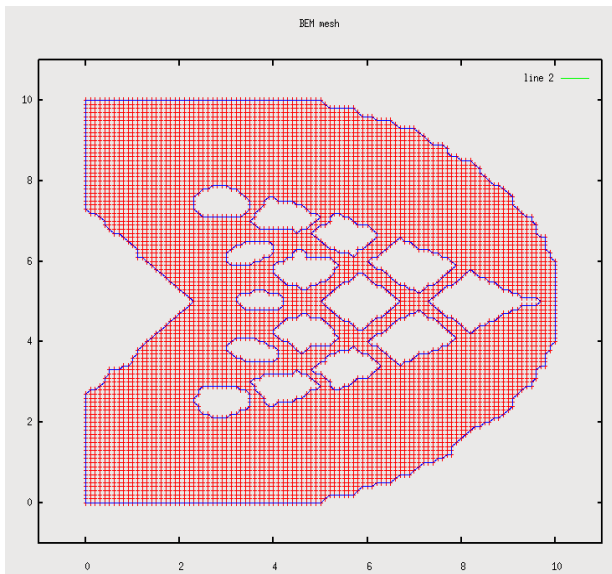


Figura 4.25: Iteración nº 5

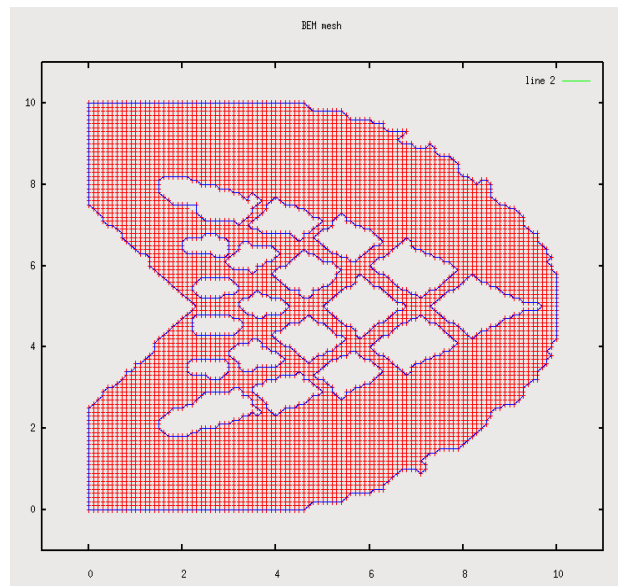


Figura 4.26: Iteración nº 6

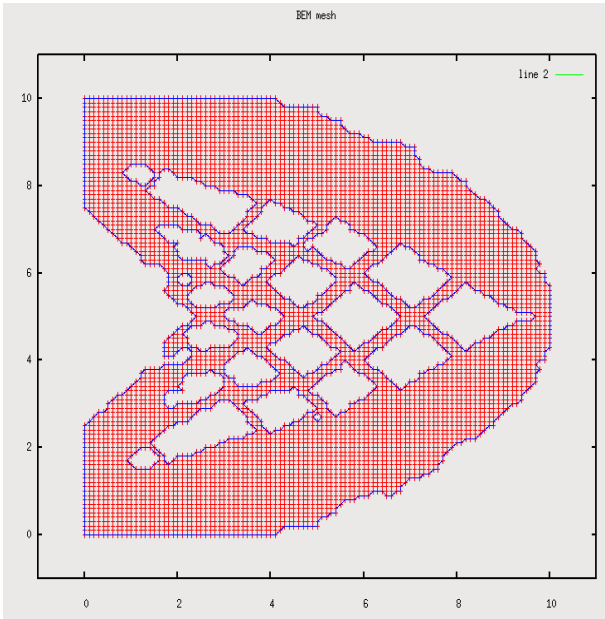


Figura 4.27: Iteración nº 7

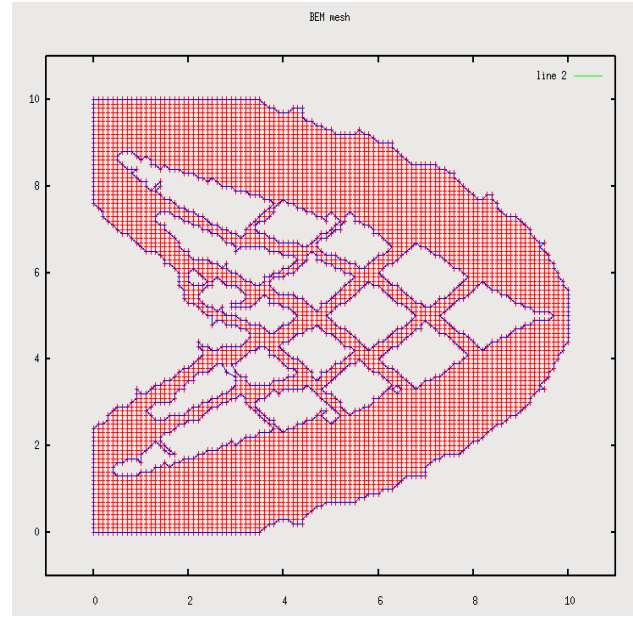


Figura 4.28: Iteración nº 8

En este ejemplo el algoritmo también quita de las esquinas, en este caso de la derecha y posteriormente crea una especie de rejilla, o celosía de barras, para conseguir la máxima resistencia con el menor área. Se hizo quitando muchos puntos por iteración ya que si se quitaba de poco en poco simplemente quedaban dos barras diagonales como se muestra a continuación:

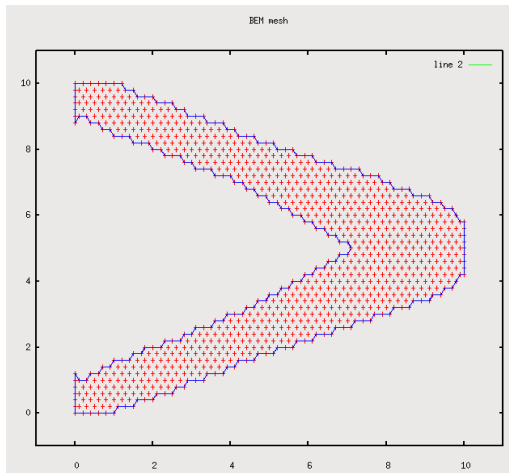


Figura 4.29

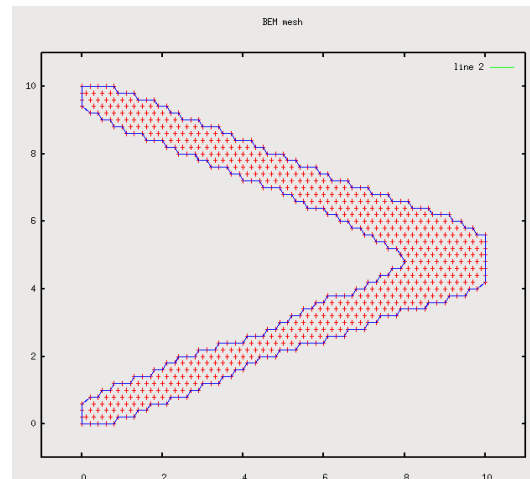


Figura 4.30

Vemos como quedaría en el caso de quitar muy poco puntos por cada iteración. Comenzaría el algoritmo a eliminar los puntos de las esquinas y de la parte central del lado del que está sujeta la pieza.

- Ejemplo de una placa de 100x100 sujeta en su lado izquierdo y sometida a una fuerza vertical en la esquina superior derecho.

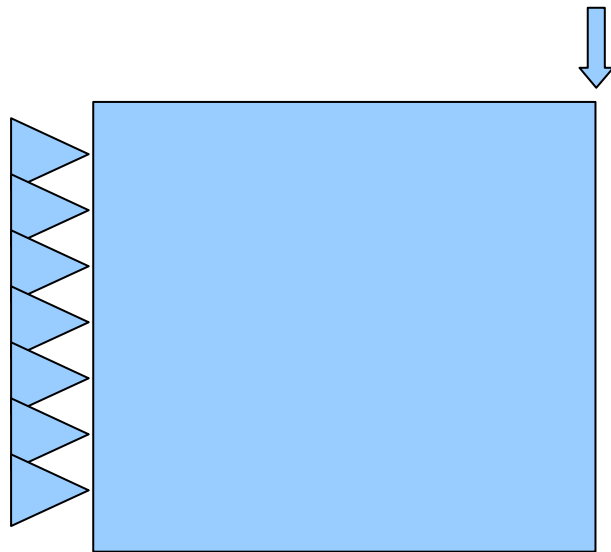


Figura 4.31

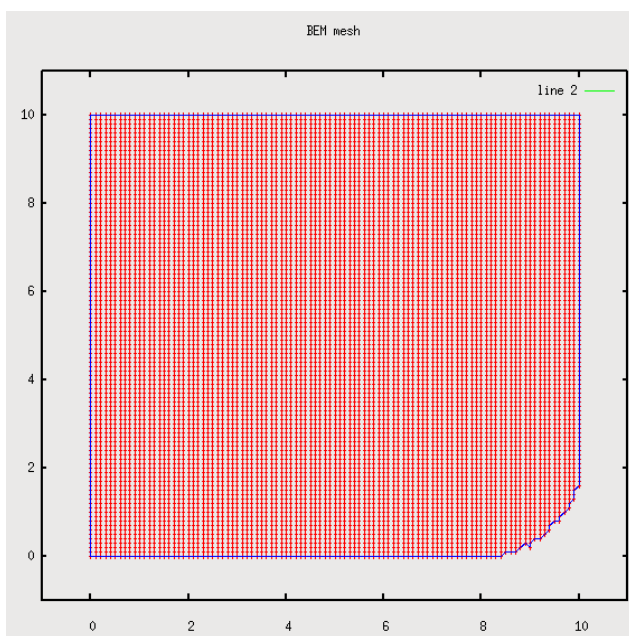


Figura 4.32: Iteración nº10

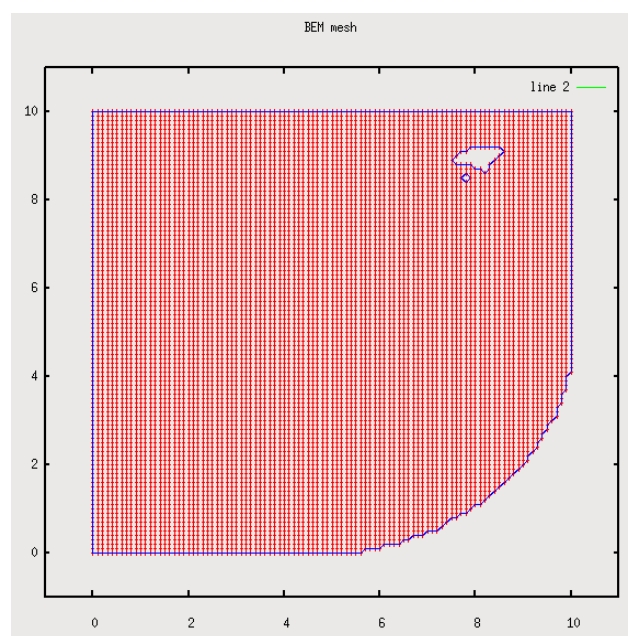


Figura 4.33: Iteración 20

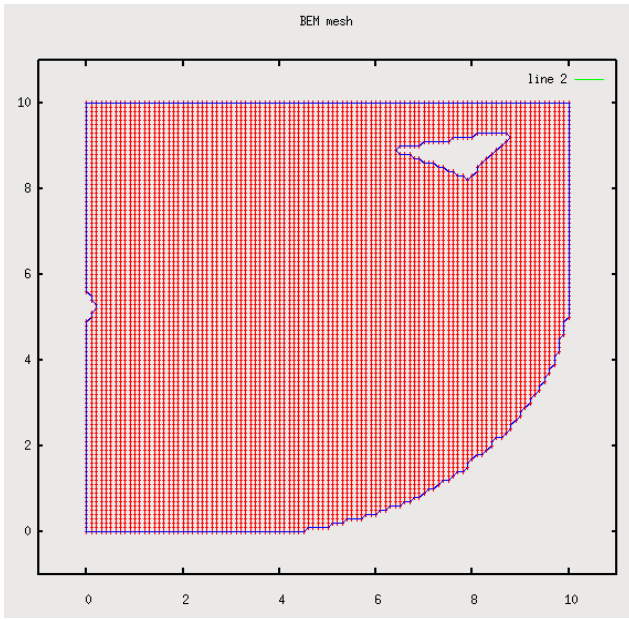


Figura 4.34: Iteración nº30

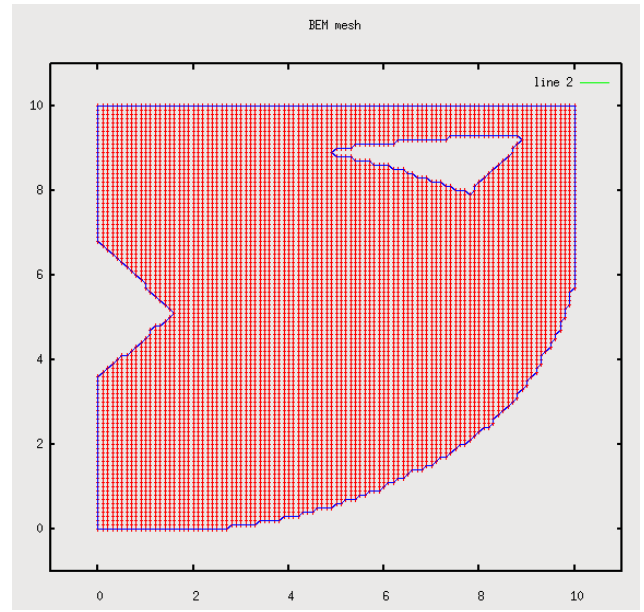


Figura 4.35: Iteración nº40

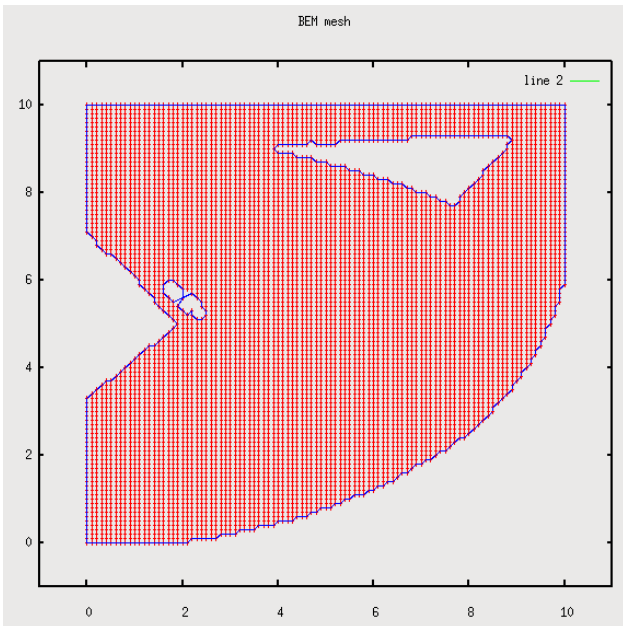


Figura 4.36: Iteración nº50

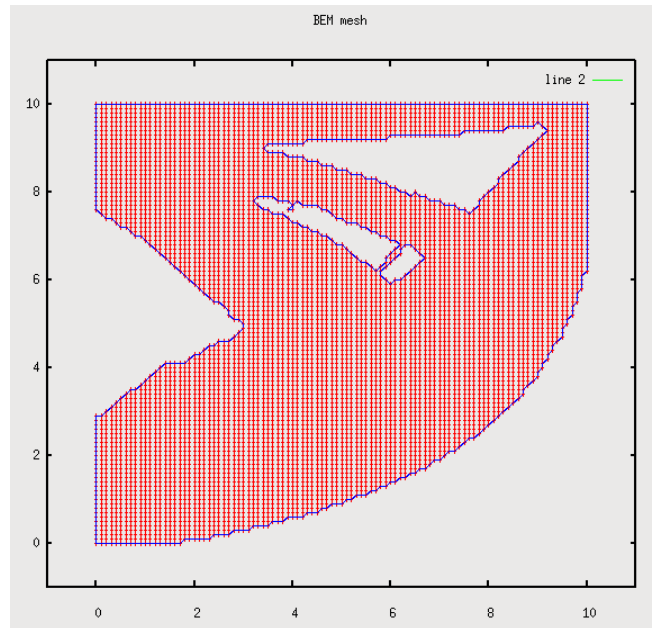


Figura 4.37: Iteración nº60

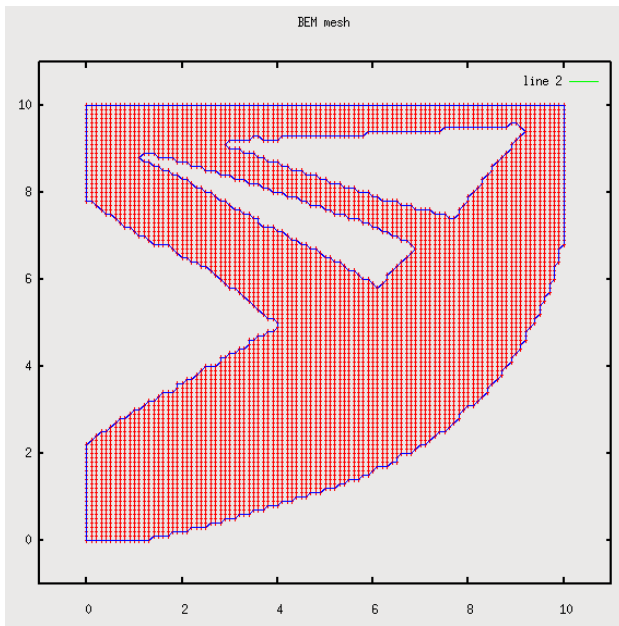


Figura 4.38: Iteración nº70

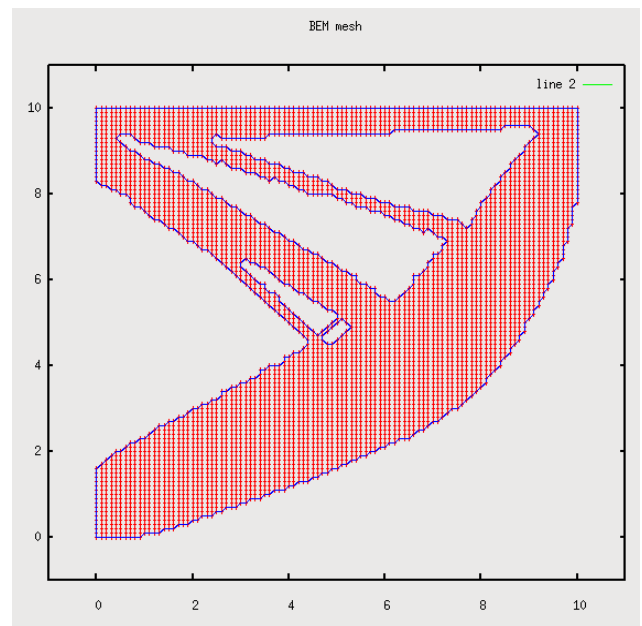


Figura 4.39: Iteración nº80

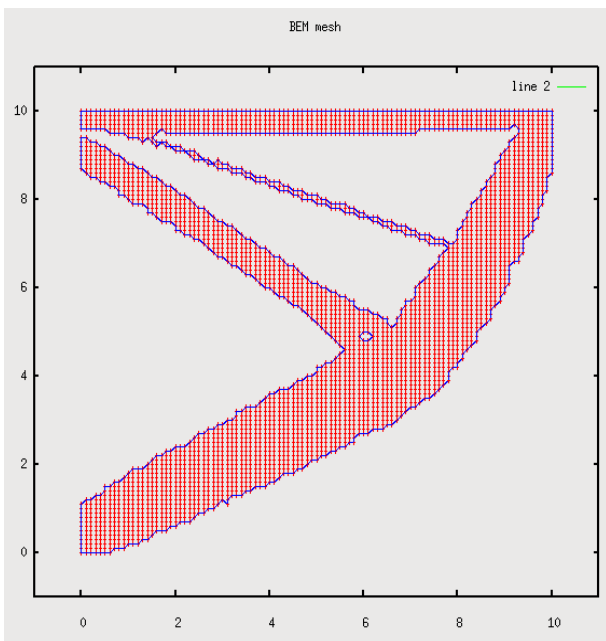


Figura 4.40: Iteración nº90

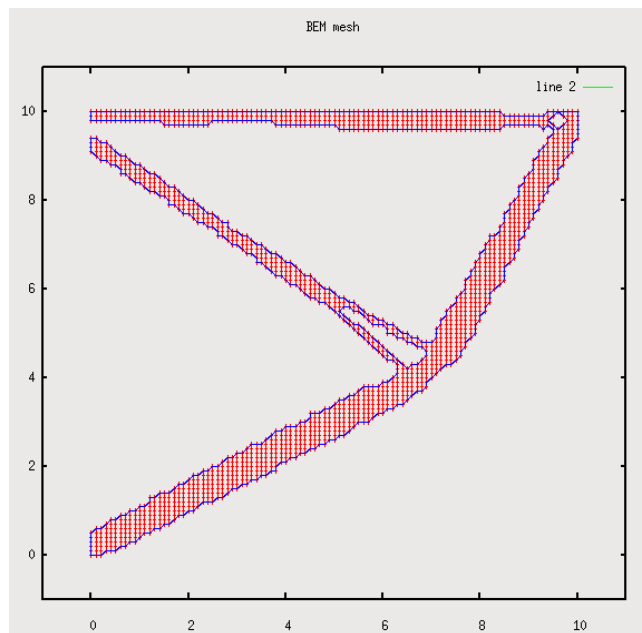


Figura 4.41: Iteración nº100

En la evolución de las siguientes figuras se observa como la estructura óptima resultante es una formada por barras en forma triangular.

- Ejemplo de la chapa con restricciones de movimiento total en una esquina de desplazamiento horizontal en la opuesta, sometida a cargas verticales puntuales. (problema de la bicicleta):

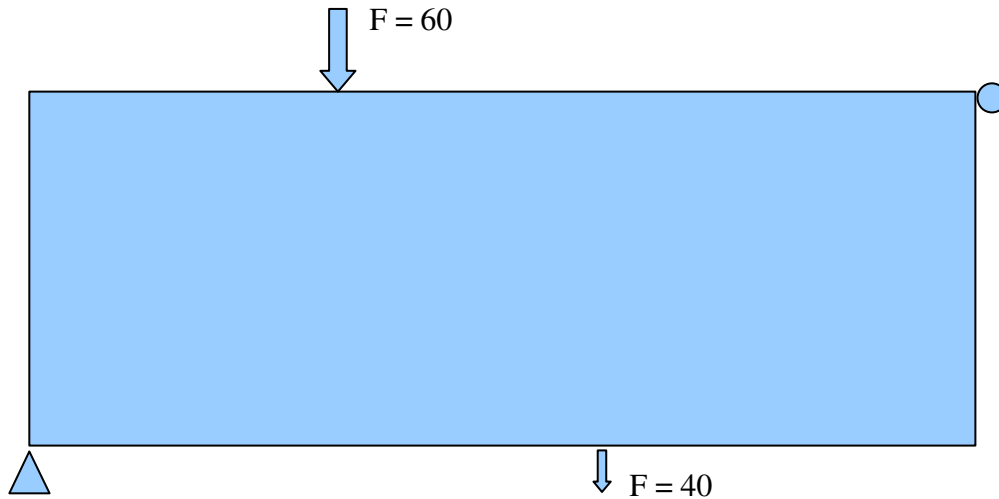


Figura 4.41

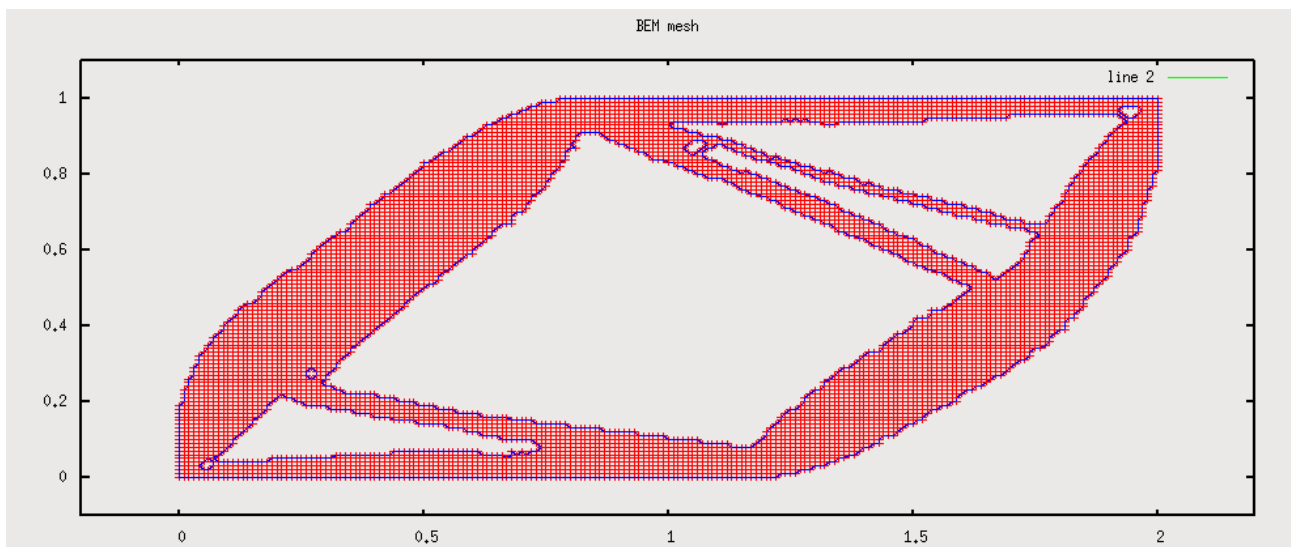


Figura 4.42

En este caso se intento modelar el cuadro de una bicicleta con un reparto de esfuerzos de 60 en el lugar que ocuparía el sillín y 40 en los pedales. El resultado de nuevo es una estructura de barras pero no sitúa ninguna barra en la zona central.

- Ejemplo de placa soldada a un tubo:

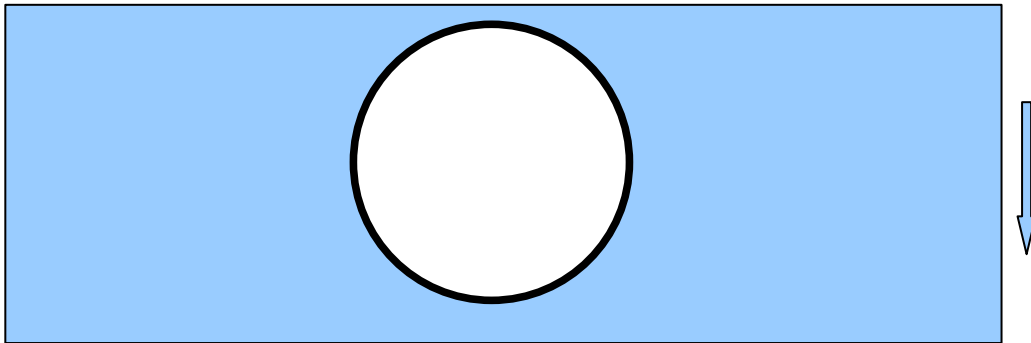


Figura 4.43

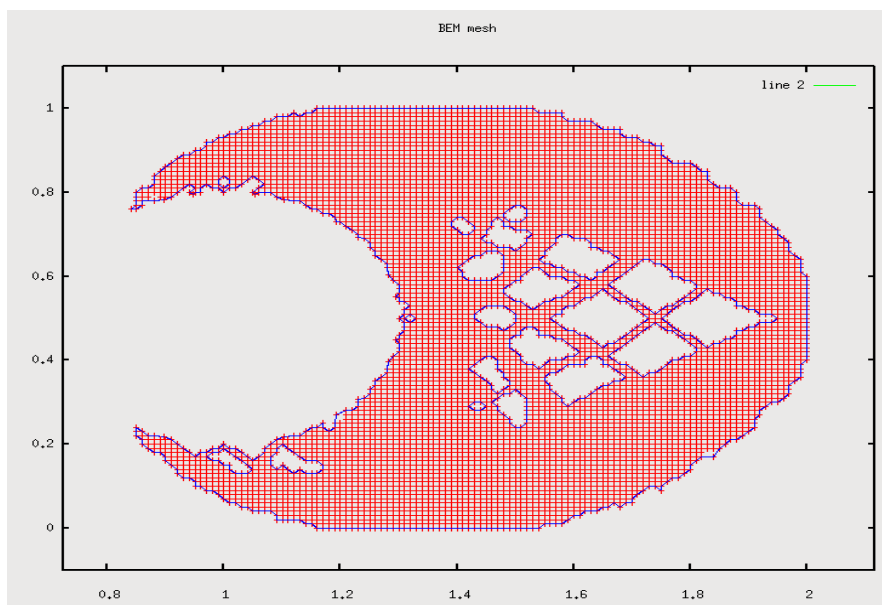


Figura 4.44

Este ejemplo es una placa rectangular de dimensiones 200x100 que tiene aplicada una fuerza vertical en el punto medio de su lado derecho. Esta sujeta por un agujero central que tiene todos sus puntos fijos. El resultado es eliminar la parte donde no esta cargada y en la zona cargada crear una rejilla con barras.

#### 4.7.- Ejemplo personal. Problema del gancho.

Como ejemplo particular de nuestra aplicación, probaremos el problema del “gancho” consistente en una placa rectangular de 2x1m con dos agujeros centrados simétricos y centrados en la chapa y una abertura para insertar el objeto a enganchar de la forma en la que se representa en la figura 4.45

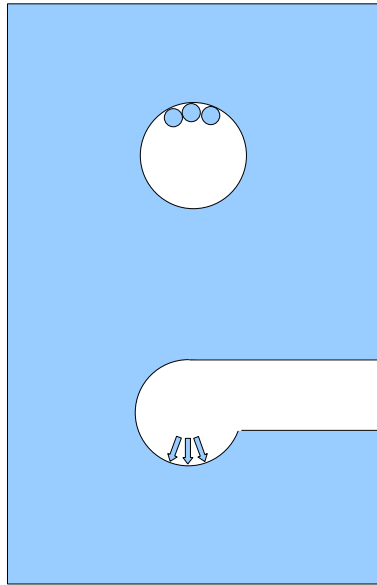


Figura 4.45  
Problema del gancho

Como solución inicial obtuvimos la siguiente forma inicial:

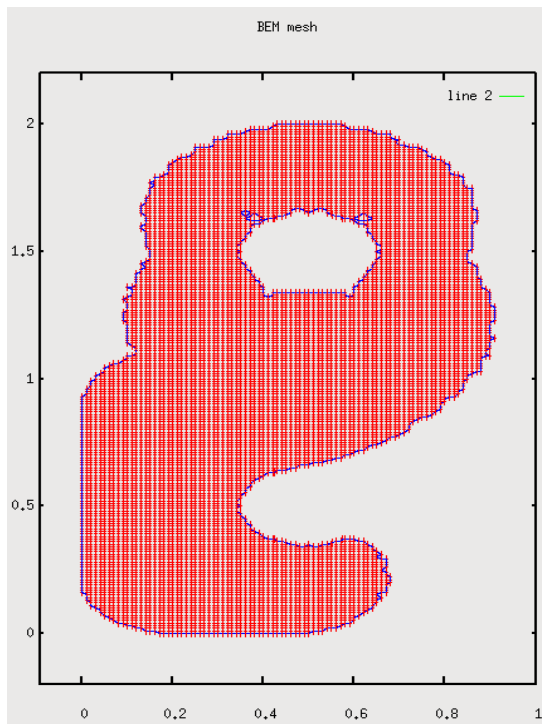


Figura 4.46  
Problema con  
falta de material.

En la figura 4.46 se puede observar como en la parte lateral izquierda hay una posible falta de material por lo que procedimos a cambiar la geometría inicial añadiendo material en dicha zona quedando el problema inicial de la siguiente forma (figura 4.45)

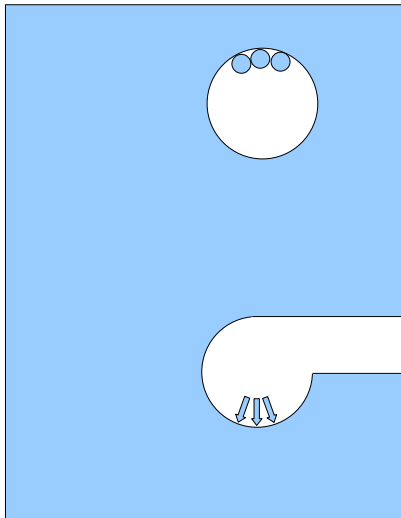


Figura 4.47 Problema del gancho modificado

Esto nos dio los resultados que se exponen a continuación de forma mas detallada:

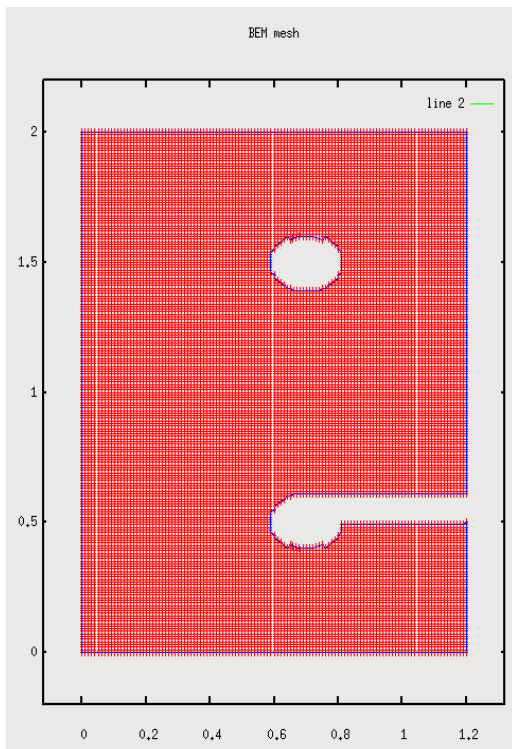


Figura 4.47: Iteración 1

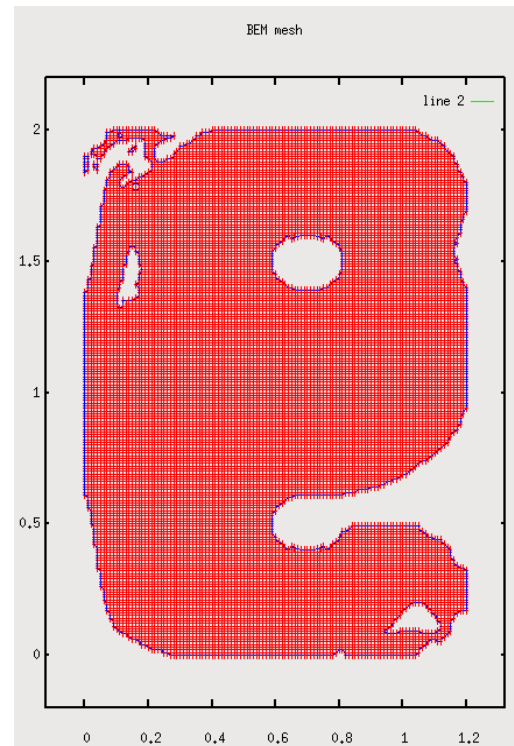


Figura 4.48: Iteración 5

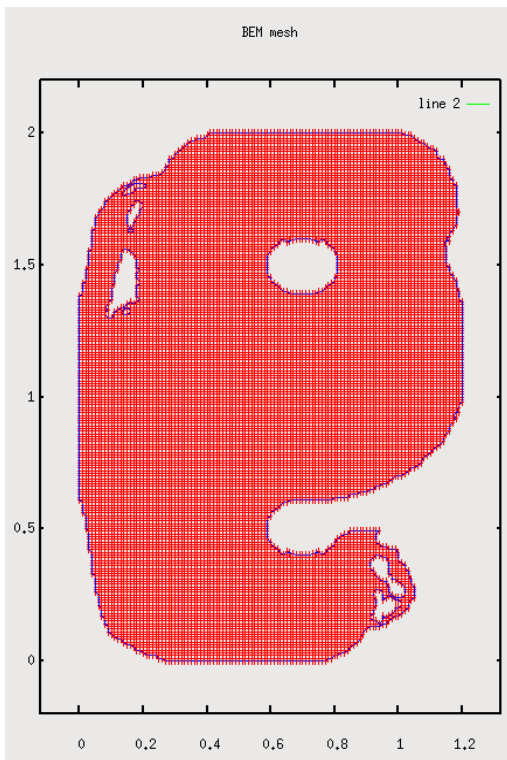


Figura 4.49: Iteración 10



Figura 4.50: Iteración 15

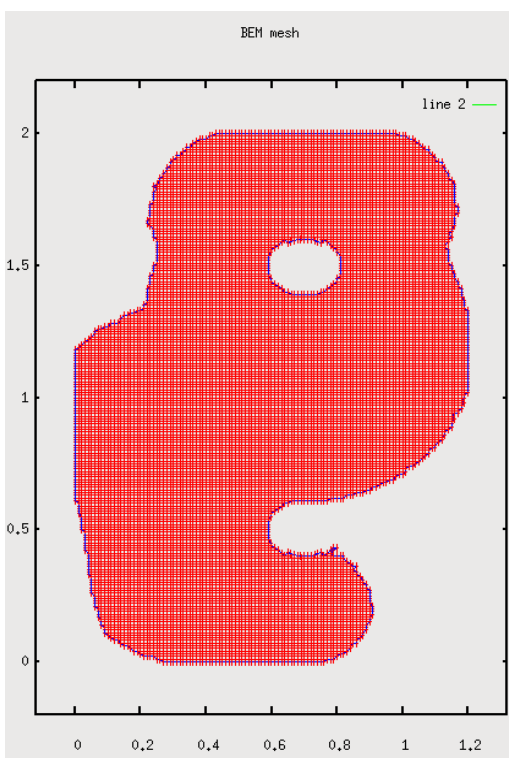


Figura 4.51: Iteración 20



Figura 4.52: Iteración 25

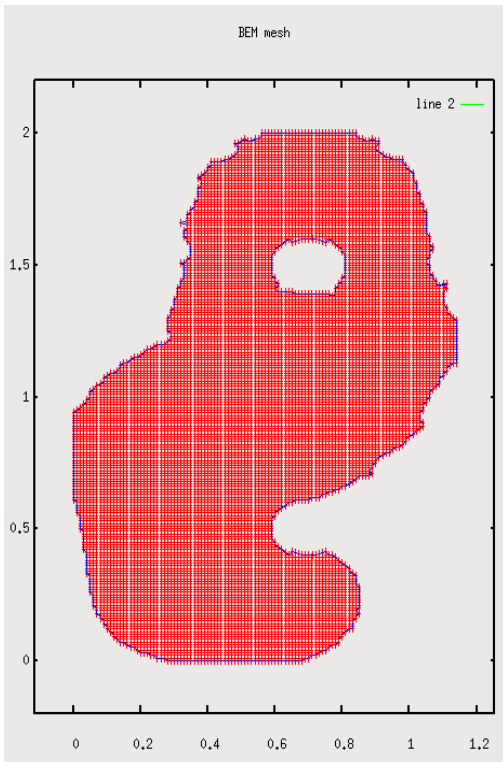


Figura 4.53: Iteración 30

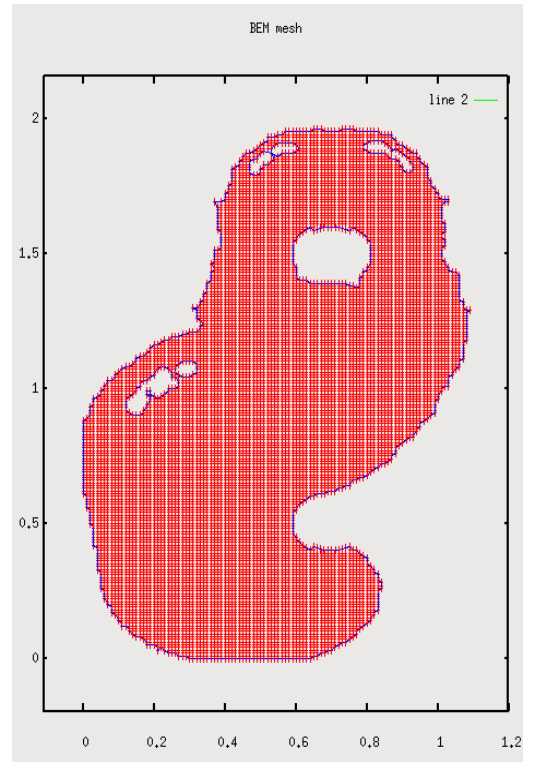


Figura 4.54: Iteración 35

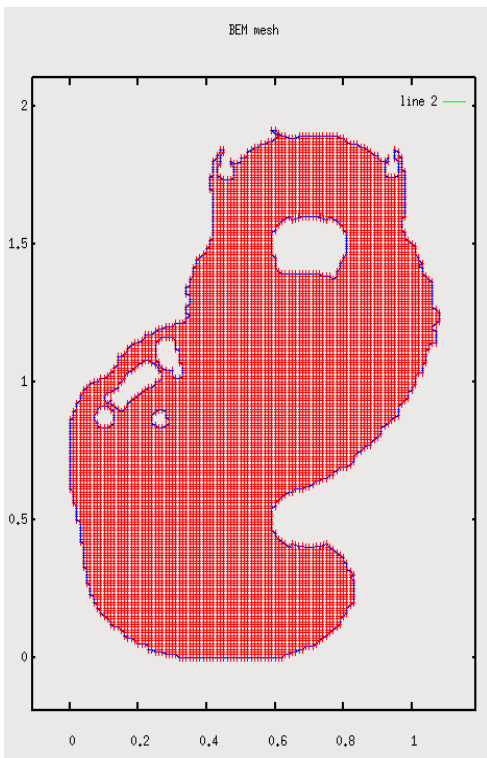


Figura 4.55: Iteración 40

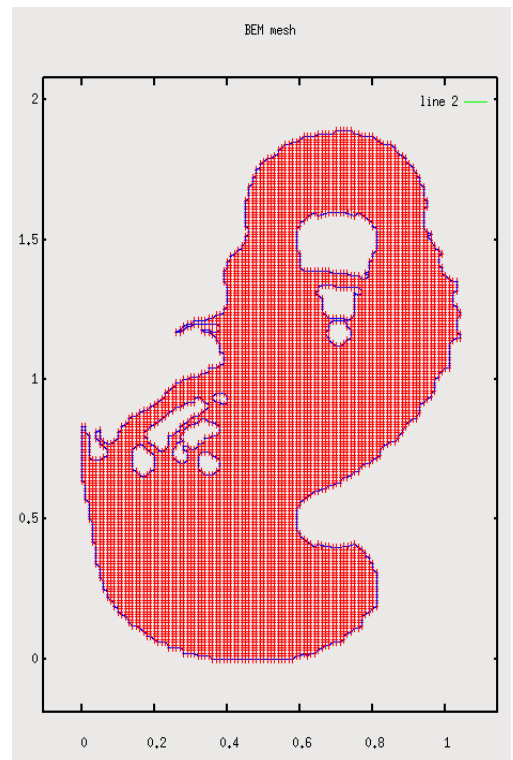


Figura 4.56: Iteración 45

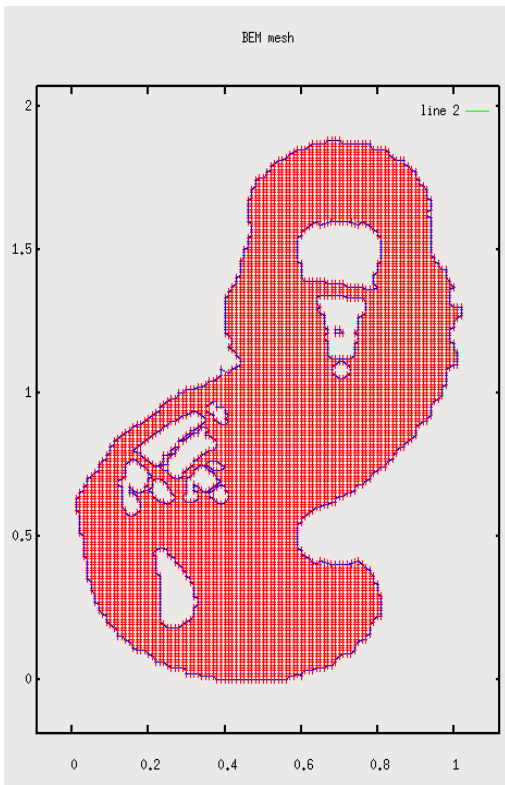


Figura 4.57: Iteración 50

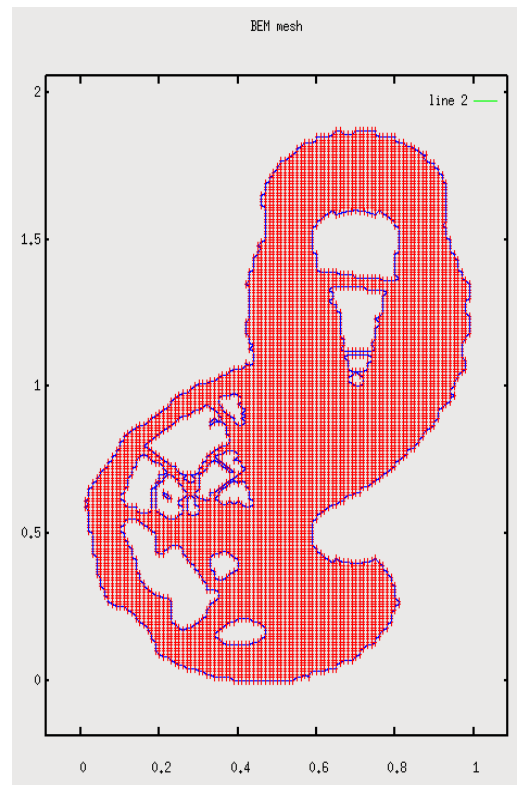


Figura 4.58: Iteración 55



Figura 4.59: Iteración 60

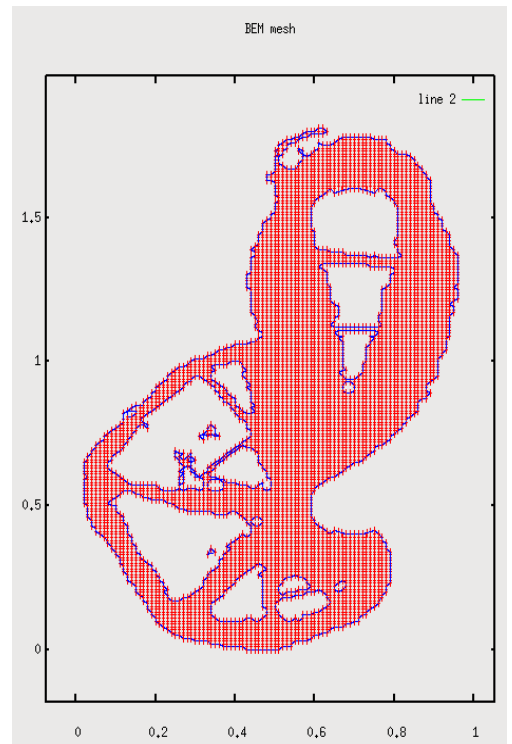


Figura 4.60: Iteración 65

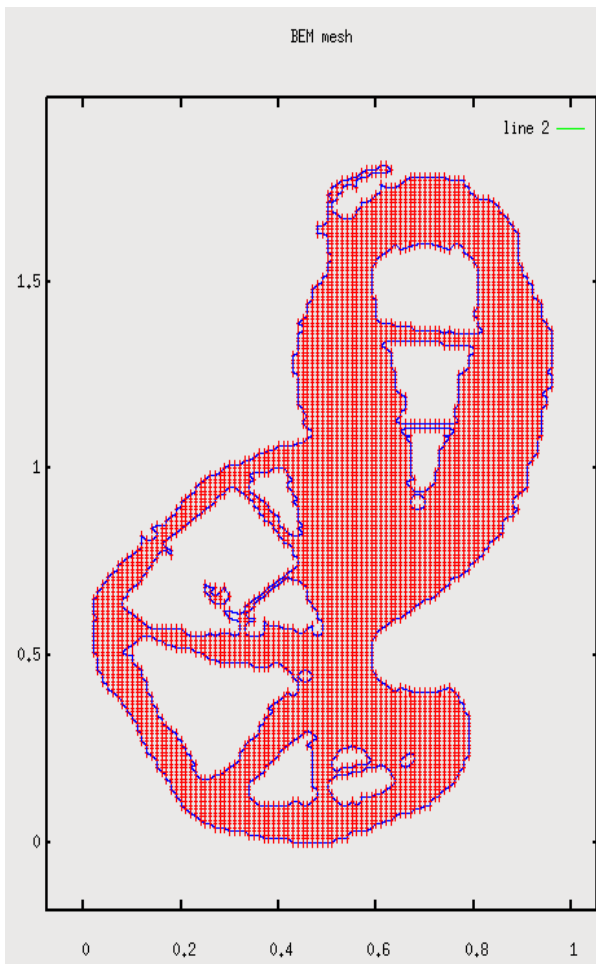


Figura 4.61: Solución final del gancho. Iteración 70

En la figura 4.61 podemos observar la estructura final a la que llega el algoritmo del proyecto. Dicha estructura resultante podríamos dividirla en tres partes diferenciadas que serían:

- la correspondiente a la zona de agarre del gancho, zona superior, donde toma forma de elipse con una cuña interior con barras transversales y en donde se insertaría el gancho
- la correspondiente al cuerpo del gancho, zona inferior izquierda, que sería la encargada de soportar las mayores tensiones y que vemos que está formada por un anillo exterior unido con barras radiales al centro de la pieza o zona de la punta.
- la correspondiente a la punta, que el algoritmo no calcula y que se tendría que diseñar en función del objeto que quisiéramos colgar en ella.

Teniendo en cuenta las observaciones anteriores podríamos diseñar un gancho óptimo con una punta cualquiera como vemos en la figura 4.62.

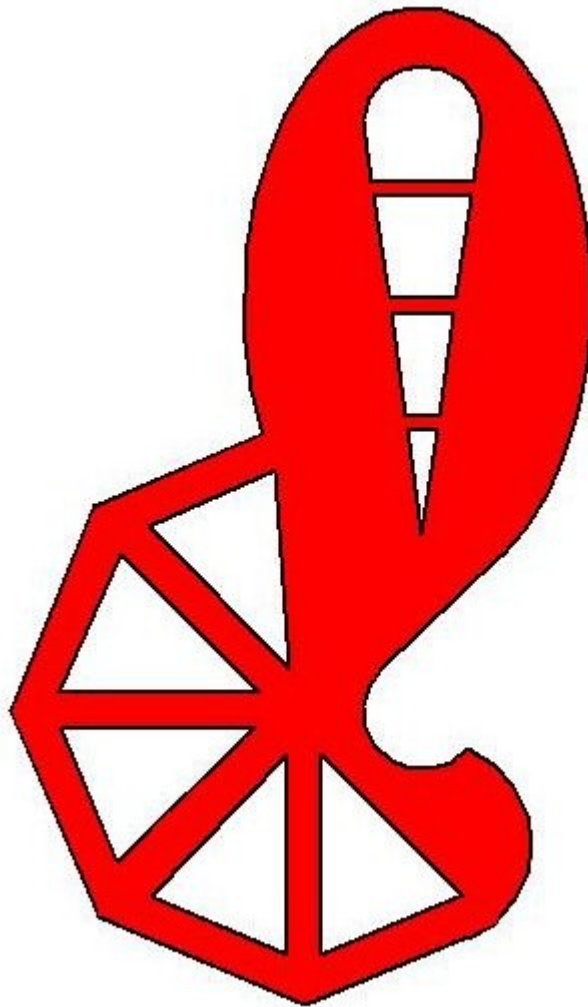


Figura 4.62: Diseño final del gancho