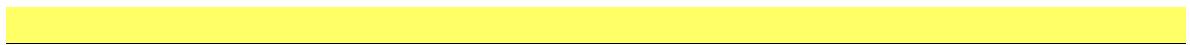


Guía del método zaxxon





Índice

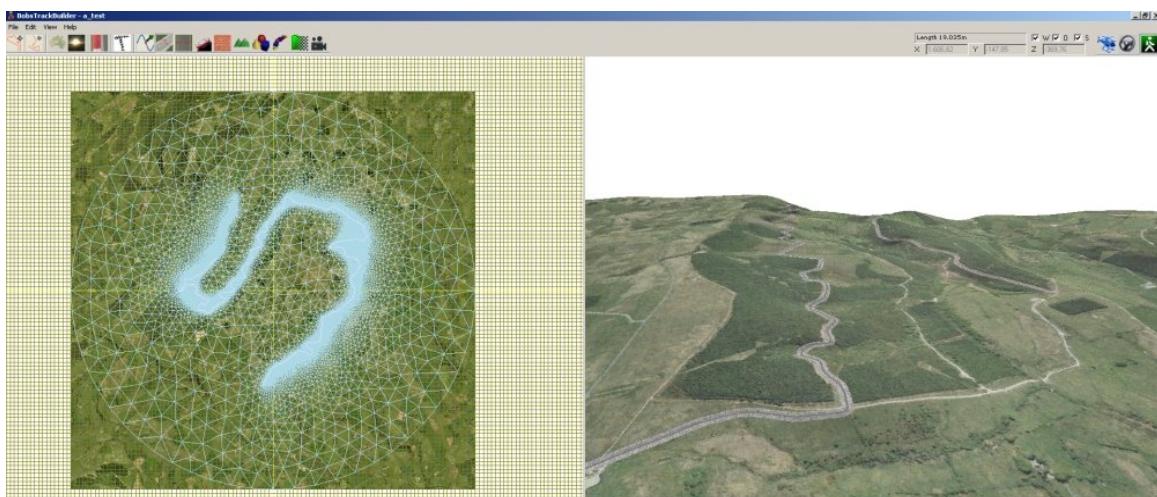
I	<u>Resumen</u>	1
I.1	<u>¿Qué es el "método zaxxon"?</u>	1
II	<u>Instalación del software</u>	3
II.1	<u>La estructura de carpetas y los scripts</u>	3
II.2	<u>Octave 3.2.3</u>	5
II.3	<u>El interfaz gráfico (GUI) para los scripts</u>	5
II.4	<u>Instalación de gmsh</u>	7
II.5	<u>WP.zip</u>	7
III	<u>Antes de usar el método</u>	8
III.1	<u>One-track o multi-track?</u>	8
III.2	<u>¿Origen de los datos de elevación?</u>	9
IV	<u>Usando los scripts</u>	13
IV.1	<u>Proyectos multi-track</u>	13
IV.2	<u>Proyectos one-track</u>	23
V	<u>Trabajar con diferentes fuentes de datos de elevación</u>	24
V.1	<u>Google Earth</u>	24
V.2	<u>AGR Data</u>	25
V.3	<u>Seamless server</u>	28
V.4	<u>Ficheros hgt</u>	29
VI	<u>Ajuste fino del perfil en altura de la carretera</u>	30
VI.1	<u>Retocar a mano el perfil</u>	30
VI.2	<u>Uso del script “corregir”</u>	32
VII	<u>Usando gmsh</u>	36
VII.1	<u>¿Qué son anchors_carretera.geo y joined.geo?</u>	36
VII.2	<u>Conceptos básicos de gmsh</u>	36
VII.3	<u>Controles básicos de gmsh</u>	36
VII.4	<u>¿Cuáles son los límites de la zona no conducible?</u>	37
VII.5	<u>¿Cómo debo crear la frontera externa del terreno no conducible?</u>	37
VII.6	<u>¿Cuál es la salida del proceso?</u>	38
VII.7	<u>¿Cómo se definen las Physical Surface?</u>	39
VII.8	<u>Los thresholds en gmsh</u>	41
VII.9	<u>Procesar joined.geo con gmsh</u>	45
VIII	<u>Imágenes de fondo</u>	58
VIII.1	<u>Conceptos básicos</u>	58
VIII.2	<u>Mezclado con las imágenes de fondo</u>	59
VIII.3	<u>Resumen</u>	62
IX	<u>Usando datos LiDAR</u>	66
X	<u>Usos avanzados de los scripts</u>	71
X.1	<u>Terreno adaptado a las imágenes de fondo</u>	71
XI	<u>Lista de comandos</u>	73
XII	<u>Notas adicionales</u>	81

XII.1	GUI	81
XII.2	Gmsh y multi-track	81
XII.3	Memory exhausted	84
XII.4	¿Gmsh se cuelga al hacer mesh/2D?	84
XII.5	¿Puedo insertar en gmsh una ruta a partir de un kml?	85
XII.6	Herramienta Translate de gmsh	86
XIII	Enlaces	88
XIII.1	Videotutoriales	88
XIII.2	Forums	89
XIII.3	Archivos de ayuda antiguos	89
XIII.4	La web de parlesportes	89
XIV	Licencia	90

I RESUMEN

I.1 ¿Qué es el "método zaxxon"?

Básicamente es una forma "rápida" de crear proyectos base para BTB, utilizando para ello datos de elevación. **Funciona con BTB Pro, pero no con BTB Evo.** Una vez te acostumbras al método puedes tener un proyecto BTB preparado para ser editado con el BTB en unas horas. Se puede incluir terreno y carreteras creados a partir de datos de elevación, y, si se quiere, texturizados con imágenes de satélite reales.



Un resumen simplificado del método, teniendo en cuenta que hay múltiples formas de hacer las cosas sería:

1. Todo empieza con el kml de la ruta. Se convierte a coordenadas BTB
2. Se obtienen datos de elevación suficientes para cubrir todas las carreteras y terreno.
3. Se le da a la carretera un perfil en altura acorde con los datos de elevación que se tienen (se puede retocar a mano)
4. Entonces se definen los límites del terreno alrededor de las carreteras y se crea un mallado para ese terreno.
5. Se le da elevación al terreno a partir de los datos de altura que se han conseguido
6. Se crean muros invisibles para evitar que el coche salga del terreno conducible (solo para RBR)
7. Se dividen las carreteras en segmentos por cuestión de eficiencia tanto en BTB

como in-game

8. Se divide el terreno empleando una rejilla m x n (también por eficiencia)
9. Se crea un Venue.xml, un fichero que BTB puede leer y que permite empezar a trabajar con el proyecto

Se pueden incluir automáticamente imágenes de fondo (satélite) en el proyecto. Quizá únicamente para tenerlas de referencia o para usarlas como textura para el terreno.

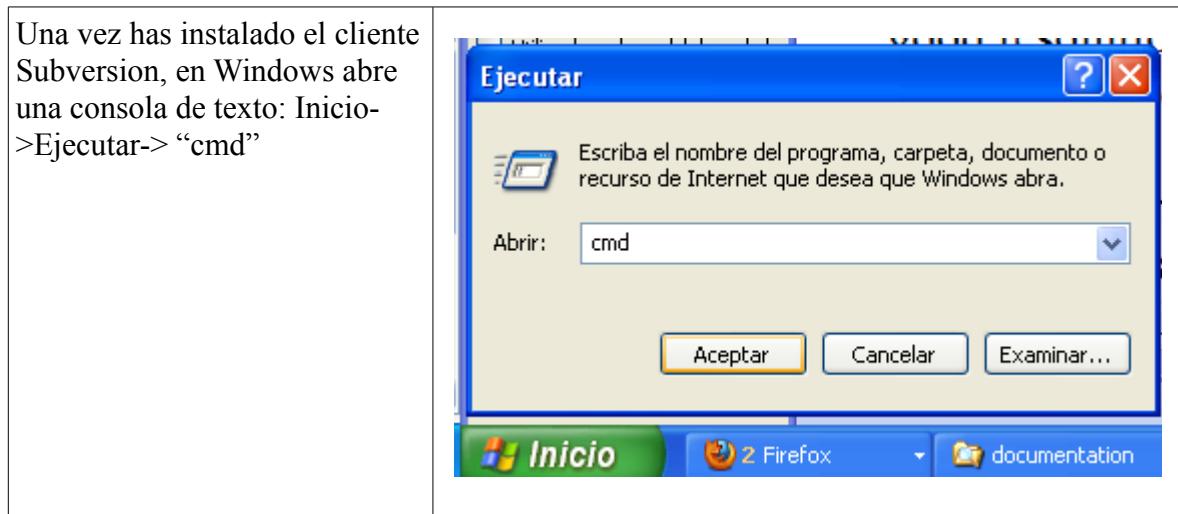
II INSTALACIÓN DEL SOFTWARE

Para usar los scripts se necesita:

1. [La estructura de carpetas y los scripts](#)
2. [Octave 3.2.3](#)
3. [El interfaz gráfico \(GUI\) para los scripts](#)
4. [Instalación de gmsh](#)
5. [WP.zip](#)

II.1 La estructura de carpetas y los scripts

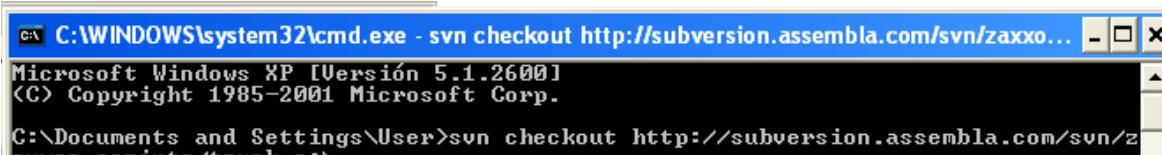
Se necesita Subversion (SVN) para descargar los ficheros. Se pueden descargar clientes gratuitos visitando: <http://subversion.apache.org/packages.html> (por ejemplo <http://www.sliksvn.com/pub/Slik-Subversion-1.7.2-win32.msi> de <http://www.sliksvn.com/pub>)



En la consola escribe:

```
svn checkout http://subversion.assembla.com/svn/zaxxon_scripts/trunk c:\
```

(O se puede descargar [download_scripts.bat](#) y ejecutarlo (doble click))

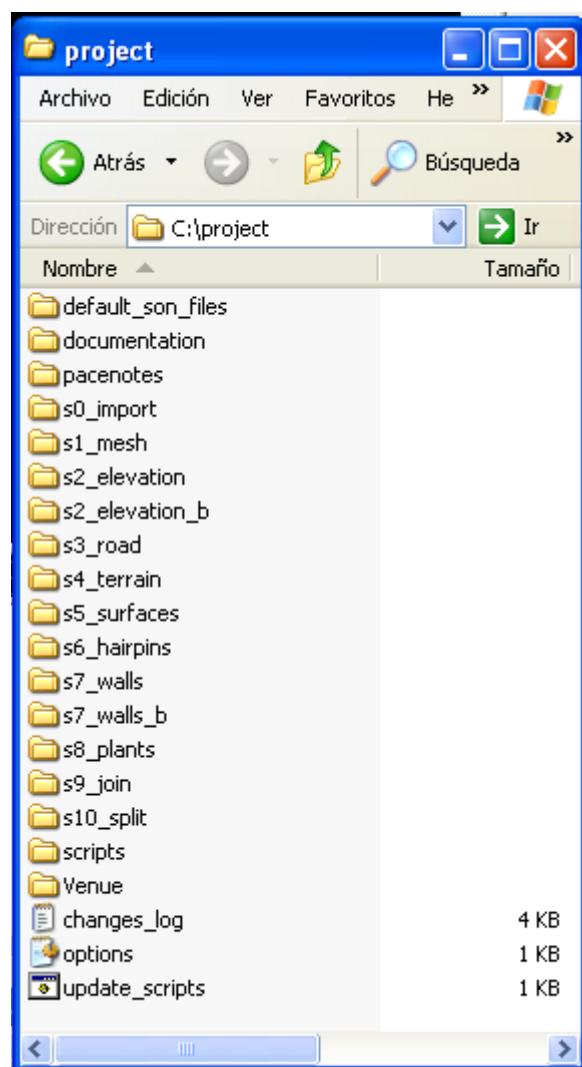


```
C:\WINDOWS\system32\cmd.exe - svn checkout http://subversion.assembla.com/svn/zaxxon_scripts/trunk c:\

Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\User>svn checkout http://subversion.assembla.com/svn/zaxxon_scripts/trunk c:\
```

Se obtendrá una copia de los ficheros necesarios en [c:\project](#)

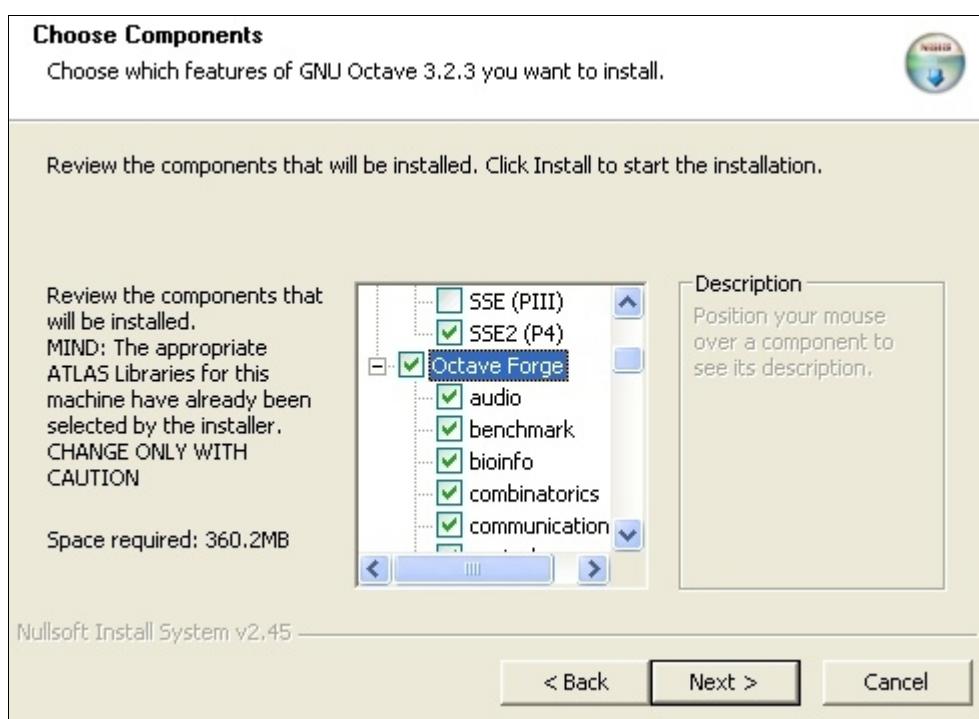


II.2 Octave 3.2.3

Se necesita [Octave 3.2.3](#) (NO instalar una versión diferente de Octave):

http://sourceforge.net/projects/octave/files/Octave_Windows%20-%20MinGW/Octave%203.2.3%20for%20Windows%20MinGW32%20Installer/Octave-3.2.3-2_i686-pc-mingw32_gcc-4.4.0_setup.exe/download

OBLIGATORIO: durante la instalación **marca Octave Forge libraries**



Tras la instalación de Octave 3.2.3, abrirlo y ejecutar:

> pkg rebuild -auto communications

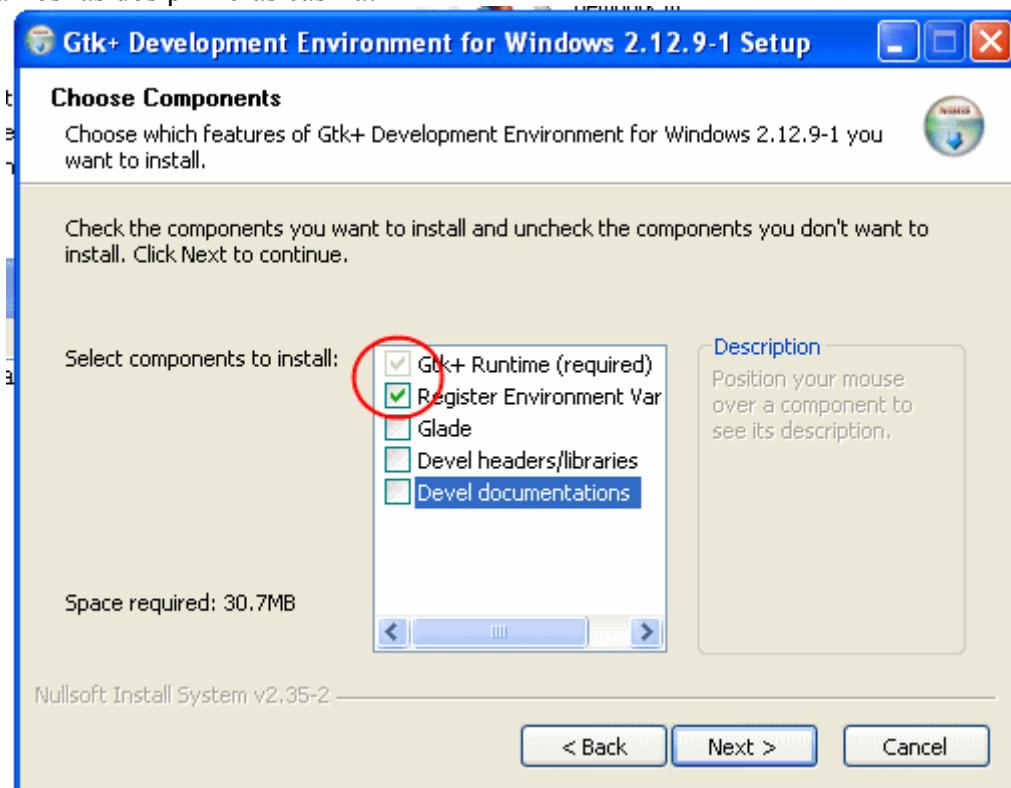
Luego cerrar Octave y abrirlo de nuevo cuando se quiera comenzar a usarlo.

II.3 El interfaz gráfico (GUI) para los scripts

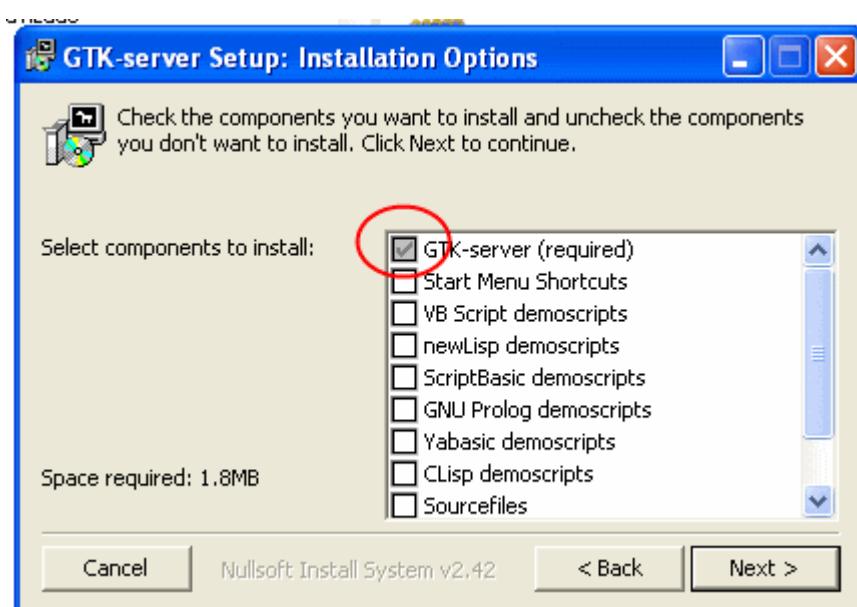
1. Instalar GTK (gtk-dev-2.12.9-win32-2.exe)

<http://sourceforge.net/projects/gladewin32/files gtk%2B-win32-devel/2.12.9/gtk-dev-2.12.9-win32-2.exe/download>

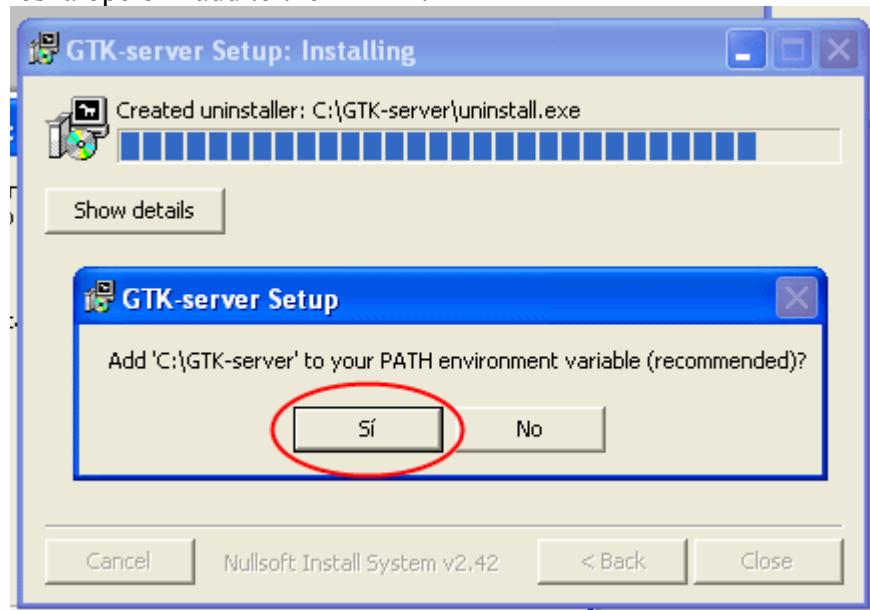
Marcamos las dos primeras casilla:



2. Instalar gtk-server (gtk-server-2.3.1-installer.exe)
<http://downloads.sourceforge.net/gtk-server/gtk-server-2.3.1-installer.exe>



Seleccionamos la opción “add to the PATH”:



II.4 Instalación de gmsh

Gmsh es un generador de mallados para elementos finitos. Lo usaremos para crear un mallado para nuestro terreno. Debes instalarlo. Basta descargarlo y extraer el contenido en la carpeta que elijas.

Gmsh: <http://geuz.org/gmsh/#Download>

II.5 WP.zip

El proyecto creado usará texturas de un XPack de nombre WP.zip. Se puede conseguir el XPack WP.zip de: <http://www.mediafire.com/?z2tcoh4wyxj>

III ANTES DE USAR EL MÉTODO

III.1 One-track o multi-track?

¿Cuántas rutas .kml tienes para las carreteras de tu proyecto? Si solo tienes una ruta hablaremos de un “proyecto one-track” y usar los scripts será muy sencillo. Si tienes más de una ruta (desvíos, cruces, rutas alternativas, etc.) el proceso es un poco más complicado, especialmente a la hora de usar gmsh.

One-track

En los proyectos one-track no hay que hacer nada especial con los ficheros que tenemos en <c:\project>. Estás listo para empezar a usar los scripts.

Multi-track

Cuando se trabaja con varias rutas los scripts se usan básicamente de la misma forma en todas ellas, cada una procesada dentro de su propia estructura de carpetas. El trabajo que se hace por separado luego se combina, creando un proyecto que incluye rutas diferentes/independientes, como desvíos sin salida (uso estético) o rutas que pueden usarse como caminos alternativos.

Una idea importante es que debemos elegir una ruta principal. A esa ruta la llamaremos el FATHER. Y las rutas secundarias (todas las demás) serán los SONS.

Para preparar un proyecto multi-track:

1. Mover el contenido de <c:\project> to <c:\project\father>
2. Si por ejemplo se tienen tres sons, abrir octave y ejecutar:

- > addpath('c:\project\father\scripts')
- > cd c:\project\father
- > create_sons(3)

Ahora tendremos los ficheros del father en <c:\project\father>, y los ficheros de los 3 sons en <C:\project\son01>, <C:\project\son02> and <C:\project\son03>. Eso es todo: ya estamos listos para emplear los scripts.

NOTA: los sons saben que tienen un father y su ubicación gracias a que en sus carpetas se ha creado un fichero **father.txt** con el contenido “father”. El father sabe que tiene 3 sons y su ubicación porque en su carpeta se ha creado un fichero llamado “**sons.txt**”, de contenido:

son01

son02

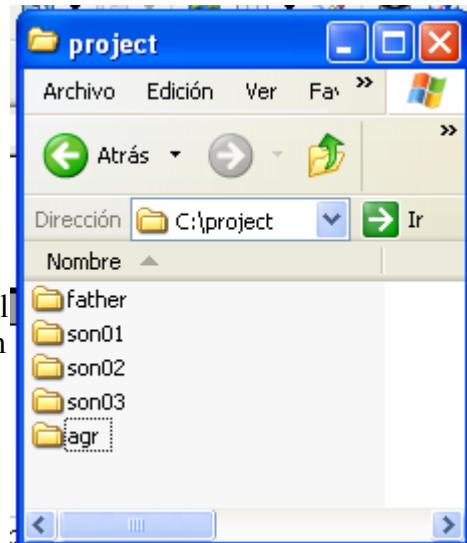
son03

III.2 ¿Origen de los datos de elevación?

Las fuentes habituales de datos de elevación son Google Earth y AGR (ASCII Grid). El GUI está preparado para usar solo estas dos fuentes. Si se quiere usar otra, se debe leer el capítulo V.

AGR

- 1) Crear una carpeta llamada “agr” en la misma carpeta en la que está "project" (o las carpetas del father's y los sons en los proyectos multi-track). Ejemplo: si trabajamos en un proyecto one-track en c:\project, crearemos la carpeta c:\agr.
- 2) Copia los ficheros .agr a la carpeta creada en el paso 1). Renombra esos ficheros con extensión .agr si es necesario.
- 3) Usa los scripts con normalidad.



NOTA: si los ficheros .AGR son demasiado grandes los scripts pueden no ser capaces de usarlos. Se puede emplear el script **split_agr** para dividir un fichero en ficheros más pequeños, como se explica en la sección V.2

Google Earth

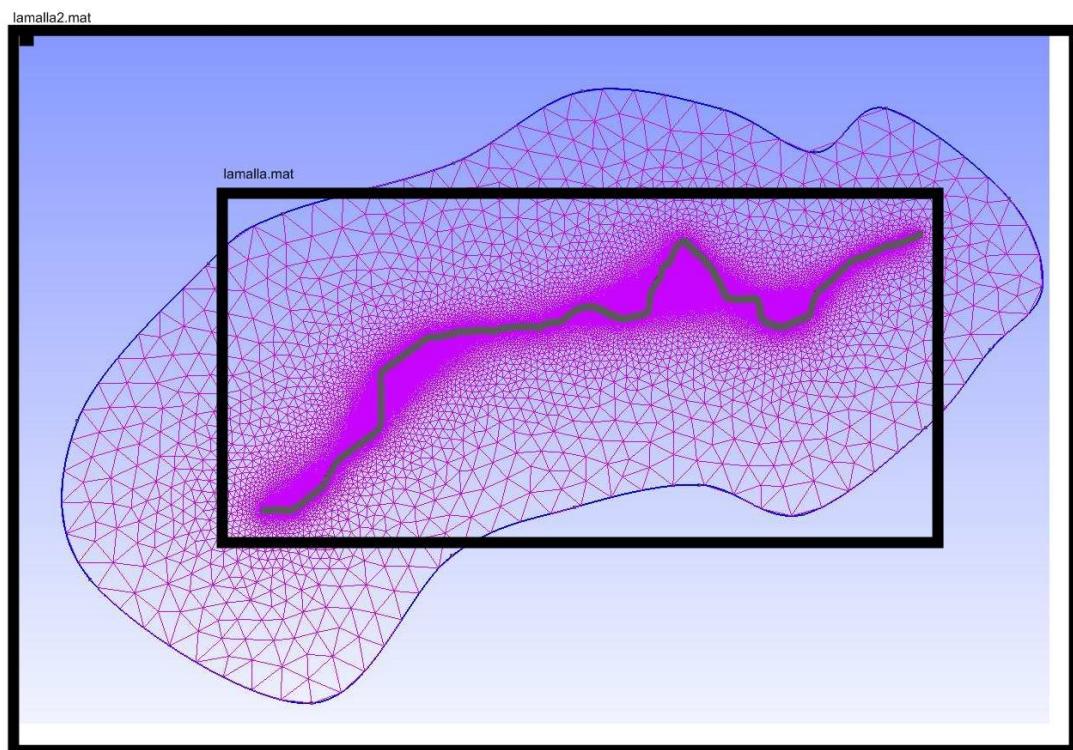
Se pueden conseguir datos de elevación de Google Earth (leer sección V.1 para ampliar). En ese caso los scripts crearán una rejilla de puntos, distribuirán esos puntos en decenas de ficheros (de nombre `gridXXX.kml`) y pedirán a GE que les de elevación. Los ficheros se procesarán uno tras otro, a mano si se usa un programa del estilo de BTBLofty o automáticamente si se usa python).

Los scripts pueden trabajar con una única rejilla de datos. Pero se recomienda usar 2. La razón para recomendar usar 2 rejillas es que queremos buena resolución para los datos cerca de las carreteras **pero es absurdo tener una resolución de 10m a 1Km de las carreteras** porque el mallado que creemos tendrá triángulos muy grandes (75-250m de lado) lejos de las carreteras. Por tanto podemos crear dos rejillas y buscar datos de elevación para ellas: una rejilla con buena resolución que debe cubrir por completo las carreteras (father y sons) y otra con peor resolución cubriendo por completo el terreno del proyecto.

La rejilla pequeña se creará dentro de la carpeta `s2_elevation`. Para la rejilla grande se empleará otra carpeta: `s2_elevation_b`. Los datos de altura se copiarán automáticamente a la carpeta `s4_terrain` con el nombre **lamalla2.mat**. Para la rejilla pequeña el nombre usado será **lamalla.mat**.

Como ya se ha dicho, los scripts también trabajarán si existe una única rejilla (creada en `s2_elevation`), pero como dicha rejilla debería cubrir todo el terreno, si tuviese que tener una buena resolución espacial, el número de puntos de la rejilla sería enorme, al igual que el fichero `lamalla.mat` (y consecuentemente el tiempo requerido para obtener los datos de altura de Google Earth).

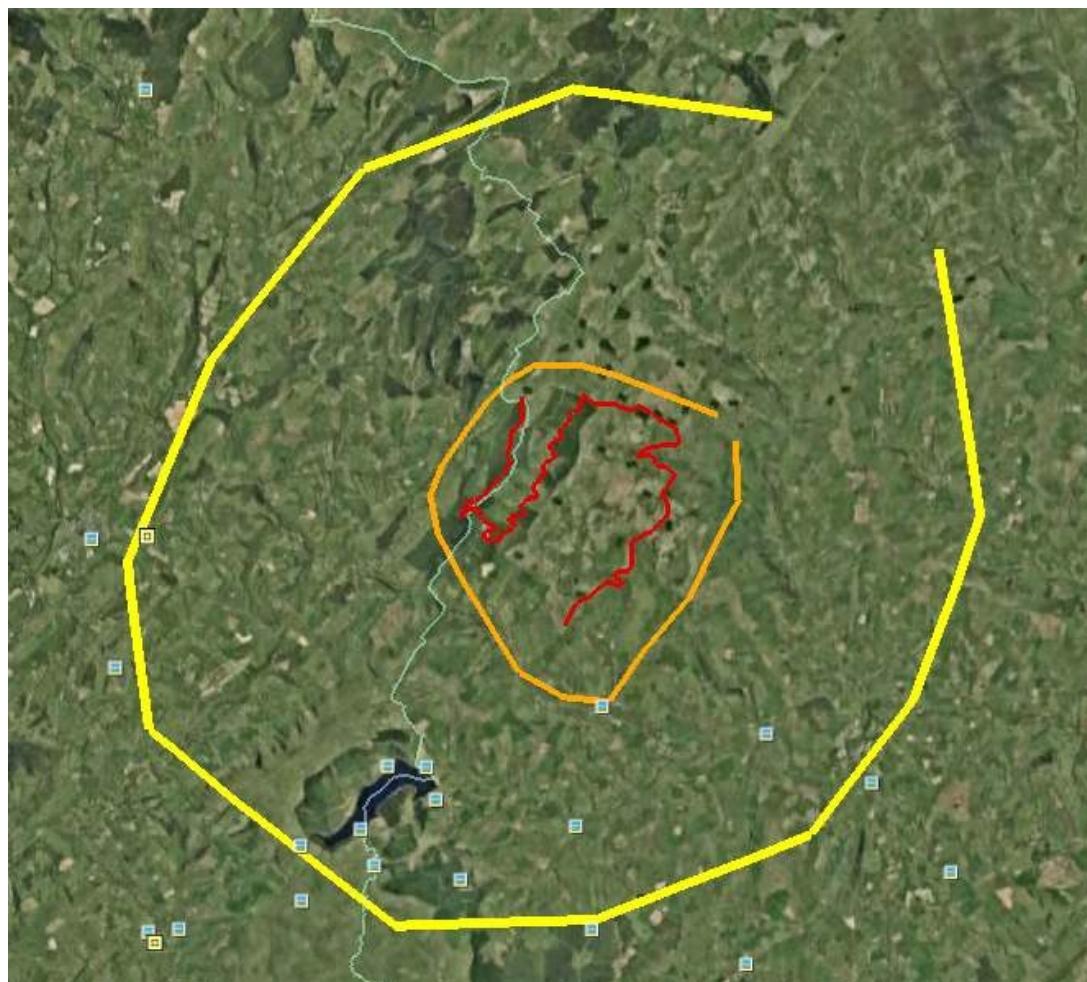
La siguiente imagen es ilustrativa:



La mejor forma de delimitar ambas rejillas es crear rutas con Google Earth. Creamos dos kmls más:

- 1) "**limits.kml**" (ruta naranja) rodeando la carretera principal y todos los desvíos. Se debe grabar en **s2_elevation**
- 2) "**limits_b.kml**" (ruta amarilla) delimitando el terreno deseado para el proyecto. Se debe grabar en **s2_elevation_b**

Las rutas no necesitan ser cerradas y los scripts crearán un par de rejillas suficientemente amplias como para cubrirlas por completo. La rejilla grande cubrirá todo el terreno encerrado por la ruta amarilla y la rejilla pequeña cubrirá todo el terreno rodeado por la ruta naranja. El color de la ruta es indiferente para los scripts y solo se menciona para ayudar a identificarlas.



IV USANDO LOS SCRIPTS

IV.1 Proyectos multi-track

NOTA: llegados a este punto se debe haber copiado los .kml del father y los sons dentro de sus respectivas carpetas s0_import. La elevación de los puntos del kml será ignorada.

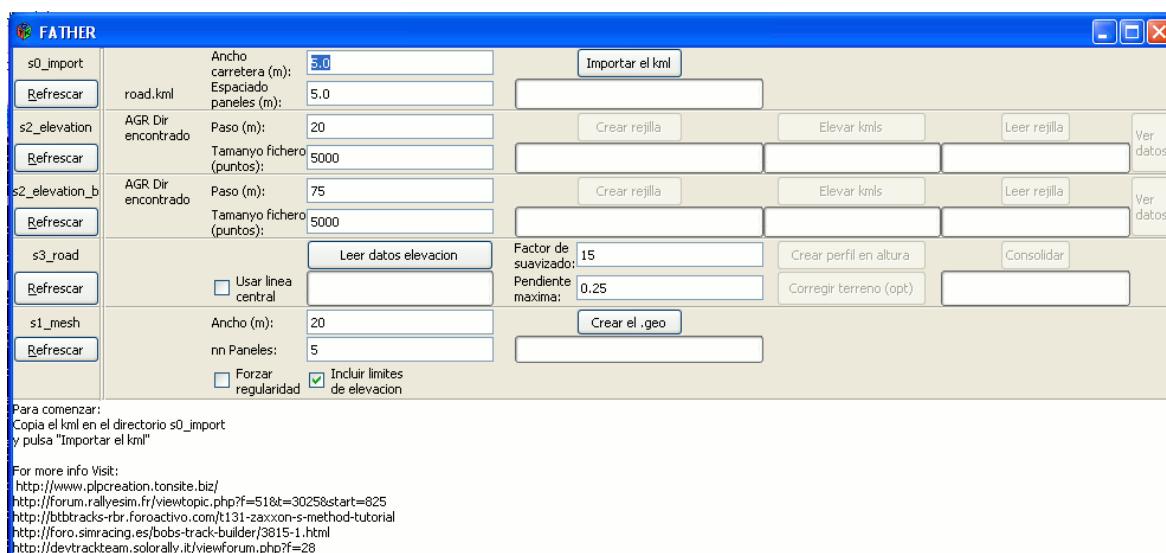
Y finalmente abrimos octave, añadimos la carpeta donde están los scripts a los paths donde octave buscará los comandos y ejecutamos **zgui**. Si por ejemplo los scripts están en c:\project\father\scripts:

```
> addpath('c:\project\father\scripts')
> cd c:\project\father
> zgui
```

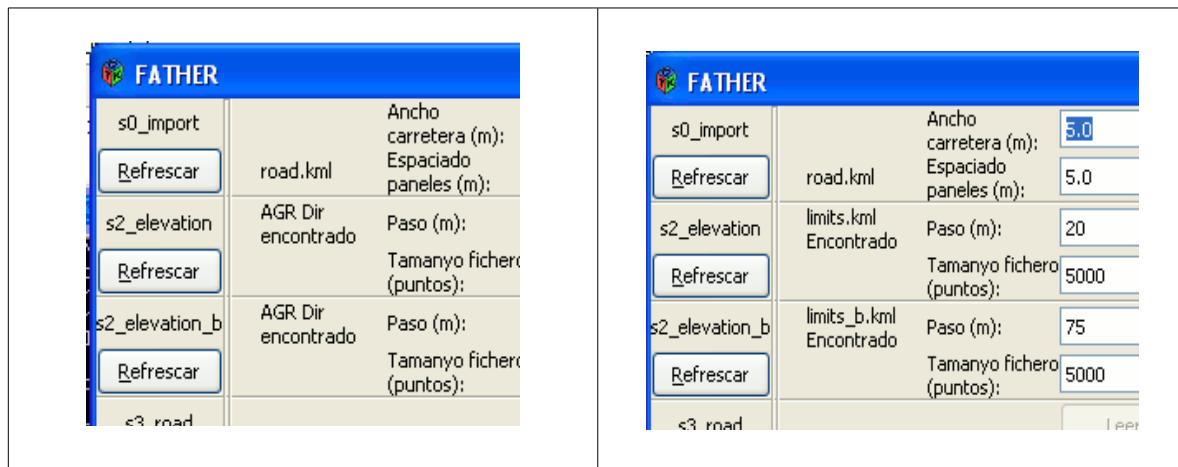
zgui es el interfaz para la primera parte de uso de los scripts (antes de usar gmsh)
zgui(1) es el interfaz para la segunda parte de uso de los scripts (cuando ya existe anchors_carretera.msh)

NORMAL y FATHER tienen 2 interfaces: zgui y zgui(1). Los SONs solo tienen zgui, pero no zgui(1).

Se debería abrir una nueva ventana en nuestro sistema, el GUI:

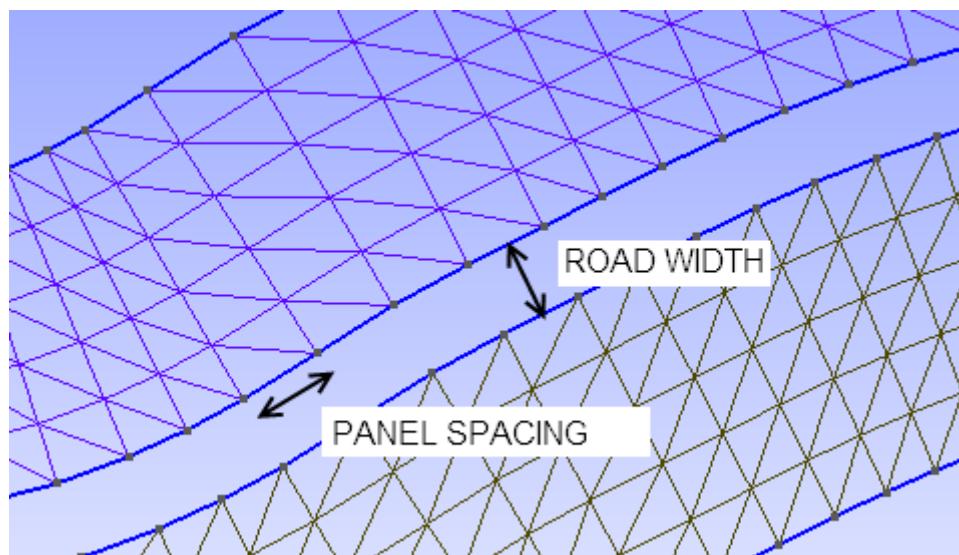


En la barra superior debería leerse la palabra “FATHER”. Y si estamos usando datos AGR deberíamos ver un mensaje en la ventana diciendo “AGR Dir encontrado”. En ese caso los pasos s2_elevation y s2_elevation_b deben saltarse. En caso contrario, si se usan datos de altura de Google Earth un mensaje en pantalla debe informarnos de que los ficheros “limits.kml” y “limits_b.kml” han sido encontrados.



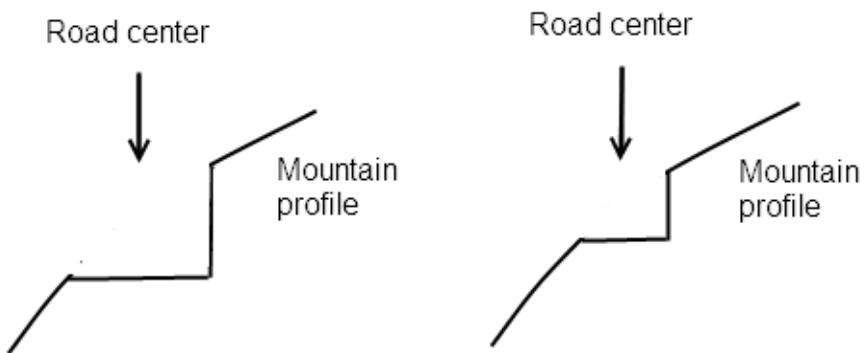
s0_import

El próximo paso es pulsar el botón “Importar el kml”. Para este botón necesitamos especificar un ancho para la carretera y el espaciado que queremos entre los puntos que unen la carretera y el terreno.



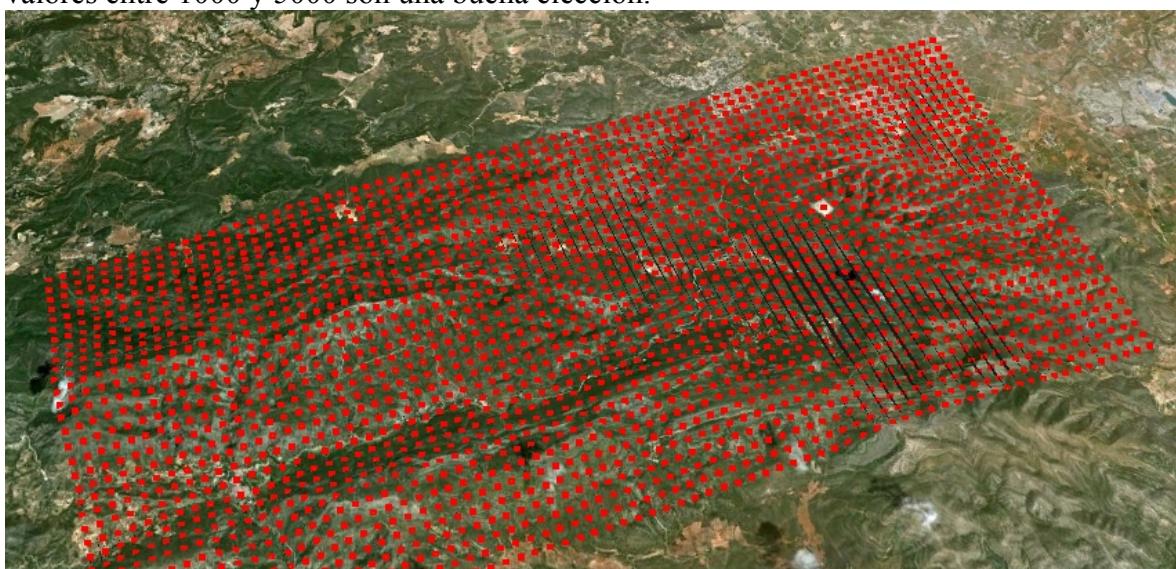
En este paso cada punto del .kml se traducirá en un nodo de la carretera BTB. Si el paso falla, se debería revisar el kml buscando pequeños lazos (puntos que retroceden, en lugar de avanzar) o puntos demasiado próximos.

El ancho de la carretera es importante porque afectará al mallaado que vamos a crear pero también porque el perfil en altura de la carretera depende del ancho de la misma. Cuando los datos de altura no son buenos a una carretera ancha los scripts le asignarán una altura menor que a una carretera estrecha. Posiblemente el mejor valor para el ancho de la carretera sea uno próximo al ancho real de la misma.



s2_elevation/s2_elevation_b

El siguiente paso es conseguir datos de altura, si se está usando GE como fuente de dichos datos. En ese caso hay que definir la separación entre los puntos de la rejilla y pulsar “Crear la rejilla”. Los puntos de la rejilla se distribuirán en varios ficheros (llamados gridXXX.kml) que se crearán dentro de s2_elevation\salida o s2_elevation_b\salida. El tamaño del fichero es la cantidad de puntos incluidos en cada fichero. 5000 es el máximo recomendado. Si se usa un valor superior a 5000 Google Earth se ralentiza drásticamente. Valores entre 1000 y 5000 son una buena elección.

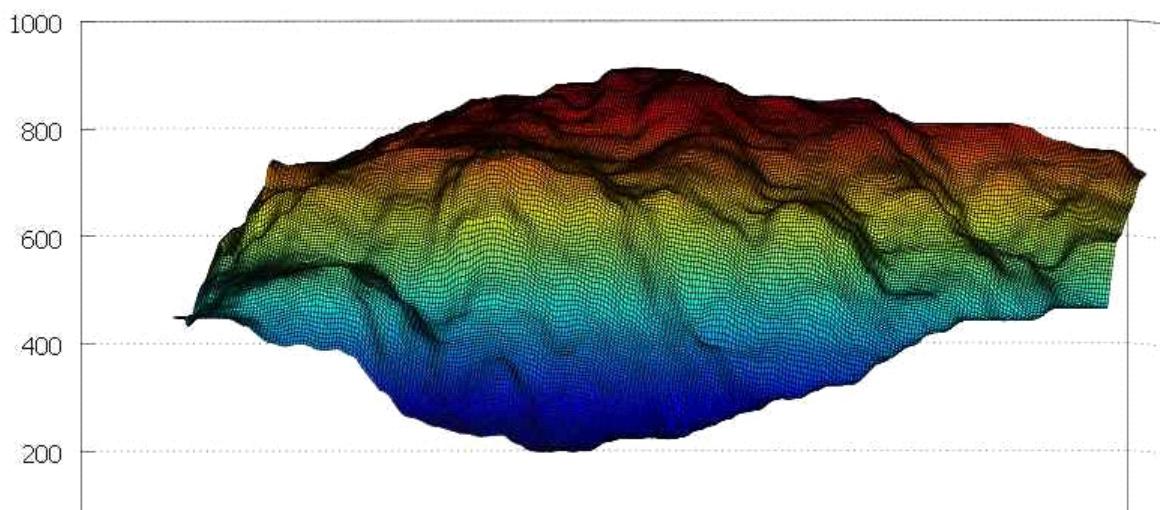


Ahora, si has instalado python (leer la sección V.1) puedes pulsar en el botón “Elevar los

kmls”. En caso contrario tendrás que elevar los ficheros empleando otra aplicación, como las listadas en la sección V.1.

Cuando el proceso finaliza cada fichero gridXXX.kml ha sido procesado y se ha creado un fichero gridXXX_relleno.kml con los mismos puntos pero con elevación. El siguiente paso es leer los ficheros gridXXX_relleno.kml y recopilarlos en la matriz de datos que usarán el resto de scripts. Haz click en el botón “Leer rejilla”. Un fichero llamado lamalla.mat se creará en la carpeta s2_elevation\salida.

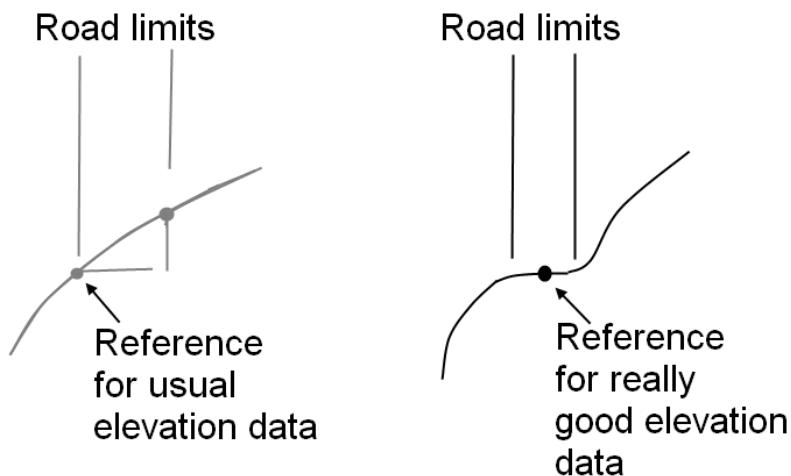
Puede ser buena idea en este momento pulsar el botón “Ver datos” y comprobar que a simple vista los datos recopilados no contienen errores.



s3_road

Ahora queremos darle un perfil en altura a la carretera. Para las fuentes de datos habituales el primer paso es calcular la altura del terreno a ambos lados de donde se sitúa la carretera (en concreto en los puntos que llamamos “anchors”). Cuando el terreno tiene distinta elevación a ambos lados de la carretera los scripts usan el más bajo de los dos como referencia para crear el perfil de elevación de la misma. No obstante si tuviésemos datos de elevación de elevada resolución (un dato cada 1m) podríamos usar el centro de la carretera como referencia de altura. Este sería el caso de emplear datos LiDAR.

Ante la duda NO marque “Usar la línea central”.



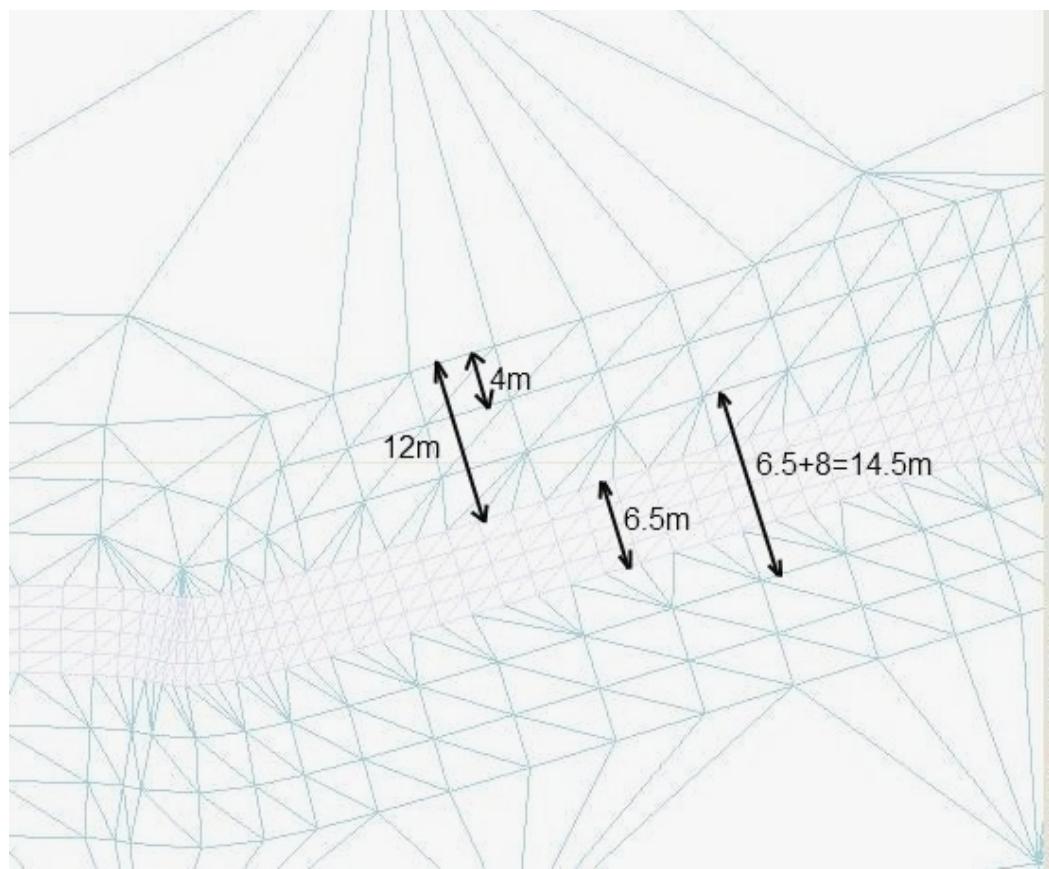
Una vez tenemos referencias de altura a lo largo de toda la carretera, las usamos para crear un perfil en altura suave para la misma. Se empleará un factor de suavizado S (S debe ser siempre impar) para crear un nuevo juego de valores de altura donde cada valor es el promedio de los S valores más próximos a ese punto. La distancia tenida en cuenta en el suavizado depende de la separación entre anchors. Por defecto la separación entre anchors es de 5m, pero se ha podido escoger otro valor en el paso s0_import. La mejor opción es probar valores (pulsando “Crear perfil en altura”) y examinar el resultado. El otro parámetro limita la pendiente de la carretera que se va a crear. 0.25 significa 25%. Se puede poner a 1 si no se quiere emplear este parámetro (1 sería limitar a una pendiente del 100%, y eso en una carretera normal no debería tener efecto).

Si te satisface el perfil en altura creado debes pulsar "e" en la ventana de texto de octave y luego <ENTER>. Finalmente hay que consolidar el perfil pulsando el botón “Consolidar”, el último del paso s3_road. Si no nos gusta el perfil, en este paso se permite al usuario ajustar a mano el perfil en altura, e incluso cambiar un poco los datos de altura (solo si se usan ficheros lamalla.mat, no en otros casos como AGR) tratando de adaptarlos a la carretera que queremos. Leer sección VI para detalles.

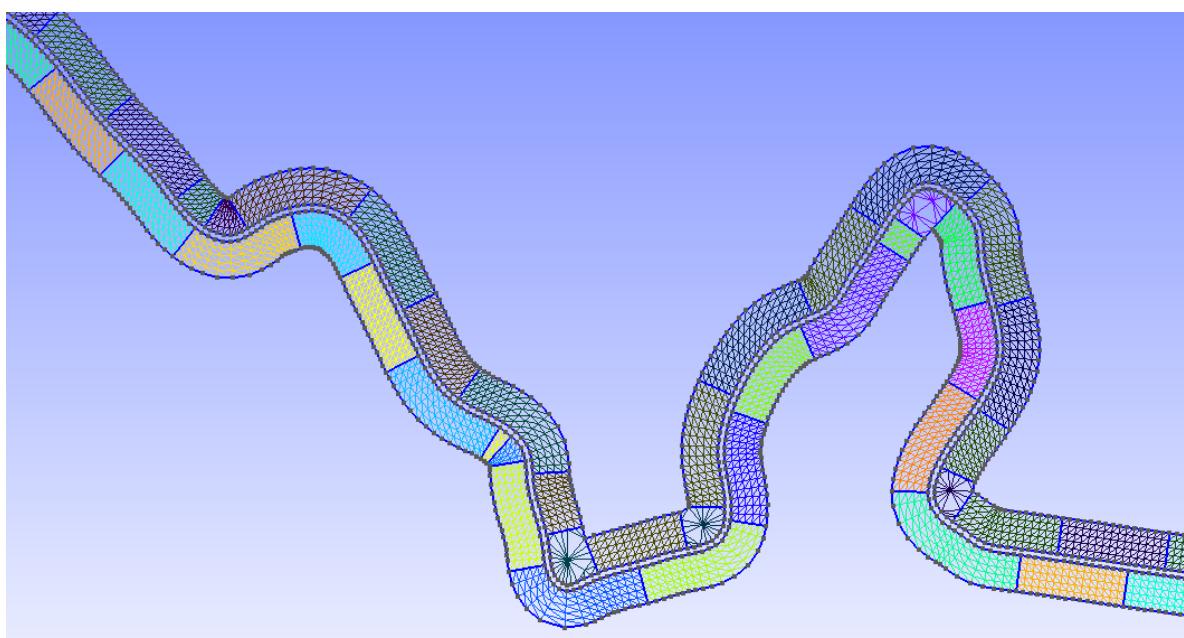
s1_mesh

Una vez tenemos nuestra carretera, el siguiente paso es crear terreno conducible a ambos lados de la misma. Tenemos que indicar dos datos: el ancho total de terreno conducible y cuántos paneles hay en dicho ancho. Hay una opción para dividir automáticamente la superficie conducible creada por los scripts en segmentos más cortos.

La siguiente imagen puede ser de ayuda para entender los parámetros: se usa un ancho de 12m y 3 paneles. La malla creada por los scripts para la parte conducible será la base para que el usuario cree otra malla para la parte no conducible. El usuario tiene que crear esa malla usando gmsh.

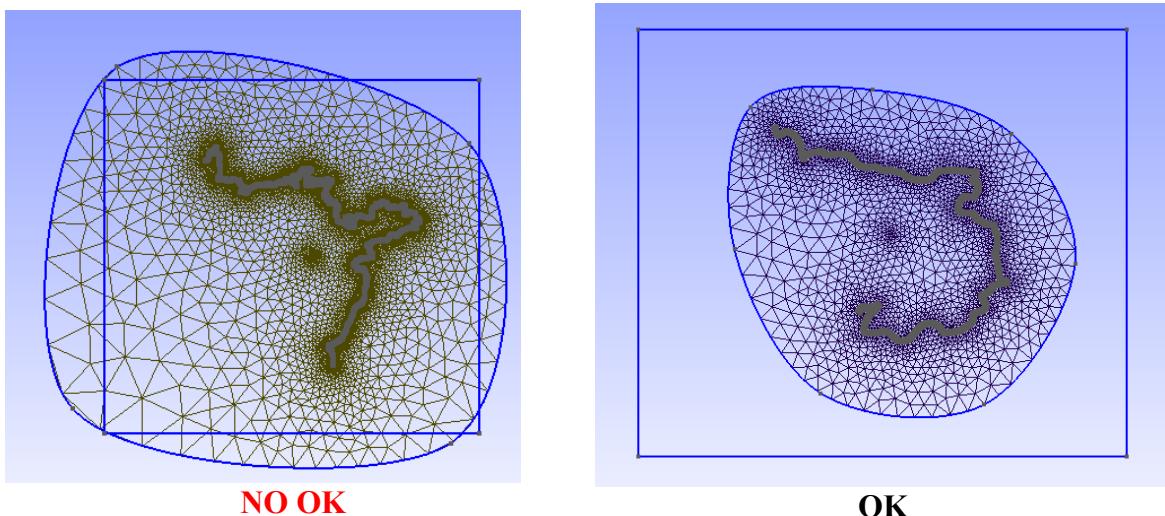


En el caso de que se divida la superficie conducible en segmentos más cortos se obtiene algo como esto:



El parámetro MINISUPS del fichero options.ini ajusta cada cuántos puntos queremos una nueva superficie.

En este paso es realmente importante seleccionar “**Incluir límites de elevación**” en caso de que se use Google Earth como fuente de datos de altura. Eso debería ayudarnos a no crear un mallado que excede los datos de elevación disponibles, un mallado que no podría ser procesado por los scripts. Por ejemplo, en la parte izquierda de la siguiente figura se muestra un mallado que excede los límites y que por tanto debe ser rehecho para que sus límites no se salgan del rectángulo (límites de datos de elevación disponibles).

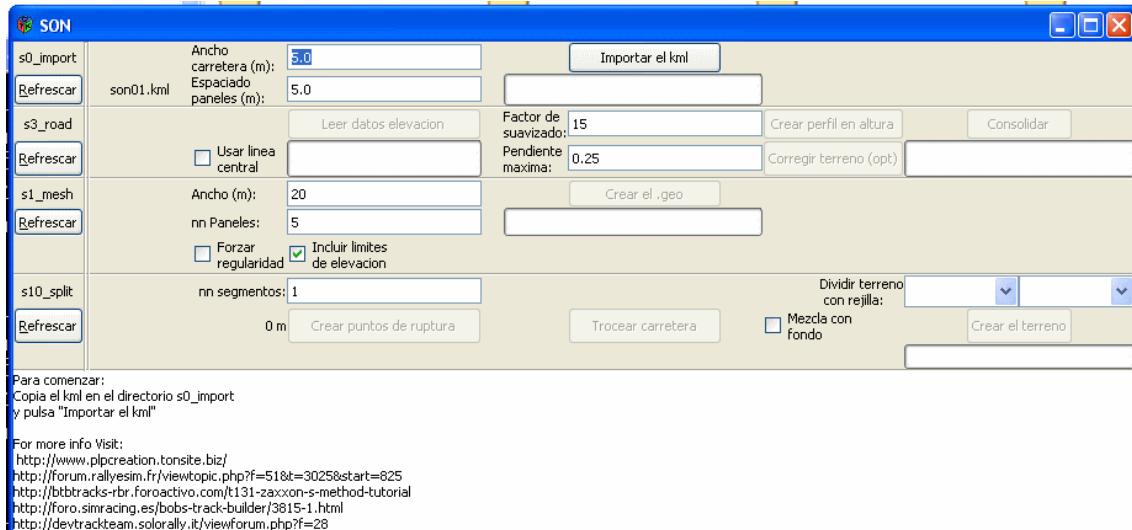


No sigas con el proceso a menos que estés seguro de que el mallado está cubierto por completo por los datos de elevación.

Una vez que hemos procesado al father (al menos parcialmente), tenemos que procesar los sons. Los pasos son similares pero no idénticos, pues en los sons no se gestionan datos de altura.

```
> cd ..  
> cd son01  
> zgui
```

La ventana del GUI es similar a la del FATHER, pero deberíamos leer “SON” en la barra superior.



El SON tiene el paso “s10_split”, un paso sencillo en el que se trocea la carretera en varios trozos por motivos de eficiencia. Segmentos de alrededor de 1Km parecen funcionar bien. Mayores longitudes pueden hacer que tanto BTB como RBT vayan realmente lentos.

A veces los SONs son realmente cortos y usar factores de suavizado normales en el paso s3_road hace que los scripts fallen (si por ejemplo el SON tiene solo 7 anchors y seleccionamos un factor de suavizado de 13 no habrá suficientes puntos para hacer los cálculos). Seleccionar un factor de suavizado pequeño (incluso 1 llegado el caso) a menudo resuelve el problema.

Tenemos que **procesar todos los SONs**.

Una vez hayamos procesado todos los SONs tenemos que juntar todas las rutas: el FATHER y los SONs:

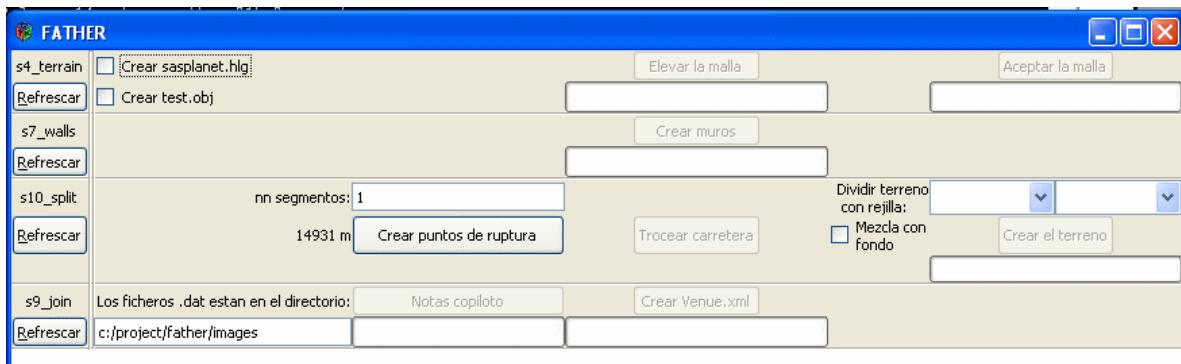
```
> cd c:\project\father\s1_mesh
> join_geos
```

La salida es c:\project\father\s1_mesh\joined.geo, un fichero que debe ser procesado con gmsh para crear el fichero .msh que necesitamos.

Procesar joined.geo no es un paso trivial así que no se explica en esta sección. La salida del proceso debería ser el fichero **joined.msh**, un mallado gmsh con 2 Physical Surfaces: 111 para la parte conducible y 222 para la no-conducible. Se puede leer más sobre gmsh en la sección VII. Se recomienda encarecidamente ver un videotutorial (sección XIII.1).

Una vez tenemos el fichero joined.msh en la carpeta s1_mesh\salida, podemos proseguir con los scripts, lanzando zgui(1) para el FATHER.

```
> cd c:\project\father
> zgui(1)
```



s4_terrain

Las opciones en el paso s4_terrain son:

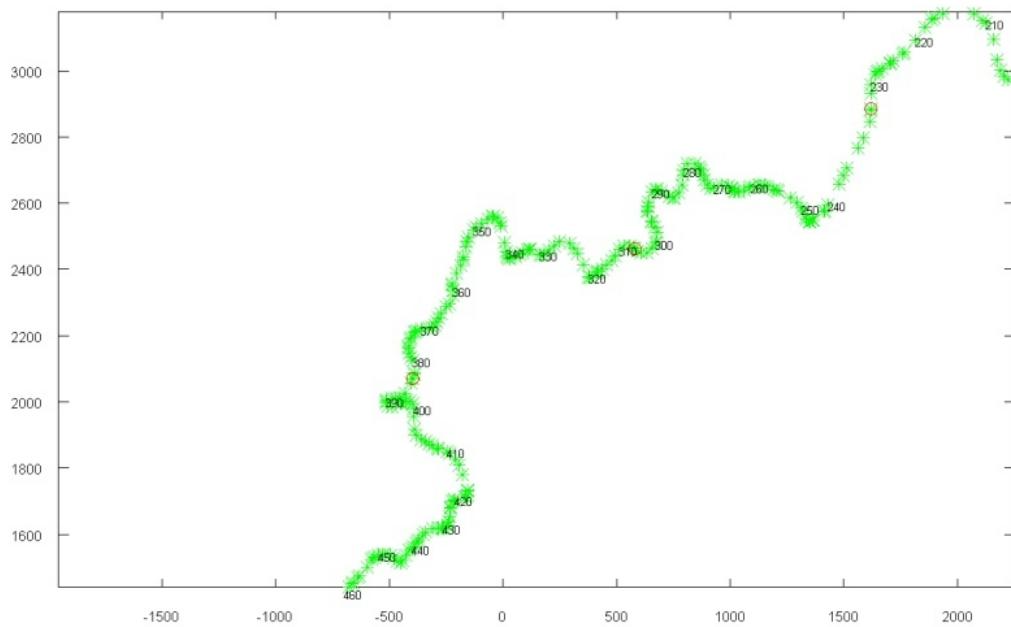
- 1) Crear sasplanet.hlg. Hay que crear este fichero si se quieren incluir imágenes de satélite en el proyecto. Este fichero está preparado para ser abierto con SASPLANET. Se puede leer más en la sección VIII.
- 2) Crear test.obj. Se crea un objeto 3D (formato .obj) en la carpeta s4_terrain\salida. Este objeto se puede emplear para comprobar la malla creada o quizás sea el resultado final que buscábamos de los scripts.

s7_walls

Se crean muros invisibles que impiden alcanzar la zona no conducible. Es opcional. Solo tiene sentido para RBR, no para rFactor.

s10_split

En primer lugar creamos puntos de ruptura (script split_track(n), siendo n el número de segmentos que queremos tener). Este botón crea un fichero llamado '**pos_nodes.txt**' que contiene los números de los nodos en los que la carretera cambia de segmento. También se muestra una gráfica que nos permite comprobar la situación de dichos nodos desde una vista en planta del trazado. Si queremos usar otros puntos de ruptura basta con **editar pos_nodes.txt** y luego pulsar en el siguiente botón "Dividir la carretera". El segundo botón (llama al script partir_track) usa pos_nodes.txt como entrada por lo que este fichero determina en última instancia cuántos segmentos tenemos (y dónde empiezan y acaban).



Una vez la carretera se ha troceado (tramos de 1Km parecen adecuados por razones de eficiencia), hay que trocear el terreno empleando una rejilla MxN grid. Si tenemos imágenes de fondo y las vamos a usar como textura para el terreno debemos emplear exactamente las mismas dimensiones MxN que hemos usado para trocear las imágenes. Se recomienda trocear por razones de eficiencia pero los usuarios de rFactor deben saber que demasiados polígonos (>1000) en un área de terreno puede hacer que BTB no consiga exportar el proyecto para rFactor.

Si se selecciona la opción “Mezclar con fondo”, las áreas de terreno tendrán las imágenes de satélite como textura. Leer sección VIII para ampliar.

s9_join

Si se desea tener imágenes de satélite como imágenes de fondo en BTB, hay que indicar el directorio correcto donde están los ficheros .dat (leer sección VIII para ampliar). En caso contrario hay que escribir un directorio que no exista. Si se selecciona la opción “Mezclar con fondo” en el paso s10_split SE DEBEN tener las imágenes de fondo, pues de lo contrario es imposible abrir el Venue.xml con BTB .

Finalmente se pueden añadir notas de copiloto a la ruta principal y crear el s9_join\salida\Venue.xml, el resultado final del proceso. Buscar la carpeta My Projects en el directorio de instalación del BTB. Crear una carpeta llamada NAME. Copiar Venue.xml dentro de NAME. Crear una carpeta dentro de NAME, llamada XPack y copiar 3 ficheros dentro: default.zip y common.zip (copiar de otros proyectos de los que hay en My Projects) y WP.zip. También hay que añadir los archivos .dds si se usan imágenes de fondo (ver capítulo VIII).

IV.2 Proyectos one-track

Los proyectos que solo tienen una ruta son una versión simplificada de los multi-track: básicamente los mismos pasos pero no hay que procesar los SONs. En lugar de crear joined.msh a partir de joined.geo, crearemos anchors_carretera.msh a partir de anchors_carretera.geo.

Cuando se trabaja con una sola ruta deberíamos leer la palabra "NORMAL", en lugar de "FATHER"/"SON" en el GUI.

V TRABAJAR CON DIFERENTES FUENTES DE DATOS DE ELEVACIÓN

V.1 Google Earth

Es posible dar elevación a los ficheros gridXXX.kml usando un wrapper de Python para Google Earth COM API. También se pueden emplear BTBLofty o 3DRouteBuilder, pero el proceso será más tedioso (y puede que más caro):

Hay varias opciones para dar elevación a los ficheros gridXXX.kml:

- 1) Comprar 3D Route Builder. A menos que se paguen 15 euros, el límite por fichero son 200 puntos
- 2) Usar BTBLofty. Funciona bien en Windows XP, pero no en Windows7. Aun así, en Windows XP hay que procesar los ficheros a mano: uno tras otro empleando una gran cantidad de tiempo.
- 3) Usar el script raise_kml. Hace lo mismo que las dos opciones anteriores. Funciona en Windows XP y Windows 7 y procesa automáticamente todos los ficheros gridXXX.kml.

Para usar raise_kml hay que realizar los siguientes pasos:

Instalar Python

1. Instalar Python 2.7 en el directorio <C:\Python27>. NO INSTALARLO EN UN DIRECTORIO DIFERENTE
<http://python.org/ftp/python/2.7/python-2.7.msi>
2. Instalar pywin32
<http://sourceforge.net/projects/pywin32/files/pywin32/Build216/pywin32-216.win32-py2.7.exe/download>
3. Instalar pygoogleearth
<http://pypi.python.org/packages/any/p/pygoogleearth/pygoogleearth-0.0.2.win32.exe#md5=7e92b3cf1dcb4a5493aacb393a9e54a2>
4. Editar el fichero c:\Python27\lib\site-packages\pygoogleearth\geapplication.py cambiando (línea 231)
return gehelper.point_dict_from_terrain_point(terrain_point)

por:

```
return terrain_point
```

V.2 AGR Data

Se aceptan ficheros ASCII Grid con formato:

```
NCOLS 601
NROWS 401
XLLCORNER 714000
YLLCORNER 4328800
CELLSIZE 25
NODATA_VALUE -999
33.381 33.279 33.102 32.982 32.809 .....
```

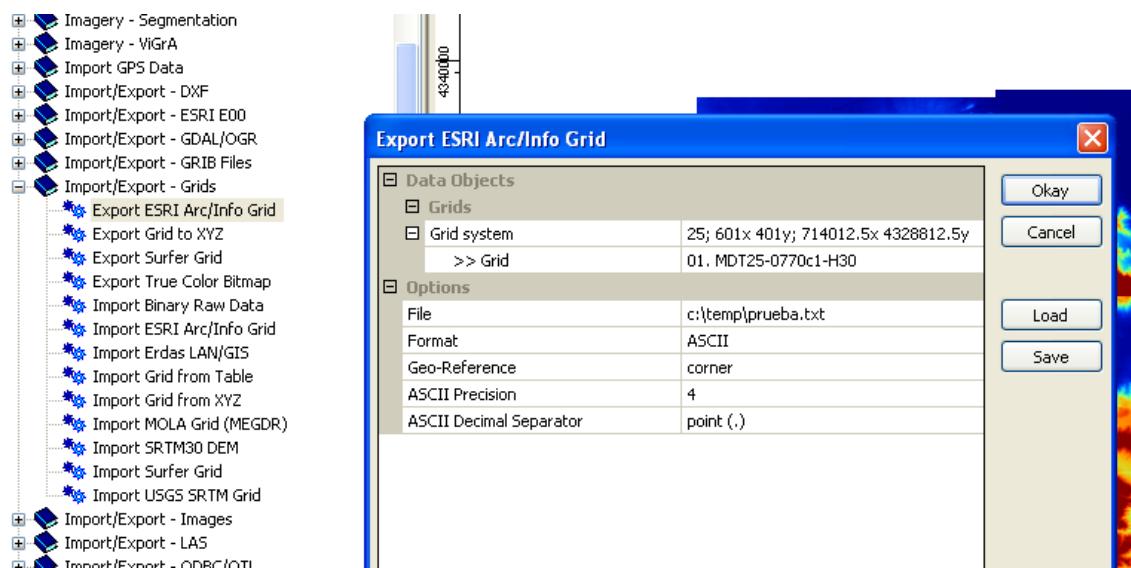
. Las coordenadas deben ser **UTM** (*Universal Transverse Mercator*).

Uso:

- 4) Crear una carpeta llamada “agr” en la misma carpeta en la que está "project" (o las carpetas del father's y los sons en los proyectos multi-track). Ejemplo: si trabajamos en un proyecto one-track en c:\project, crearemos la carpeta c:\agr.
- 5) Copia los ficheros .agr a la carpeta creada en el paso 1). Renombra esos ficheros con extensión .agr si es necesario.
- 6) Usa los scripts con normalidad. Siguiendo los pasos recomendados en la pantalla el paso s2_elevation hay que saltárselo, pues ya tenemos datos de altura.

NOTA: si se tienen datos con un formato diferente se puede emplear software libre para convertirlos (como SAGA-GIS, donde se importa la rejilla y se selecciona el *Module Import/Export - Grids | Export ESRI Arc/Info Grid*, usando formato ASCII y corner geo-reference).

<http://sourceforge.net/projects/saga-gis/>



NOTA: para España se pueden descargar de forma gratuita datos .agr con 5 o 25m de resolución. Basta crear un usuario y descargar datos MDT05 o MDT25.

<http://centrodedescargas.cnig.es/CentroDescargas/>

De la siguiente gráfica se puede sacar el código adecuado a la zona que nos interesa:
http://centrodedescargas.cnig.es/CentroDescargas/equipamiento/cuadricula_MTN50.png

The search interface has a header 'Búsqueda Avanzada'. It includes three main search fields: 'Selección de Productos' (set to 'Modelo Digital del Terreno - MDT25'), 'Selección División administrativa' (dropdown menu), and 'Selección Hoja del MTN50' (set to '70'). Below these is a link 'Ver cuadricula del MTN50'. At the bottom is a 'Ver lista de Productos' button.

The results page shows a table with the following data:

Producto	Archivo	Formato	Tamaño(MB)	Descargar
Modelo Digital del Terreno - MDT25	MDT25-0070c4.zip	AGR	1,32	
Modelo Digital del Terreno - MDT25	MDT25-0070c1.zip	AGR	1,32	
Modelo Digital del Terreno - MDT25	MDT25-0070c2.zip	AGR	1,35	
Modelo Digital del Terreno - MDT25	MDT25-0070c3.zip	AGR	1,30	

NOTA: para Cataluña se pueden descargar datos con 15m de resolución de:
http://www.icc.cat/vissir2/?lang=es_ES

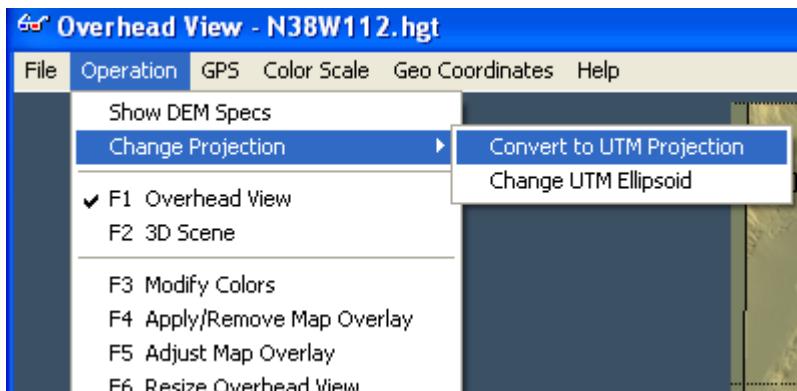
clickando en el mapa y seleccionando "otras". El formato es exactamente el mismo que usado por el cnig, pero la resolución es diferente. La extensión de los ficheros debe cambiarse de ".txt" a ".agr".

NOTA: Los ficheros MDT05 pueden ser demasiado grandes para usarlos directamente con los scripts. El script split_agr puede usarse para trocear un fichero MDT05 en ficheros más pequeños. Por ejemplo, para trocear **MDT05-0667-H30.ASC** y crear 5x5 ficheros (con extensión .AGR):

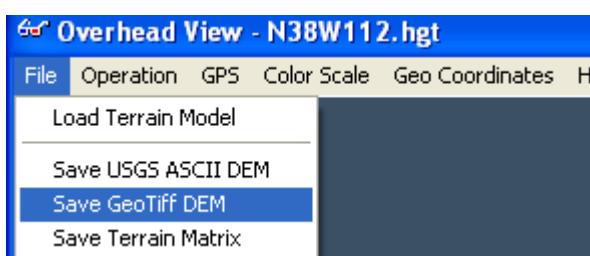
```
split_agr('MDT05-0667-H30',5)
```

NOTA: los ficheros .hgt se pueden abrir con 3dem para proyectarlos a UTM (Operation\Change Projection)

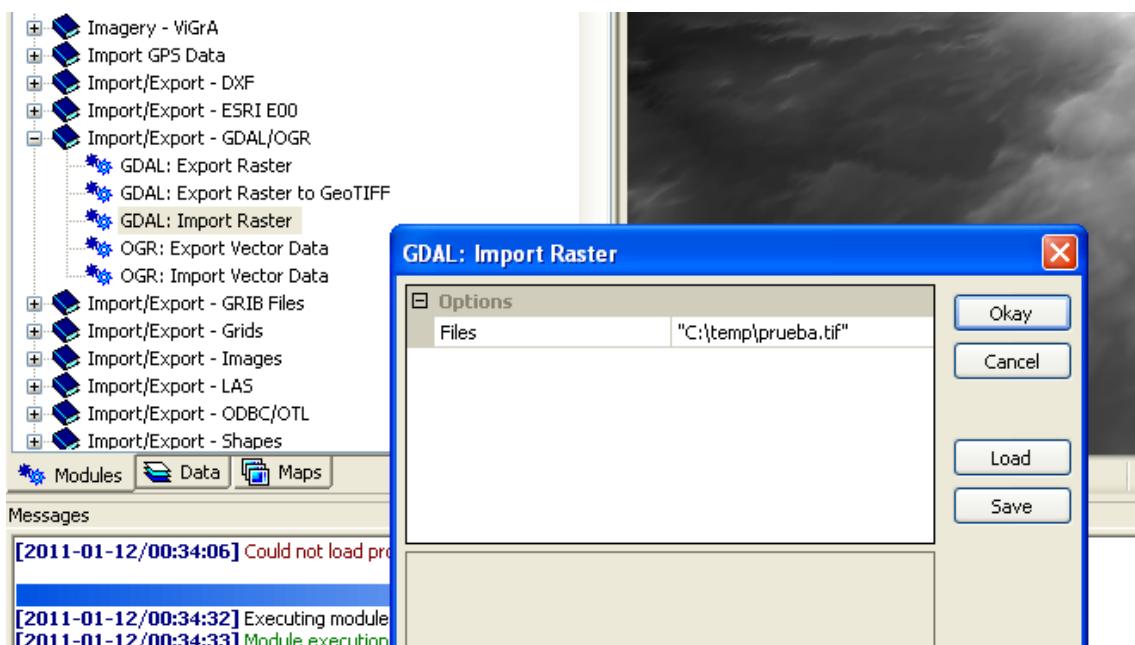
<http://www.viewfinderpanoramas.org/3dem.zip>



Luego se guardan con formato Geotiff dem:



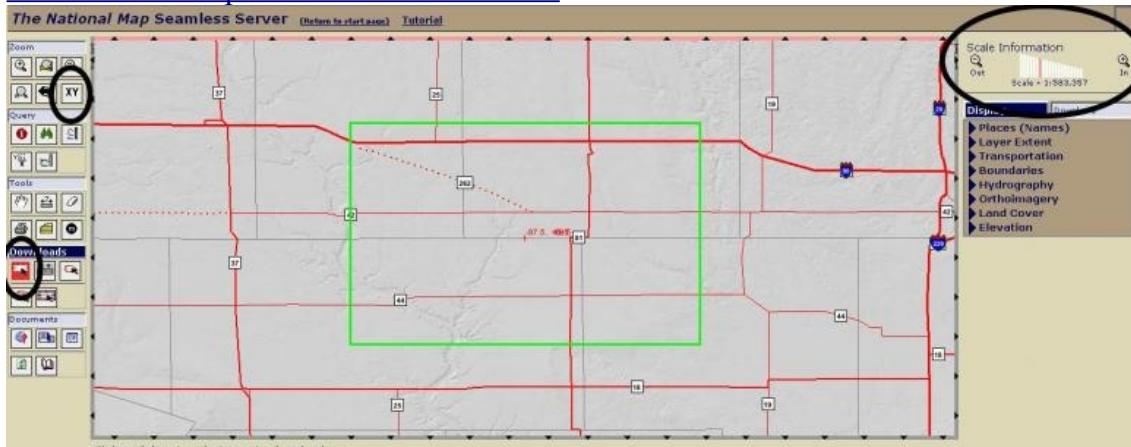
Y se importan con SAGA GIS:



V.3 Seamless server

Esta web ofrece datos cada 10 o 30m para USA y 90m (y a menudo con algunos puntos sin altura) para el resto del mundo.

[The National Map Seamless Server Viewer](#)



Para comprobar el tipo de datos que ofrecen para nuestra zona de interés seleccionamos la opción **Zoom->XY** y escribimos las coordenadas de un punto. Luego hacemos zoom para ampliar (parte superior derecha de la pantalla). Con el **Downloads->Rectangle** enmarcamos la zona de interés. Si el rectángulo es demasiado grande, se pondrá rojo. En la ventana pop-up clickamos en el botón **Modify Data Request**. Buscamos la sección **Elevation** y seleccionamos la fuente de datos,. Por

ejemplo "**National Elevation Dataset (NED) 1/3 Arc Second**" y elegimos formato "**GRIDFLOAT**". Clickamos en el botón "**Save Changes & Return to Summary**" y si no hay ningún problema el botón "**Download**" aparecerá ante nosotros. Es una forma sencilla y rápida de obtener buenos datos de elevación. Si no nos interesa un tramo de USA siempre podemos recurrir a GE a través de 3DRoute Builder, BTBLofty o raise_kml.

Nota:

3 arcosecond \approx 90m

1 arcosecond \approx 30m

1/3 arcosecond \approx 10m

Los datos en formato gridfloat se pueden usar para crear lamalla.mat mediante el script **leer_gridfloat**. Este script no puede usarse hasta que se haya completado el paso s0_import.

Ejemplo:

```
> cd ..\s2_elevation  
> leer_gridfloat("file.hdr","file.flt")
```

V.4 Ficheros hgt

Ejemplo:

```
> cd ..\s2_elevation  
> lee_hgt("N41W009.hgt",[41 42],[-9 -8])
```

Los ficheros .hgt pueden transformarse en lamalla.mat usando el script **lee_hgt**. Este script no puede usarse hasta que el paso s0_import se haya completado.

VI AJUSTE FINO DEL PERFIL EN ALTURA DE LA CARRETERA

Hay dos acciones avanzadas relacionadas con el perfil en altura de la carretera:

- 1) Retocar a mano el perfil propuesto por los scripts
- 2) Cambiar el terreno para que se adapte a la carretera que hemos creado

VI.1 Retocar a mano el perfil

Hay un tutorial en javascript (ver sección XIII.1).

Ejecutar dar_altura con normalidad

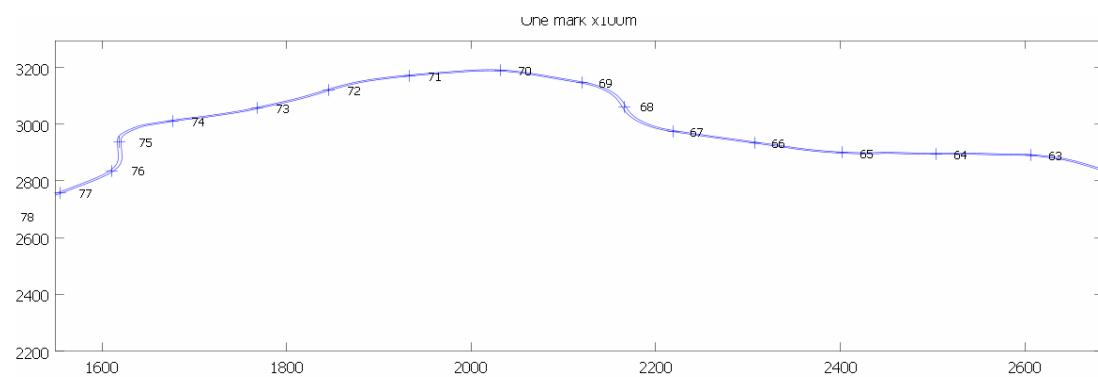
```
> dar_altura(15,0.3,-0.3)
```

```
> dar_altura(15,0.3,-0.3)
Leyendo ..\anchors.mat
Leyendo alturas_track1.mat
Leyendo el fichero retoques.txt
Cerrando el fichero
    e-_ end          n-_ new segment
```

Ahora **dar_altura** espera a que el usuario teclee “n” o “e” y luego <ENTER>

- “n” si se quiere modificar un tramo del perfil en altura
- “e” si hemos terminado la edición

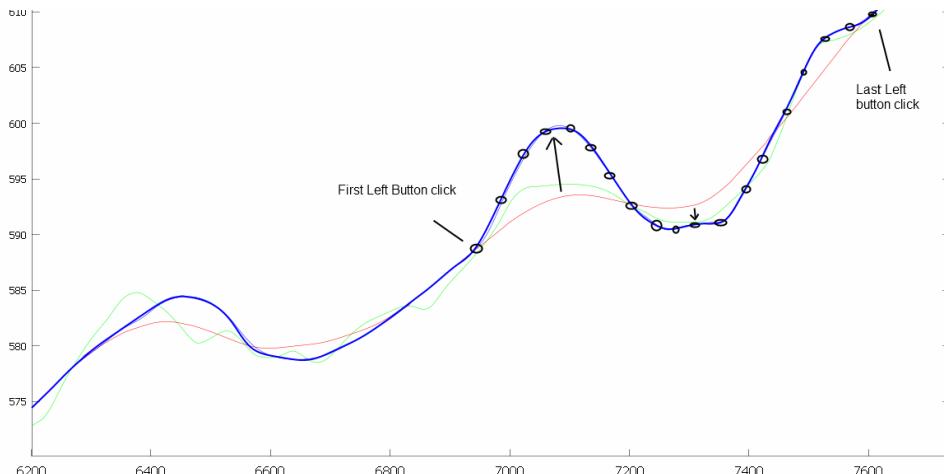
Pero **antes de pulsar “n”, debemos ampliar la zona de interés.** Y antes de ampliar probablemente necesitaremos echarle un vistazo a la otra gráfica creada por dar_altura. Muestra la distancia recorrida en cada punto de la ruta (multiplicar por 100 para tener la distancia en metros)



Si por ejemplo quieras cambiar el perfil en altura entre las posiciones 6800m y 7600m, hay que hacer un zoom de esa zona (se selecciona con el botón derecho) en la otra figura. Tras hacer el zoom se puede pulsar “n” en la ventana de texto de octave. El mensaje en la ventana de texto debería cambiar:

```
Leyendo el fichero retoques.txt
Cerrando el fichero
      e-_ end          n-_ new segment
n
Left click to add point. Right click to finish
```

Ahora debemos hacer click en la gráfica de elevación con el botón izquierdo del ratón. **De izquierda a derecha**. Se puede usar cualquier número de puntos (≥ 3). Para acabar basta con hacer click en cualquier parte de la ventana con el botón derecho del ratón.



Se nos vuelve a pedir que escojamos “e” o “n”.

Si pulsamos “e” el script dar_altura termina su labor. Dentro de “s3_road\salida” se puede encontrar un fichero “Venue.xml” que se podría abrir con BTB para comprobar la forma de la carretera.

Importante: Todos los retoques manuales se graban en un fichero llamado “retoques.txt”. Es un fichero de texto que podría editarse a mano llegado el momento. Cada línea del fichero se compone por el número de puntos de un segmento y las coordenadas (x,y) de dichos puntos.

Cada vez que se llama a dar_altura ‘retoques.txt’ es leido y su información se aplica al perfil. Los nuevos cambios se añaden al final de ‘retoques.txt’.

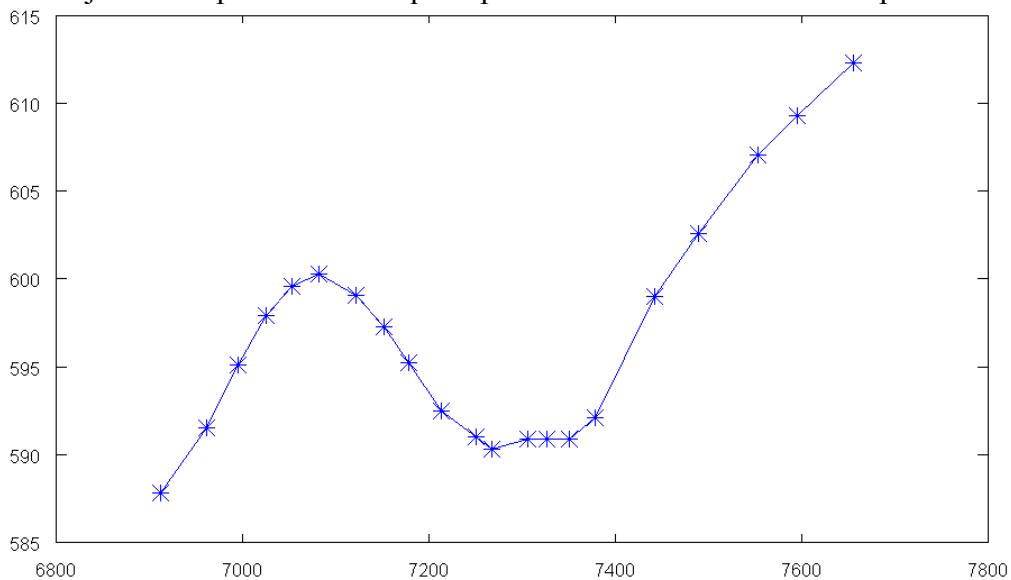
Si se quieren borrar los cambios hechos a mano hay que borrar el fichero “retoques.txt”. Los cambios contenidos en retoques.txt se aplican en el mismo orden en que aparecen en

el fichero.

Ejemplo de línea contenida en retoques.txt:

```
21 6911.3 587.8 6961.6 591.5 6994.3 595.1 7024.5 597.9 7052.2 599.6 7081.2 600.3
7121.5 599.1 7151.7 597.3 7178.1 595.2 7213.4 592.5 7249.9 591.0 7267.5 590.3 7305.2
590.9 7326.6 590.9 7350.6 590.9 7378.2 592.1 7442.4 599.0 7489.0 602.6 7553.2 607.1
7594.7 609.3 7655.2 612.3
```

Si dibujamos los puntos vemos que representan uno de los cambios aplicados:

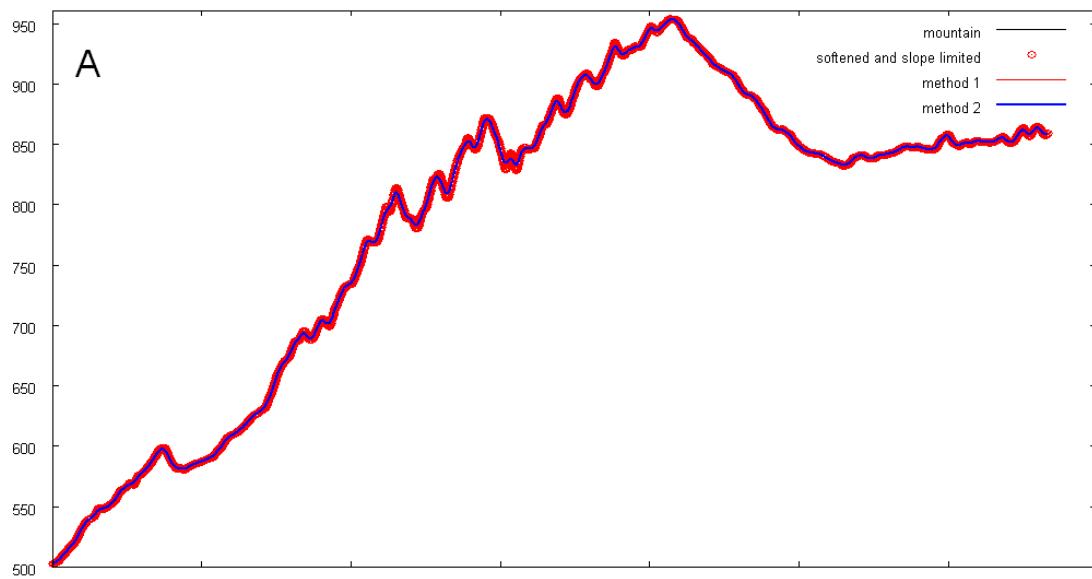


El primer parámetro de dar_altura es un factor de suavizado. Debemos escoger un valor que minimice el número de ediciones manuales requeridas. Siempre debe ser impar.

El perfil final se creará con un spline sacado de un dato de elevación cada 25m. Eso significa que *las pequeñas irregularidades creadas al editar a mano NO son importantes*. Esa distancia (25m por defecto cuando no se usa el GUI) se puede cambiar usando un 4º parámetro en el script dar_altura.

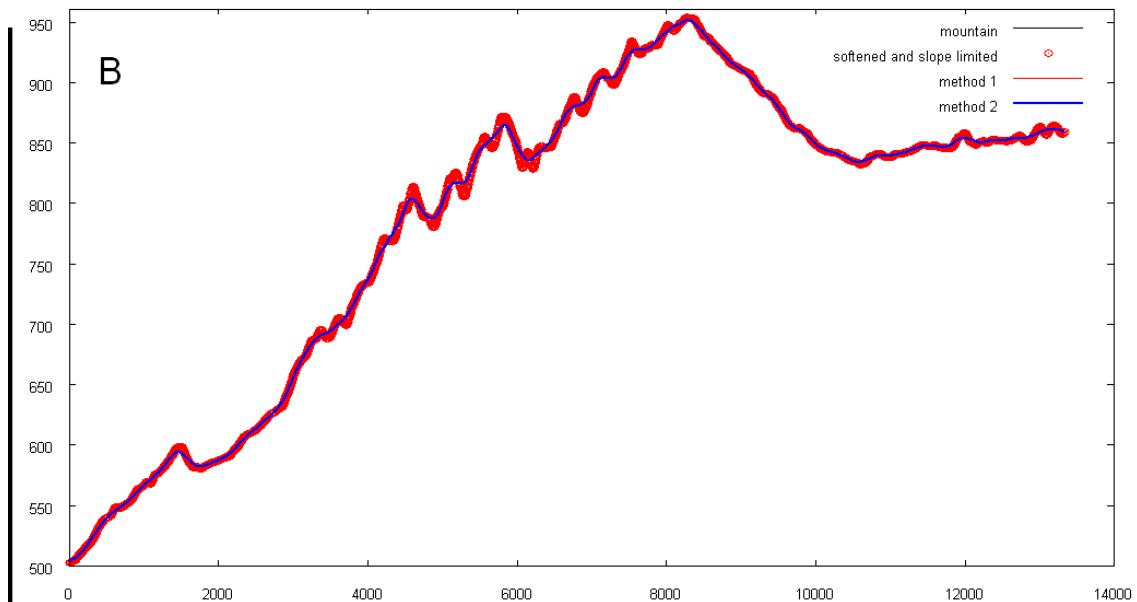
VI.2 Uso del script “corregir”

Supongamos que queremos crear un tramo pero al ejecutar “dar_altura” comprobamos que los datos de altura no son nada buenos y que la carretera sube y baja sin sentido (ver Fig A). Los cambios de elevación se producen obviamente por malos datos de altura (podríamos acudir a Google Earth para confirmarlo).

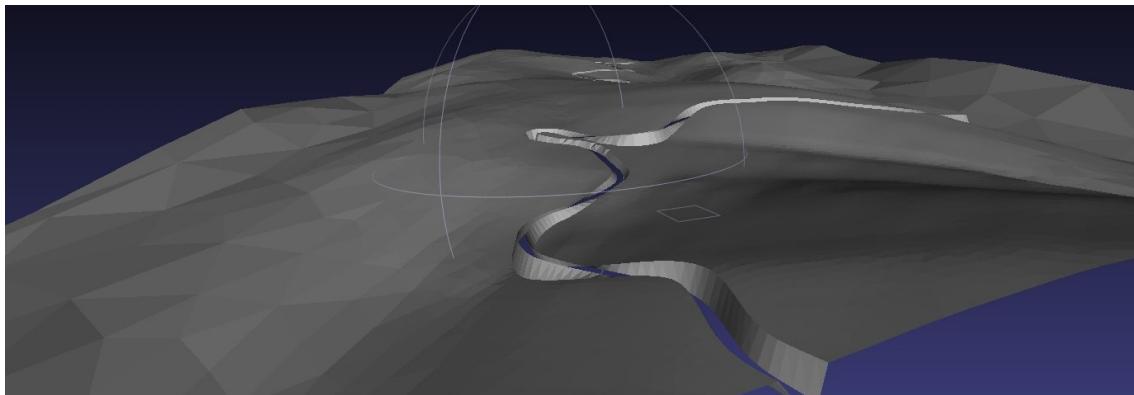


El script “corregir” puede cambiar los datos de altura de la siguiente forma. En primer lugar creamos un perfil suave para la carretera con dar_altura (ver Fig B).

```
> dar_altura(53,1,-1,100)
```



Esta carretera no se ajusta bien a los datos de altura que tenemos. Si siguiésemos con el proceso usando esta carretera, obtendríamos algo como esto:



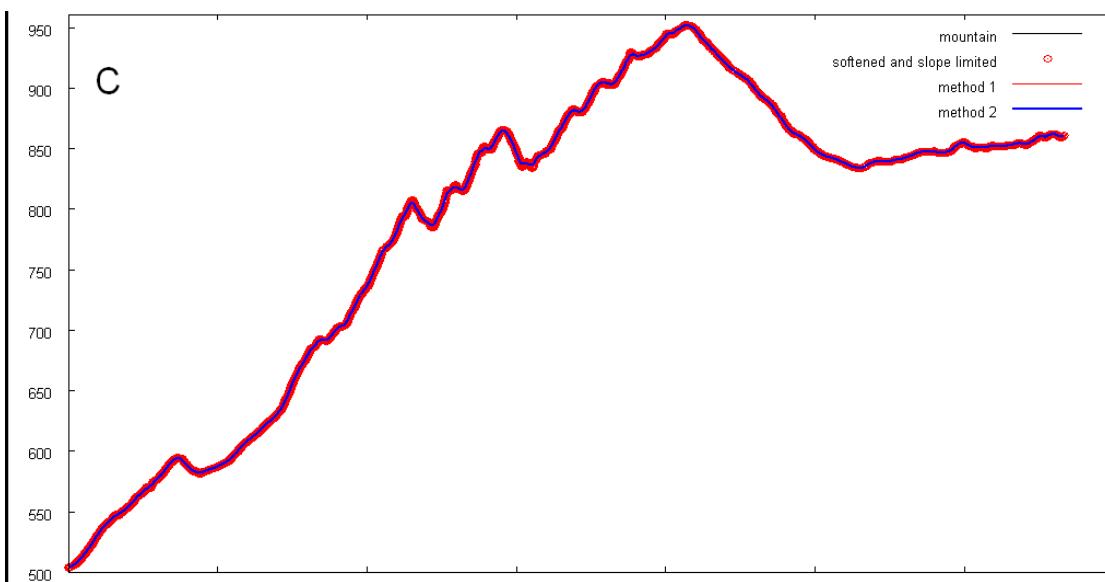
Ahora queremos modificar los datos de altura para que se adapten a nuestra deseada "carretera suave":

> corregir

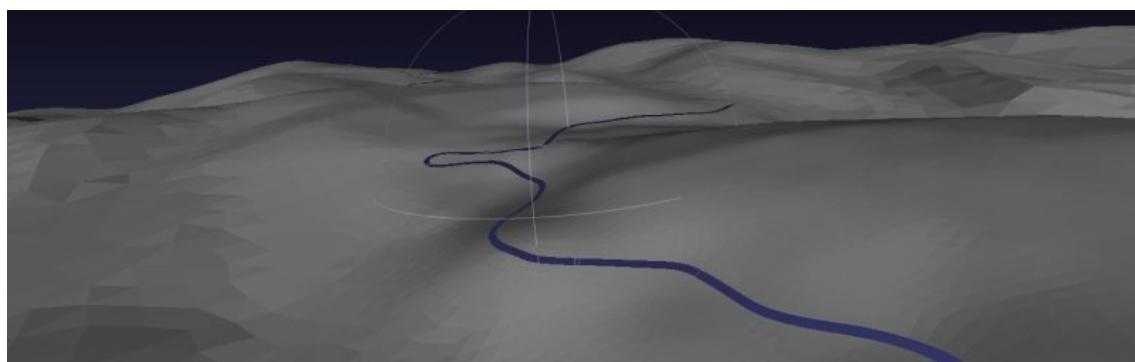
Y ejecutamos de nuevo dar_altura para crear la carretera final, una carretera adaptada a los nuevos datos de elevación:

> dar_altura(21,0.25,-0.25)

Se puede ver (Fig C) que ahora la carretera (curva azul) es prácticamente tan suave como la deseada, y la montaña (círculos rojos) no tiene grandes diferencias en elevación con dicha carretera. Se pueden comparar los círculos rojos y la curva azul en Fig B y Fig C. La carretera es muy parecida, **pero “corregir” ha cambiado los datos de elevación para adaptarlos a la carretera.**



Una vez tenemos una carretera con un perfil en altura que nos guste, podemos seguir con los siguientes scripts. El resultado puede ser tan bueno como:



Scripts que se han ejecutado (los parámetros podrían haber sido diferentes):

```
cd ..\s3_road
s3_coge_datos
creartrack1
dar_altura(53,1,-1,100)
corregir
dar_altura(21,0.25,-0.25)
```

Si se quiere comenzar el proceso desde el principio basta con ejecutar de nuevo s3_coge_datos en s3_road, y tendremos los datos de elevación originales listos para ejecutar creartrack1 y el resto de los scripts. Si no se quiere usar “corregir”, basta con usar dar_altura con normalidad.

VII USANDO GMSH

VII.1 ¿Qué son anchors_carretera.geo y joined.geo?

anchors_carretera.geo y joined.geo son ficheros que deben ser procesados con gmsh.

Contienen:

- TerrainAnchors, los puntos en que carretera y terreno se unen. Estos puntos están unidos con rectas.
- Puntos auxiliares, a ambos lados de la carretera y a una distancia configurable como parámetro en el diseño.
- “Plane surfaces” ya creadas para la zona conducible.

El usuario tiene que definir la superficie no conducible. Ésta es la única parte con cierta dificultad del “método zaxxon”, pero hay varios videotutoriales que muestran los pasos a seguir.

VII.2 Conceptos básicos de gmsh

Si dos puntos de la frontera de una malla están unidos con una recta, ambos puntos formarán parte del mallado.

Si dos puntos de la frontera de una malla están unidos con un spline, esos puntos no necesariamente formarán parte del mallado. gmsh buscará la posición óptima de los nodos de los triángulos en esa frontera.

Cuando se trabaja con gmsh cada punto tiene una "**characteristic length**", que es la longitud deseada para los lados de los triángulos en ese punto. Por ejemplo, en el fichero anchors_carretera.geo (es un fichero de texto), los Anchors de carretera tienen una characteristic length llamada "c11", con un valor de 20m. No obstante los Anchor están unidos con rectas por lo que los triángulos en contacto con la carretera tendrán un lado definido por la posición de los Anchors (típicamente separados 5m). Por defecto los scripts fijan el tamaño de los triángulos mediante un mecanismo denominado thresholds y no empleando characteristic lengths (ver sección VII.8).

VII.3 Controles básicos de gmsh

- Botón derecho del ratón y arrastrar: mover el proyecto
- Ctrl + botón izquierdo del ratón y arrastrar: hacer zoom de una zona
- Clickar 1:1 en la esquina inferior izquierda: zoom ajustado
- Clickar Z en la esquina inferior izquierda: restaurar vista desde arriba
- Botón izquierdo del ratón y arrastrar: rotación 3D
- Rueda del ratón: zoom

VII.4 ¿Cuáles son los límites de la zona no conducible?

La zona no conducible no debería exceder los datos de elevación disponibles. De otro modo scripts como procesar_nodostxt acabarán con errores. Si no has seleccionado en el GUI la opción de ver los límites de los datos de elevación disponibles y deseas verlos en la pantalla de gmsh, ejecuta addgrid en la carpeta s1_mesh:

```
> addgrid(1,1)
```

Donde los dos parámetros son el número de divisiones en horizontal y vertical de la rejilla que se está añadiendo. Este comando solo funcionará si se están usando ficheros lamalla.mat files, como es el caso cuando se sacan datos de altura de Google Earth, pero no si se usan datos AGR. Abre el .geo de nuevo para ver los cambios.

NO se deben alcanzar los límites de los datos de elevación con los mallados hechos en gmsh. De otro modo los scripts fallarán al no encontrar datos de altura para algunos de los puntos del mallado.

VII.5 ¿Cómo debo crear la frontera externa del terreno no conducible?

Para poder mallar primero hay que crear las “Plane surfaces” que mallar. Para crear las plane surfaces primero hay que definir sus límites y para definir sus límites necesitamos antes definir algunos puntos.

Seleccionamos:

Geometry->Elementary Entities->Add->New->Point

Una vez el razón está en la posición deseada para el punto mantenemos pulsado “shift” en el teclado y las coordenadas de congelan. Llevamos el puntero del ratón hasta la ventana “Contextual Geometry Definitions” y allí ponemos la coordenada Z=0 y la Characteristic Length a cl3 (cl3 es un valor definido en la parte inicial de anchors_carretera.geo). Clickeamos en añadir.

Una vez definidos los puntos que vamos a usar, seleccionamos:

Geometry-> Spline

y creamos un spline seleccionando los puntos por los que debe pasar. Una vez cerrada la curva podemos definir la “Plane surface” que tiene el spline como su borde externo y la superficie conducible como su hueco interno.

Geometry -> Plane Surface

Clickamos en la frontera externa hasta que esté toda roja. Si se seleccionado por completo gmsh nos pedirá que seleccionemos el borde del hueco interno (“Select hole boundaries”). Debemos seleccionar todos los segmentos del borde externo de la zona conducible, hasta que todos ellos tengan color rojo. En ese momento la opción deshacer (“undo”) desaparecerá de la parte superior de la ventana de gmsh. Pulsamos “e” para crear la “Plane surface”.

Podemos mallar para comprobar que todo es correcto hasta ahora:

Mesh -> 2D

y todas las plane surfaces ya definidas serán malladas.

Vamos a tener dos tipos de superficies:

- Conducibles: entre los puntos auxiliares y la carretera
- No conducibles: entre la frontera externa y los puntos auxiliares.

Podemos usar la opción Tools->Statistics para averiguar cuántos triángulos tendrá nuestro diseño.

VII.6 ¿Cuál es la salida del proceso?

Si salvamos el mallado, un fichero llamado joined.msh/anchors_carretera.msh será creado. Este fichero contiene dos tipos de líneas:

1) Nodos. Por ejemplo el nodo #14:

14 2149.778 5113.46 0

2) Triángulos (gmsh les llama elements, y BTB faces). Por ejemplo el triángulo 21222

pertenece a la physical surface de los conducibles (111) y está definido por tres nodos: 1490, 17130 and 1491:

```
21222 2 3 111 31672 0 1490 17130 1491
```

VII.7 ¿Cómo se definen las Physical Surface?

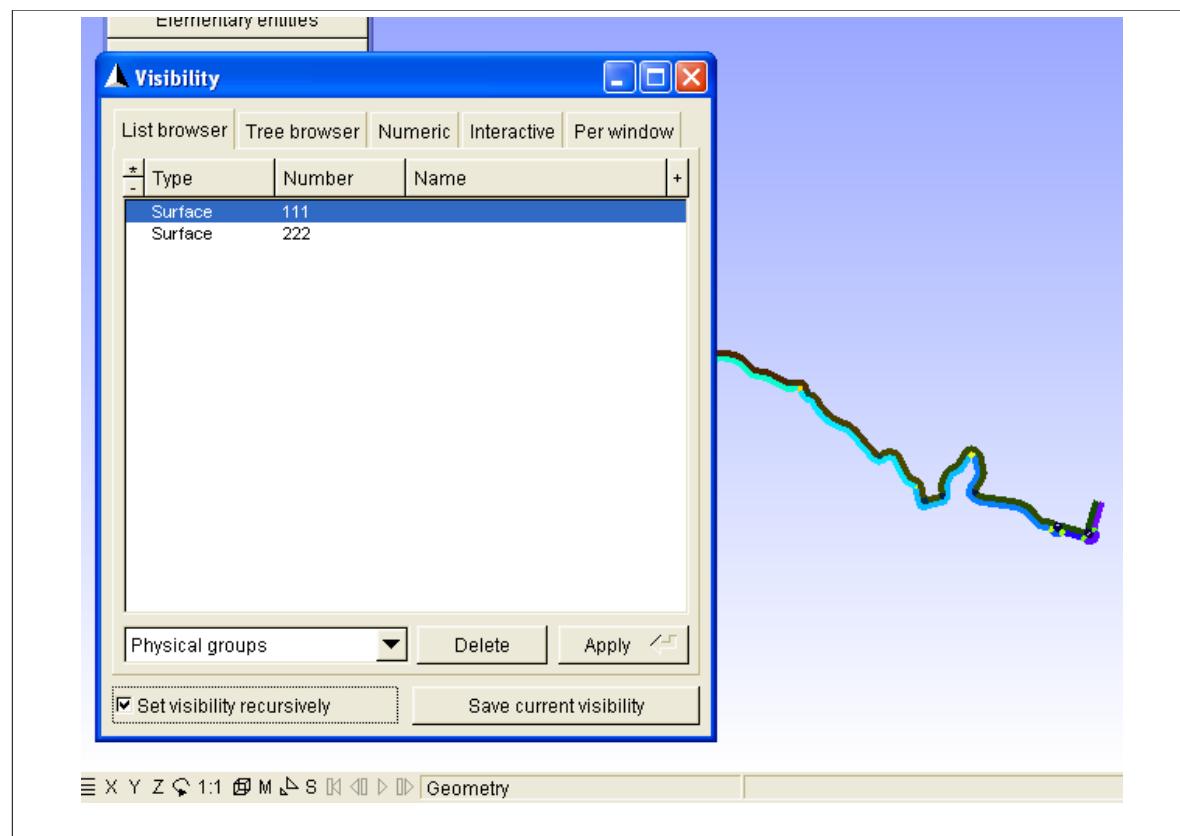
Tenemos que crear 2 physical surfaces, conducible (numerada 111) y no conducible (numerada 222). Cada una de ellas se define como la lista de “plane surfaces” que pertenecen a esa physical surface.

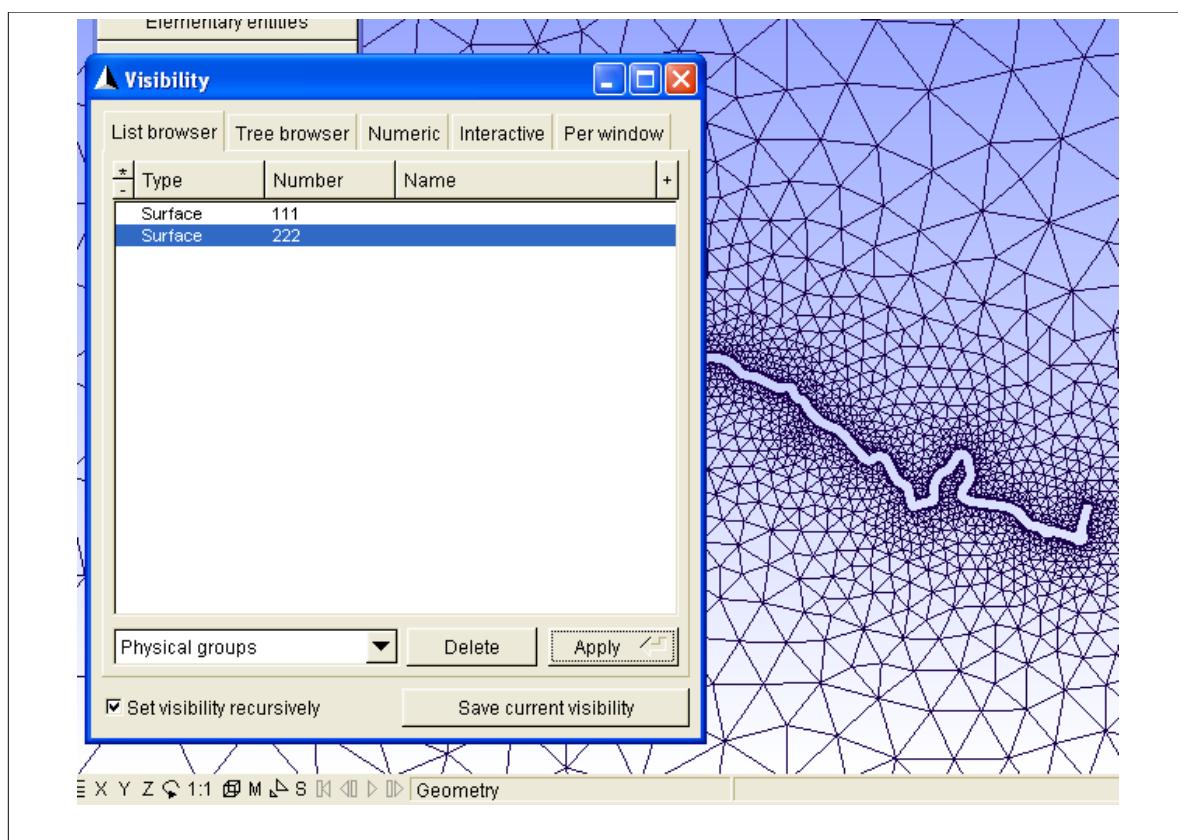
La lista inicial de plane surfaces para la Physical Surface 111 es el contenido del fichero salida\phys111.txt (la lista está ahí desde que los scripts permitieron los proyectos multi-track). Si tu proyecto no tiene rutas adicionales, simplemente añade el contenido de phys111.txt al final de anchors_carretera.geo:

“Physical Surface(111)= {contenido de phys111.txt};”.

Añadimos “Physical Surface(222)={X};” al final del fichero, donde X es el código de la superficie no conducible, la que en su declaración tiene dos números a la derecha del igual (suele ser la última Plane Surface definida en anchors_carretera.geo).

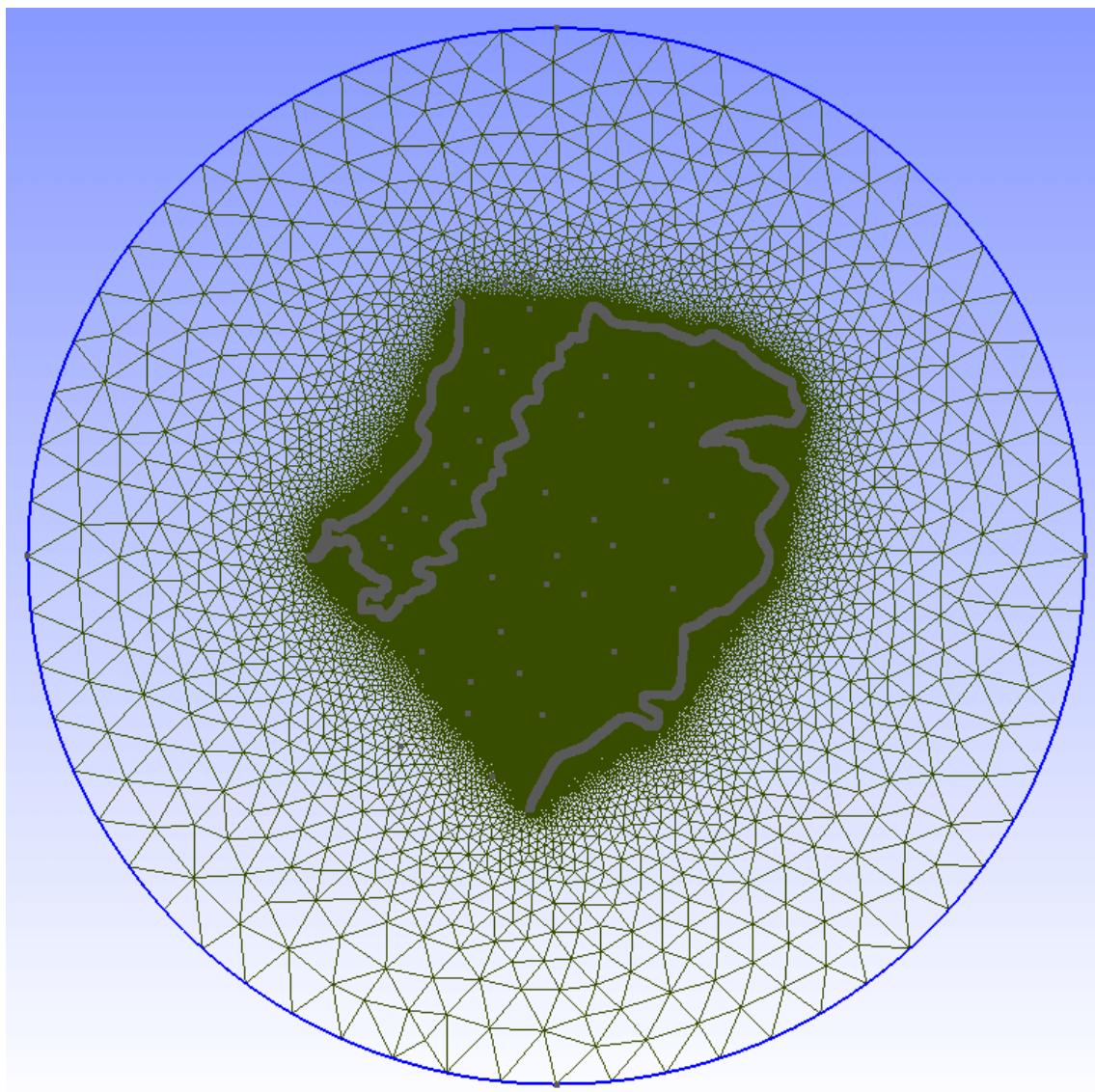
Solo las plane surfaces que pertenecen a una physical surface llegarán al .msh, así que es importante prestar atención a la hora de definir las physical surfaces. **Una vez que se ha terminado de mallar se recomienda abrir anchors_carretera.msh/joined.msh con gmsh para comprobar que todas las mallas están ahí.** Usar la herramienta Visibility (y en ella la opción de Physical Groups en la pestaña de List Browser) puede ayudar en la comprobación pues permite seleccionar la superficie física que se quiere ver.



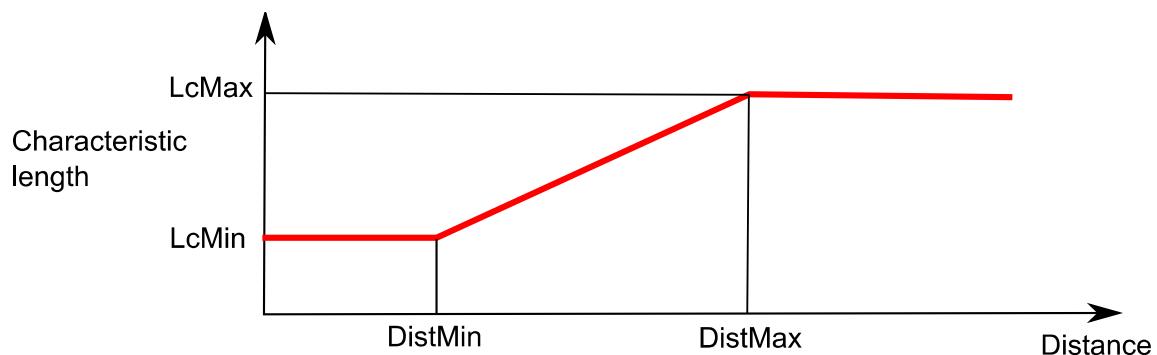


VII.8 Los thresholds en gmsh

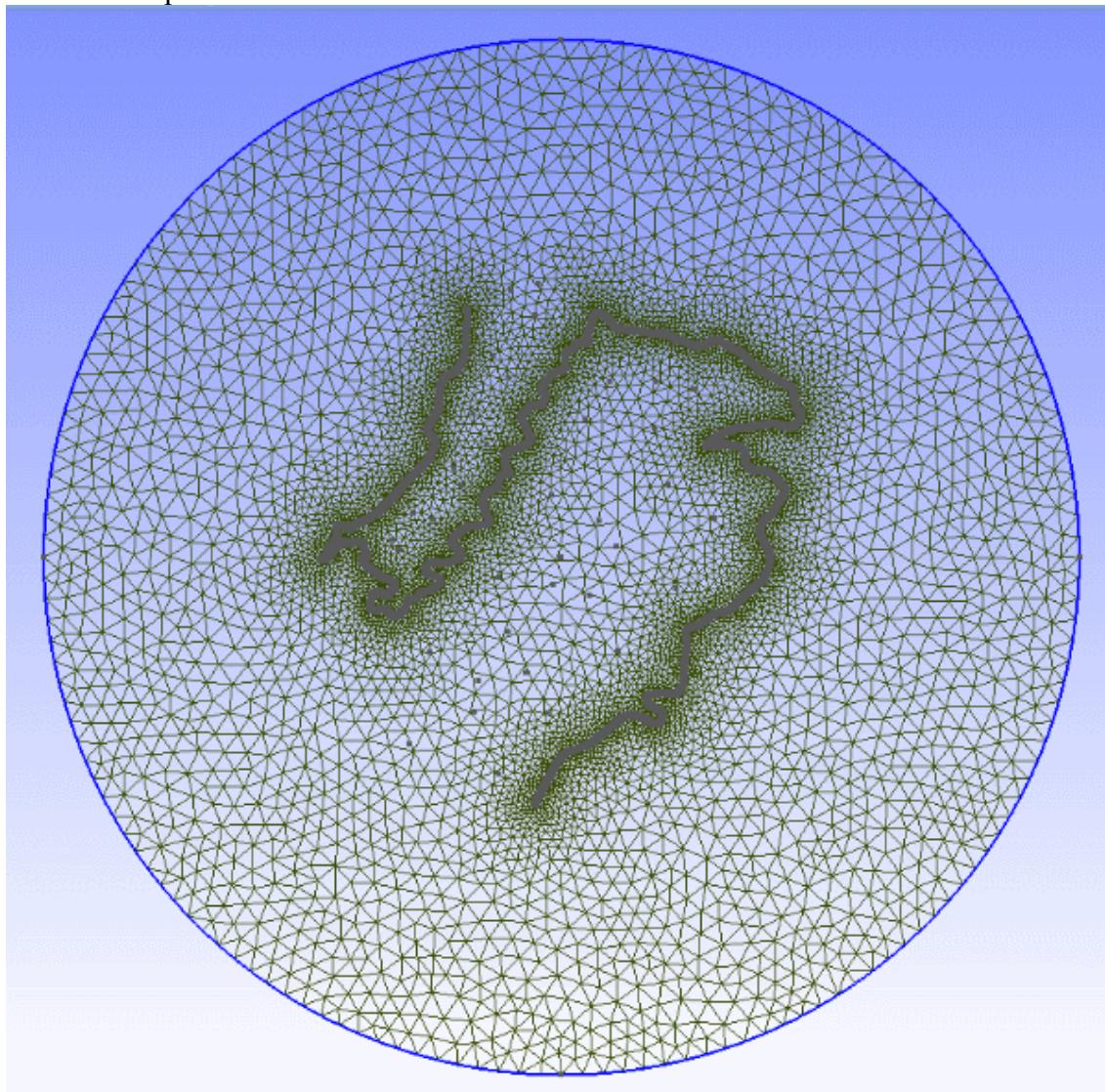
Una forma de controlar el tamaño de los triángulos en gmsh es asignar a los puntos una longitud característica (characteristic length) (4º parámetro en la definición de los mismos). El problema de las longitudes características es que gmsh crea demasiados triángulos en las rutas (o segmentos de ruta) con forma de U. Por ejemplo:



Otra forma de controlar el tamaño de los triángulos son los *threshold fields* (*campos de umbrales*). Por ejemplo se puede pedir a gmsh que asigne a cada triángulo un tamaño que sea función de la distancia de ese triángulo a la carretera.



El resultado puede ser tan bueno como se muestra:



El código que usa gmsh para un threshold field es:

```

Field[1] = Attractor;
Field[1].NodesList = {1:3342}; //List of point of the track

Field[2] = Threshold;
Field[2].IField =1;
Field[2].LcMin = 20;
Field[2].LcMax = 2000;
Field[2].DistMin = 1;
Field[2].DistMax = 10000;
Field[2].StopAtDistMax = 0;

Mesh.CharacteristicLengthExtendFromBoundary = 0;

Background Field=2;

```

Los scripts mallado_regular o join_geos insertan (en anchors_carretera.geo o joined.geo) el código necesario para usar threshold fields. Está **activo por defecto**. Si no se quiere tener los thresholds activos, basta con abrir el fichero .geo con un editor de textos y buscar el código **If (1)** y cambiarlo por **If (0)**.

Solo puede haber un threshold field activo a menos que se use un Min Field (leer notas al final de este documento). El threshold que se inserta automáticamente usa todos los puntos definidos hasta ese momento en el .geo, por lo que todos los tracks son considerados en el threshold en los proyectos multi-track.

Los parámetros por defecto en el .geo son (ver s1_mesh\thresholds.txt):

```

LcMin = 20; //(Characteristic length at a distance LcMin and below)
LcMax = 2000; //(Characteristic length at a distance LcMax and above)
DistMin = 1;
DistMax = 10000;

```

NOTA:

También es posible cambiar los parámetros del threshold desde los menús de gmsh:

Seleccionar **Mesh > Define > Fields**.

En la ventana Fields clickar en **Threshold**.

Por ejemplo podemos reducir el tamaño de los elementos cerca de la frontera exterior:

1) Enter LcMax = 1000

2) Apply

3) Clickear mallado 1D.

Nota: hay que hacer mallado 1D antes del 2D cada vez que se altera un parámetro de los

Threshold.

4) Clickear mallado 2D.

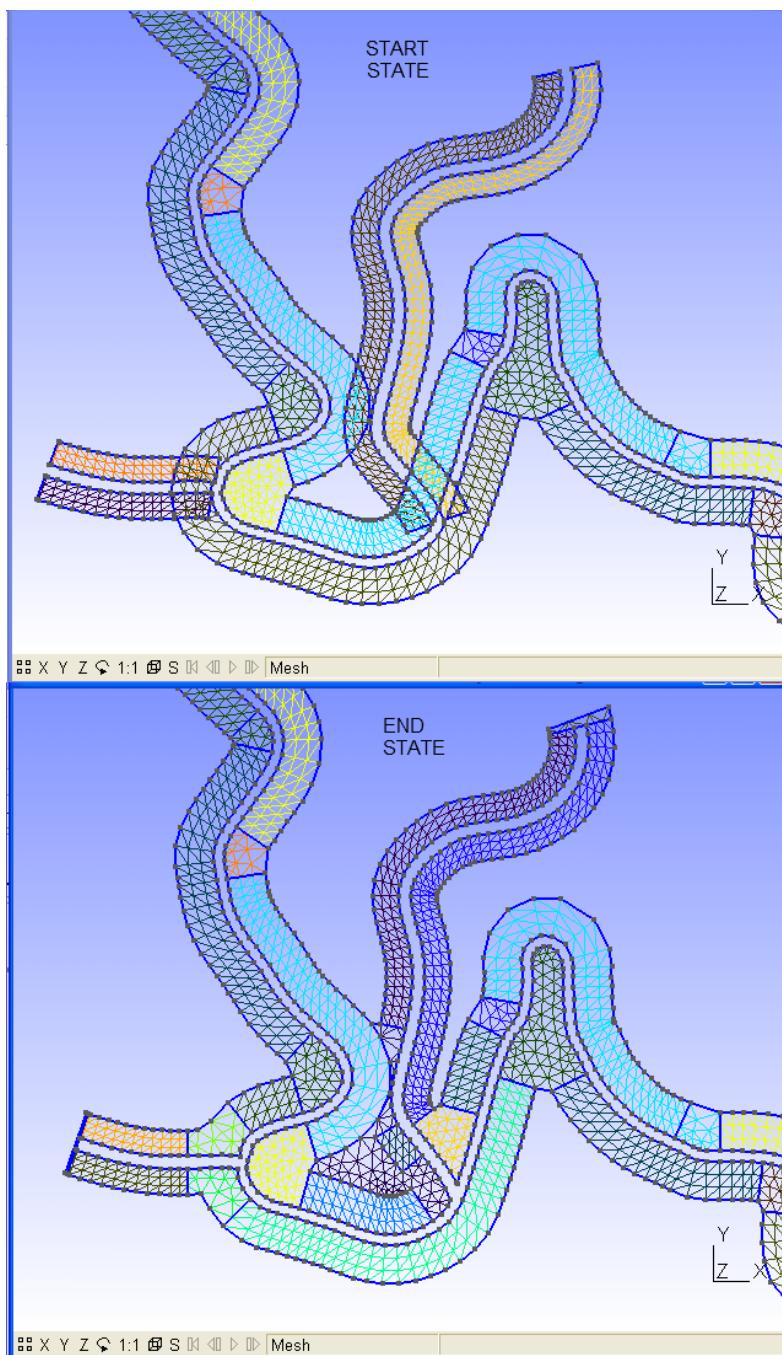
VII.9 Procesar joined.geo con gmsh

Cuando abrimos joined.geo con gmsh, veremos que las superficies conducibles de los SONs se solapan con las del father. Eso hay que arreglarlo a mano empleando gmsh y un editor de textos.

Los pasos a seguir son:

- 1) Arreglar el solapamiento de superficies conducibles
- 2) Crear la Physical Surface(111)
- 3) Crear la superficie no conducible y la Physical Surface(222)
- 4) Comprobación final

1) Arreglar el solapamiento



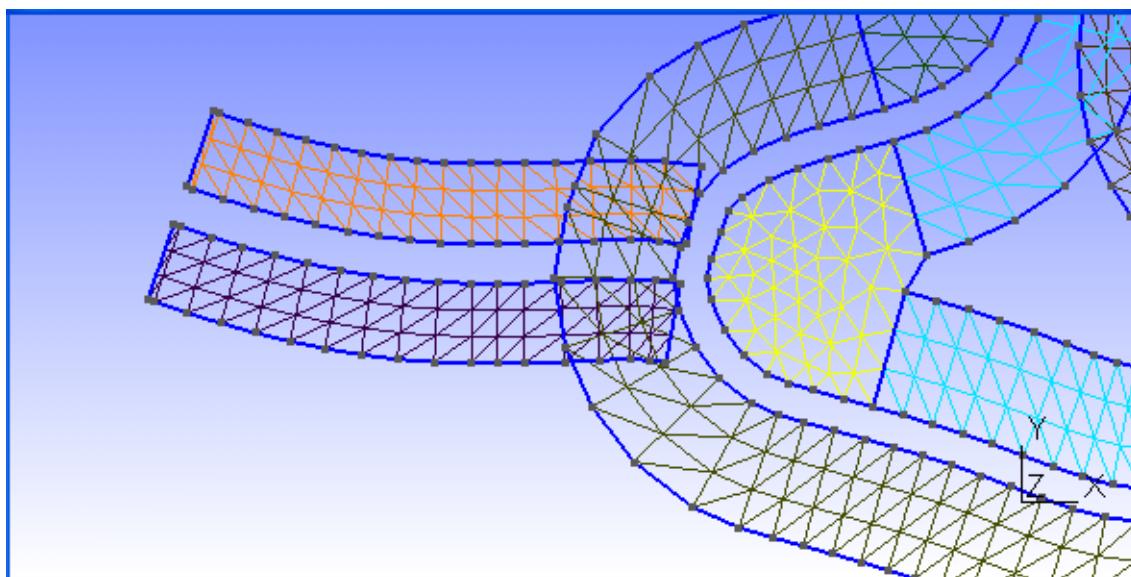
Tenemos tres opciones para arreglar el solapamiento: 0) la rápida, 1) la fácil, y 2) la no tan fácil.

RÁPIDA WAY

Ésta puede ser una buena opción. Se deja el trabajo para hacerlo más tarde en BTB:
<http://btbtracks-rbr.foroactivo.com/t511-gmsh-editing#3345>

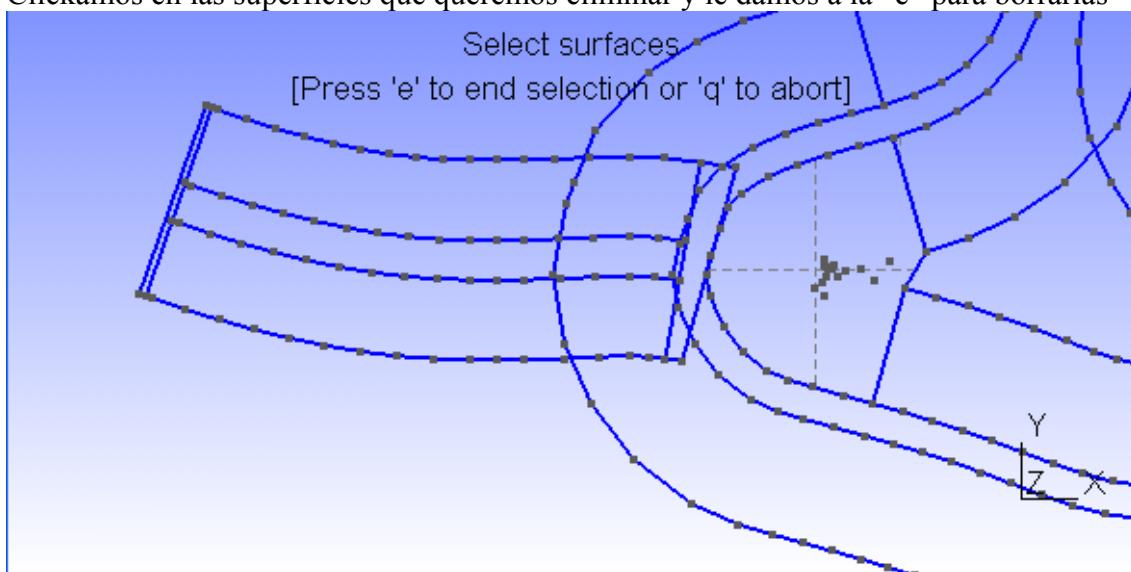
FÁCIL

- Eliminamos las superficies que se solapan
- Borramos líneas no necesarias (solo para ver mejor la zona de trabajo)
- Creamos nuevas plane surface



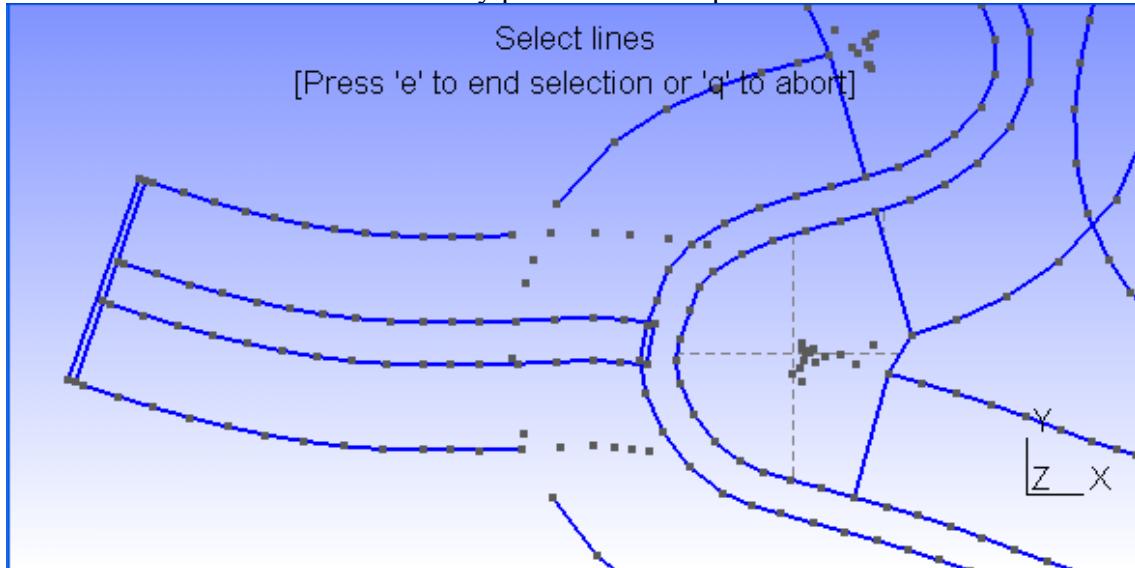
Geometry\Elementary entities>Delete\Surface.

Clickamos en las superficies que queremos eliminar y le damos a la "e" para borrarlas



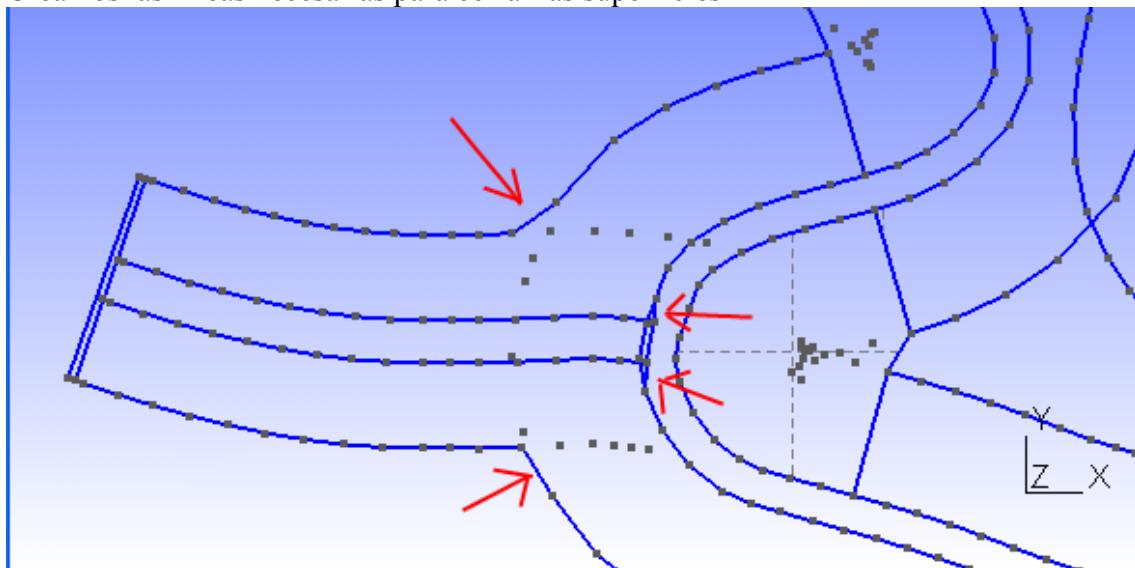
Geometry\Elementary entities>Delete\Line

Clickamos en las líneas no deseadas y pulsamos la “e” para eliminarlas



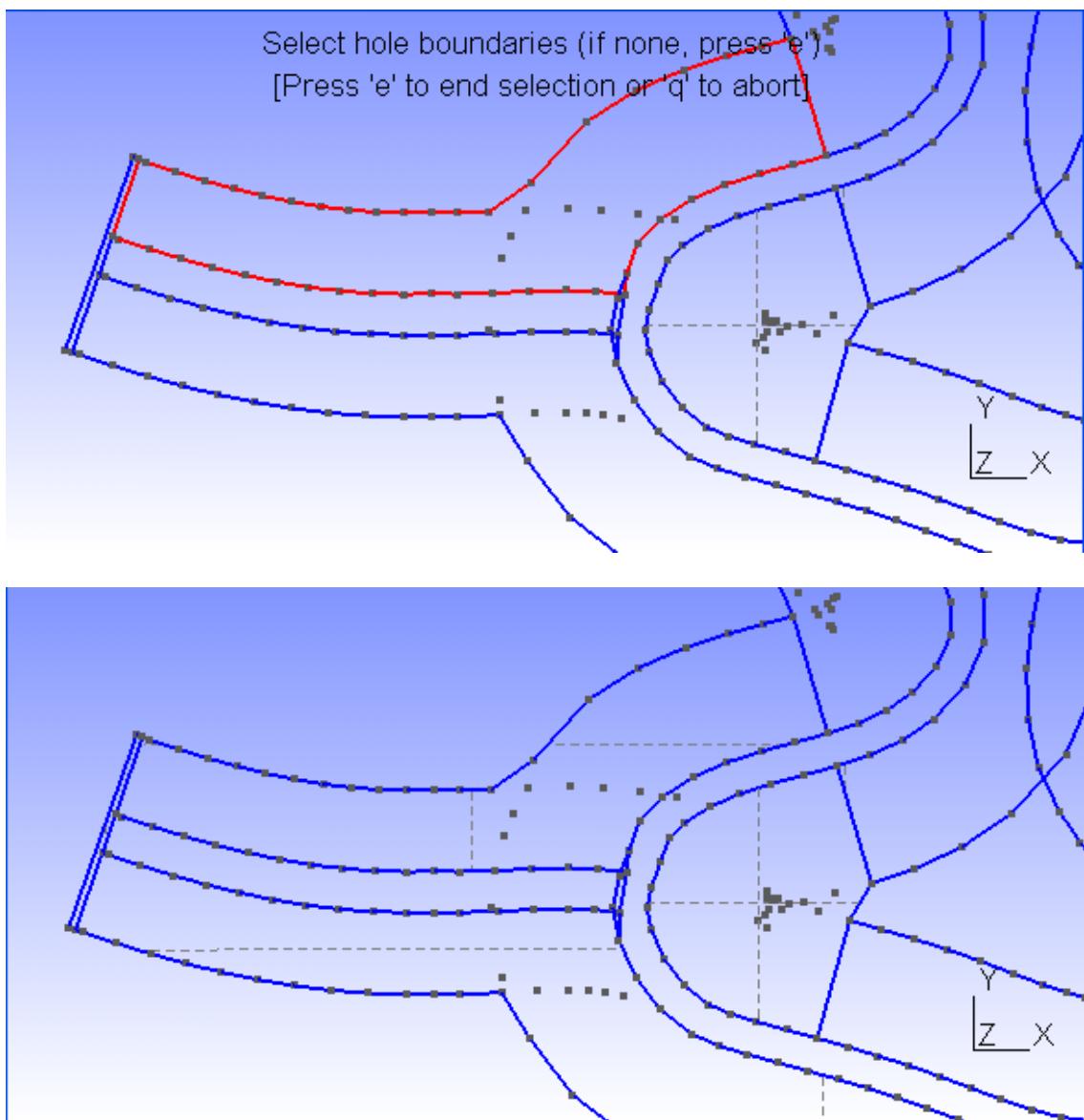
Geometry\Elementary entities>Add>New\Straight Line

Creamos las líneas necesarias para cerrar las superficies

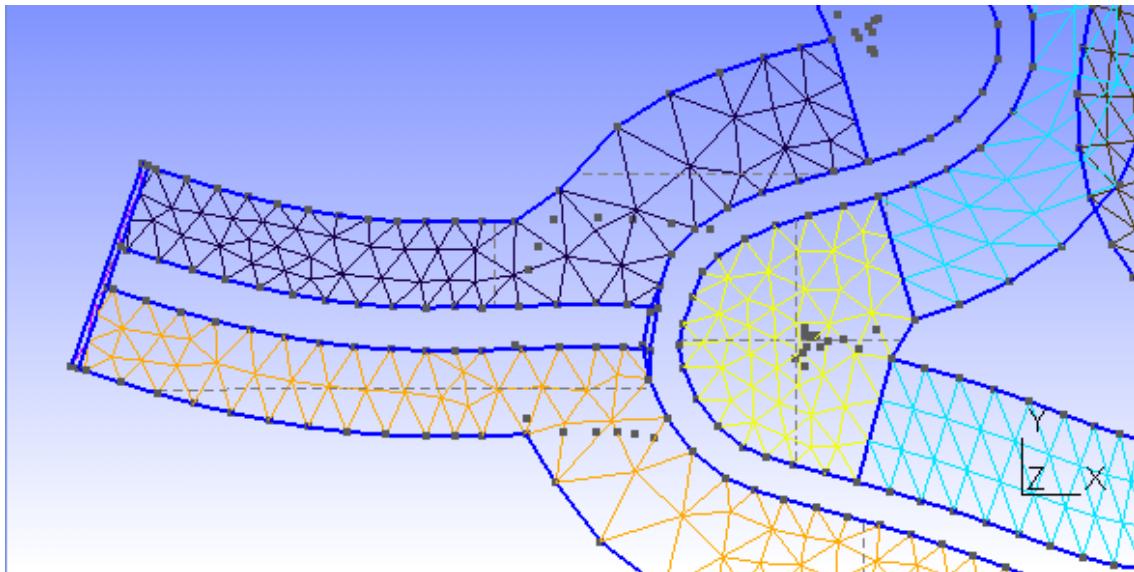


Geometry\Elementary entities>Add>New\Surface

Seleccionamos la frontera de la superficie a crear y pulsamos la “e”. Cuando todos los segmentos de la frontera están seleccionados la opción “u” desaparece del mensaje de la parte superior de la pantalla. Repetimos lo anterior para todas las superficies que lo necesiten.



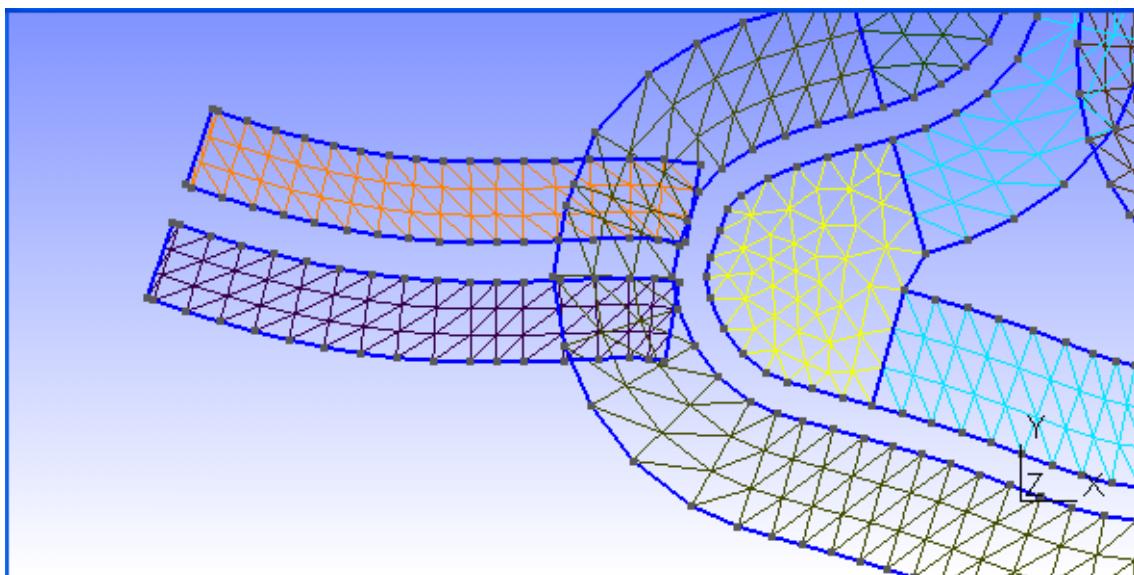
Mesh\2D
Para comprobar el mallado resultante



Como se puede ver, con el método “fácil” se pierde el patrón regular de las mallas conducibles. El método "no tan fácil" trata de preservar el patrón, con un poco de esfuerzo extra.

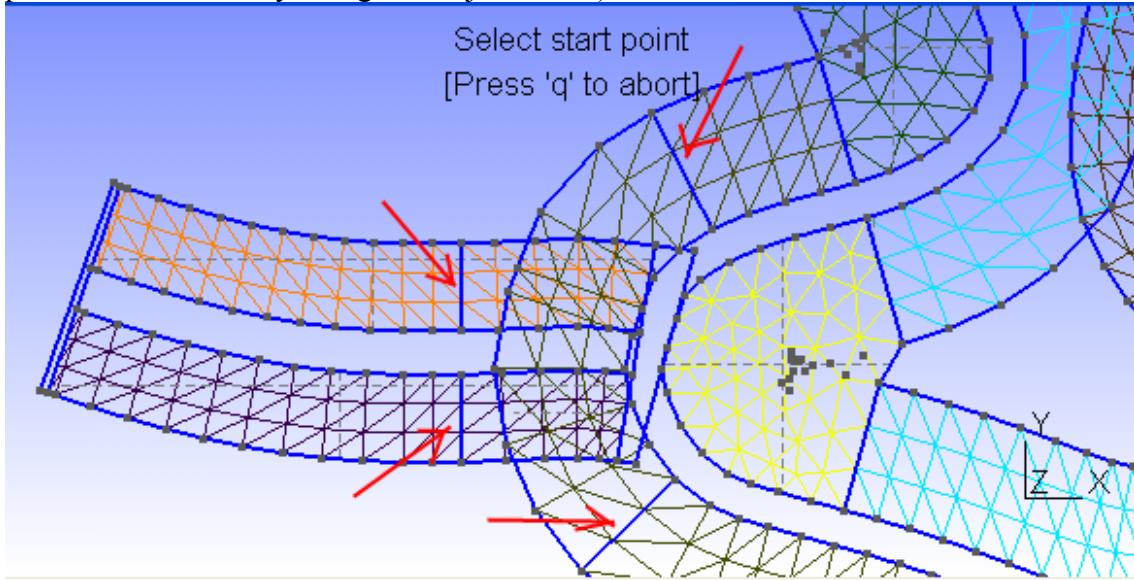
NO TAN-FÁCIL

- Eliminamos las superficies que se solapan
- Borramos líneas no necesarias (solo para ver mejor la zona de trabajo)
- Creamos superficies regulares
- Creamos superficies irregulares



Geometry\Elementary entities>Add\New\Straight Line

Creamos las líneas transversales en las que acabará el patrón regular (marcamos el primer punto en la carretera y el segundo lejos de ella).



Abrimos joined.geo con un editor de textos y declaramos las líneas recién creadas como transfinite. Se puede usar un lazo for para hacer este paso más rápido (se copia y pega el lazo y se cambian las líneas inicial y final).

```
Line(99992) = {97703, 97742}; #M
Line(99993) = {97685, 97724}; #M
Line(99994) = {1853, 31853}; #M
Line(99995) = {1845, 31845}; #M
```

gmsh has written this

```
For h In {99992:99995}
  Transfinite Line(h)= 4 using Progression 1;
EndFor
```

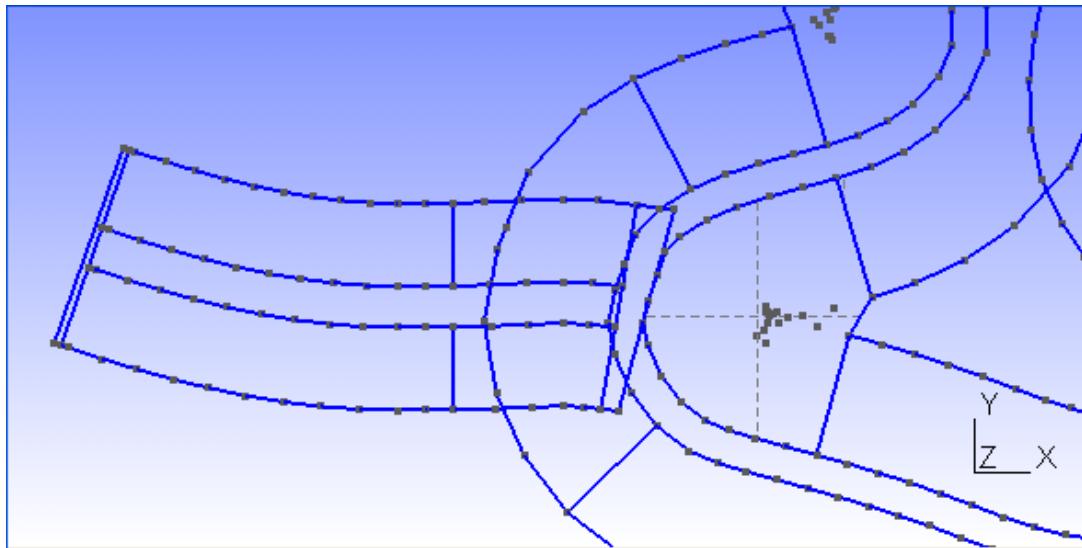
we add this

3 panels

En este ejemplo se han creado 4 líneas que tenían 3 paneles, por lo que pueden compartir “lazo for”. Es buena idea crear las líneas en tandas, todas seguidas, del mismo número de paneles, para que puedan compartir el lazo for. Si se aplica un número erróneo de paneles a una línea gmsh no creará un mallado regular en la superficie implicada.

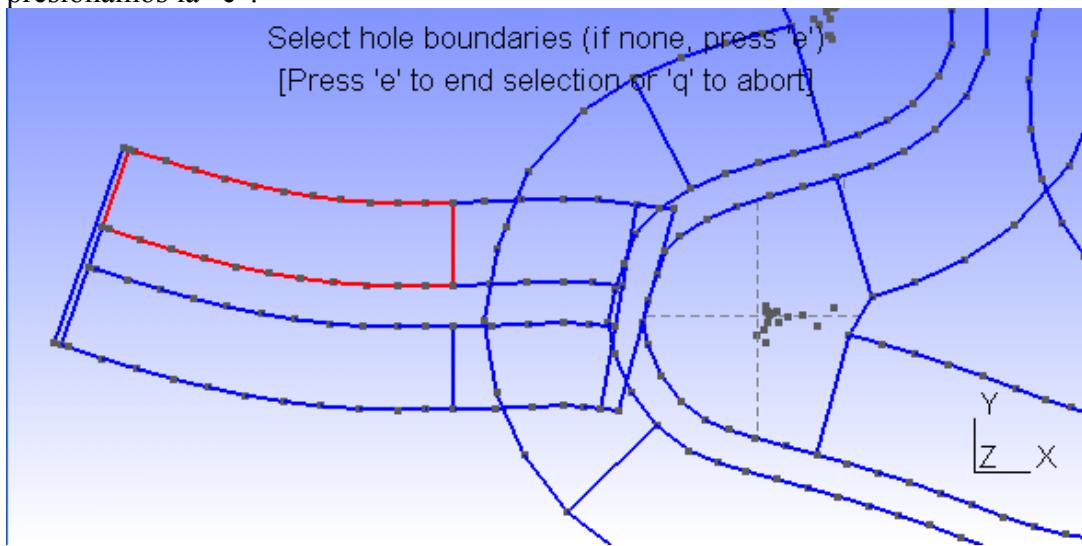
Geometry\Elementary entities>Delete\Surface.

Clickamos en las superficies que queremos eliminar y le damos a la ”e” para borrarlas



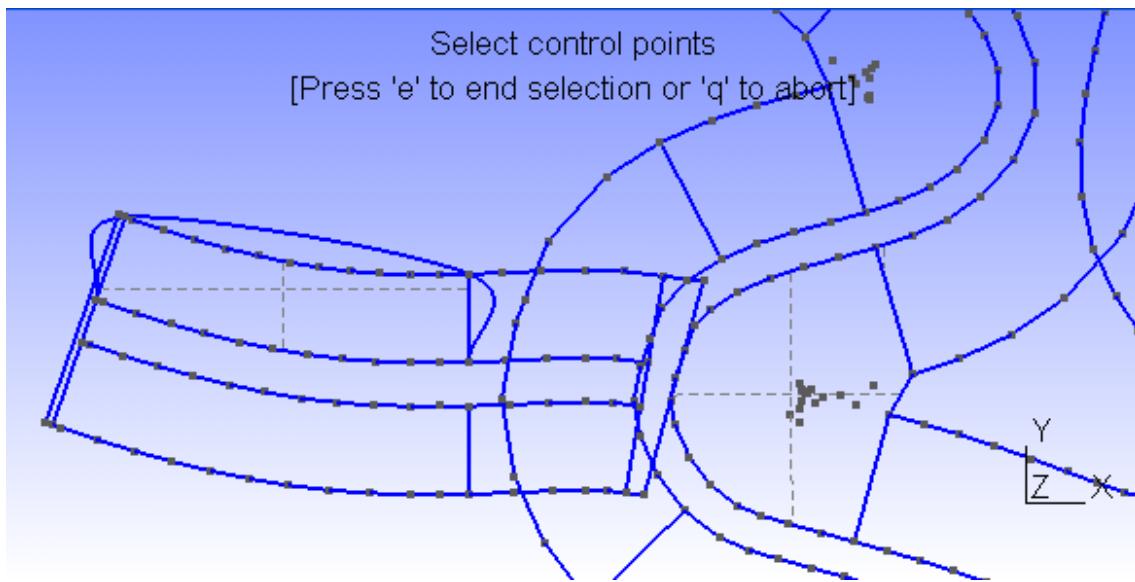
Geometry\Elementary entities>Add>New\Plane Surface

Seleccionamos la frontera de una de las superficies con patrón regular que queremos crear y presionamos la "e".



Geometry\Elementary entities>Add>New\Spline

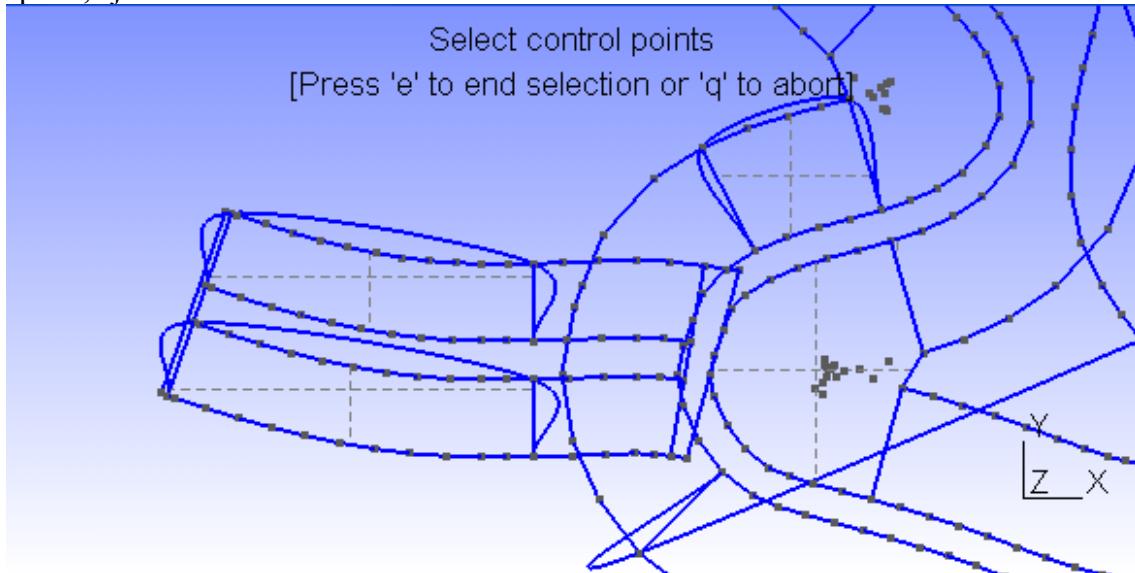
Creamos un spline usando las esquinas de la superficie que acabamos de crear y pulsamos la "e"



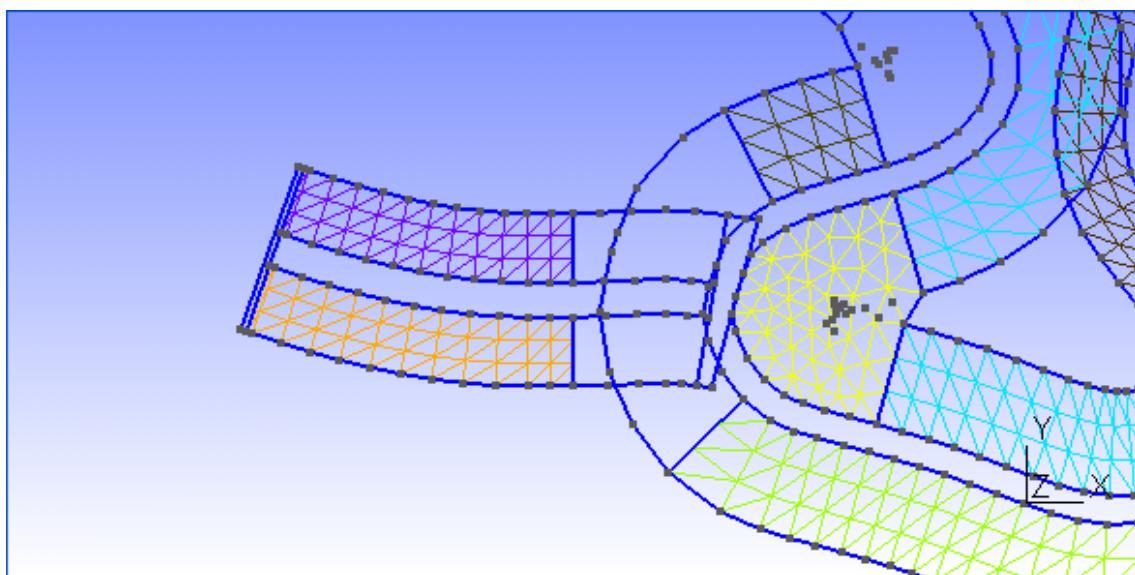
Ahora ejecutamos el script “addt” dentro de la carpeta s1_mesh

```
octave-3.2.3.exe:7:f:\vmdata\tramos\tudons\tudons_father\s1_mesh
> addt
Leyendo el fichero salida\joined.geo
Cerrando el fichero
Escribiendo el fichero salida\joined.geo
Cerrando el fichero
octave-3.2.3.exe:8:f:\vmdata\tramos\tudons\tudons_father\s1_mesh
>
```

Repetimos el procedimiento para todas las superficies regulares: crear superficie, crear spline, ejecutar “addt”. NO se debe cambiar el orden de esas tres acciones.



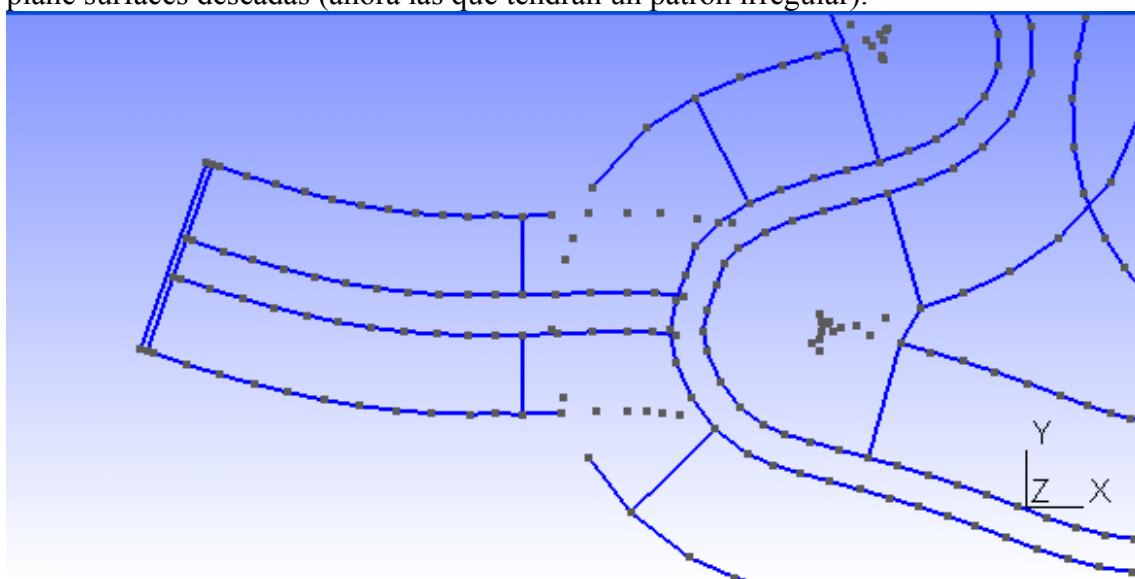
Ahora cierra gmsh, abre de nuevo joined.geo con gmsh y “Mesh\2D”

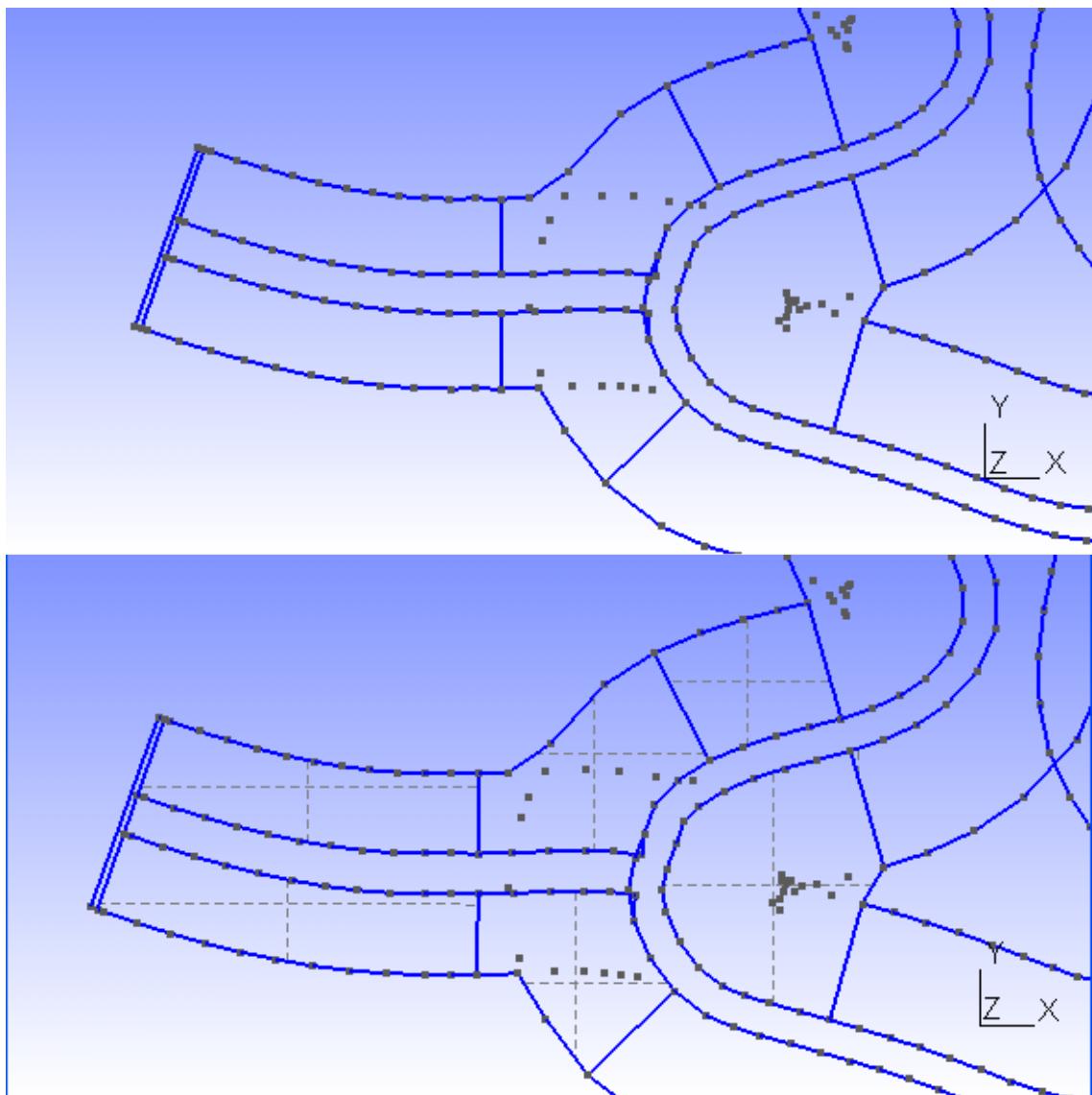


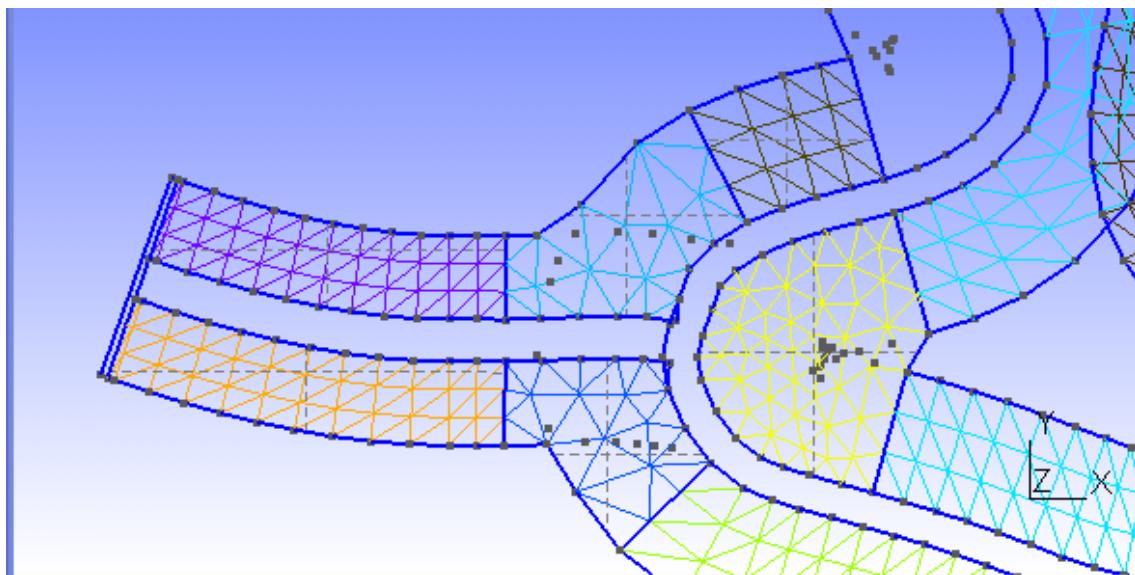
Como se puede ver, las superficies creadas tienen un patrón regular.

Ahora debemos hacer los mismos pasos con todos los desvíos, o podemos proceder a finalizar éste.

Si se quieren completar todos los pasos para este desvío, basta con proceder como se ha explicado en el método “fácil”: borrar las líneas no deseadas, crear las necesarias y crear las plane surfaces deseadas (ahora las que tendrán un patrón irregular).







2) Crear la Physical Surface(111)

El script join_geos ya ha incluido en joined.geo una lista de superficies conducibles. Pero hemos borrado algunas de ellas y creado otras nuevas. Si se abre joined.geo con un editor de textos, todas las declaraciones Plane Surface creadas tras el texto

```
***** END OF JOINED .GEO FILES*****
```

deben ser incluidas en la lista de la Physical Surface(111). Así que hay que mover la declaración de esta Physical Surface al final del fichero y añadir a mano el código de todas las nuevas plane surfaces a la lista. Para ayudar en esta tarea se puede ejecutar un script llamado “**listc**” que crea un fichero llamado “**listc.geo**”, con la lista de todas las plane surfaces creadas tras el texto que hemos comentado antes.

```
octave-3.2.3.exe:11:f:\vmdata\tramos\tudons\tudons_father\s1_mesh
> listc
Leyendo el fichero salida\joined.geo
Cerrando el fichero
Escribiendo el fichero salida\listc.geo
Cerrando el fichero
octave-3.2.3.exe:12:f:\vmdata\tramos\tudons\tudons_father\s1_mesh
>
```

3) Crear la superficie no conducible y la Physical Surface(222)

La zona no conducible y la Physical Surface 222 se deben crear como siempre: una frontera externa con un agujero definido por la frontera exterior de la zona conducible. Cuando hayamos creado la superficie el usuario ha de insertar la declaración de la Physical Surface(222)

4) Comprobación final

Una vez pienses que joined.geo está preparado, ábrelo con gmsh, Mesh\2D, File\Save mesh y abre joined.mesh con gmsh. Comprueba las mallas: comprueba todo el proyecto buscando superficies que falten. Si todo está correcto puedes seguir con el proceso (trocea_malla).

VIII IMÁGENES DE FONDO

VIII.1 Conceptos básicos

Los scripts pueden ayudar a introducir imágenes de fondo en el proyecto BTB. Si se quiere incluso se pueden usar dichas imágenes como textura del terreno no conducible.

Para trabajar con imágenes de satélite usaremos SASPLANET:

<http://sasgis.ru/programs/sasplanet/nightly.php>

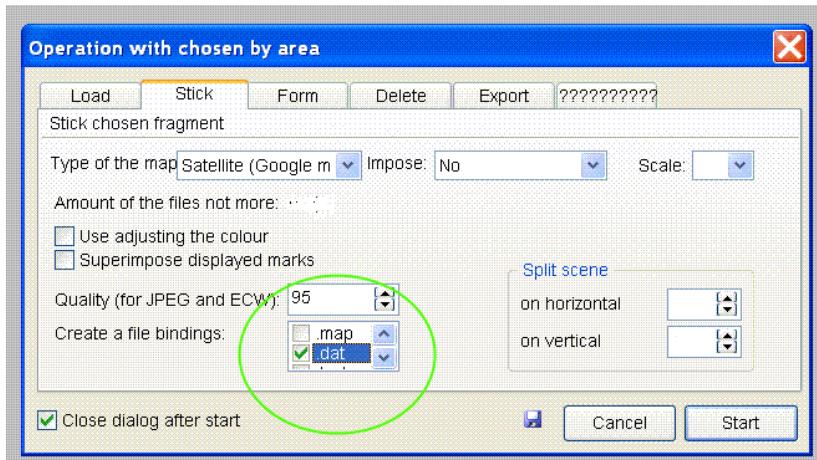
En el paso s4_terrain habrá que marcar la opción “Crear sasplanet.hlg”. Un fichero llamado s1_mesh\sasplanet.hlg se creará, con las coordenadas de un rectángulo en el que encaja nuestro terreno.

Una vez el paso s4_terrain se ha completado se puede abrir s1_mesh\sasplanet.hlg con **SASPlanet**:

Operations -> Select -> Load from file -> Load -> Start

Cuando el proceso finaliza, grabamos las imágenes en una carpeta (c:\example_folder) dividiendo las imágenes según una rejilla (5x3, por ejemplo). **IMPORTANTE: si quieres emplear las imágenes como textura del terreno se tendrá que dividir el terreno empleando exactamente la misma rejilla (paso s10_split)**. Use “**sat**” como prefijo para las imágenes: tienen que llamarse sat_1-1.jpg, etc.

Operations -> Select -> Previous Selection -> Stick -> Start (SE DEBE seleccionar .dat file bindings)



Las imágenes de satélite se han dividido y la información de las coordenadas de cada trozo se encuentran en los ficheros .dat que se han creado.

En el paso s10_split hay que indicar en qué directorio están los ficheros .dat. En el paso s10_split los scripts (add_dat_to_geo) crean una lista de las imágenes de fondo con el formato que necesita el BTB. Se llama **s1_mesh\list_bi.txt**. Si existe, se inserta automáticamente en el Venue.xml cuando se ejecute join_all. **Si finalmente no se quieren insertar imágenes en el proyecto hay que borrar o renombrar este fichero antes de ejecutar join_all.**

Las imágenes deben llamarse **sat_X-Y.dds** (nótese que deben estar en formato .dds) y deben estar ubicadas en el directorio **My Project\XPacks\Common\Textures** (*My project* es el nombre de la carpeta de su proyecto).

VIII.2 Mezclado con las imágenes de fondo

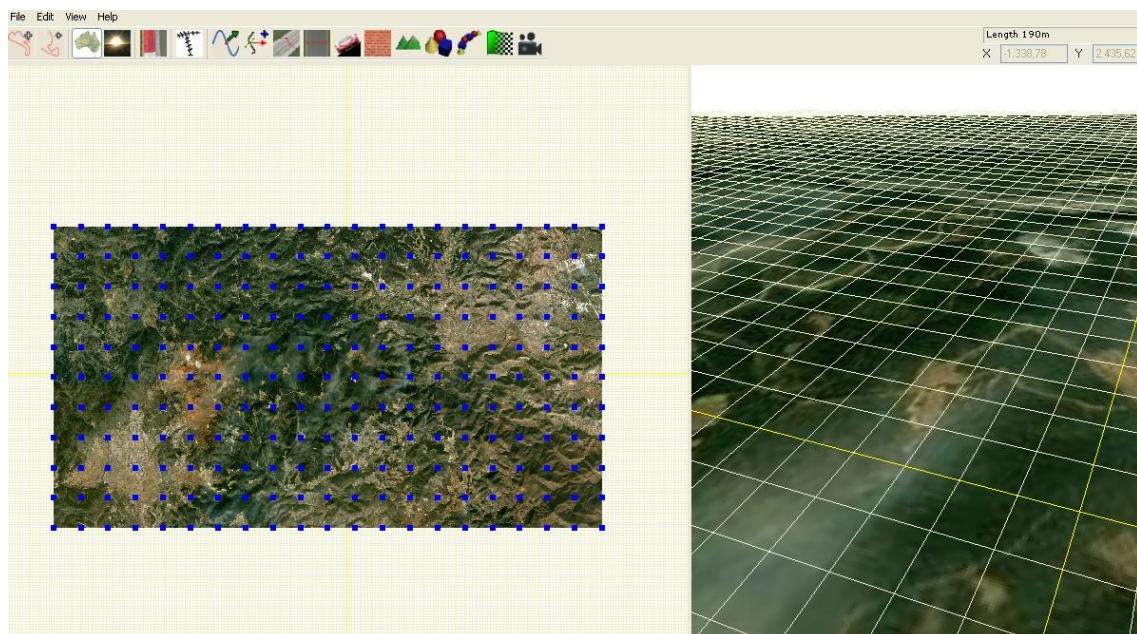
En la creación del terreno BTB, los scripts (procesar_elementstxt_mt) pueden proceder a usar automáticamente las imágenes de fondo como textura para la zona no conducible, si el terreno se ha troceado **usando una rejilla de exactamente las mismas dimensiones que las empleadas en SASPLANET (al crear list_bi.txt)**.

Por ejemplo:

procesar_elementstxt_mt(10,5,0) divide el terreno empleando una rejilla 10x5, pero NO asocia las imágenes de fondo a las zonas de terreno. Si list_bi.txt existe y las imágenes están ubicadas en la carpeta Common\Textures, el proyecto tendrá imágenes de fondo, pero éstas no estarán asociadas al terreno (como su textura).

procesar_elementstxt_mt(10,5,1) divide el terreno empleando una rejilla 10x5, usando las imágenes de fondo como parte de la textura del terreno. Antes de abrir el Venue.xml con

BTB es obligatorio 1) crear list_bi.txt (dándole la ubicación de los ficheros .dat) antes de crear el Venue.xml y 2) copiar las imágenes a la carpeta Common\Textures. De otro modo BTB no abrirá el proyecto.



Ejemplo de list_bi.txt

```

<BackgroundImages count="50">
    <BackgroundImage>
        <Path>Common\Textures\sat_1-1.dds</Path>
        <Plane>
            <Position x="-11232.375390" y="-0.5" z="5636.028398" />
            <Scale x="2100.341106" y="1" z="-2311.287859" />
            <Rotation x="0" y="0" z="0" />
        </Plane>
    </BackgroundImage>
    <BackgroundImage>
        <Path>Common\Textures\sat_1-2.dds</Path>
        <Plane>
            <Position x="-11232.375390" y="-0.5" z="3324.740539" />
            <Scale x="2100.341106" y="1" z="-2311.287859" />
            <Rotation x="0" y="0" z="0" />
        </Plane>
    </BackgroundImage>
    etc.
</BackgroundImages>

```


VIII.3 Resumen

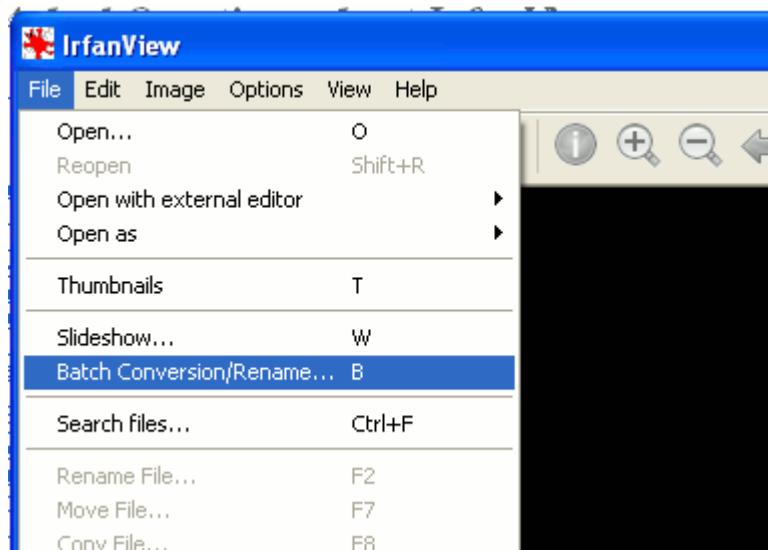
Para conseguir las imágenes usando SASPlanet se necesita un fichero .hlg con la información de la extensión de nuestra malla.

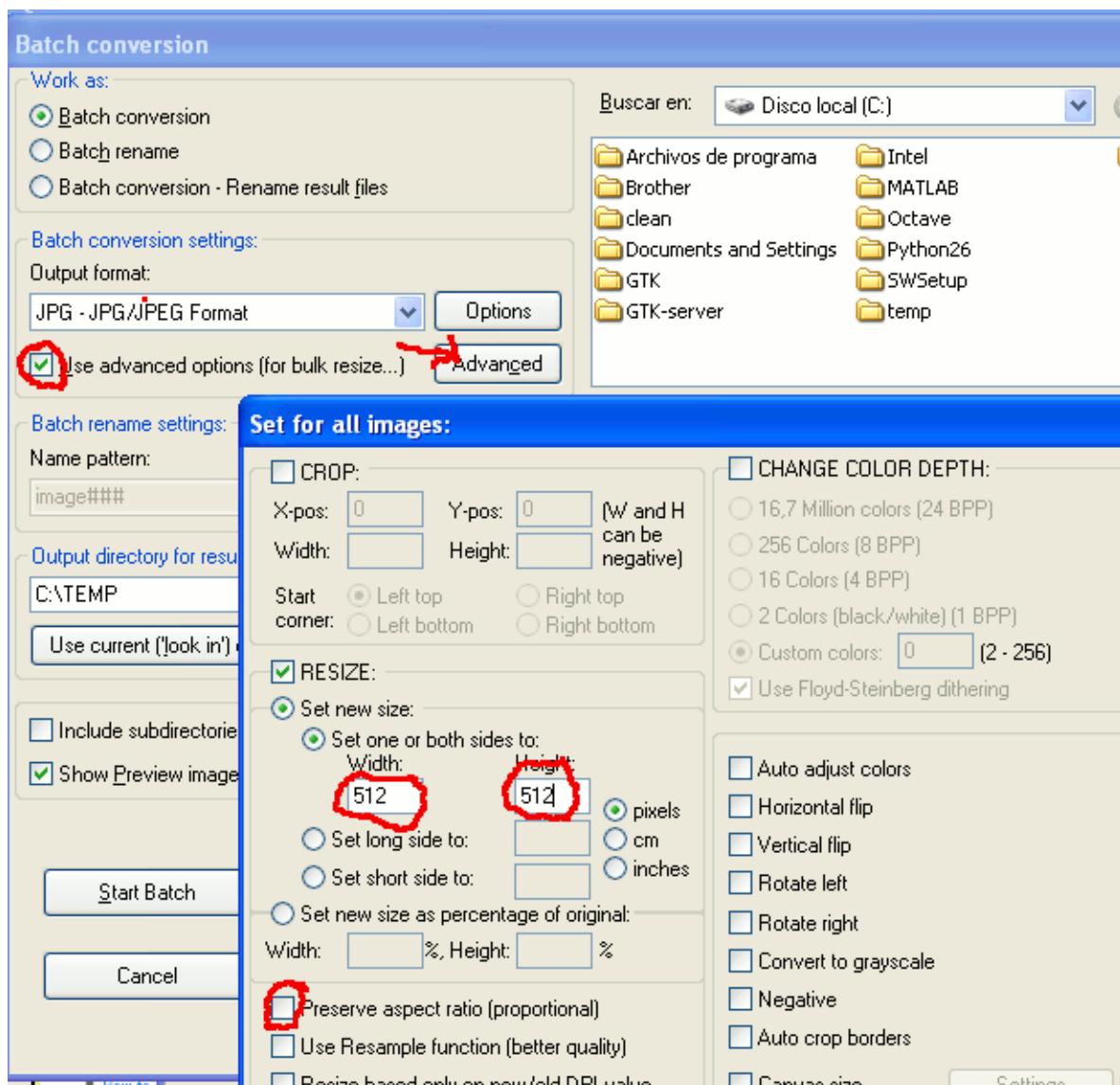
Seleccionando la opción adecuada en el paso s4_terrain (llamando al script **create_hlg**) realiza esa tarea: se lee la información de la malla (creada a partir del .geo, salvada como .msh y troceada como trocea_malla) y escribe el .hlg para SASPlanet

Cuando se abre sasplanet.hlg con SASPlanet, se siguen los pasos descritos en este capítulo y se obtienen los ficheros .jpg. Por cada imagen .jpg se ha de guardar un fichero .dat. Los ficheros .dat contienen la información de las coordenadas terrestres de la imagen. En el paso s10_split, un script (add_dat_to_geo) puede leer los ficheros .dat y convierte las coordenadas terrestres a coordenadas BTB y crea un fichero listo para ser insertado en el Venue.xml de tal forma que las imágenes sean insertadas en el fondo del proyecto. El fichero resultante es list_bi.txt. Este fichero puede ser insertado en el paso s9_join(por el script join_all) en el Venue.xml y tendremos las imágenes como fondo.

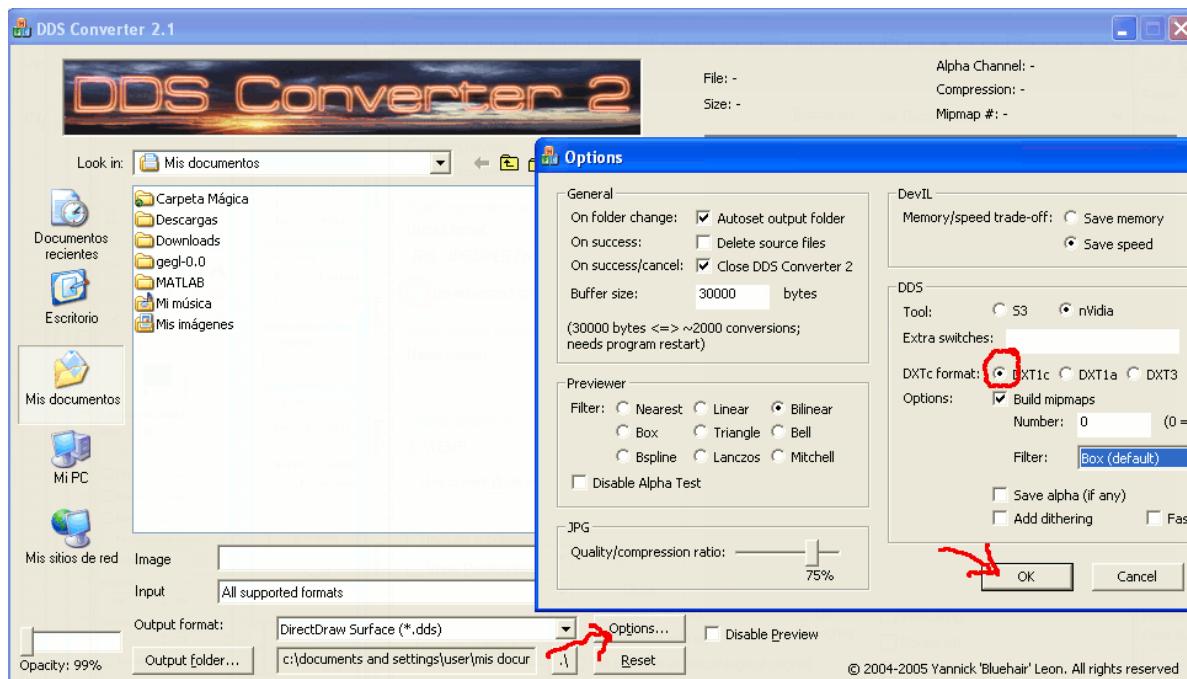
Los pasos anteriores son sencillos de seguir. Podemos complicar las cosas un poco más si queremos un proyecto en el que se usen las imágenes de fondo como textura del terreno (si ése es el caso, hay que asegurarse de que se crea list_bi.txt indicando el directorio correcto para localizar los ficheros .dat en el paso s10_split). Pues bien, es sencillo: en el paso s10_split hay que seleccionar la opción “Mezclar con fondo”(eso hará que se llame al script procesar_elementstxt_mt con 3 parámetros: dimensiones de la matriz de imágenes, y un “1”, para decirle a los scripts que queremos usar como textura las imágenes de fondo, e.g procesar_elementstxt_mt(10,5,1)).

NOTA: Yo uso Irfanview para redimensionar en lote los .jpg a potencias de 2, y DDSConverter para convertir en lote las imágenes a .dds. En Inrfanview selecciono la opción “batch conversion” y específico el tamaño deseado para las imágenes (importante desmarcar la opción “preserve aspect ratio”).





En DDSConverter selecciono el formato DXT1c (las imágenes de fondo no tienen canal alpha).



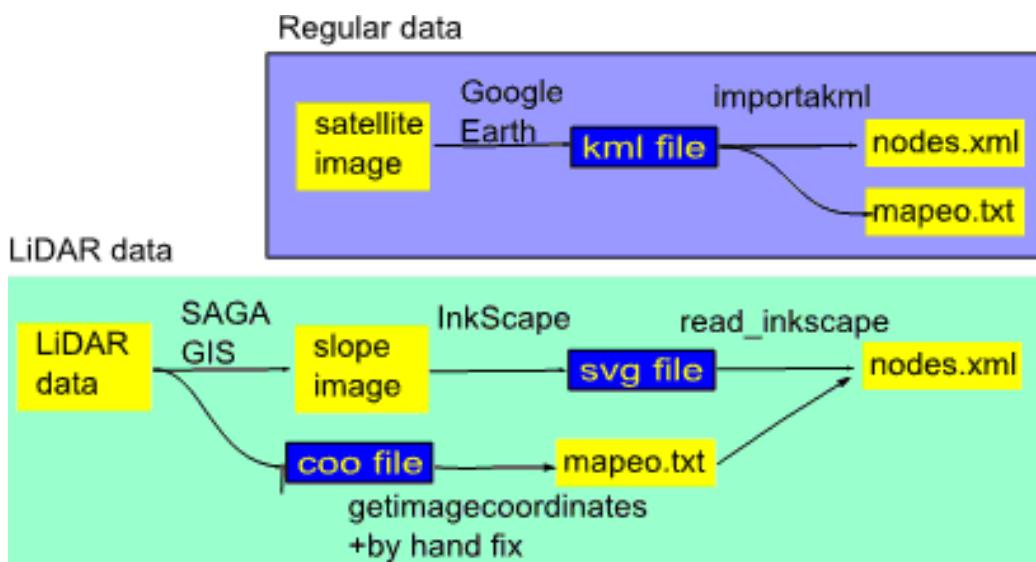
NOTA: DDSConverter no parece funcionar en Windows 7

IX USANDO DATOS LiDAR

Ejemplo de ficheros empleados con InkScape (carpeta "waste previous") y con los scripts:

http://www.mediafire.com/file/nu5hgpf9jbs4b24/waste_part1.7z
http://www.mediafire.com/file/gmy4n4xmzc5616s/waste_part2.7z

Este tipo de datos tiene su propia sección en este documento porque el proceso con ellos es especial. La diferencia respecto de un proyecto normal creado con los scripts es que es que al tener muy buena resolución en los datos de altura **no** vamos a confiar en las rutas .kml creadas a partir de las imágenes de satélite. En lugar de eso **usaremos los datos de altura para crear una imagen** que nos diga dónde están las carreteras y **usaremos esa imagen para crear carreteras que se adapten a las que se vean en esa imagen**. De esta forma situaremos las carreteras con gran precisión.



Paso 1)

Use SAGA-GIS (<http://sourceforge.net/projects/saga-gis/>) para importar el fichero .ASC

- Import/Export - Grids\Import ESRI Arc/Info grid
- Terrain Analysis-Morphometry\Local Morphometry

Si se parte de ficheros .las, también se puede usar SAGA-GIS para importar los datos. Por ejemplo, abrimos SAGA y seguimos estos pasos:

- Import/Export-LAS\Import LAS Files
- Shapes\Point Cloud to Grid (1m grid)
- Grid Tools\Close gaps

- Terrain Analysis – Morphometry\Local Morphometry (slope)

Y finalmente exportamos los “**slope**” datos usando el formato .jpg:

- Import/Export Images\Export Image

Paso 2)

Usamos FUSION (http://forsys.cfr.washington.edu/fusion/FUSION_Install.exe) para transformar las rejillas ASCII a rejillas DTM

```
c:\FUSION\ASCII2DTM N4414A3.dtm m m 1 0 0 0 N4414A3.asc
```

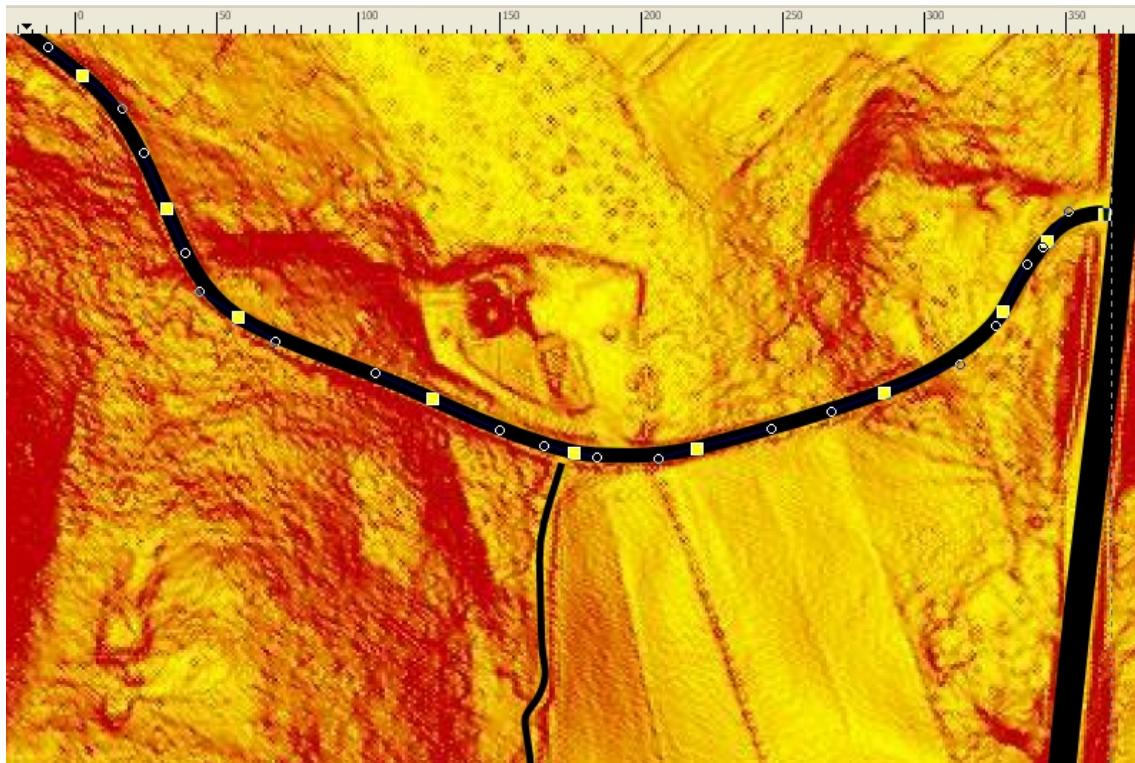
Los ficheros .dtm deben copiarse a una carpeta llamada “lidar” ubicada en la misma carpeta donde está “project” o los FATHER y SONs en un proyecto multi-track (e.g. si trabajamos one-track en c:\project, los copiaremos a c:\lidar).

Si se parte de ficheros .las recomiendo filtrar los datos para eliminar los rebotes en la vegetación (en el ejemplo que se muestra se filtran datos con unidades en pies para crear una rejilla de resolución 1m. Si se trabaja en metros habría que usar 1 en lugar de 3.28084):

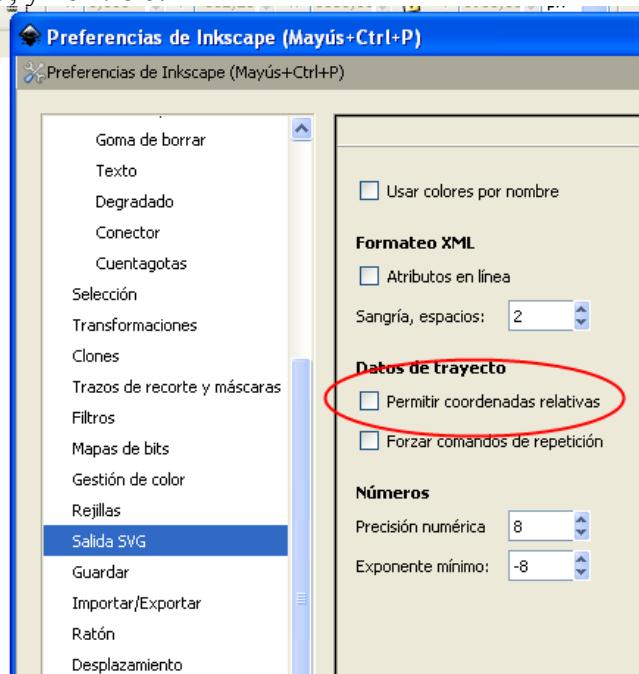
```
c:\FUSION\GroundFilter %1f.lda 3.28084 %1.las  
c:\FUSION\GridSurfaceCreate %1f.dtm 3.28084 f f 1 0 0 0 %1f.lda
```

Paso 3)

Importe el **slope** .jpg con InkScape (**no redimensione ni mueva la imagen**) y cree rutas que sigan los caminos claros (zonas de baja pendiente) que son los que muestran dónde están las carreteras. Cree un camino para cada ruta de su proyecto. Ajuste el ancho en pixels de la ruta para que encaje en las zonas claras y así tendrá una orientación del ancho real de al carretera (este valor puede ser útil posteriormente al ejecutar el script btb06).

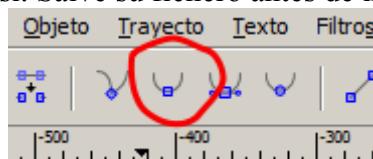


NOTA: en **opciones generales** de Inkscape, sección **SVG output**, seleccione que las coordenadas relativas no estén permitidas. De esta forma las curvas bezier solo deberán contener letras *M* y *C*, y no *m* o *c*.



Es posible que tras cambiar las opciones generales de Inkscape algún camino ya existente de inkscape siga teniendo coordenadas relativas por lo que recomiendo usar **Edit\Edit XML** dentro de InkScape y hacer un cambio simbólico en los datos del camino (añadir un espacio en blanco y luego borrarlo) y darle a “Apply”). Tras esa operación **todas las letras del camino deberían ser mayúsculas: M o C.**

Los nodos del camino deben ser “s(suaves)”-type. Se puede forzar que todos los nodos sean de tipo-s seleccionando todos los nodos del camino (ctrl-a) y luego clickando en el icono adecuado de la barra superior (shift-s hace lo mismo pero en mi experiencia **InkScape a menudo se cuelga** al hacerlo así. Salve su fichero antes de intentarlo).



Paso 4)

En el fichero .svg solo se debe incluir **una imagen**. Si tenemos las coordenadas de la imagen podremos traducir las rutas a coordenadas reales. Por tanto para esta imagen hay que crear un fichero .coo que contenga las coordenadas UTM de la imagen: (lasinfo de lastools, <http://www.cs.unc.edu/~isenburg/lastools/download/lastools.zip>, nos puede dar esa información si usamos lastools, en caso contrario SAGA-GIS también nos informa de las características de los datos importados). La coordenada Z (elevación) no será empleada, pero los scripts la requieren en el .coo por cuestiones de compatibilidad. El contenido del fichero .coo debería ser (X1, Y1, Z1, X2, Y2, Z2 sustituidos por los valores reales de nuestra imagen):

```
min x y z X1 Y1 Z1
max x y z X2 Y2 Z2
```

El fichero .coo puede ser empleado para crear una primera versión de mapeo.txt
getimagecoordinates('fichero.coo',1).

El fichero mapeo.txt se puede retocar a mano para centrar el terreno (o quizás para cambiar de pies a metros, si ése es nuestro caso).

Paso 5)

El fichero .svg debe ser procesado junto con el .coo de la imagen usada (en los ficheros que dejé como ejemplo junté 6 imágenes en una y creé un .coo para la imagen resultante):

```
read_inkscape ('pennsylvania.svg','combinado.coo')
```

Antes de procesar el fichero .svg recomiendo abrirlo con un editor de textos y comprobar que en los caminos solo aparece la letra “s” (ejemplo: *sodipodi:nodetypes= "ssssssssssssssssssssssss"*). Si vemos otra letra, como “c”, debemos abrir el .svg de nuevo con InkScape y forzar los nodos del camino a ser “suaves”. Los nodos “c” al comienzo o

final de la cadena son normales y no deberían ser problemáticos.

La salida de **read_inkscape** es un fichero .xml por cada camino incluido en el .svg. Esos ficheros .xml deben ser renombrados como nodes.xml, copiados a la carpeta Venue de los scripts y procesados con btb06. Repetimos el proceso para todas las rutas. Si por ejemplo el ancho de una ruta es de 5m:

```
> cd Venue  
> btb06(5,1)
```

Y el proyecto se puede procesar con normalidad usando el GUI, a partir del paso s3_road.

Pero **create_sons** puede hacer parte de este paso por nosotros, como se explica más adelante. Antes de usar el resto de scripts el fichero mapeo.txt tiene que ubicarse en la carpeta raíz del proyecto (o la raíz del father si es multi-track). Luego podemos continuar con el uso de los scripts de la misma forma que si hubiéramos empleado importakml.

Paso 6)

Descargo los scripts usando subversion. Copio mapeo.txt en la carpeta raíz del father.

Luego **create_sons** puede usarse para crear los SONs automáticamente a partir de los ficheros .xml localizados en una carpeta (e.g. *create_sons('c:\temp\pennsylvania')*). También copia los ficheros .xml como Venue\nodes.xml para cada SON creado.

Pero también podemos hacer estos pasos a mano:

- Usamos **create_sons(N)** en la carpeta raíz del father para crear N sons
- Copiamos los ficheros .xml como nodes.xml dentro de las carpetas Venue del father y de los sons

Paso 7)

Usa los scripts con normalidad. FUSION debe instalarse en C:\FUSION. FUSION usará ficheros .dtm para dar elevación a nuestros puntos cuando sea necesario.

En el script **creartrack1** se recomienda usar 1 como parámetro: **creartrack1(1)**. Esto hacer que la elevación de la carretera se calcule a partir de la alturas del terreno en el centro de la misma. De otro modo el perfil en elevación de la carretera se calcula a partir de la elevación del terreno en los laterales de la carretera.

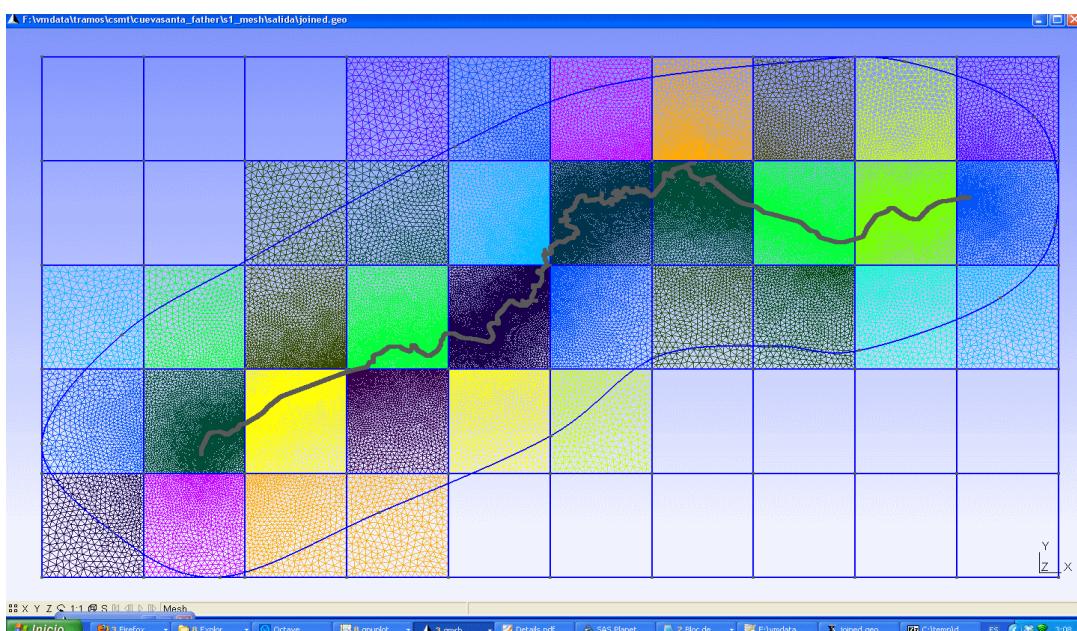
NOTA: Se pueden conseguir datos LiDAR para algunos estados de USA desde:

http://lidar.cr.usgs.gov/LIDAR_Viewer/viewer.php

X USOS AVANZADOS DE LOS SCRIPTS

X.1 Terreno adaptado a las imágenes de fondo

Quieres tener áreas de terreno que encajen exactamente en las imágenes de fondo, tal y como hice en mi tramo Cueva Santa. De acuerdo, si ése es el caso, es necesario insertar los límites de nuestras imágenes en el joined.geo, el .geo con el que estamos trabajando. Es sencillo: ejecutamos add_dat_to_geo en la carpeta s1_mesh y seleccionamos la opción "add the grid to joined.geo". Ahora al abrir joined.geo con gmsh veremos la rejilla con los límites de las imágenes descargadas con SASPlanet. Ahora hay que hacer un poco de trabajo combinando las nuevas líneas con el terreno existente, creando superficies más pequeñas para el terreno no conducible. No es difícil realizar estos pasos si se tiene un poco de práctica con gmsh. En la imagen se pueden ver los límites originales del mallado (el spline grande), y las nuevas mallas ajustadas a los límites de las imágenes descargadas.



Las Plane Surfaces que no están en contacto con el terreno conducible no dan trabajo. Y para las que se solapan con ese terreno se pueden seguir estos pasos:

- 1) Asegurarnos de que estamos usando la vista en planta clickando en la Z (esquina inferior izquierda)
- 2) Crear un nuevo punto cerca del terreno de forma que ese punto pertenezca a la línea que está en contacto con el terreno conducible
- 3) Eliminar esa línea

- 4) Crear una nueva línea, la sustituta de la que acabamos de eliminar, pero ésta acabando en el punto que acabamos de crear, y por tanto sin llegar a tocar el terreno conducible
- 5) Crear una línea desde el nuevo punto hasta un punto cercano del límite externo del terreno conducible. Trataremos de que esta nueva línea esté alineada todo lo que se pueda con la línea que hemos creado antes
- 6) Repetimos los pasos anteriores para todas las líneas problemáticas
- 7) Creamos las plane surfaces y añadimos sus códigos a la lista de la physical surface 222

Una vez tenemos nuestra nueva malla, seguimos con el proceso con los scripts con normalidad.

XI LISTA DE COMANDOS

Script	Ejecutar en
accept_mesh.m acepta anchors_carretera.msh como la malla definitiva, evitando el procesado adicional con MeshLab. Se ejecuta en lugar de simplificar+MeshLab+juntar_mallas	s4_terrain
add_dat_to_geo.m Lee los ficheros .dat creados por SASPlanet al dividir las imágenes de satélite y crea una rejilla que se puede salvar como grid.geo o ser añadida a salida\joined.geo. Este script también genera una lista de imágenes de fondo (s1_mesh\list_bi.txt) lista para ser insertada en el Venue.xml. NOTA: este script internamente llama a addgrid, por lo que addgrid.hlg se sobrescribe.	s1_mesh
addgrid.m Crea una rejilla con formato .geo. Dos posibilidades: addgrid(numx,numz) Quieres ver los límites de los datos de altura disponibles usando una rejilla numx X numz addgrid(xmin,xmax,zmin,zmax,step) Crea una rejilla con los límites especificados y la distancia entre líneas deseada. Si se añade otro parámetro al comando, valga lo que valga, el código .geo definirá los puntos y líneas de forma individual (en lugar de usar un “lazo for”). addgrid crea un fichero addgrid.hlg pensado para ser abierto con SASPLanet y así obtener imágenes de satélite de esa área.	s1_mesh
add_sobject.m Crea una lista de SObjects que habría que insertar a mano en el Venue.xml. add_sobject(num_points) El parámetro es el número máximo de puntos usados por cada SObject. Los SObjects demasiado largos se partirán.	s7_walls_b
addt.m Abre joined.geo y sustituye la última aparición de una Plane Surface seguida de un Spline con el código adecuado para que la plane surface tenga un mallado regular (Transfinite)	s1_mesh

btb_a_coor.m Devuelve las coordenadas terrestres de un punto BTB <pre>> [mapeo]=textread('mapeo.txt','%f'); > x=2380.47; > z=-2350.67; > [longitud altura latitud]=BTB_a_coor(x,0,z,mapeo)</pre>	directorio base
btb06.m Crea los puntos en ambos lados de la carretera donde la carretera y el terreno se unirán (se llaman anchors). El parámetro es la separación entre los anchors del lado izquierdo y derecho. Afectará a la malla creada por mallado_regular	venue
coor_a_btb.m Devuelve las coordenadas BTB de un punto a partir de sus coordenadas terrestres <pre>> [mapeo]=textread('mapeo.txt','%f'); > longit= -73.67; > latit= 41.47; > [x1 y1 z1]=coor_a_BTB(longit,latit,elevation,mapeo)</pre>	directorio base
corregir.m Para una carretera dada, compara el perfil en elevación obtenido al ejecutar dar_altura y el obtenido de los datos de altura (de lamalla.mat), y cambia los datos de elevación (lamalla.mat) para que se adapten al perfil en elevación obtenido con dar_altura. corregir también acepta un fichero .kml como parámetro y usa sus coordenadas y altura para cambiar lamalla.mat. Esto podría ser útil si tuviéramos un kml con una altitud fiable, pero que supondría tener una carretera elevada o hundida respecto del terreno que tenemos. <code>corregir('file.kml')</code>	s3_road
creartrack1.m Obtiene valores de elevación para una carretera a partir de sus coordenadas y de los datos de altura (lamalla.mat)	s3_road
create_hlg Crea el fichero s1_mesh\sasplanet.hlg (debe abrirse con SASPlanet) a partir de las coordenadas límite (encajándola en un rectángulo) de anchors_carretera.msh (debe ejecutarse trocea_malla antes de usar create_hlg)	s1_mesh

<p>create_sons Crea la estructura de carpetas y ficheros para uno o varios SONs en un proyecto multi-track.</p> <p>create_sons(number_of_sons) Crea las carpetas son01, son02, etc. en la misma carpeta en la que se encuentra el father</p> <p>create_sons('c:\temp\kmls',keep_names) Crea un SON por cada kml que encuentra en el directorio empleando como primer parámetro. Si keep_names es 0, las carpetas se llamarán son01, son02, etc. Si keep_names es 1, las carpetas se nombrarán en base al nombre de los respectivos kmls</p>	Directorio base del FATHER
<p>cut_lamalla.m Reduce el tamaño de lamalla.mat. Útil si los datos se han obtenido para un área demasiado extensa.</p> <p><code>cut_lamalla([xmin xmax],[zmin zmax])</code></p>	s2_elevation s2_elevation_b
<p>dar_altura.m Suaviza la salida de creartrack1 y les da a los nodos de la carretera una altura y elevación que encajan en esa curva suave.</p> <p>dar_altura(smooth_factor,pos_slope,neg_slope,step,interactive)</p> <ul style="list-style-type: none"> - smooth es un factor de suavizado, cuando más grande más suave (debe ser impar) - pos_slope y neg_slope son la máxima y mínima pendiente permitida (1 significa 100% o 45 grados) - el perfil de elevación final se obtiene a partir de un punto cogido cada "step" metros. Debe usarse un valor pequeño para preservar los detalles del perfil, un valor grande para suavizarlos. Se usa 25m si se omite - Si interactive==0, el script no le da al usuario la opción de editar el perfil a mano y termina 	s3_road
<p>fix_project.m Este script lee las coordenadas de los puntos contenidas en joined.geo y crea nuevos ficheros porcentajes.mat y anchors.mat para todas las rutas del proyecto. De esta forma quizás un proyecto en el que las rutas y el terreno no están correctamente enganchados pueda ser arreglado. Después de ejecutar este script todos los pasos desde juntar_mallas hasta el final deben rehacerse.</p> <p>Este script no puede funcionar si se han añadido o borrado SONs desde que se creó joined.geo.</p> <p>Haga una copia de seguridad antes de usar este script.</p>	s1_mesh\salida

importakml.m Lee un fichero kml y crea establece una relación entre las coordenadas BTB y terrestres. importakml(kml_file) Todos los puntos del kml original se convertirán en nodos de la carretera importakml(kml_file,'decimate',factor) Se queda con 1 de cada “factor” puntos en el kml como nodos de la carretera. Por ejemplo, si el kml tiene 100 puntos y se usa factor==2, la carretera tendrá 50 nodos. importakml_old(kml_file,tolerance) Usa el algoritmo antiguo”. Una carretera <i>ideal</i> suave con una gran cantidad de puntos que siguen las coordenadas del kml (usando splines akima). Finalmente algunos nodos se eliminan. Un nodo se elimina si al hacerlo la carretera no se desvía más de "tolerance" metros de “camino ideal”	s0_import
join_all.m Paso final del proceso. Une todas las carreteras, terreno, notas de copiloto y muros creando un fichero de nombre Venue.xml. Para abrir este fichero con BTB hace falta buena suerte y el XPack WP.zip.	s9_join
join_geos.m Junta los ficheros anchors_carretera.geo creados con mallado_regular para todas las rutas, creando el fichero joined.geo dentro de la carpeta s1_mesh\salida. Este fichero debe ser editado con gmsh.	s1_mesh
juntar_mallas.m Lee i.ply, c.ply y n.ply de s4_terrain\salida y los une en una única malla (ficheros anchors_contaltura.txt y elements.txt)	s4_terrain
leehgt.m Crea lamalla.mat a partir de un fichero .hgt (1 grado x 1 grado) leehgt(fichero,latitud,longitud) La extensión de los datos debe ser latitud a latitud+1 y de longitud a longitud+1	s2_elevation s2_elevation_b
leehgt2.m Igual que leehgt, pero une dos ficheros .hgt adyacentes leehgt2(file1,latit1,longit1,file2,latit2,longit2) Si latit1==latit2, longit1 debe ser <longit2 Si longit1==longit2, latit1 debe ser <latit2	s2_elevation s2_elevation_b
leer_gridfloat.m Crea lamalla.mat a partir de ficheros gridfloat. El primer parámetros es el fichero .hdt y el segundo el .flt	s2_elevation s2_elevation_b
leetif.m Crea lamalla.mat a partir de un fichero geotiff	s2_elevation s2_elevation_b

listc.m Lee salida\joined.geo y crea un fichero de nombre listc.geo con los códigos de todas las Plane Surfaces creadas dentro de joined.geo tras su creación (la última línea de joined.geo en su creación es la referencia usada por listc)	s1_mesh
make_grid.m Crea varios ficheros conteniendo una rejilla regular de puntos con coordenadas terrestres. Esos ficheros deben ser elevados con BTBLofty o una aplicación similar y guardados con otro nombre: grid001.kml elevado debe salvarse como grid001_relleno.kml en la misma carpeta <code>make_grid(xmin, xmax, zmin, zmax, step, file_size)</code> Los parámetros son la x y z mínima y máxima, así como la separación entre los puntos de la rejilla (metros). El valor máximo de file_size depende de la aplicación que se vaya a usar para elevar los ficheros. 5000 como máximo es lo recomendado. Otra posibilidad para make_grid es crear una ruta en un kml y pedirle a make_grid que cree una rejilla que cubra por completo esa ruta: <code>make_grid('limits.kml', step)</code>	s2_elevation s2_elevation_b
mallado_regular.m Crea terreno conducible a ambos lados de la carretera. La posición de los bordes de la carretera (anchors) se toma de ejecutar btb06. Junta la carretera se creará un terreno del ancho especificado, dividido en dirección transversal en el número deseado de paneles. El ancho del terreno (metros) es el primer parámetro y el número de paneles es el segundo. Si se desea forzar un patrón regular (transfinite) para toda la zona conducible, se debe usar 1 como 3º parámetro. En caso contrario con 2 parámetros es suficiente. Se puede emplear un 4º parámetro "nseg" para forzar que cada nseg anchors se cree una nueva superficie conducible (ver sección IV.1)	s1_mesh
muro_pegado.m Crea muros a ambos lados de la carretera (de principio a fin). La lista de muros se puede encontrar en el directorio salida y debe ser insertado a mano en el fichero Venue.xml(actualizando la cuenta total de muros, si es el caso) muro_pegado(tam_wall,offset) Los parámetros son el número de puntos por muro y el desplazamiento en metros desde en dirección saliente desde el borde de la carretera (se emplea el parámetro usado en btb06 como ancho para determinar la posición del borde)	s7_walls_b

msh2btb Crea un terreno BTB a partir de un fichero s10_split\salida\anchors_carretera.msh . El terreno creado no se mezclará con imágenes de fondo y no se conectarán a las carreteras. Este comando presupone que es usado en lugar de procesar_elementstxt_mt ply2btb(cells_x,cells_z) Se divide el terreno empleando una rejilla cells_x X cells_z	s10_split
pacenotes.m Saca la forma de la ruta del fichero driveline.ini. La salida de este script será usada por pacenotes_2	pacenotes
pacenotes_a.m Obtiene la forma de la ruta de los anchors creados por btb06. La salida de este script será usada por pacenotes2_a	pacenotes
pacenotes2.m Crea un nuevo fichero pacenotes.ini a partir del antiguo y de la salida de pacenotes.m pacenotes2(sensitivity,distance) Los parámetros son la sensibilidad en la detección de curvas y la distancia con la que desplazar las notas respecto del comienzo de la misma. 10 se traduce en unos 50m.	pacenotes
pacenotes2_a.m Crea una lista de notas de copiloto en formato BTB listas para ser insertadas en el fichero Venue.xml. Join_all busca estas notas de copiloto y si las encuentra las inserta en el Venue.xml. Los parámetros son los mismos que para pacenotes2	pacenotes
partir_track.m Divide una carretera en varios segmentos. Lee los puntos de ruptura de pos_nodes.txt	s10_split
plot_lamalla.m Representa en pantalla el contenido de salida\lamalla.mat como una superficie.	s2_elevation s2_elevation_b
ply2btb Crea un terreno BTB a partir del fichero s10_split\salida\n.ply . El terreno creado no se mezclará con imágenes de fondo y no se conectarán a las carreteras. Este comando presupone que es usado en lugar de procesar_elementstxt_mt ply2btb(cells_x,cells_z) Se divide el terreno empleando una rejilla cells_x X cells_z	s10_split
poner_muro.m Crea muros en la frontera entre las superficies conducibles y las que no lo son. Los muros se insertan automáticamente en el Venue.xml al ejecutar join_all	s7_walls

procesar_elementstxt_mt.m Crea terreno en formato BTB a partir de la malla creada con gmsh y la salida de partir_track. Por defecto el terreno se dividen empleando una rejilla 10x10, pero se puede usar otro tamaño. procesar_elementstxt_mt(cells_x,cells_z,do_mapping) Divide el terreno empleando una rejilla cells_x X cells_z, y si do_mapping es 1, el terreno se creará usando las imágenes de fondo como textura (ver add_dat_to_geo).	s10_split
procesar_nodostxt.m Los nodos en la malla anchors_carretera.msh reciben un valor de elevación sacado de lamalla.mat, si es factible, o en su defecto de lamalla2.mat	s4_terrain
process_sons.m Este script procesa todos los SONs de un proyecto multi-track. Debe ser editado antes de usarlo para configurarlo con los valores adecuados en los scripts a los que llama.	Directorio base del FATHER
raise_kml.m Llama a una API de Google Earth para conseguir valores de elevación para los ficheros gridXXX.kml que hay en s2_elevation\salida. La salida se llamará gridXXX_relleno.kml y estará lista para ser procesada por el script read_grid. Este script necesita Google Earth y Python27 instalados en el sistema.	s2_elevation s2_elevation_b
read_grid.m Lee los ficheros gridXXX_relleno y crea lamalla.mat, recolectando todos los datos de elevación	s2_elevation s2_elevation_b
readkml.m Traduce una ruta de un fichero kml a formato gmsh y con coordenadas BTB. El fichero resultado se escribe en la carpeta salida, con el mismo nombre que la entrada pero extensión .geo. readkml('file.kml',curve) El segundo parámetro puede ser "t", para añadir líneas rectas, "s" para añadir un spline, o "st" para añadir ambos. No usar un segundo parámetro significa no crear curvas, solo puntos.	s1_mesh
readkml Bat.m LLama a readkml para todos los ficheros .kml encontrados en la carpeta especificada. readkml Bat('d:\folder',curve)	s1_mesh
simplificar.m Divide anchors_carretera.msh en tres partes que deben ser procesadas con MeshLab: intocables.ply, conducibles.ply y noconducibles.ply También crea una carpeta nc splitted con un .ply independiente para cada superficie de la zona no conducible, permitiendo simplificarlas de forma individual.	s1_mesh

split_agr.m Divide un fichero con extensión .ASC en ficheros más pequeños con extensión .AGR y el mismo formato. Ejemplo: <code>split_agr('MDT05-0667-H30',5)</code> divide el fichero <code>MDT05-0667-H30.ASC</code> en 5x5=25 ficheros con extensión .AGR	agr
split_track.m Crea los puntos de ruptura que se usarán para dividir la carretera en varios segmentos. Escribe esos puntos en el fichero pos_nodes.txt, permitiendo al usuario modificarlos a mano antes de ejecutar partir track	s10_split
start.m Llama a importakml para procesar s0_import\road.kml Llama a make_grid con s2_elevation\limits.kml como parámetro Llama a make_grid con s2_elevation_b\limits_b.kml como parámetro Los pasos por defecto para make_grid son 25 y 75m, respectivamente. Por defecto los gridXXX.kml se limitan a 5000 puntos por fichero start start(step,step_b,file_size) Los parámetros son los pasos en metros usados en make_grid y el número de puntos por kml.	
terrain_noise.m Añade un valor aleatorio a la elevación de los nodos del terreno. El valor aleatorio estará en el rango especificado. Úsese justo antes de join_all terrain_noise([ymin ymax])	s4_terrain
trocea_malla.m Trocea anchors_carretera.msh en 2 partes: listado de nodos en la malla (nodos.txt) y triángulos (elements.txt)	s1_mesh
vercontorno.m Muestra un diagrama de curvas de nivel empleando los datos de elevación (lamalla.mat) y la posición de la carretera (salida de btb06)	s2_elevation s2_elevation_b

XII NOTAS ADICIONALES

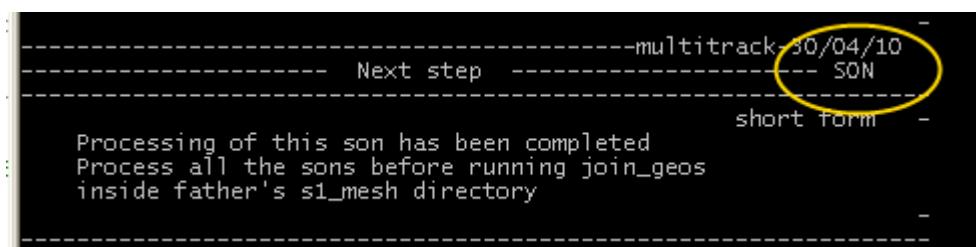
XII.1 GUI

A veces el interfaz se niega a completar un paso sin razón aparente. Si eso sucede cierre el interfaz e inténtelo de nuevo. Si eso no arregla el problema cierre el GUI y teclee los comandos en la ventana de texto de octave. Los mensajes de error en pantalla ayudarán a encontrar la causa del problema.

Si se obtiene un mensaje de error relativo a iconv.dll quizá copiar ese fichero de "c:\GTK\bin" a "c:\GTK-server" resuelva el problema.

XII.2 Gmsh y multi-track

- 1) Si no se ven las **palabras clave SON y FATHER** en el “Next step message”, no se debe seguir con los scripts hasta corregir el problema



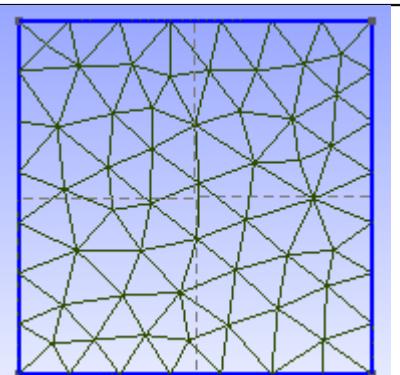
- 2) **Los cruces** se pueden procesar fácilmente si son tratados como **2 ramas independientes**. Así que al crear los ficheros kml es mejor crear 2 kmls en lugar de 1.
- 3) Hay un script llamado **process_sons** que puede ser usado para procesar varios SONs en lote siempre que comparten parámetros como el ancho de carretera o el número de paneles en la zona conducible. El fichero scripts\process_sons.m se puede editar fácilmente para adaptarlo a nuestras necesidades. Antes de usar el script hay que ejecutar “cd C:\project_father”. Tenga en cuenta que este script usa el primer .kml que encuentra en los directorios s0_import (por tanto conviene no grabar más que un fichero .kml en esos directorios).
- 4) Cuando se crea o elimina un objeto con gmsh este programa anota el cambio inmediatamente en el fichero .geo. Una forma de deshacer cambios puede ser cerrar gmsh, abrir el fichero .geo con un editor de textos y borrar el código de las operaciones

indeseadas.

- 5) En gmsh no se puede borrar un punto que pertenece a una línea. No se puede borrar una línea que pertenece a una superficie. Si se quiere borrar una línea que pertenece a una superficie primero hay que borrar la superficie.
- 6) ¿Se puede tener una isla de terreno no conducible rodeada por terreno conducible? Creo que poner_muro fallará si esa situación se produce. Siempre se puede crear la superficie como conducible y luego en BTB independizar ese terreno y cambiarle las propiedades. No obstante en ese caso no se crearán los muros protectores para esa "isla".
- 7) No ejecute **listc** después de haber creado la Plane Surface para la zona no conducible. Si se hace, recuerde eliminar el identificador de esa superficie de la lista o se tendrá esa superficie duplicada en el proyecto.
- 8) He procesado joined.geo y ahora quiero añadir otro SON al proyecto pero no quiero rehacer todo el trabajo hecho en msh, ¿hay alguna forma de hacerlo? Creo que sí. Empiece haciendo una copia de seguridad de todo. Elimine la línea con el texto "**END OF JOINED .GEO FILES**" de joined.geo e inserte en su parte final el anchors_carretera.geo creado para el nuevo SON. Intente abrir el .geo con gmsh. Si no funciona, pida ayuda.
- 9) ¿Para qué sirve el script addt? ¿Por qué creamos un spline si no queremos tener un spline? Aclaremos esto. ¿Cómo se crea en gmsh una superficie con patrón regular?

Ejemplo:

```
Point(1) = {-1, 1, 0}
Point(2) = {-1, -1, 0}
Point(3) = {1, -1, 0}
Point(4) = {1, 1, 0}
Line(1) = {1, 4}
Line(2) = {4, 3}
Line(3) = {3, 2}
Line(4) = {1, 2}
Line Loop(5) = {2, 3, -4, 1}
Plane Surface(6) = {5}
```



Si queremos tener un patrón horizontal regular tenemos que declarar las líneas a izquierda y derecha como Transfinite (indicando el número de puntos que la malla usará en ellas) y también declararemos la superficie como Transfinite.

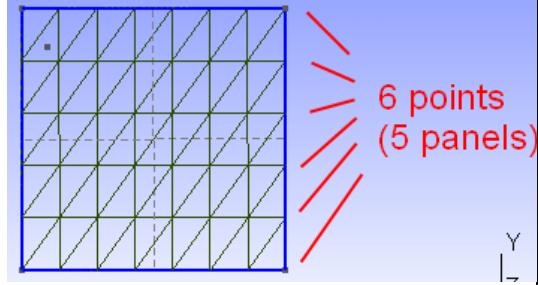
```

Point(1) = {-1, 1, 0};
Point(2) = {-1, -1, 0};
Point(3) = {1, -1, 0};
Point(4) = {1, 1, 0};
Line(1) = {1, 4};
Line(2) = {4, 3};
Line(3) = {3, 2};
Line(4) = {1, 2};
Line Loop(5) = {2, 3, -4, 1};

Transfinite Line(4)= 6 Using Progression 1;
Transfinite Line(2)= 6 Using Progression 1;

```

Plane Surface(6) = {5};
Transfinite Surface(6)={1,2,3,4};



El script “addt” (add transfinite) busca la última aparición de “Plane Surface” en joined.geo y si tras ella viene un spline, **sustituye el spline por una declaración de Transfinite Surface** usando los valores que lee de la declaración de la plane surface y del spline. Por ejemplo, si el contenido del joined.geo es:

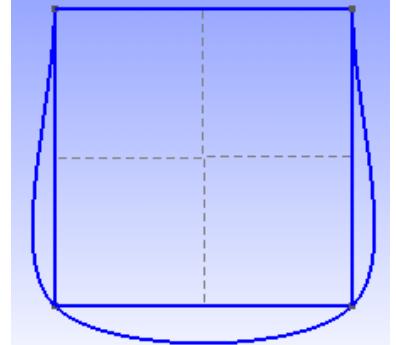
```

Point(1) = {-1, 1, 0};
Point(2) = {-1, -1, 0};
Point(3) = {1, -1, 0};
Point(4) = {1, 1, 0};
Line(1) = {1, 4};
Line(2) = {4, 3};
Line(3) = {3, 2};
Line(4) = {1, 2};
Line Loop(5) = {2, 3, -4, 1};

Transfinite Line(4)= 6 Using Progression 1;
Transfinite Line(2)= 6 Using Progression 1;


```

Plane Surface(6) = {5};
Spline(7)={1,2,3,4};



y ejecutamos “addt”, joined.geo cambiará para tener el mismo código que el visto en el ejemplo anterior (el que tenía declarada una Transfinite Surface).

Ésta es la razón por la que el orden de ejecución siempre debe ser: crear superficie, crear spline, ejecutar “addt”. Si se olvida ejecutar “addt” siempre se puede abrir joined.geo más tarde con un editor de textos, buscar “Spline” y hacer el cambio a mano.

Si no queremos usar “addt”, podemos crear la Plane Surface con gmsh, y sin cerrar gmsh abrir joined.geo con un editor de textos, y añadir la declaración de Transfinite Surface a mano, usando los números de las esquinas de la plane surface a la que queremos dar un perfil regular, y guardar los cambios. La próxima vez que mallemos veremos el resultado.

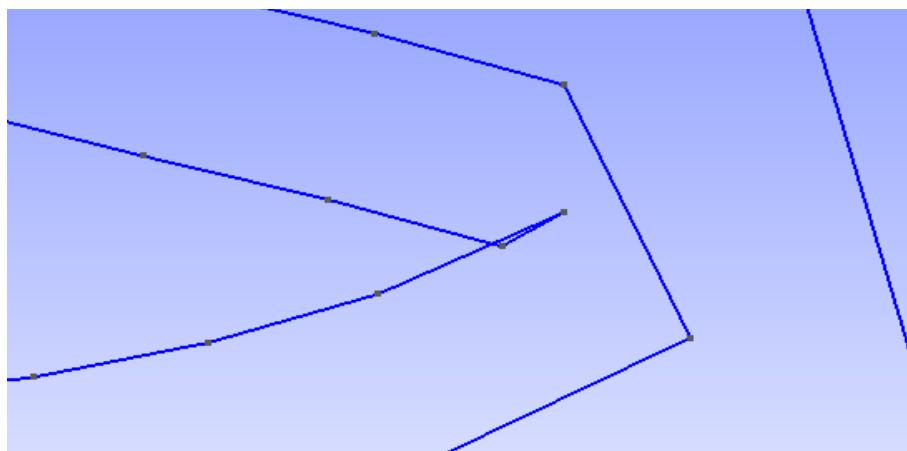
XII.3 Memory exhausted

A veces el terreno tiene demasiados triángulos y join_all, el último paso, no consigue crear el Venue.xml. Si ése es el caso, un pequeño truco puede resolver el problema:

1. Renombramos s10_split/salida/lis.txt como real_lis.txt
2. Creamos un fichero s10_split/salida/lis.txt cuyo contenido sea el texto "LITTLE_TRICK".
3. Ejecutamos join_all
4. Editamos el Venue.xml con un editor de textos, buscamos el texto "LITTLE_TRICK", y lo sustituimos con el contenido de real_lis.txt.

XII.4 ¿Gmsh se cuelga al hacer mesh/2D?

A lo mejor estás teniendo problemas con gmsh. Cuando intentas hacer mesh/2D gmsh se cuelga. Si ése es el caso, casi siempre el problema es que tienes líneas que se cruzan (ver imagen abajo) en los bordes de la carretera. Usa la herramienta translate (ver XII.6) o cambia las coordenadas de los puntos a mano para evitar el cruce.



XII.5 ¿Puedo insertar en gmsh una ruta a partir de un kml?

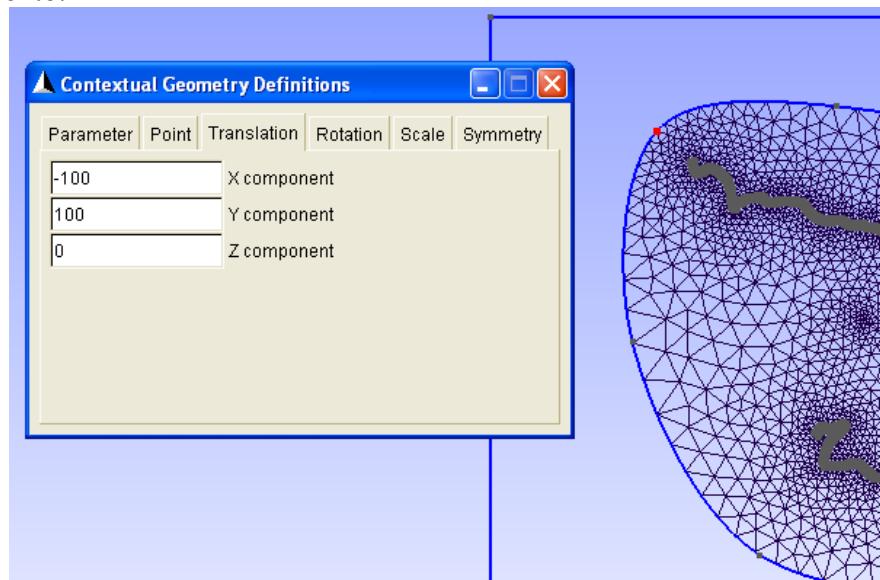
Sí, se puede. Si quieres añadir puntos/líneas en el .geo a partir de un .kml échale un vistazo al script **readkml**. Puedes añadir la salida de este script a tu .geo (anchors_carretera.geo o joined.geo) y puedes usarla para definir tus plane surfaces.

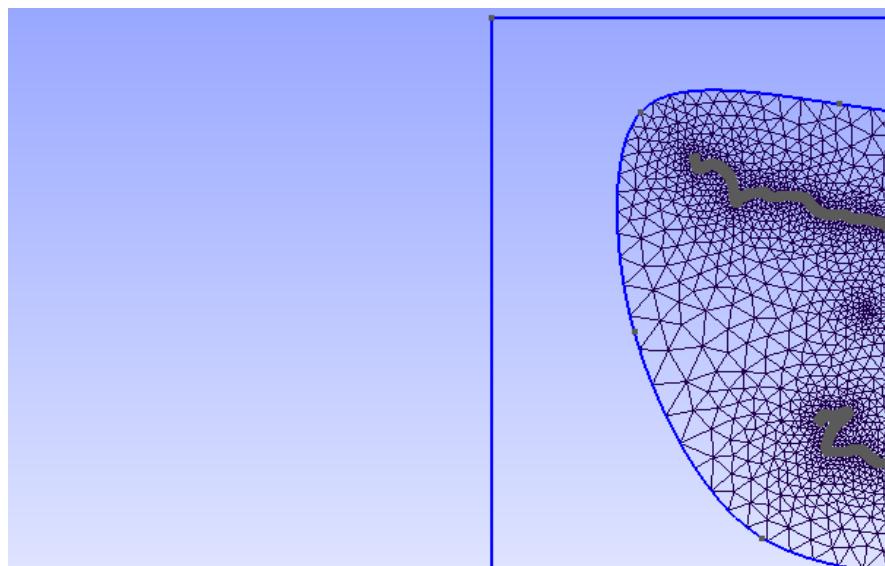
XII.6 Herramienta Translate de gmsh

A veces cuando creamos las mallas necesitamos cambiar un puntos de sitio. La herramienta que lo permite se llama “Translate”. En los menús de gmsh seleccionamos Geometry->Elementary Entities->Translate->Point.

Las mallas que creamos solo tienen coordenadas X,Y. La coordenadas Z debe ser siempre 0 por lo que al trasladar un punto **el desplazamiento en la dirección Z siempre debe ser 0**.

Seleccionamos el punto a mover, especificamos el desplazamiento en las coordenadas X e Y y tras seleccionar la ventana principal de gmsh pulsamos la “e” para aplicar el desplazamiento.





Los cambios se graban en el .geo y serán aplicados cada vez que abramos el .geo con gmsh.

Ejemplo de código añadido al .geo por la herramienta translate:

```
Translate {-100, 100, 0} {
    Point{5644};
}
```

XIII ENLACES

XIII.1 Videotutoriales

Instalación del interfaz gráfico (GUI)

<http://www.mediafire.com/?myip7mte8rjgpgm>

(extraer los ficheros y abrir gui.htm con un navegador web. Se requiere un Flash player.)

(Wink Source: <http://www.mediafire.com/file/w0zmi53e8cu9nm2/gui.wnk>)

2 rutas usando datos de elevación AGR

Videotutorial 1/3: http://www.mediafire.com/file/afdod089q47dgvc/agr_example.7z
(wink source: http://www.mediafire.com/file/da7rj6xvvdfa8p0/agr_source.7z)

Videotutorial. 2/3: <http://www.mediafire.com/?8d0s63wqaqu24xh>
(wink source: http://www.mediafire.com/file/kb8t9gqd93kghk9/agr_source_b.7z)

Videotutorial. 3/3: http://www.mediafire.com/file/7rspbi46lf207ya/agr_example_c.7z
(wink source: http://www.mediafire.com/file/y263vv5g4yuiwn4/agr_source_c.7z)

Videotutoriales de Parlesportes.

Están muy bien, aunque no se hable francés.

<http://www.mediafire.com/?tspub5ub5ncqe>

Videotutorial de liquido (uso de gmsh en un proyecto one-track)

Audio en español: <http://www.mediafire.com/?fcgy67999cdal3>
(también se puede ver en Vimeo: <http://vimeo.com/37193937>)

Subtítulos en inglés: <http://www.mediafire.com/?7k5qd59cqbbiy4e>

Cambiar a mano el perfil en elevación de la carretera:

<http://www.mediafire.com/?pdrdg6q9kp5zv3x>

XIII.2 Forums

- 1) Méthode zaxxon in Rallyesim forum (francés o inglés):
<http://forum.rallyesim.fr/viewtopic.php?f=51&t=3025>
- 2) El foro de SISCO (español o inglés): <http://btbtracks-rbr.foroactivo.com/dudas-sobre-el-metodo-zaxxon-f37/>
- 3) simracing's forum (español): <http://foro.simracing.es/bobs-track-builder/4093-dudas-y-preguntas-del-ma-zaxxon.html>
- 4) devtrackteam (inglés): <http://devtrackteam.solorally.it/viewtopic.php?f=28&t=69>

XIII.3 Archivos de ayuda antiguos

<http://www.multiupload.com/5B4KUL1ZO0>

o

<http://www.mediafire.com/?iuh4bp81hlnsf77>

XIII.4 La web de parlesportes

Web en francés creada por parlesportes con interesante material sobre BTB, 3DStudioMax y también sobre el método zaxxon. Siempre tiene tutoriales actualizados.

<http://plpcreation.tonsite.biz/>

<http://plpcreation.tonsite.biz/index.php/btb/methode-zaxxon>

XIV LICENCIA

Ni el tutorial ni el método descritos ni los scripts desarrollados por el autor son software libre. La redistribución o el uso comercial del tutorial, del método o de los scripts está prohibida. Solo se permiten modificaciones de los scripts para uso personal.

El autor no acepta ningún tipo de responsabilidad por cualquier daño o perjuicio causado por el uso de este método.