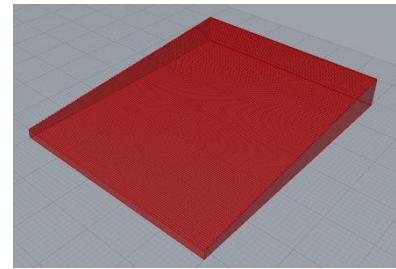


A.I- Impresión de una cuña



A priori, el diseño e impresión de una cuña, la primera pieza que proponemos en este trabajo, podría parecer trivial, pero existen dos razones fundamentales de su elección. La primera de ellas radica en la didáctica del proceso, ya que es una pieza adecuada para explicar y analizar, de una forma fácil y práctica, el procedimiento seguido. La segunda razón, más técnica, es que la geometría de la cuña, en particular su pendiente, nos permite destacar la mejora de la calidad superficial obtenida mediante el presente trabajo.

Por esta razón, en nuestra cuña, el ángulo formado entre la piel y la cama de impresión (que coincide con el plano XY) será de gran importancia. Cuanto más pequeño sea este ángulo mayor ventaja e importancia cobrará la impresión no planar, hecho que trata destacar este trabajo.

Diseño de la piel

Comencemos con la geometría; el procedimiento a seguir consistirá, en primer lugar, en crear la piel a imprimir lo más parametrizada posible. De tal forma, la superficie de la cuña será creada a partir de cuatro líneas, dados los puntos extremos que definen a éstas. El componente *Boundary Surface* creará una superficie plana limitada por una serie de curvas planas y con continuidad, si éstas forman una línea cerrada, entendiéndose que el inicio y fin de la curva se definen en el mismo punto (Fig. A. I-1).

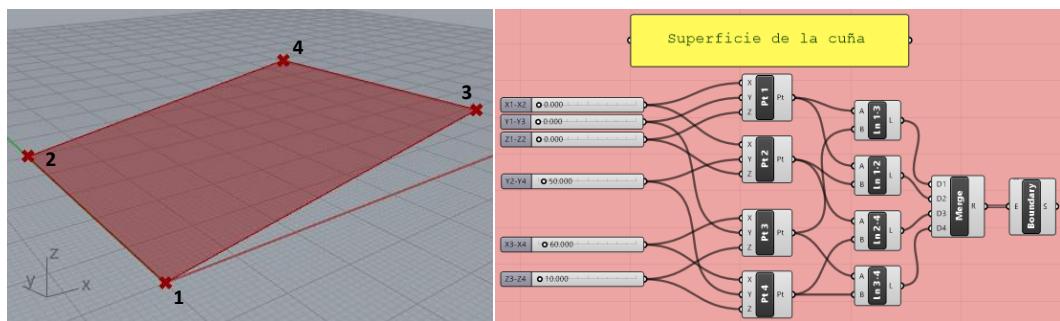


Fig. A.I-1. Las líneas serán generadas a partir de 4 puntos que definiremos como parámetros de entrada. Una vez obtenida una línea continua y cerrada a partir de las líneas anteriores (*Merge*), el componente *Boundary Surface* creará una superficie (superficie limitada por la línea cerrada).

Otra particularidad por definir en el diseño será el ángulo de inclinación de la piel a imprimir respecto al plano XY. El conocimiento de este ángulo nos permitirá saber si hay

o no colisión, y por lo tanto dictaminar si la impresión no-planar es válida. Nos limitaremos en el algoritmo a obtener la información de este ángulo sin que éste sea un parámetro de entrada.

Para ello, tal como se indica en la Fig. A. I-2, utilizaremos la longitud de la línea que conecta el punto 1 con el 3 (Ln 1-3), así como la diferencia en Z de los puntos 1-3. El componente *Evaluate* evaluará una expresión dadas unas variables de entrada, de esta forma obtendremos el ángulo buscado.

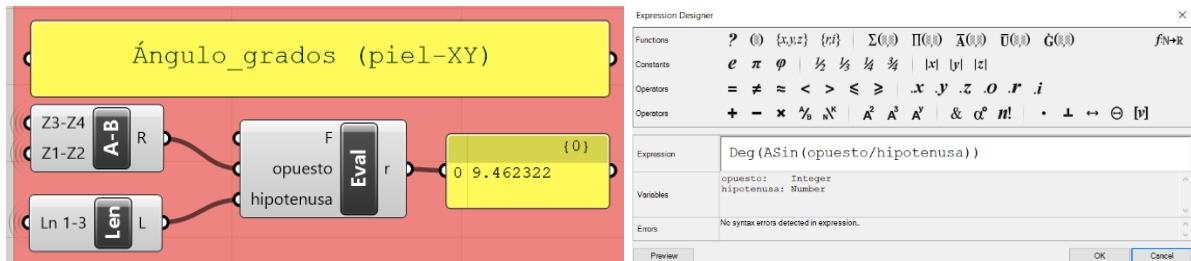


Fig. A.I-2 Haciendo uso del componente *Evaluate*, obtendremos a través de la expresión indicada el ángulo en grados entre la superficie a imprimir y el plano XY.

Traslación de la geometría a su zona de impresión

Al igual que haríamos si programáramos el contorno de una pieza para su fresado mediante CNC, es habitual establecer unas coordenadas absolutas respecto a un cero pieza (un origen), tomado éste generalmente en la coordenada (0,0,0). Es decir, nos olvidamos a la hora de diseñar y programar dónde posicionaremos la pieza en el volumen de fabricación de la fresadora. El mismo procedimiento usaremos en fabricación aditiva.

Ahora bien, a la hora de fabricar nuestra pieza será necesario establecer una correlación entre las coordenadas pieza y máquina. Este hecho se podría resumir preguntándonos dónde quedará la coordenada (0,0,0) en nuestra base de impresión.

En la Witbox 1 la coordenada máquina (0,0,0) está establecida en la esquina inferior izquierda de la cama de impresión, pero a nada que hayamos hecho unas cuantas impresiones antes nos daremos cuenta de que en la mayoría de las piezas el laminador las sitúa en el centro de la cama, punto que es el centro de gravedad de la masa de la cama, así donde se minimizan los posibles errores de nivelación de la cama.

De tal forma, si nosotros queremos imprimir la pieza en la zona central de la cama tendremos que trasladarla de la coordenada (0,0,0), donde la hemos diseñado, a un punto central de la mesa de impresión. Este punto central en la Witbox 1 se puede definir como la coordenada (148.5,105,0) dado que las dimensiones de la cama son 297 mm x 210 mm.

Antes de mover la geometría es necesario indicar dos aspectos. El primero de ellos deriva en que si nuestro vector de traslación lo tomamos como (148.5,105,0) moverá nuestro punto 1 de la pieza (ver fig. A. I-1), punto que está situado en la esquina inferior izquierda al centro de la cama, y este proceso no supone posicionar la pieza en el centro de la cama.

Para que sí esté en el centro de la cama habrá que restarle a este vector la mitad de las longitudes de las líneas que definen el largo y ancho de la cuña, teniendo en cuenta que lo importante son las longitudes de las líneas proyectadas en el plano XY.

El segundo aspecto por considerar es consecuencia de que la impresión de la piel necesita de un soporte, lo que hemos denominado base de la pieza. Si tomamos como ejemplo la línea 1-2 (ver fig. A. I-1), vemos que ésta está en $Z=0$ lo que implica que se imprimiría sobre la misma cama de impresión. Si tenemos en cuenta que es necesario dejar un espacio entre la boquilla y la cama (altura de capa, que definiremos en un valor de 0.2 mm), y que debajo de la piel es necesario soporte, nuestro vector de traslación nunca podrá tener una componente cero en Z. Nosotros vamos a definir un valor de 0.2 mm para la altura de capa y 2 mm para el soporte-base. (Es necesario puntualizar que si nuestro deseo fuera el de optimizar el tiempo de impresión se podría minimizar la altura de la base).

Ambos aspectos están recogidos en el cálculo del vector de traslación (Fig. A. I-3).

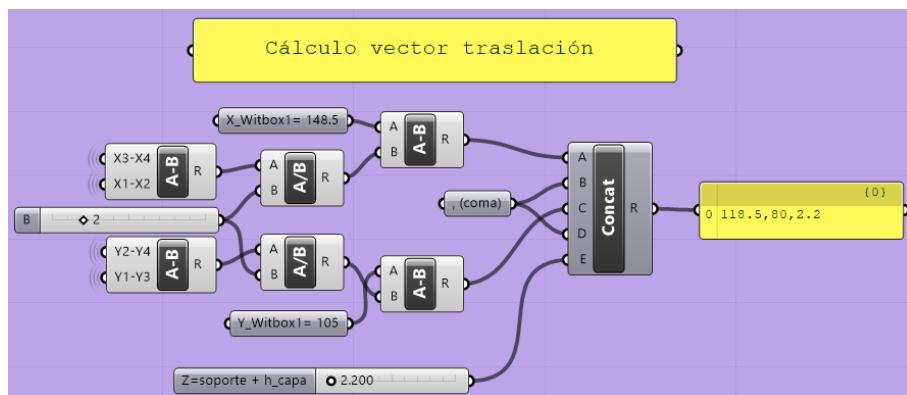


Fig. A.I-3. Algoritmo para el cálculo del vector de traslación.

Teniendo en cuenta lo anteriormente explicado, el componente *Move* nos permite trasladar una geometría dada indicando un vector de traslación. (Fig. A. I-4) y (Fig. A. I-5).

A partir de aquí, el procedimiento lo dividiremos en dos partes. La primera de ellas derivará en el diseño de la base; la segunda parte, la más compleja y novedosa, nos permitirá obtener las trayectorias para la impresión de la piel.

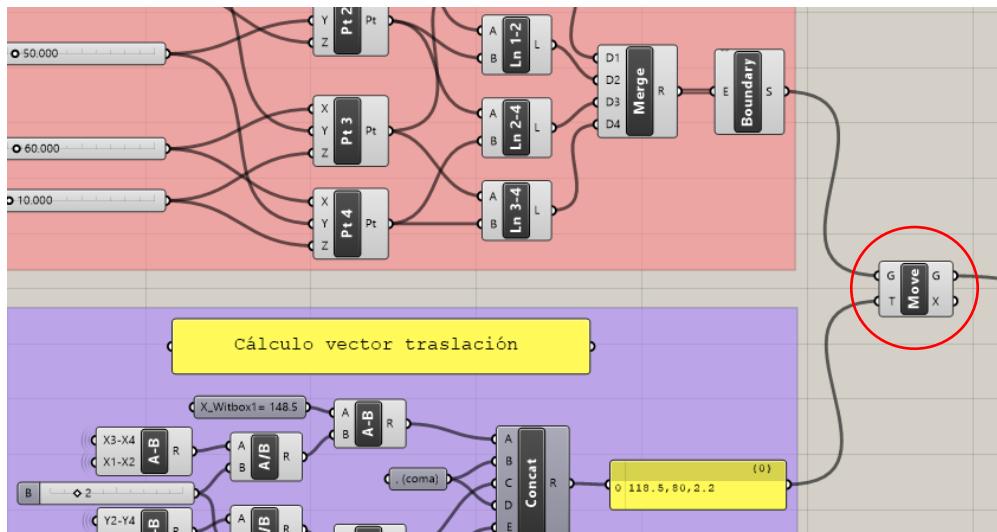


Fig. A.I-4. El componente *Move* (indicado con un círculo rojo) trasladará la geometría de la piel anteriormente creada a una nueva posición definida por el vector de traslación.

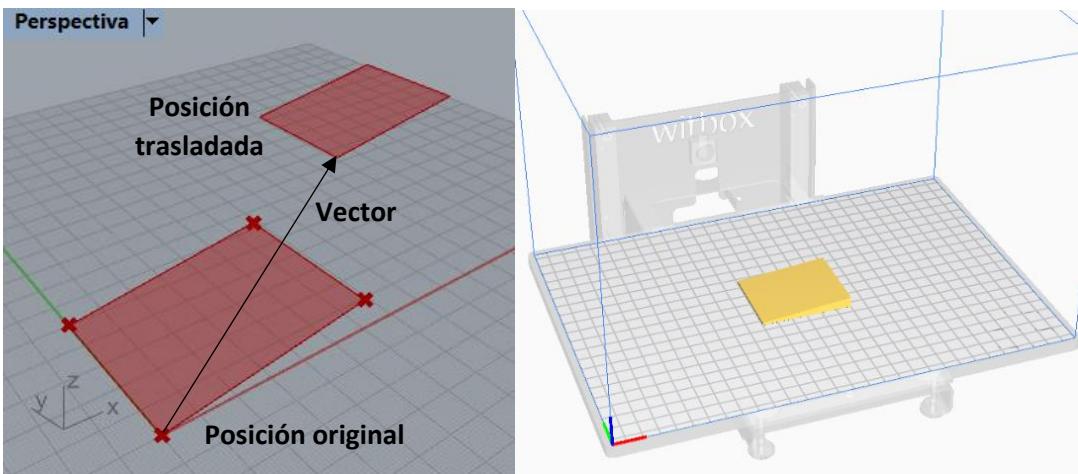


Fig. A.I-5. Izquierda: Todos los puntos que conforman la piel son reposicionados, haciendo uso de un vector de traslación, a la zona donde se quiere imprimir la pieza. Derecha: La cuña será impresa en el centro de la cama de impresión. Si no hicieramos esta traslación la cuña se fabricaría en el origen de coordenadas de la máquina, que corresponde con la esquina inferior izquierda de la mesa de impresión.

Obtención de la base

Empecemos por el diseño de la base; Para ello será necesario convertir la superficie “piel” en un BRep. El término BRep (Boundary Representation) es una forma de definir una geometría usando sus fronteras o límites, algo semejante a lo que hemos hecho para obtener nuestra superficie a partir de las cuatro líneas que eran su límite. Yendo más allá, este hecho implica que un sólido o una polisuperficie está compuesta de una colección de aristas y caras, y lo que es más importante, las podemos obtener. Para ello, el componente *DeBrep* deconstruye un BRep previamente diseñado.

Dada la importancia de este componente a la hora de obtener la base, y para una mejor comprensión del concepto de BRep, se ha creído conveniente a modo de ejemplo mostrar la deconstrucción de nuestra piel (Fig. A. I-6). Puntualizo aquí que, si bien la conversión de la piel en un BRep sí que es necesaria en este momento, la deconstrucción de ésta se llevará a cabo más adelante.

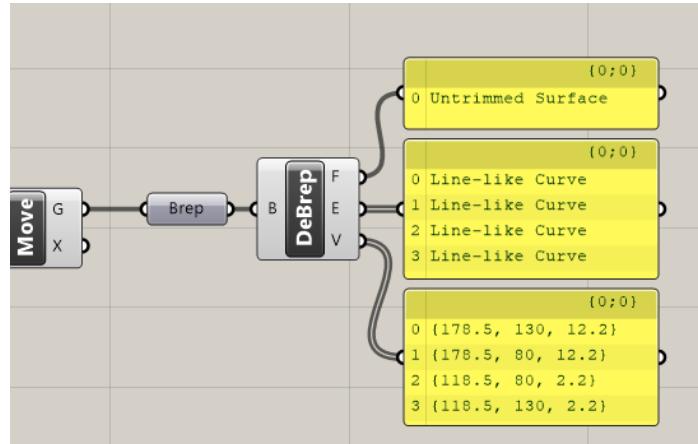


Fig. A.I-6. Deconstrucción de la piel previa conversión de ésta en un BRep (Boundary Representation). Como cabría esperar el componente DeRep descompone y nos indica que la piel previamente diseñada está formada por una superficie (Face, F), cuatro aristas (Edges, E) y cuatro vértices (Vertices, V).

Continuando con el diseño de la base, la idea para crear la misma es construir un volumen cerrado cuya superficie superior sea un offset de la piel y la base inferior sea la proyección de la piel en el plano XY. Estas dos superficies, semejantes a las tapas de nuestro volumen, formarán junto a las paredes un volumen de impresión cerrado, lo que hemos denominado base.

Para comprender cómo es el proceso de creación de la base a partir de la piel será necesario fijarse en el algoritmo indicado en la (Fig. A. I-7), y dentro de ésta en los números a modo de referencia que se han creado. A continuación, basándonos en esas referencias explicaremos detalladamente el proceso.

Partiremos, como hemos dicho con anterioridad, de un BRep que será la piel ①. A partir de ésta queremos crear la parte superior de la base que tiene exactamente la misma geometría que la piel, pero situada en una cota Z menor, cota que equivale a la altura de capa (en nuestro ejemplo 0.2 mm). Para realizar esto utilizamos el componente *Offset* dándole a éste una distancia de 0.2 mm ②. En este punto hay que puntualizar que el *Offset* crea una proyección a través de un vector normal a la piel. Este hecho no es adecuado, ya que lo importante es que sea un offset a través de un vector unidad Z. Para solucionar esto definiremos la superficie offset creada como superficie de proyección y proyectaremos en ésta el BRep piel a través del vector Z ③.

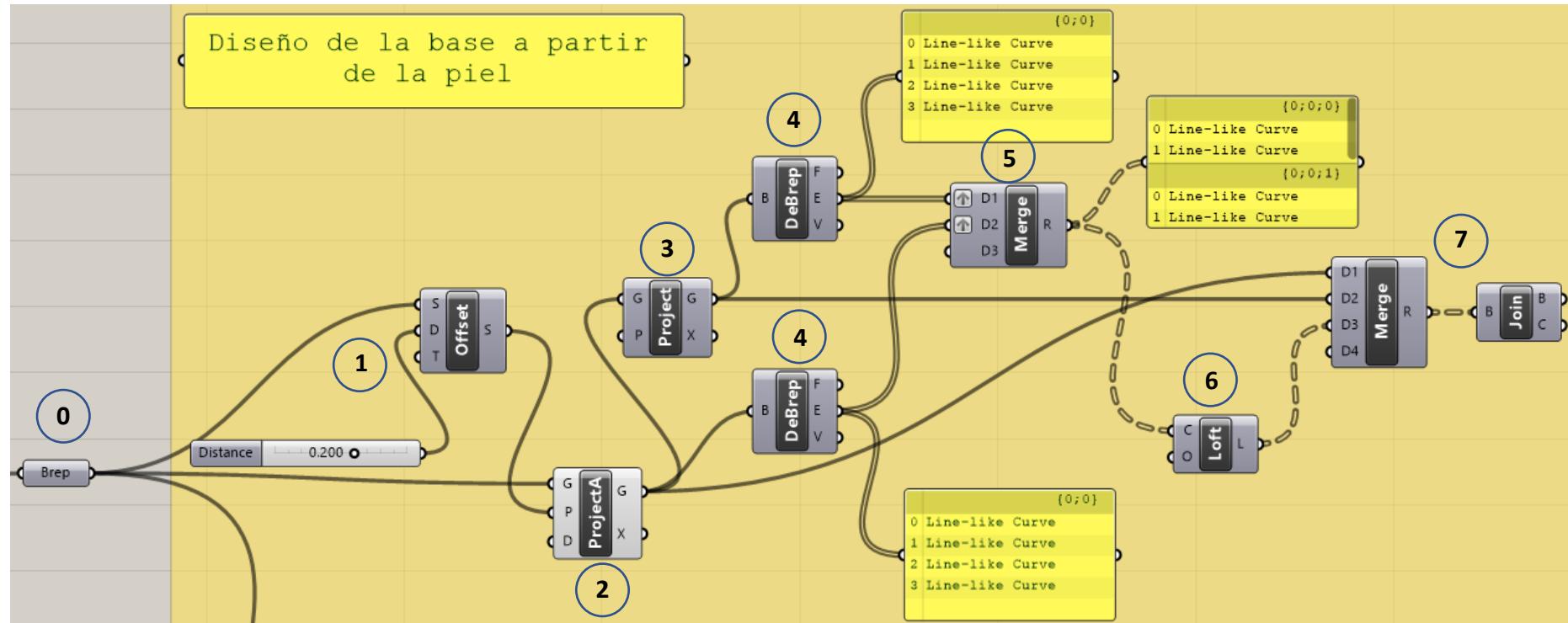


Fig. A.I-7. Algoritmo en Grasshopper® para la creación de la base a partir de la piel.

En la (Fig. A. I-8) podemos ver estos pasos de proyección de una forma más detallada.

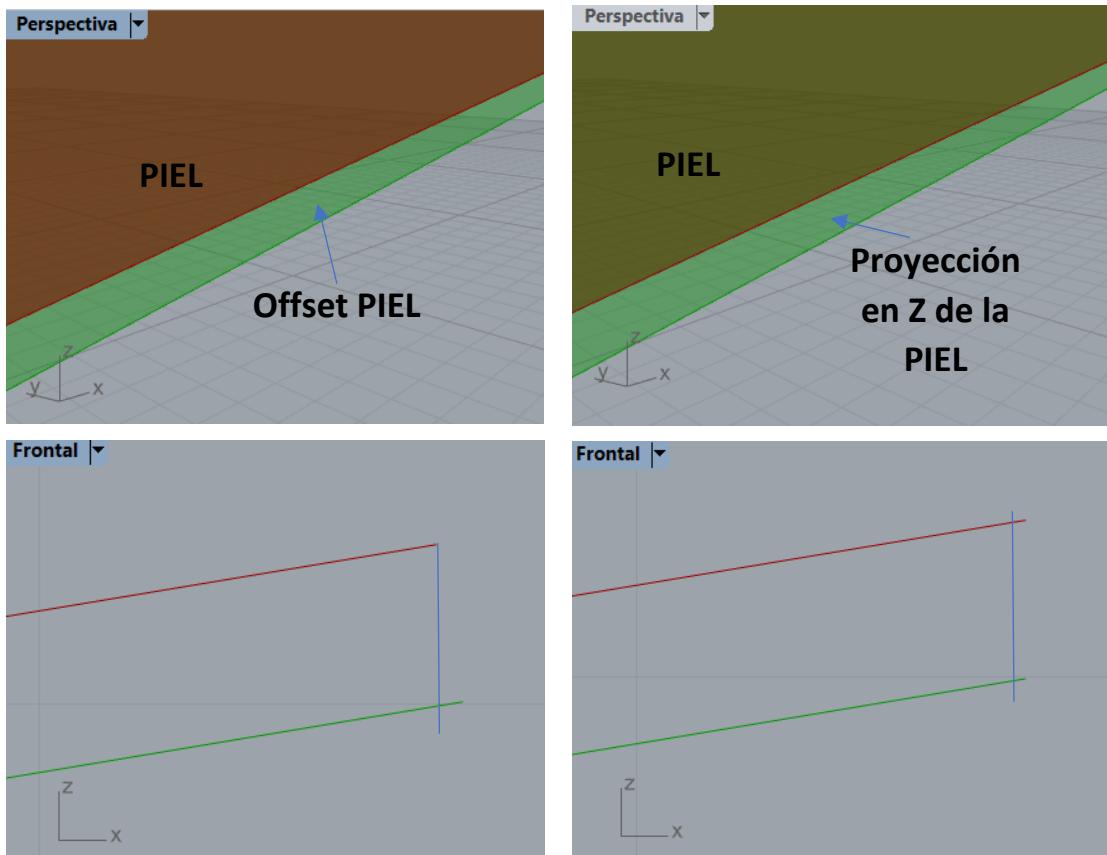


Fig. A.I-8. Diseño de la parte superior de la base a partir de la piel. Izquierda: Si realizamos un offset de la piel la geometría creada no es válida. En la vista frontal de la acción Offset, y ayudándonos de una línea azul, vemos como las superficies no finalizan en la misma cota x. Derecha: Para solucionar este hecho utilizamos la superficie offset creada como destino de proyección de la piel a través de un vector Z de la misma. En la vista frontal de dicha acción vemos como el error ha sido solucionado.

El siguiente paso consistirá en realizar la parte inferior de la base. Para ello proyectamos la superficie obtenida en el paso anterior sobre el plano XY a través de un vector Z ③. Esta superficie obtenida será la que irá adherida a la cama de impresión (Fig. A. I-9).

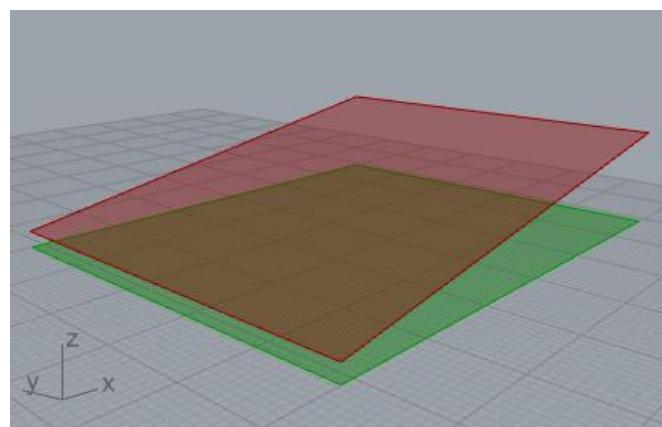


Fig. A.I-9. En verde la superficie obtenida de proyectar la superficie en rojo sobre el plano XY.

Obtenidas las superficies vamos a deconstruir éstas, como ya hemos explicado en párrafos anteriores, aislando sólo las líneas o aristas ④. Tras fusionar dos a dos éstas con el componente *Merge* ⑤, entendiendo dos a dos la fusión de la línea de arriba perteneciente a la superficie superior con la inmediatamente inferior, realizaremos un *Loft* obteniendo las superficies o caras laterales de la base ⑥. El componente *Loft* crea una superficie a partir de un conjunto de curvas (Fig. A. I-10).

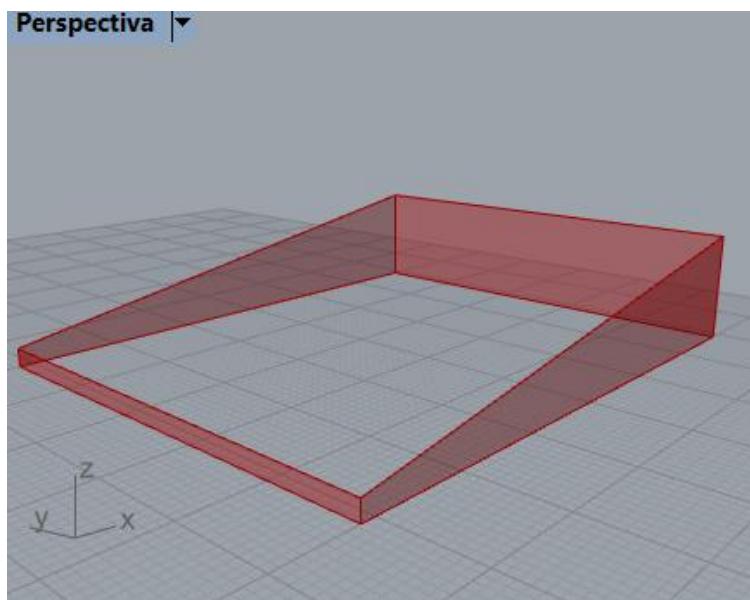


Fig. A.I-10. Obtención de las caras laterales de la base mediante el componente *Loft*.

Para terminar, solo nos queda fusionar y unir (*Merge* y *Join*) las tres superficies creadas ⑦, obteniendo de esta forma un volumen hueco, pero perfectamente estanco, quedando diseñada la geometría denominada como base. En este punto hay que recordar que para imprimir un modelo éste debe de ser estanco/hermético o sólido evitando de esta forma dobles superficies y posibles colisiones entre ellas (Fig. A. I-11).

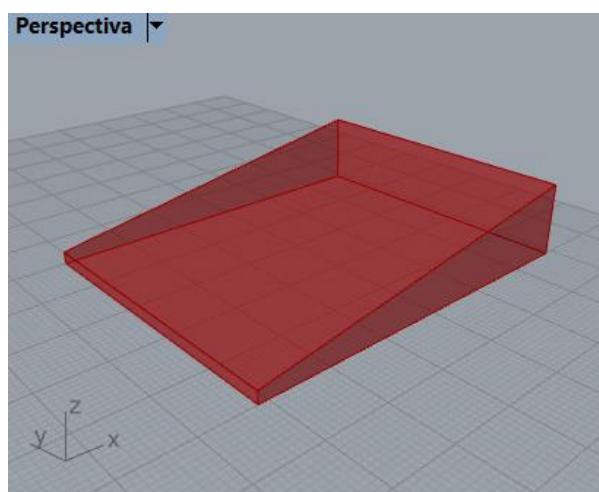


Fig. A.I-11. Tras fusionar y unir las tres superficies se obtiene un volumen perfectamente estanco adecuado para imprimirlo.

Una vez obtenida la base en Grasshopper® es necesario realizar una operación conocida como *Bake* (cocinar el diseño). En este momento la geometría pasa a formar parte del entorno de Rhinoceros® (Fig. A. I-12).

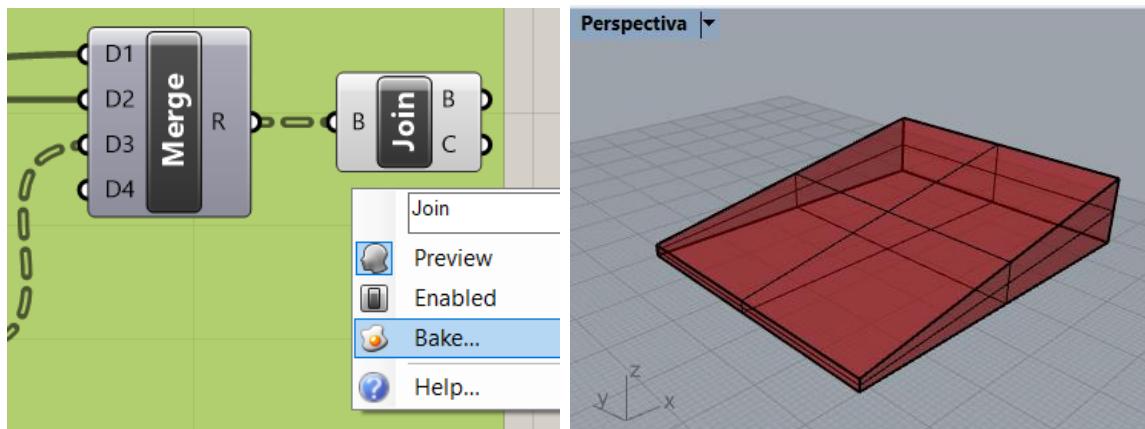


Fig. A.I-12. Tras cocinar (*Bake*) en Grasshopper® la base que sustentará la piel, la geometría pasará a formar parte del entorno de Rhinoceros®.

Para imprimir la base diseñada, dado que vamos a utilizar un procedimiento de impresión FDM tradicional, capa a capa, seguiremos el procedimiento explicado en el punto 2.4. “Del diseño a la obtención de la pieza real”. Recordemos que en Rhinoceros® habrá que exportar el fichero como *.stl, laminar éste utilizando un software específico, en este caso CURA®, y finalmente imprimirla en la impresora 3D.

Trayectorias para la impresión de la piel

Una vez diseñada e impresa la base, continuaremos con la segunda parte en la que hemos dividido el proceso, que consiste en obtener las trayectorias para la impresión de la piel.

Nuestro punto de partida, al igual que en la obtención de la base, será el BRep que representa la piel. A partir de aquí, en primer lugar, obtendremos una serie de líneas que surgen de la operación de cortar la piel mediante un conjunto de planos, en base a una dirección y una separación entre estos. Para que se entienda esta operación, vayámonos a la impresión tradicional, donde un software laminador corta nuestra geometría, el volumen 3D, en capas o secciones perpendiculares al eje Z, separadas estas secciones una distancia llamada altura de capa. Aquí, procederemos de la misma forma, pero en vez de laminar un volumen, laminaremos una “geometría 2D”, una superficie que es la piel (aunque ésta adquiera geometría compleja en los tres ejes). Y laminaremos o cortaremos dicha superficie en base a planos perpendiculares a un vector y una distancia entre ellos, obteniendo de esta forma una serie de líneas.

Para ello, utilizaremos el componente *Contour*, el cual introduciendo un BRep y los ya nombrados parámetros de vector de dirección y separación entre cortes, obtendrá una lista de curvas, en el caso de la cuña líneas rectas.

En la figura (Fig. A. I-13) se muestra el laminado de la piel en la cuña, así como el algoritmo utilizado para ello. En este punto es necesario advertir que el vector de dirección tomado

es [1,0,0] y la separación entre secciones de 0.4 mm. La elección de estos parámetros no es aleatoria. Por un lado, la dirección del vector es importante de cara a medir y comparar la rugosidad entre modelos realizados mediante impresión tradicional y no planar, objetivo del presente trabajo. Por otro lado, podemos fabricar pieles formadas por capas sucesivas donde cada una de las capas tenga una orientación diferente. Esta filosofía de trabajo formaría parte del campo de la resistencia de materiales y el diseño de nuevos materiales bajo un concepto de anisotropía controlada.

En cuanto a la distancia entre secciones de corte, podemos trabajar bajo la hipótesis de que dicha distancia es igual al diámetro de *nozzle*. Esto equivaldría a suponer que la sección transversal del filamento extruido es redonda, lo cual no es cierto ya que se produce un pequeño “aplastamiento” de este redondo. Aun así, vamos a adoptar la forma redonda como hipótesis de partida. Si teniendo en cuenta distintas publicaciones técnicas (*papers*), tomamos una geometría más precisa como sección del filamento extruido, podremos optimizar más este parámetro “distancia” cuya finalidad principal es que no existan huecos entre las fibras contiguas obteniéndose así una superficie sin huecos.

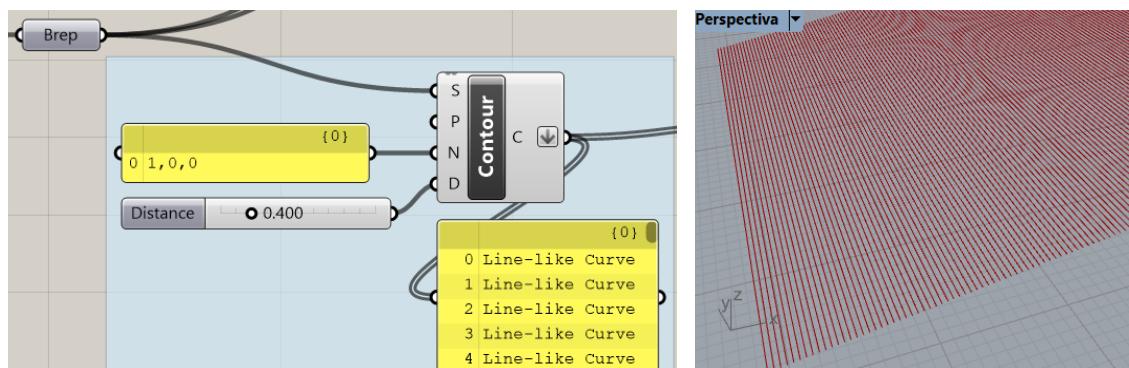


Fig. A.I-13. Izquierda: El componente *Contour* tiene como parámetros de entrada el BRep de la piel, un vector de dirección y una distancia entre planos de corte sucesivos. Tras procesar la información obtendremos una lista de curvas. Derecha: La piel se convierte en una sucesión de curvas, en este caso rectas, que son el primer paso para la definición de trayectorias (prototrayectorias).

Las líneas obtenidas son en realidad un primer paso para la creación de las futuras trayectorias de la boquilla de impresión, lo que podemos llamar prototrayectorias. De tal forma, y con este pensamiento en mente, la boquilla se posicionaría en el inicio de la línea y se trasladaría depositando material hasta el punto final de la línea. Luego se tendría que trasladar en vacío al punto inicial de la siguiente línea y volvería a depositar material hasta el punto final de dicha línea, y así proseguiría el proceso hasta el punto final de la última línea.

Aunque este planteamiento de trayectorias en ocasiones podría ser el correcto, para la impresión de la piel de la cuña se puede optimizar. La manera de optimizar las trayectorias consiste en minimizar esa traslación en vacío de forma que la boquilla se posiciona en el inicio de la primera línea, se traslada depositando material hasta el final de la línea y una vez llegado a este punto en vez de irse al inicio de la siguiente línea en vacío se mueve al final de la siguiente línea, con lo que la distancia se ha reducido

drásticamente. Una vez en este punto, se trasladaría depositando material hasta el que era el inicio de la segunda línea (Fig. A. I-14).

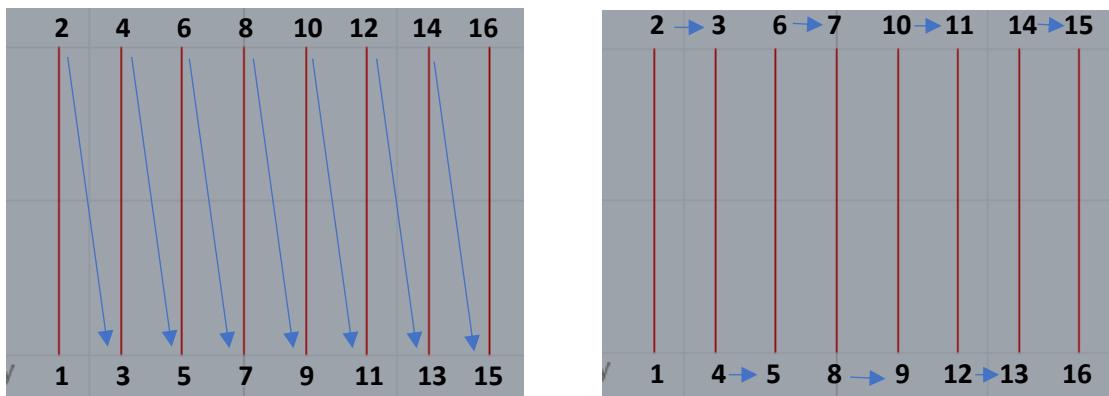


Fig. A.I-14. Izquierda: Camino seguido por la boquilla sin optimizar la trayectoria. Derecha: Camino que sigue la boquilla optimizando la trayectoria. Véase que los desplazamientos en vacío se reducen drásticamente y con ello el tiempo de impresión.

Para realizar esta optimización de trayectorias hemos diseñado un algoritmo (Fig. A. I-15) que funciona de la siguiente manera; como entrada del proceso tenemos una lista de líneas que proceden de la salida del componente *Contour*, visto anteriormente. Estas líneas las vamos a etiquetar y clasificar según un patrón par-impar, en realidad es un patrón TRUE-FALSE, de tal forma que las líneas pares irán a una lista y las líneas impares irán a otra (hay que recordar que la primera línea es la línea 0 en lista). Esta asociación de cada una de las líneas a un grupo u otro lo haremos con el comando *Dispatch*.

Si pensamos en cada una de las líneas como una trayectoria entonces éstas tendrán inicio y final. Aplicaremos al grupo formado por las líneas impares el comando *Flip* que invertirá el sentido de las líneas de dicho grupo. De tal forma, en dicho grupo donde estaban los inicios estarán ahora los finales y viceversa.

Finalmente volveremos a crear un solo grupo, una sola lista, cogiendo la primera línea del grupo par, luego la primera línea del grupo impar, después la segunda línea del grupo par y la segunda línea del grupo impar, y repetiremos el procedimiento hasta completar la lista. Esta última operación la haremos con el comando *Weave*.

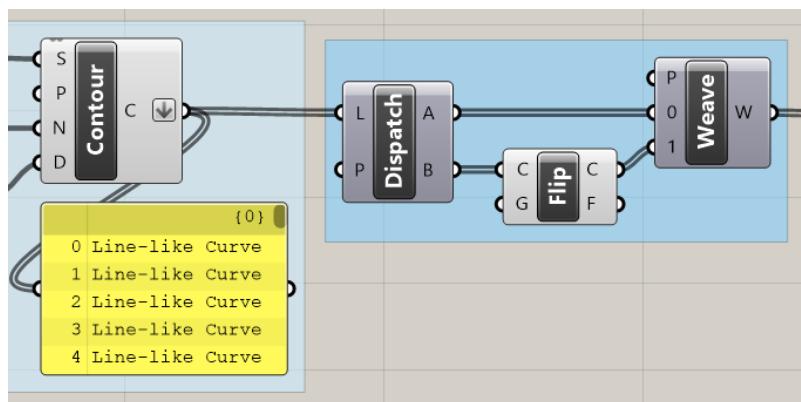


Fig. A.I-15. Algoritmo usado para optimizar las trayectorias de la boquilla.

Hasta aquí, cuando hemos hablado de trayectorias de la boquilla lo hemos hecho basándonos en la existencia de líneas previamente definidas en base a unos algoritmos. Ahora bien, las máquinas CNC, donde incluimos a las impresoras 3D, no entienden de líneas sino de puntos, para ser más exactos de las coordenadas de dichos puntos.

Este tipo de máquinas se mueve entre puntos y siempre en línea recta (aquí obviaremos pequeñas curvaturas en la microgeometría como consecuencia de aceleraciones y deceleraciones de los motores). Esto implica que, si tenemos una línea recta y hemos definido con dos puntos el inicio y final de esa línea, la impresora se trasladará del punto inicial al punto final en línea recta, dado que dicha línea está definida por dos puntos.

En este punto surgen dos aspectos para tener en cuenta, los cuales requieren de una solución. El primero es cómo actuamos ante la posibilidad de que esas líneas por donde se desplazará la boquilla no sean rectas sino curvas, aquí ya no es válido indicar a la máquina el punto inicial y final de la línea. El segundo aspecto para tratar se deriva de optimizar el movimiento del cabezal, algo visualizado en la Fig. A. I-14, donde esos pequeños recorridos obtenidos serán depositando material, y por lo tanto nos interesa obtener una línea continua (sin levantar la boquilla), compuesta de una serie de líneas rectas de diferente longitud y orientación, lo que se define como polilínea.

La solución a lo anteriormente visto es crear un algoritmo que cubra estos dos aspectos. Para ello, el proceso comienza discretizando cada una de las líneas obtenidas. Recordemos que en este primer ejemplo de la cuña las líneas obtenidas son rectas, pero en ejemplos posteriores obtendremos curvas de diferentes grados. Para discretizar la línea utilizaremos el componente *Divide*, el cual, a partir de una línea, ya sea esta recta o curva, fracciona la línea en tantos segmentos como lo deseas mediante la introducción de un número natural. El componente *Divide* da como resultado (output) los puntos de inicio y final de cada uno de los segmentos rectos obtenidos (Fig. A. I-16).

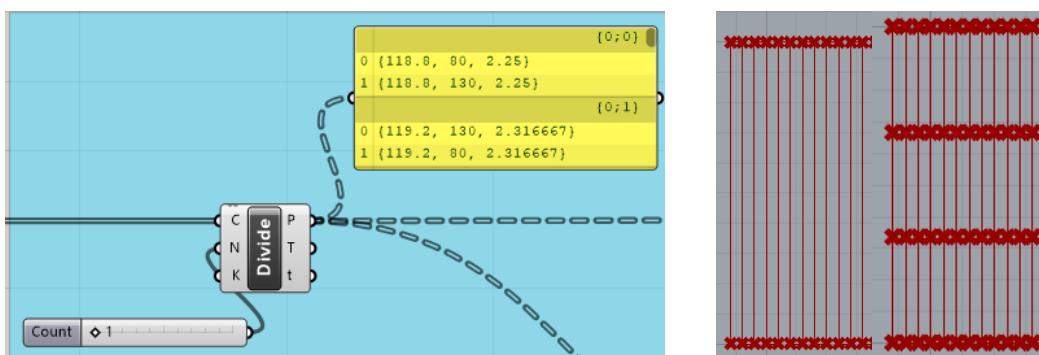


Fig. A.I-16. Izquierda: El componente *Divide* de Grasshopper® secciona una línea dada en tantos segmentos como se indique a través de un número natural, obteniéndose de tal forma las coordenadas del punto inicial y final que definen a dichos segmentos. Derecha: División de las líneas en uno y tres segmentos respectivamente. En el caso de una línea recta no es óptimo obtener más de un segmento.

En el caso de líneas rectas, como en el ejemplo de la cuña, lo óptimo es discretizar una línea a través de un solo segmento, obteniendo el punto inicial y final del mismo. La

división de una línea recta en múltiples segmentos solo añadirá más código y puede generar pulsos innecesarios en la trayectoria de la máquina.

En el caso de curvas, la discretización es mucho más interesante desde el punto de vista ingenieril. Pongamos un ejemplo; situemos puntos separados la misma distancia a lo largo de la curva y tracemos líneas rectas entre estos puntos, obteniendo de esta forma una polilínea. Pronto nos daremos cuenta de que la polilínea obtenida se ajustará más a la curva primitiva cuantos más puntos situemos, es decir, cuantos más segmentos obtengamos ya que estos serán de menor longitud.

Esta filosofía de discretización tiene la ventaja de que es muy sencilla de implementar, pero en geometrías donde se alternen pequeñas curvaturas con otras más grandes la calidad del ajuste será deficiente. En estos casos necesitaríamos que en la zona con más curvatura la longitud de los segmentos sea menor.

En este trabajo, se seguirá el camino más sencillo de discretización que consiste, como ya hemos comentado, en dividir una curva en segmentos rectos de la misma longitud. Para optimizar esta discretización consiguiendo mejores ajustes entre la polilínea obtenida y la curva teórica, García Cuevas et al. [24] exponen discretizaciones de curvas basadas en tolerancias de distancia, ángulo y longitud mínima y máxima de cada segmento en que queda dividida la curva.

Siguiendo con el algoritmo utilizado en el presente trabajo, tras dividir la línea, que será recta en la cuña y curva en casos posteriores, obtendremos puntos, pertenecientes éstos al inicio y final de cada segmento en que ha sido discretizada la línea. Estos puntos tienen dos cometidos que son; la utilización de éstos en un proceso posterior como coordenadas en la trayectoria de la boquilla, y la generación de los segmentos, valiéndonos de dichos puntos, con la finalidad de calcular la longitud de dichos segmentos que nos serán de utilidad para establecer el parámetro de extrusión, E.

Cálculo del parámetro de extrusión “E”

Afrontemos ahora cómo calcular el parámetro E. Recordaremos, de un capítulo anterior, que las impresoras 3D disponen de cuatro motores paso a paso; Tres de ellos serán los encargados de posicionar la boquilla en un punto X, Y, Z del volumen de impresión. El motor restante será el encargado de hacer girar la rueda dentada del extrusor que, junto con un rodamiento u otra rueda dentada, arrastrará el filamento hacia el hotend y de este a la boquilla. Resumiendo, este último motor, que podemos denominar “motor E”, controlará el flujo de material extruido.

En el GCode que interpretará la impresora, en el caso de un movimiento del cabezal de extrusión donde se quiera extruir material (G1), además de las coordenadas del punto a donde se desplaza será necesario insertar en la misma línea el parámetro E, que representa la cantidad de filamento en mm (a menos que esté seleccionado un modo de extrusión volumétrica) al que tiene que alimentarse al hotend para obtener un hilo o fibra impresa con la geometría deseada.

Teniendo en cuenta la geometría de la sección transversal del filamento depositado [21], y su reducción a una sección circular, además de la hipótesis de conservación, en la que el material que entra es el mismo que será extruido, se pueden realizar los cálculos necesarios para obtener dicho parámetro E [25].

Volviendo al algoritmo, vamos a explicar de manera detallada, apoyándonos en la figura Fig. A. I-17, para una mejor comprensión, el procedimiento a seguir para la obtención del parámetro E necesario en el GCode.

Empecemos con la parte marcada como ① en la figura. Se trata del componente *Expression* que evalúa una expresión matemática. En este caso, disponemos de unas variables expuestas en ② que actúan como *inputs* de la expresión. Tras evaluarlas, a través de dicha expresión obtendremos el parámetro E.

Como apreciamos en ②, las variables altura de capa, anchura de extrusión, diámetro de filamento y multiplicador de extrusión son variables iniciales por introducir pero que se mantiene constantes a lo largo del proceso. Es decir, el diámetro del filamento, por poner un ejemplo, es una variable que introducir según utilicemos una máquina u otra, sin embargo, este diámetro se mantendrá constante a lo largo del proceso. Lo mismo podemos decir de las otras tres variables: altura de capa, anchura de extrusión y multiplicador de extrusión.

Sin embargo, existe otro input que es la longitud del segmento, lo que se traduce en la longitud de trayectoria, que variará en cada línea de código. Esto es así porque el parámetro de extrusión E se calcula de forma absoluta y no relativa, lo que implica que se define un E0 ($E=0$) cuando empieza la impresión y se van sumando las longitudes a medida que transcurre la impresión. Por ejemplo, si el primer y segundo segmento miden 50mm, la primera E se calcula en base a 50 mm y la segunda E en base a 100 mm.

Para obtener esta longitud variable, siempre incremental si no existen retracciones de material y si lo que queremos conseguir es una deposición continua, tomaremos como punto de partida los puntos inicial y final de los segmentos que hemos obtenido mediante el comando *Divide* (puntualizamos que lo que nos interesa son los puntos no los segmentos). Una vez esto, uniremos estos puntos formando una polilínea (*PLine* ③). La polilínea así obtenida la podemos equiparar al recorrido o trayectoria que realizará la boquilla.

A continuación, explosionamos la polilínea con el comando *Explode* ④ consiguiendo de esta forma, ahora sí, los segmentos por separado que conforman la polilínea anterior. Si no lleváramos a cabo estos pasos en el orden establecido no podríamos averiguar la longitud del desplazamiento entre las curvas obtenidas al hacer el corte de la piel (Fig. A. I-13). Este recorrido entre curvas, en este primer caso líneas rectas, se lleva a cabo en G1 depositando material y por lo tanto es necesario obtener su longitud para luego introducirla en el cálculo de E.

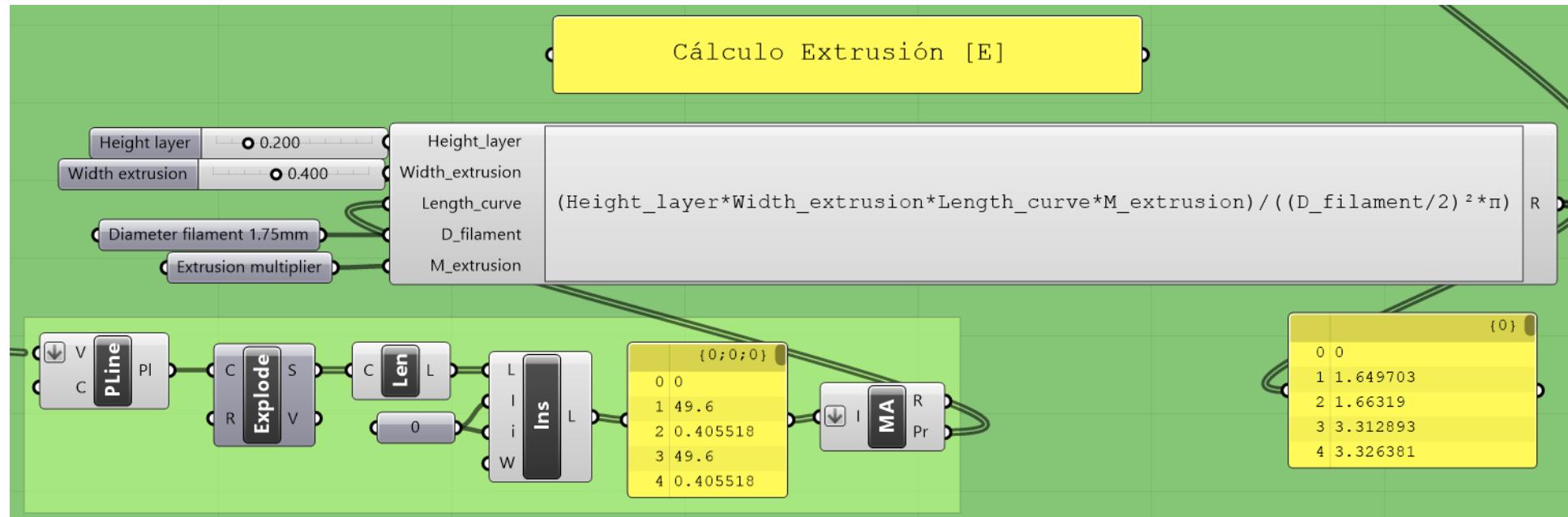


Fig. A.I-17. Algoritmo utilizado para el cálculo del parámetro de extrusión, E.

Conocida la longitud de cada segmento con el comando *Len* (5) se trata ahora de realizar una lista inicializada en 0 donde se listan de manera ordenada las longitudes de cada segmento (6). Evidentemente, la trayectoria que seguirá la herramienta tiene un sentido, con un punto de inicio y un final. El punto de inicio es un punto y como tal no tiene longitud, de tal forma la entrada cero de la lista es un cero. A partir de aquí se van listando las longitudes de cada segmento cuyo orden es el sentido de recorrido de la herramienta.

Pero ya hemos comentado anteriormente que el parámetro E tiene que ser “sumativo o absoluto”. El comando *Mass Addition (MA)* (7) logrará este cometido al introducir el listado anterior.

Una vez obtenida la variable de entrada “longitud de la curva”, junto con los otros *inputs* establecidos anteriormente, se evaluará la expresión para el cálculo de E. Conseguiremos de esta forma un listado (8) con el parámetro E que insertaremos, junto con las coordenadas, en cada una de las líneas del código GCode que precisen de un desplazamiento G1.

Obtención del *.gcode para la impresión de la piel

En el programa de código G que interpretan las impresoras 3D, al igual que en otras máquinas basadas en CNC, podemos distinguir tres partes bien diferenciadas. La primera de ellas es lo que denominamos la cabecera del programa o protocolo de inicio. En esta parte se ejecutan, entre otras, instrucciones relativas a calentar el hotend, definir las unidades en milímetros o posicionar el cabezal en el punto definido por la máquina como inicial (*home*).

Tras la cabecera, se encuentra el cuerpo del programa. Éste está formado por la secuencia de desplazamientos necesarios para imprimir la pieza, o parte de ésta, mediante unas trayectorias definidas. De tal forma, encontraremos líneas donde se dan movimientos depositando material (G1) y líneas con movimientos de posicionamiento en vacío (G0). Junto con el G0 o G1 aparecerá en la misma línea la velocidad de avance (F), las coordenadas a alcanzar con el movimiento (X, Y, Z) y el parámetro (E) relativo a la extrusión de material.

La tercera, y última parte, en la que dividimos el programa es denominada como fin de programa o protocolo final. En esta parte bajaremos la cama para que nos sea más fácil extraer la pieza, llevaremos el cabezal a la posición *home* y enfriaremos el hotend, entre otras instrucciones.

Cada máquina tiene un protocolo establecido en cuanto al código que aparece en cada una de las partes y cómo éste aparece. Además, este protocolo también sufre variaciones según el laminador utilizado. Para conocer el protocolo que establece el laminador Cura® en la impresora 3D que vamos a utilizar en este trabajo, Witbox 1, abriremos en el bloc de notas de Windows el archivo *.gcode obtenido al laminar el *.stl de la base de la cuña. En este punto podemos decir que cualquier gcode generado a partir de la laminación de una pieza que queramos imprimir, siempre establecidos el laminador Cura® y la

impresora Witbox 1, nos ayudará a obtener y comprender el protocolo que estableceremos en cada una de las partes del programa.

A continuación, nos centraremos en elaborar el cuerpo del programa a través de Grasshopper®.

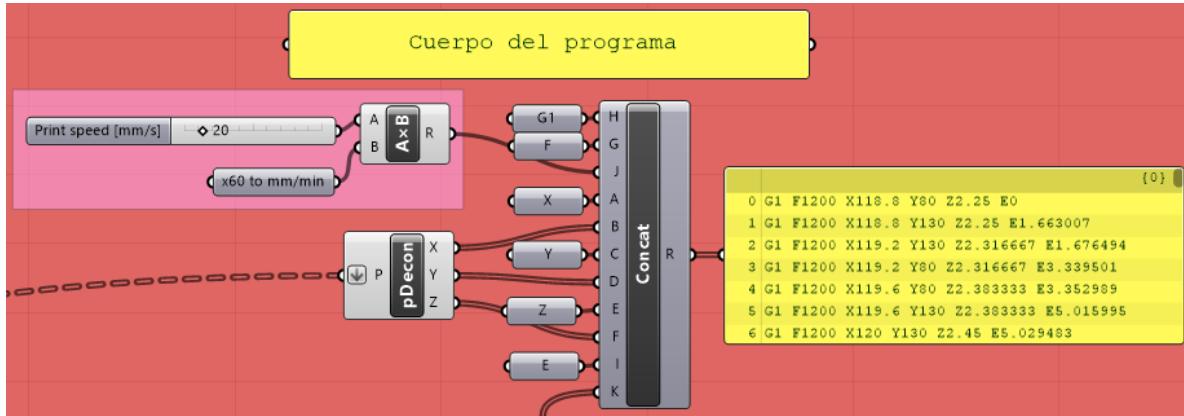


Fig. A.I-18. Algoritmo para la creación del cuerpo del programa. Lo que obtendremos es un texto (en amarillo) formado por una secuencia de líneas.

Como apreciamos en la Fig. A. I-18, el cuerpo del programa es una secuencia de líneas con formato alfanumérico. Podríamos decir que es un texto ordenado escrito bajo unas reglas estrictas que son necesarias para que la impresora pueda entender las instrucciones.

El comando *Concatenate (Concat)* de Grasshopper® permite enlazar varias cadenas de texto según un orden establecido. El formato de cada línea, ya comentado anteriormente, estará formado en primer lugar por el código G de desplazamiento. En este proyecto, la impresión de la piel la haremos bajo el requerimiento de que la deposición del material sea continua. De tal forma, utilizaremos el código G1 en cada uno de los desplazamientos. Esta hipótesis podría cambiar según nuestras necesidades de diseño, o en la optimización de impresiones futuras, pero siempre se establecerá al comienzo de cada línea de código un G0 o G1.

Tras el G1 y un espacio necesario para que la máquina interprete las instrucciones, se indica el avance del cabezal (F). Es habitual que el usuario introduzca el valor en mm/s y que la máquina tenga que interpretar dicha variable en mm/min. De tal forma, nosotros como usuarios estableceremos como variable el avance en mm/s que podremos variar según las características de la impresión, y una simple conversión permitirá introducir dicha variable en mm/min en el cuerpo del programa.

Después del avance, se introducen las coordenadas X, Y, Z, en este orden, a donde se desplazará la punta de la boquilla. Para ello, nos serviremos de las coordenadas de los puntos obtenidos mediante el comando *Divide* al dividir las líneas (Fig. A. I-16). El comando *Deconstruct (pDecon)* utilizado aquí es de suma importancia al descomponer la coordenada en sus tres componentes.

Por último, introduciremos al final de cada línea de código del cuerpo del programa el parámetro E calculado anteriormente.

Una vez obtenido el cuerpo del programa, solo nos queda fusionar, en orden y mediante el comando *Merge*, la cabecera del programa, el cuerpo y el fin de programa, obteniendo finalmente el programa que imprimirá la piel sobre la base ya fabricada anteriormente (Fig. A. I-19).

Para obtener el fichero *.gcode que interpretará nuestra impresora bastará con seleccionar y pulsar sobre el botón derecho el programa final (recuadro amarillo que hemos denominado “Programa Piel”) y una vez esto seleccionar *Copy Data Only*.

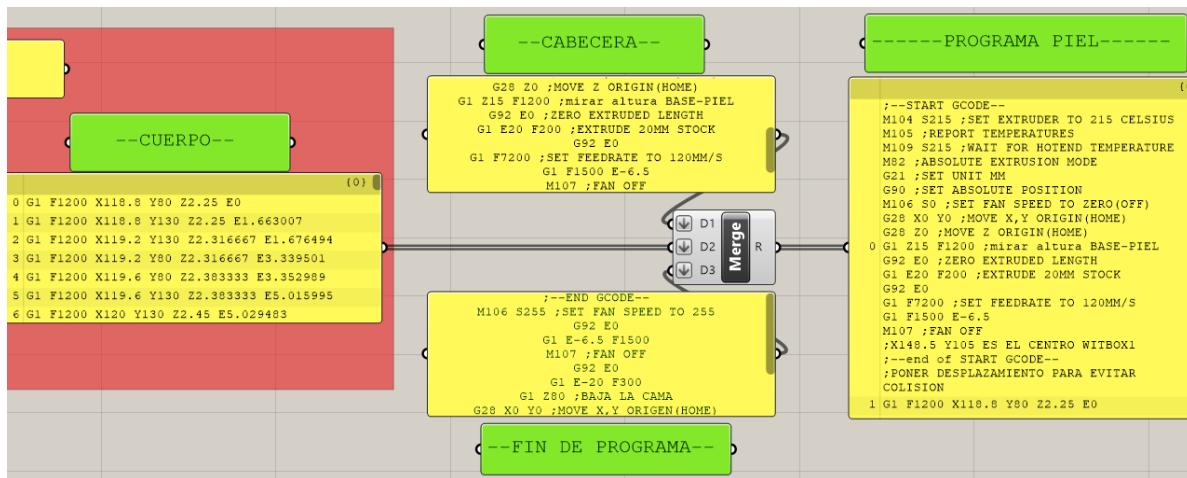


Fig. A.I-19. Fusionando en un solo texto la cabecera, el cuerpo y el fin de programa obtenemos el programa final que imprimirá la piel sobre la base que ya ha sido impresa en una etapa anterior.

Tras abrir, por ejemplo, el bloc de notas de Windows® u otro editor similar, copiaremos en éste los datos que habíamos obtenido en el paso anterior. Por último, es importante que renombremos la extensión del fichero *.txt por una extensión *.gcode (Fig. A. I-20).

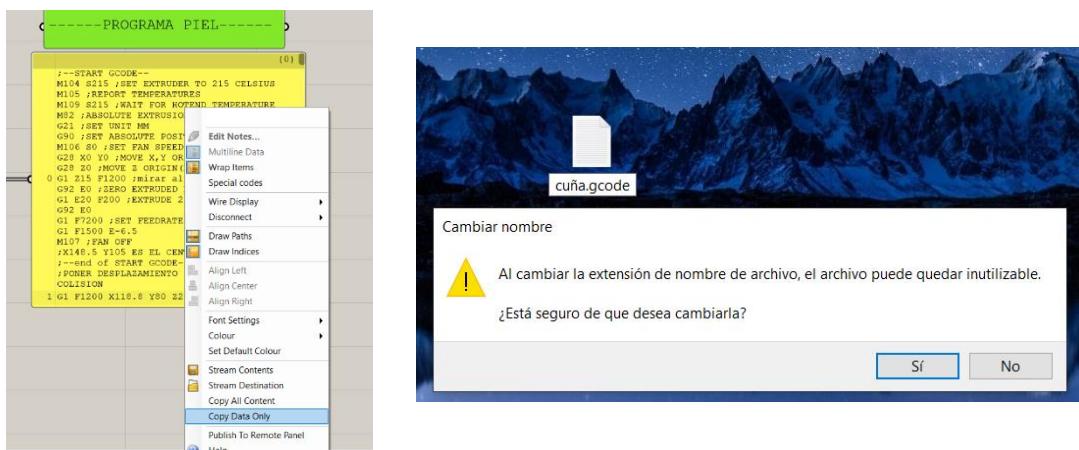


Fig. A.I-20. Izquierda: El fichero *.gcode interpretable por la impresora Witbox 1 se obtiene copiando todas las líneas de código del programa final de la piel sobre un bloc de notas o similar. Derecha: será necesario renombrar el fichero de texto con una nueva extensión *.gcode.

Superposición de pieles

Es habitual en fabricación aditiva FDM que la parte superior de la pieza impresa esté formada por varias capas de material, algo similar a si dispusiéramos varias pieles una encima de la otra. Lo que no es habitual, y es algo novedoso que se propone en este trabajo, es que la impresión de dichas capas superiores se realice de forma no planar, con diferentes orientaciones en cada capa e incluso con diferentes materiales en cada una de ellas.

Para este fin, imprimir varias pieles de forma no planar una encima de la otra, se pueden llevar a cabo diferentes estrategias. En este trabajo se proponen tres de ellas.

La primera estrategia consiste en reutilizar el mismo programa de Grasshopper® y, por lo tanto, el mismo algoritmo que el usado para la obtención de una única piel. En este programa que llamaremos, por ejemplo, “piel2”, se modifica la coordenada Z del vector de traslación sumándole la altura de capa (en nuestro caso +0.2 mm) respecto la piel sobre la que se asiente. Si dado el caso, quisieramos imprimir una tercera capa, ésta tendría un vector de traslación incrementado 0.4 mm en Z respecto al programa original, y así sucesivamente. Junto con la modificación del vector de traslación, se puede variar la orientación de las fibras en cada capa.

Esta primera estrategia, que será la que nosotros utilicemos en el presente trabajo, tiene la gran ventaja de que es muy sencilla y es fácil de implementar. El procedimiento consistirá en ir guardando distintos ficheros con la coordenada Z modificada según el número de piel e ir lanzándolos a impresión uno detrás de otro (primero la base que ya había sido lanzada, luego la piel 1, después la piel 2, a continuación, la piel 3, y así sucesivamente según el número de capas). Sin embargo, esta estrategia también lleva consigo una serie de inconvenientes, y es que es necesario inicializar todo el proceso de impresión cada vez que lancemos la impresión de una nueva capa.

Esta reinicialización se torna indispensable si deseamos cambiar de material en cada una de las capas, pero si utilizamos el mismo material en toda la impresión supone pérdidas de tiempo y posibles imprecisiones en Z debidas a que se debe tocar el final de carrera repetidas veces.

Para paliar estos inconvenientes, y siempre que se utilice un mismo material para todas las pieles, se puede utilizar una segunda estrategia en la cual se elabora un único programa en Grasshopper® que contiene la impresión de todas ellas. Este hecho sí que supondrá una modificación sustancial del algoritmo, a la vez que no se puede utilizar el mismo programa utilizado en la impresión de dos capas para la impresión de otro número de capas. Es decir, el algoritmo pierde flexibilidad.

Existiría una estrategia intermedia, que nuevamente solo podríamos utilizar si usamos el mismo material en todas las pieles, basada en crear mediante la primera estrategia el cuerpo del programa de cada piel y luego insertar secuencialmente en un *.gcode

completo final todos esos cuerpos de programa. Esta tercera opción no es compleja pero sí que requiere cierto conocimiento del código para la manipulación de éste.

El algoritmo completo en Grasshopper® para el diseño e impresión de la cuña se puede visualizar en el Anexo I- A. (A.I-A: Algoritmo Grasshopper® de la cuña).

Impresión de la pieza

En la tabla A.I-1 se muestra el valor asignado a cada uno de los parámetros necesarios para la impresión 3D de la pieza tratada en este anexo. Evidentemente, estos parámetros hacen referencia a los utilizados en el algoritmo Grasshopper® para el diseño e impresión de esta pieza desde un punto de vista no planar, dejando de lado otros parámetros que podemos considerar generales en una impresión 3D tradicional.

CUÑA	
<i>Parámetros generales importantes</i>	
Temperatura BASE [°C]	215
Temperatura PIEL [°C]	215
Velocidad impresión (pieles) [mm/s]	20
Multiplicador de extrusión	1
Ángulo de la superficie de la piel superior respecto al Plano XY [°]	4.76
Nozzle de serie con Witbox 1 [mm]	0.4

ESTRUCTURA	
	Coordenadas Grasshopper®
	Orientación (Contour)
	Distancia (Distance)
	Segmentos curva (Divide)
	Altura capa (Height layer)
	Ancho extrusión (Width extrusion)
BASE	118.5, 80, 2 [*]
1ª PIEL	118.5, 80, 2
2ª PIEL	118.5, 80, 2.2

[*] Según el algoritmo desarrollado, la última capa de la base (superficie de ésta) se situará 0.2 mm por debajo de la primera piel.

Tabla A.I-1. Parámetros de impresión.

Por otra parte, es de gran importancia la orientación que le hemos dado a las pieles. Al definir la orientación de dichas pieles se ha pretendido que la última piel tuviera la misma orientación que la superficie de la pieza impresa mediante el método FDM clásico. De tal forma, sí que podemos hacer una comparación real entre ambas piezas a la hora de evaluar la calidad superficial. El proceso de impresión, así como la orientación de cada una de las pieles se puede observar en la siguiente figura (Fig. A. I-21).

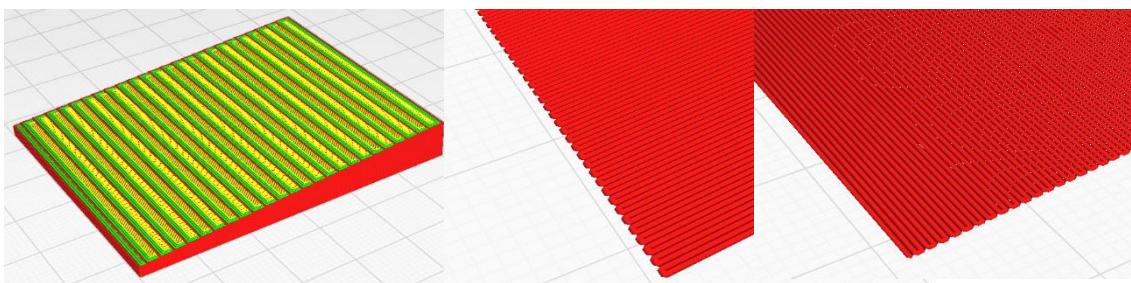


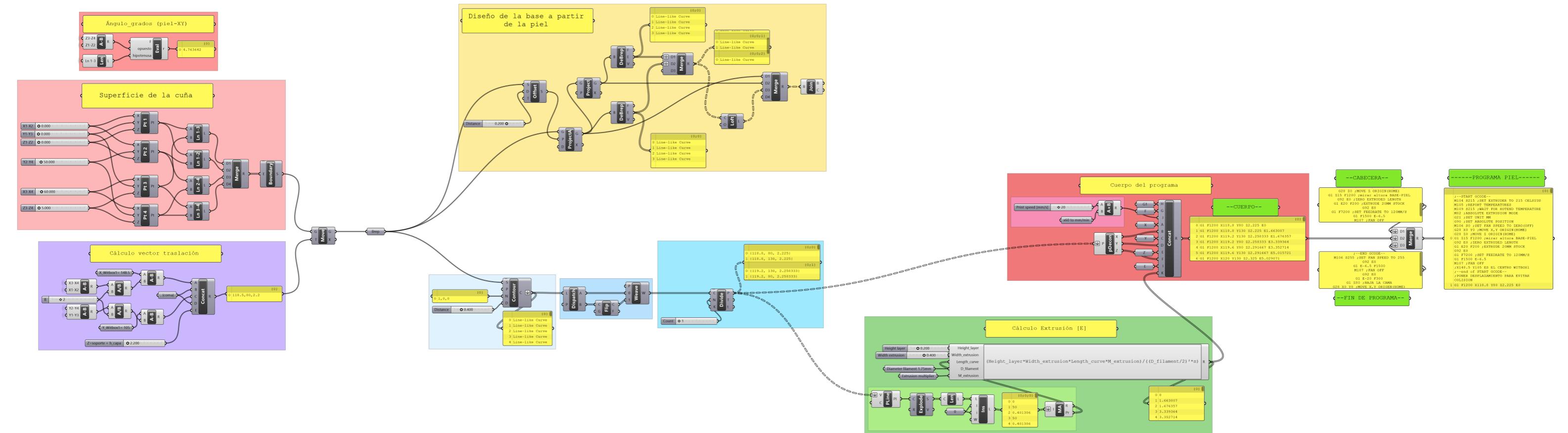
Fig. A.I-21. (Izquierda) Disposición de las capas en la base. Hay que tener en cuenta que la base, además de ser el soporte para las capas no planas, es también idéntica a la pieza que servirá de comparación superficial entre las dos filosofías de impresión tratadas en este trabajo, plana-no plana. (Centro) La primera piel, con una orientación a 0° , amortiguará el escalonamiento. De tal forma, la segunda piel, que se apoyará sobre ésta, tendrá una mejor calidad superficial. (Derecha) La segunda, y última piel, tendrá una orientación a 90° , semejante a la de la base. Así, la comparación de la calidad superficial entre ambas piezas será real.

Por último, se muestran dos fotografías correspondientes a las piezas finales, una vez impresas, utilizando los dos métodos (Fig. A. I-22).

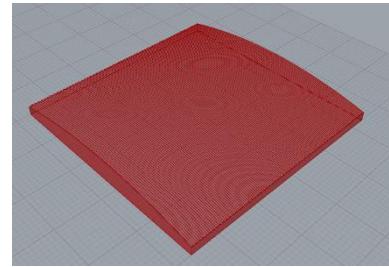


Fig. A.I-22. (Izquierda) Pieza impresa por el método FDM tradicional. (Derecha) La misma pieza impresa por un método FDM no-plano en sus últimas capas.

A.I-A: Algoritmo Grasshopper de la cuña



A.II- Impresión de un arco extruido



En esta segunda pieza, con el fin de aumentar ligeramente la complejidad en la impresión, vamos a introducir una superficie con una pequeña curvatura. Esta superficie es un arco extruido definido por tres puntos (inicio, centro y fin del arco) y una longitud de extrusión. Con el fin de imprimir la pieza con el *nozzle* que viene de serie en la Witbox 1, similar a los *nozzles* usados habitualmente en la impresión 3D por el usuario general, la diferencia de altura entre los puntos inicio-final y el punto central del arco no va a ser elevada, y será de 3 mm. Esto nos permitirá que no existan colisiones entre la boquilla y la pieza impresa, así como destacar, una vez más, la impresión no planar sobre la impresión FDM clásica.

Por otra parte, tanto para el diseño de la piel y la base, así como para la definición de las trayectorias que seguirá la boquilla y la obtención de los programas de impresión, nos basaremos en el algoritmo general utilizado y analizado detalladamente en el punto anterior. Por lo tanto, no tiene sentido explicar nuevamente este algoritmo, y solo nos detendremos a estudiar las variaciones surgidas como consecuencia de que ambas geometrías son distintas. Este planteamiento será el mismo que seguiremos en los puntos posteriores con la impresión de las otras piezas.

De tal forma, comenzemos con el diseño de la superficie “arco extruido”. Como ya hemos introducido, se trata de una superficie generada a partir de la extrusión de un arco definido por tres puntos (inicio, centro y fin). Se ha preferido que las coordenadas en Z del punto inicial y final sean idénticas, y que la coordenada Y del punto central estuviera en la mitad del arco (Fig. A. II-1). Este hecho disminuye la complejidad de la impresión, pero facilita la comprensión del diseño y deja margen a futuras impresiones basadas en este mismo modelo.

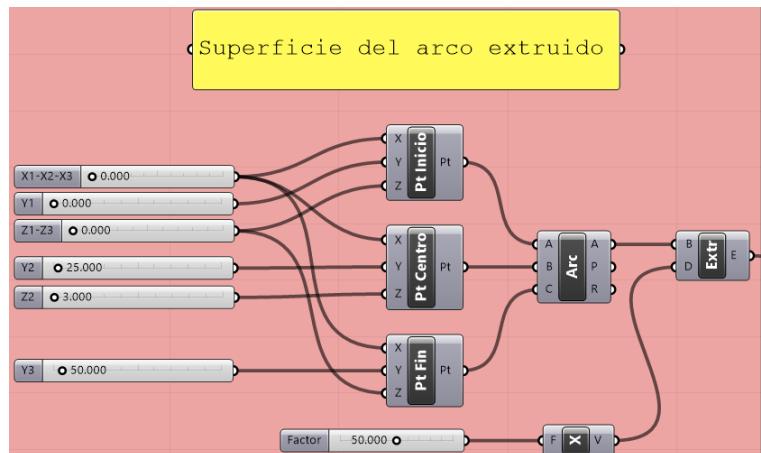


Fig. A.II-1. Algoritmo en Grasshopper® utilizado para el diseño del arco extruido.

Tras trasladar el diseño a una coordenada que coincide con la zona central de la cama de impresión, lugar donde se imprimirá la pieza, recordemos que iniciábamos ahora el proceso para la obtención de la base a partir de la superficie diseñada con anterioridad, que en realidad es la piel o fase superior. En este punto, el proceso de proyección ha sido modificado en la parte inicial haciéndolo más intuitivo.

En la Fig. A. II-2, podemos observar como a partir del BRep de la piel realizamos un movimiento de ésta en el eje Z, en sentido negativo y una distancia igual a la altura de capa. De tal forma, la parte superior de la base se constituye con una copia exacta de la piel. El proceso que sigue es igual al de la cuña y nos permite obtener el archivo *.STL para la impresión de la base tras el laminado en CURA®.

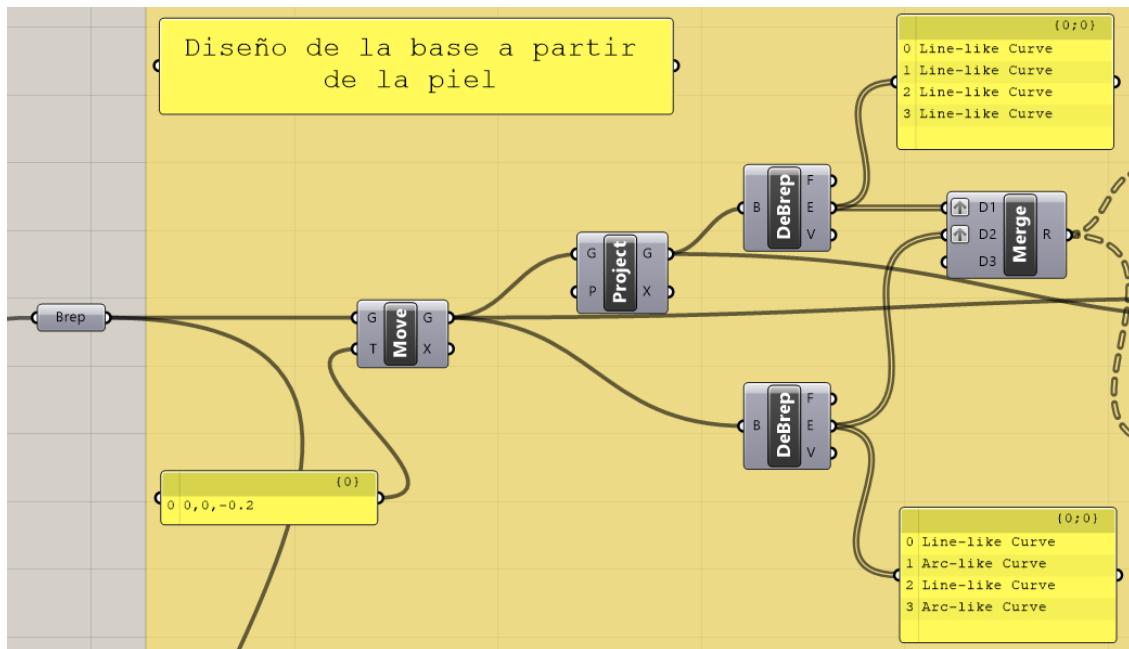


Fig. A.II-2. Simplificación del proceso para la creación de la base. Al inicio de dicho proceso, donde se obtiene la parte superior de la base, se sustituyen los comandos *Offset* y *ProjectA* utilizados en el diseño de la cuña, por un movimiento en -Z de la geometría de la piel.

En cuanto a la otra parte del proceso, por la cual se obtienen las trayectorias que seguirá la boquilla en el proceso de impresión de la piel, también ha sido necesario realizar un ajuste. Se trata de la discretización del arco en segmentos rectos para crear una polilínea.

La superficie de la cuña era una superficie plana y tras seccionar a ésta mediante planos perpendiculares obteníamos líneas rectas. La discretización era trivial y se basaba en que a una línea recta le correspondía un segmento; más segmentos no suponían que la polilínea obtenida se ajustaría mejor al perfil real.

En el caso del arco, ya no tenemos una línea recta sino curva, por lo cual, si tenemos que aproximarnos con una polilínea al perfil real curvo ya no podemos utilizar una polilínea de uno, dos o tres segmentos ya que el ajuste no va a ser adecuado. Sin embargo, la creación de una polilínea que contenga más segmentos de los necesarios tampoco mejorará la calidad superficial y llevará a un incremento del tamaño del archivo *.gcode.

Veamos en la Fig. A. II-3 (Izquierda) lo que ocurre al discretizar el arco con una polilínea formada por tres segmentos; efectivamente como cabía esperar el ajuste a la curva real es deficiente. Para mejorar este ajuste, proponemos incrementar a quince el número de segmentos Fig. A. II-3 (Derecha).

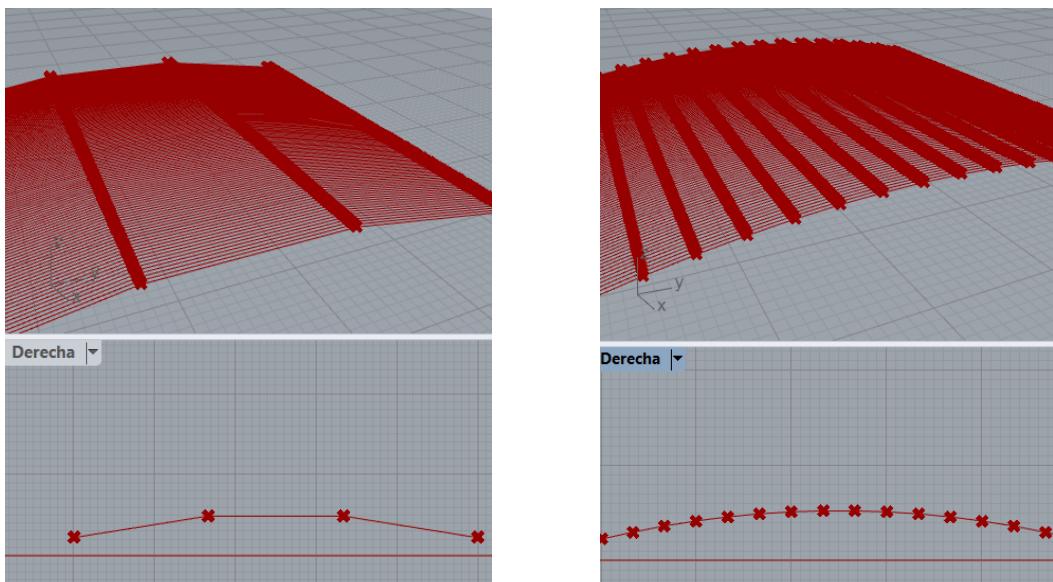


Fig. A.II-3. Discretización de la curva real en segmentos rectos para la obtención de la polilínea, que es la trayectoria que seguirá la boquilla. Izquierda: El ajuste mediante una polilínea de pocos segmentos, en el ejemplo de tres, crea un ajuste deficiente. Derecha: El ajuste mejora incrementando el número de segmentos que crean la polilínea; en el ejemplo 15 segmentos.

Calcular el número exacto de segmentos que forman la polilínea, de tal forma que exista un equilibrio óptimo entre el ajuste a la curva real, lo que conllevará una mejora en la calidad superficial, y un número excesivo de segmentos que no mejoran ésta, pero sí incrementan el peso de cálculo y del archivo, queda fuera del presente trabajo. No obstante, en esta segunda pieza, se ha discretizado la curva de tal forma que cada segmento tenga una longitud menor que el ancho de extrusión, que es 0.4 mm.

Con esto presente, en la variable *Count* que introduciremos como *input* del comando *Divide* seleccionamos un valor de 250 (Este proceso es análogo al que podemos ver en la Fig. A. I-16 referente a la cuña).

Todas las demás partes del algoritmo, así como el procedimiento llevado a cabo para obtener el programa *.gcode que nos llevará a la impresión de la piel o pieles en sus diferentes orientaciones, es idéntico al del caso anterior con la cuña.

El algoritmo completo en Grasshopper® para el diseño e impresión de la cuña se puede visualizar en el Anexo II- A. (A.II-A: Algoritmo Grasshopper® del arco extruido).

Impresión de la pieza

En la tabla A.II-1 se muestra el valor asignado a cada uno de los parámetros necesarios para la impresión 3D de la pieza tratada en este anexo. Evidentemente, estos parámetros hacen referencia a los utilizados en el algoritmo Grasshopper® para el diseño e impresión de esta pieza desde un punto de vista no planar, dejando de lado otros parámetros que podemos considerar generales en una impresión 3D tradicional.

ARCO EXTRUIDO	
<i>Parámetros generales importantes</i>	
Temperatura BASE [°C]	210
Temperatura PIEL [°C]	210
Velocidad impresión (pieles) [mm/s]	20
Multiplicador de extrusión	1
Nozzle de serie con Witbox 1 [mm]	0.4

ESTRUCTURA						
	Coordenadas Grasshopper®	Orientación (Contour)	Distancia (Distance)	Segmentos curva (Divide)	Altura capa (Height layer)	Ancho extrusión (Width extrusion)
BASE	123.5, 80, 2.2 [*]	-	-	-	0.2	-
1ª PIEL	12.5, 80, 2.2	90 ° [1,0,0]	0.4	250	0.2	0.42
2ª PIEL	12.5, 80, 2.4	0 ° [0,1,0]	0.4	1	0.2	0.42

[*] Según el algoritmo desarrollado, la última capa de la base (superficie de ésta) se situará 0.2 mm por debajo de la primera piel.

Tabla A.II-1. Parámetros de impresión.

Por otra parte, es de gran importancia la orientación que le hemos dado a las pieles. Al igual que en la cuña, la pieza anterior, al definir la orientación de dichas pieles se ha pretendido que la última piel tuviera la misma orientación que la superficie de la pieza impresa mediante un método FDM clásico. De tal forma, sí que podemos hacer una comparación real entre ambas piezas a la hora de evaluar la calidad superficial. El proceso de impresión, así como la orientación de cada una de las pieles se puede observar en la siguiente figura (Fig. A. II-4).

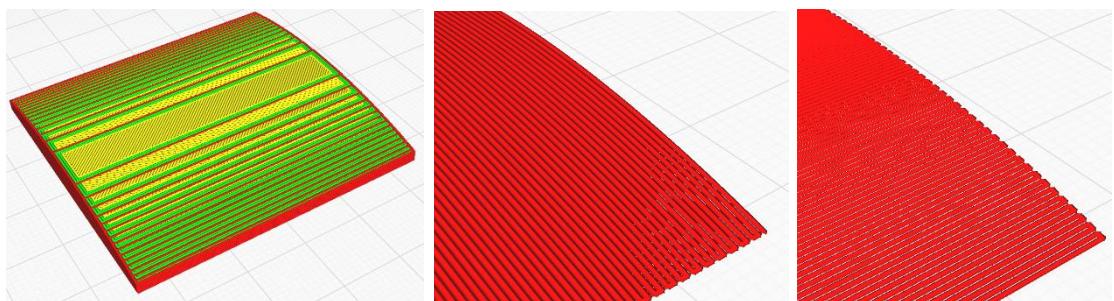


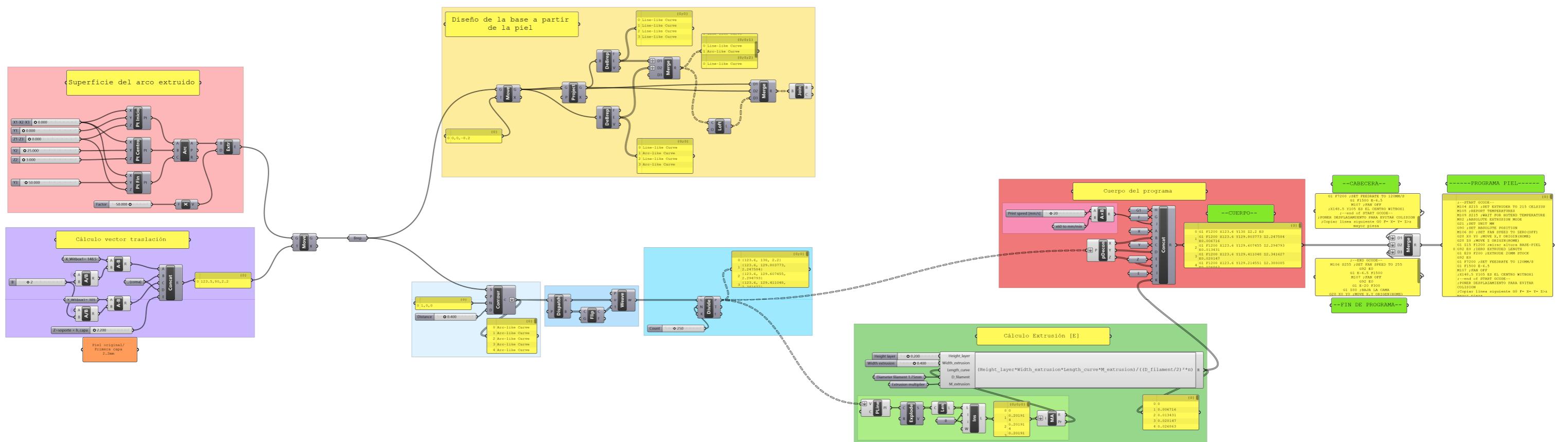
Fig. A.II-4. (Izquierda) Disposición de las capas en la base. Hay que tener en cuenta que la base, además de ser el soporte para las capas no planas, es también idéntica a la pieza que servirá de comparación superficial entre las dos filosofías de impresión tratadas en este trabajo, plana-no plana. (Centro) La primera piel, con una orientación a 90° , amortiguará el escalonamiento. De tal forma, la segunda piel, que se apoyará sobre ésta, tendrá una mejor calidad superficial. (Derecha) La segunda, y última piel, tendrá una orientación a 0° , semejante a la de la base. Así, la comparación de la calidad superficial entre ambas piezas será real.

Por último, se muestran dos fotografías correspondientes a las piezas finales, una vez impresas, utilizando los dos métodos (Fig. A. II-5).

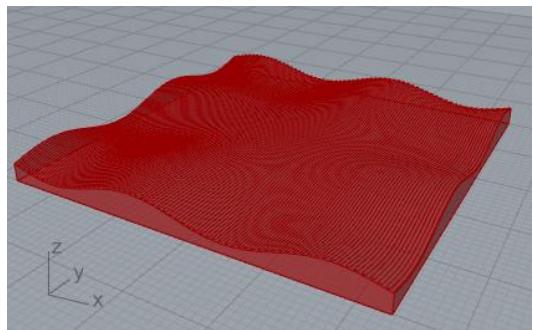


Fig. A.II-5. (Izquierda) Pieza impresa por el método FDM tradicional. (Derecha) La misma pieza impresa por un método FDM no-plano en sus últimas capas.

A.II-A: Algoritmo Grasshopper del arco extruido



A.III- Impresión de una superficie compleja a partir de sus ecuaciones paramétricas



Tras el diseño e impresión de la cuña y el arco extruido de pequeña curvatura, la tercera pieza que se propone resulta un salto de complejidad en el diseño de la superficie, además de en la impresión de ésta, al existir un continuo cambio en el vector de dirección e inclinación de la trayectoria que seguirá la boquilla.

Se trata de la definición de una superficie compleja a través de sus ecuaciones paramétricas. Dicha superficie, definida como “PIVA” en el *Proyecto Educativo Descartes* [27], posee una variable que hace modificar su geometría. La elección de dicha variable (parámetro) se ha tomado con el fin de aplanar la geometría y que la impresión de ésta sea posible, a la vez que permitiera destacar las ventajas de la impresión no plana en este tipo de figuras.

Excepto el diseño de la superficie, todo el proceso sigue la misma dinámica de piezas anteriores, haciendo uso del mismo algoritmo. Debido a este hecho, y con el fin de no repetir la explicación de los procesos, nos centraremos en este anexo en el estudio del diseño de la superficie.

Diseño de la superficie a partir de sus ecuaciones paramétricas

El diseño de la superficie se realiza a partir de dos aspectos fundamentales: las dimensiones de ésta en planta (largo y ancho) y las ecuaciones que la definen. Ambos datos son obtenidos del sitio web al cual se ha hecho referencia y son plasmados en el algoritmo mediante los componentes *Construct Domain* y *Expression*.

Para entender cómo funciona dicho algoritmo de creación de la superficie vamos a analizar la Fig. A. III-1. En primer lugar, será necesario definir un dominio ①, tanto en el eje X como en el eje Y. Esta acción requiere de la utilización del componente *Construct Domain* para cada uno de los citados ejes. En nuestro diseño, y tal y como indicaban en [27], vamos a utilizar un dominio que va desde -2π a 2π , en ambos ejes. Por lo tanto, la superficie en planta tendrá unas dimensiones de 4π por 4π .

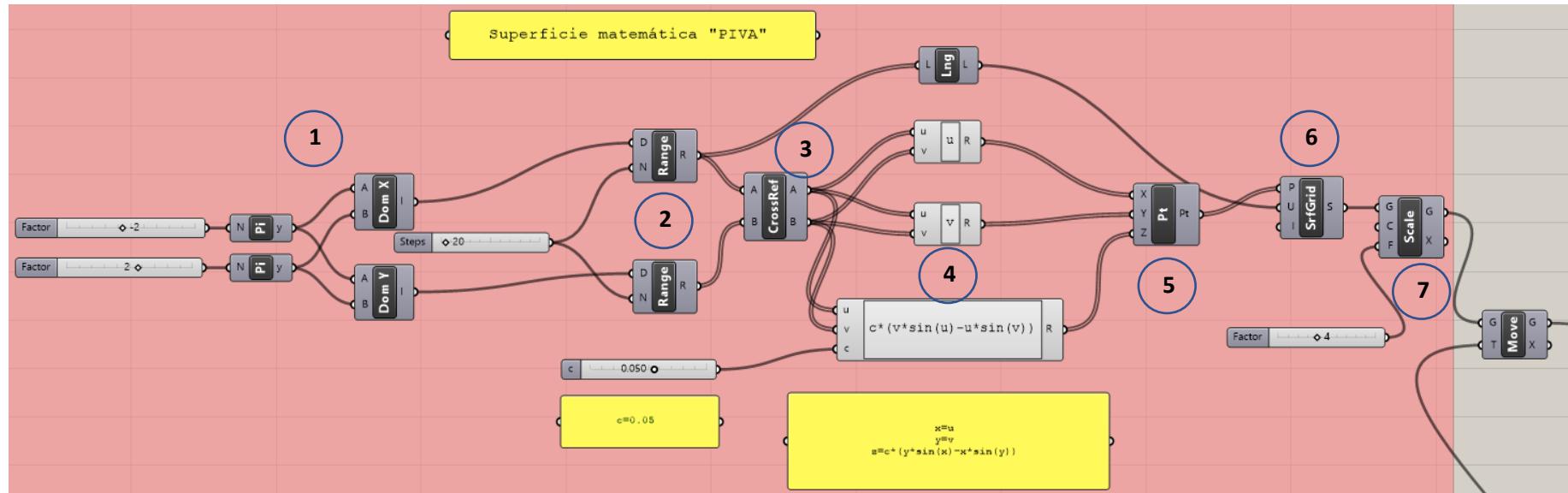


Fig. A.III-1. Algoritmo en Grasshopper® utilizado para el diseño de la superficie compleja a partir de sus ecuaciones paramétricas.

Establecido el dominio, el siguiente paso consiste en subdividir a éste en una serie de divisiones por medio del componente rango (*Range*) ②. Así, podemos suponer que nuestros dos ejes, X e Y, han sido divididos. En este diseño, tomamos un único valor de rango (*step*) para ambos ejes. Cuanto mayor es el valor del *step* mayor es el coste computacional.

Con estas dos operaciones, construcción del dominio y rango, hemos plantado la semilla para la definición de un mallado, por así decirlo ficticio, que crearemos a través del componente *Cross Reference* ③.

A partir de aquí, podemos imaginar una malla de largo y ancho el dominio, y con tantas subdivisiones como las establecidas en el rango. La característica principal de esta malla imaginaria es que es plana, de tal forma que lo que haremos en el siguiente paso es darle forma, “levantarla”, en sus puntos de unión, donde se cruzan las líneas que dividen el eje X e Y.

La materialización de dichos puntos se hará con el comando *Construct Point* ⑤, pero sin duda lo más interesante es que las coordenadas de dichos puntos están definidas mediante ecuaciones paramétricas ④. En el ejemplo que nos ocupa, las coordenadas X e Y no sufren variación respecto al valor del punto de unión del mallado que ya habíamos establecido, pero el valor Z sí que sufre dicha variación. Dicho valor, Z, será definido a partir de una ecuación paramétrica que depende de u (eje X), v (eje Y) y de una variable c .

Lo que tenemos ahora, tal y como podemos observar en la Fig. A. III-2, es un conjunto de puntos cuya disposición en el espacio tridimensional es función de tres ecuaciones paramétricas, una por cada eje.

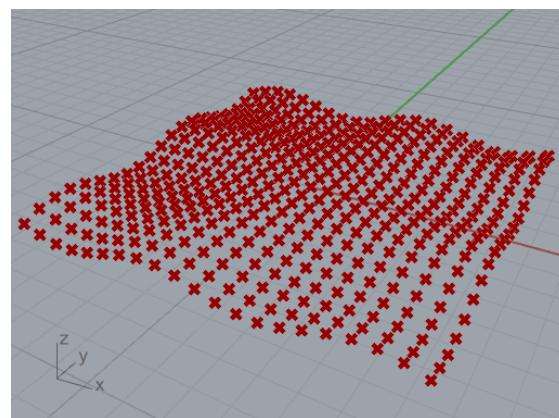


Fig. A.III-2. Nube de puntos definida a través de ecuaciones paramétricas.

A continuación, y a partir de los puntos creados, definiremos una superficie geométrica. El componente *Surface From Points (SrfGrid)* ⑥ crea una superficie a partir de una nube de puntos (Fig. A. III-3).

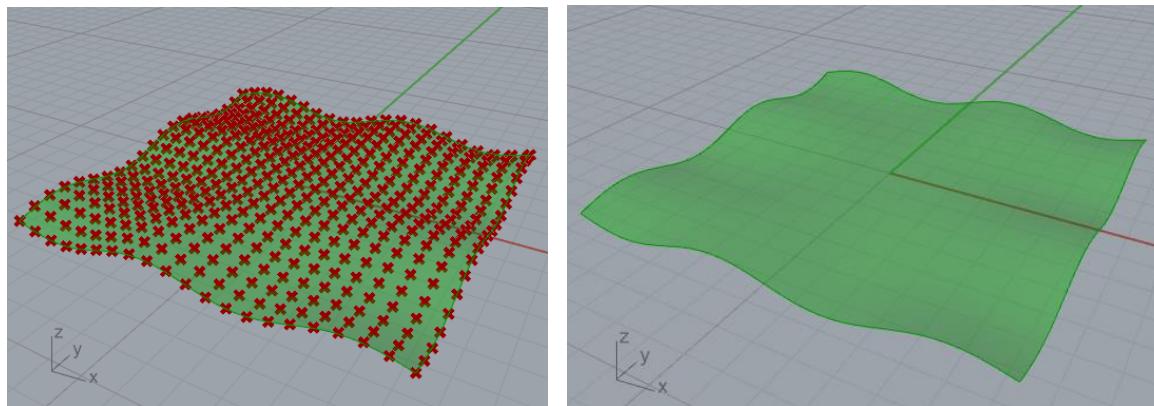


Fig. A.III-3. (Izquierda) Definición de la superficie a partir de la nube de puntos. (Derecha) Superficie una vez ocultos los puntos.

Por último, y dado que consideramos que la superficie resultante es pequeña para la impresión, vamos a escalar la misma (*Scale*) ⑦. Este último paso, se podría sustituir por una ampliación del dominio, pero hemos considerado que esta forma de proceder era más sencilla, visualmente hablando.

Además, el valor de la variable c , anteriormente citada, fue establecido en 0.05 para conseguir un alisado de la superficie que resaltaría las ventajas de la impresión no plana. Recordemos que en curvas con mucha inclinación esta ventaja, sin desaparecer, toma menos protagonismo.

El algoritmo completo en Grasshopper® para el diseño e impresión de la superficie compleja a partir de sus ecuaciones paramétricas se puede visualizar en el Anexo III- A. (A.III-A: Algoritmo Grasshopper® superficie compleja).

Impresión de la pieza

En la tabla A.III-1 se muestra el valor asignado a cada uno de los parámetros necesarios para la impresión 3D de la pieza tratada en este anexo. Evidentemente, estos parámetros hacen referencia a los utilizados en el algoritmo Grasshopper® para el diseño e impresión de esta pieza desde un punto de vista no planar, dejando de lado otros parámetros que podemos considerar generales en una impresión 3D tradicional.

SUPERFICIE COMPLEJA “PIVA”	
<i>Parámetros generales importantes</i>	
Temperatura BASE [°C]	210
Temperatura PIEL [°C]	210
Velocidad impresión (pieles) [mm/s]	20
Multiplicador de extrusión	1
Nozzle de serie con Witbox 1 [mm]	0.4

ESTRUCTURA						
	Coordenadas Grasshopper®	Orientación (Contour)	Distancia (Distance)	Segmentos curva (Divide)	Altura capa (Height layer)	Ancho extrusión (Width extrusion)
BASE	148.5, 105, 2.2 [*]	-	-	-	0.2	-
1ª PIEL	148.5, 105, 2.3 [**]	0 ° [0,1,0]	0.4	200	0.2	0.42
2ª PIEL	148.5, 105, 2.6 [**]	90 ° [1,0,0]	0.4	200	0.2	0.42

[*] Según el algoritmo desarrollado, y con los datos expuestos, la última capa de la base (superficie de ésta) se situará 0.3 mm por debajo de la primera piel.

[**] Tal y como indica Jin et al. [21], la altura entre capas sucesivas en una impresión no plana va a depender del sentido de la trayectoria de impresión, así como de la inclinación de la superficie, según unas fórmulas matemáticas establecidas. En este sentido se ha utilizado una altura en las pieles de 0.3 mm, aunque el cálculo de E se haya realizado con 0.2 mm.

Tabla A.III-1. Parámetros de impresión.

Al igual que en los dos casos anteriores la orientación de las pieles es importante. Sin embargo, dada la geometría compleja de esta superficie “PIVA”, es imposible establecer una correspondencia directa y precisa entre la orientación de las fibras en la pieza “plana” y la pieza no-plana. Esto implica, como se ve en el capítulo 5: “Evaluación de la calidad superficial”, que en estos casos no se debe medir la rugosidad en una línea sino en un área.

En la siguiente figura (Fig. A. III-4) se puede observar el proceso de impresión, así como la orientación de cada una de las pieles.

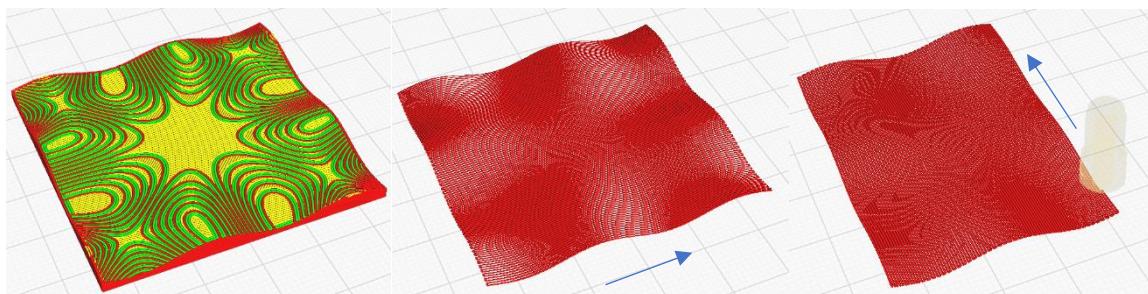


Fig. A.III-4. (Izquierda) Disposición de las capas en la base. Hay que tener en cuenta que la base, además de ser el soporte para las capas no planas, es también idéntica a la pieza que servirá de comparación superficial entre las dos filosofías de impresión tratadas en este trabajo, plana-no plana. (Centro) La primera piel, con una orientación a 0° , amortiguará el escalonamiento. De tal forma, la segunda piel, que se apoyará sobre ésta, tendrá una mejor calidad superficial. (Derecha) Proceso de impresión en la segunda piel. Esta tendrá una orientación a 90° .

Por último, se muestran dos fotografías correspondientes a las piezas finales, una vez impresas, utilizando los dos métodos (Fig. A. III-5).

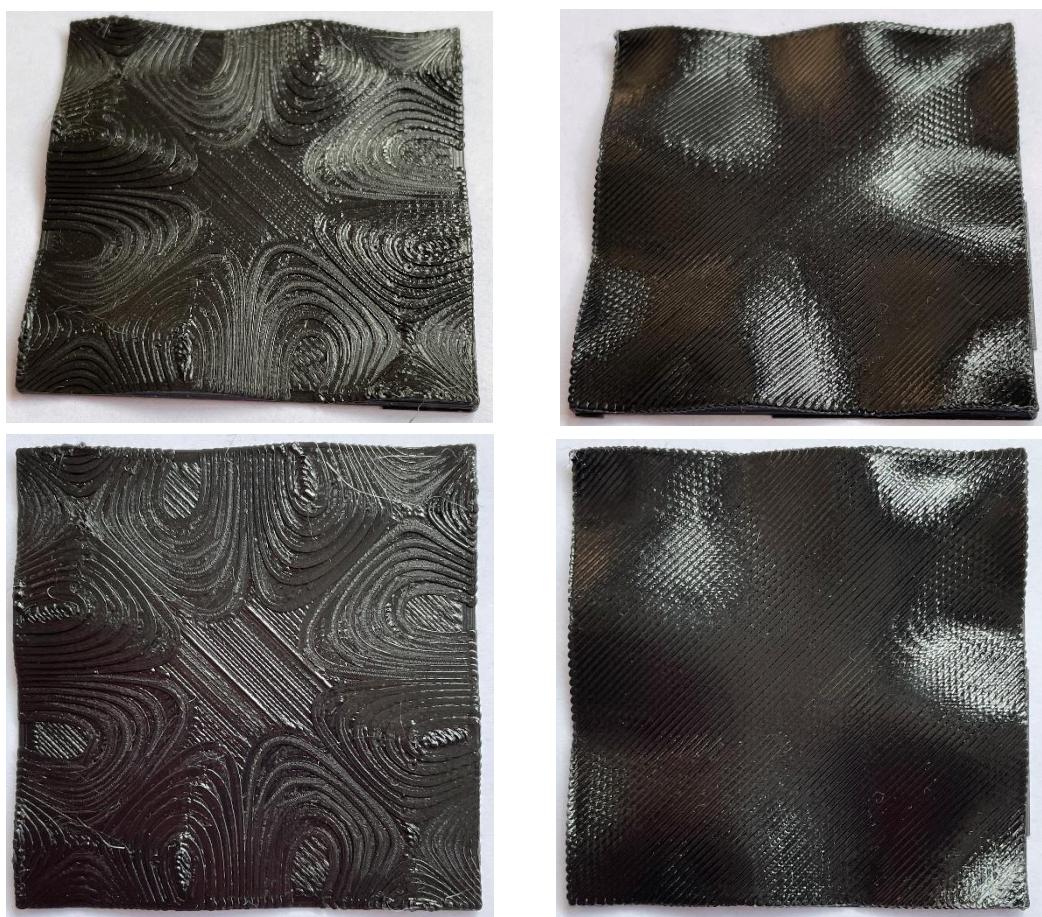
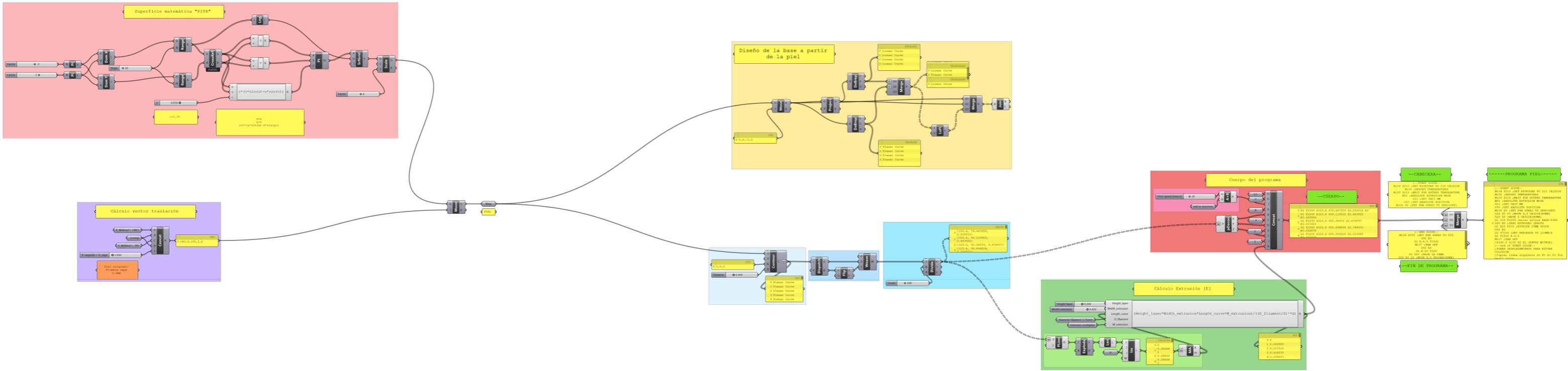
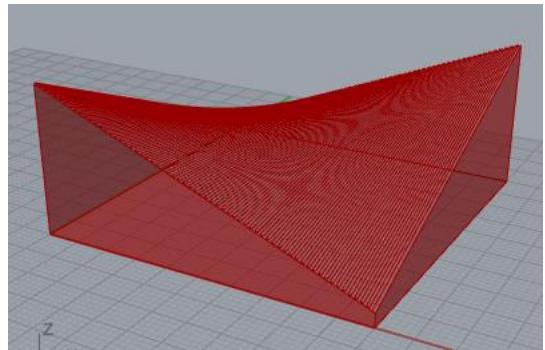


Fig. A.III-5. (Izquierda) Pieza impresa por el método FDM tradicional. (Derecha) La misma pieza impresa por un método FDM no-planar en sus últimas capas.

A.III-A: Algoritmo Grasshopper superficie compleja



A.IV- Impresión de un paraboloide hiperbólico



En la cuarta, y última pieza, se propone el diseño e impresión de una superficie de la forma paraboloide hiperbólico, también llamada “silla de montar”. Se ha querido continuar en la línea ascendente de dificultad que se ha seguido durante todo el trabajo, y la impresión de esta última pieza implica un nivel de dificultad extra.

Detengámonos a explicar las peculiaridades de esta superficie. Un paraboloide hiperbólico es una superficie tridimensional infinita cuyas secciones con un plano horizontal y vertical dan como resultado hipérbolas y parábolas, respectivamente. Por otra parte, se trata de una superficie reglada, lo que implica que aun siendo una superficie curva se puede construir con líneas rectas.

En cuanto a la parte física y de resistencia de materiales, estamos hablando de la estructura bidimensional que mejor resiste los esfuerzos de presión-tensión. Este hecho lo sabían arquitectos como Gaudí y Félix Candela, y por qué no, el inventor de las patatas Pringles (Fig. A. IV-1).

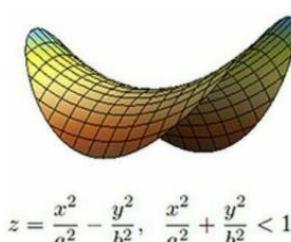


Fig. A.IV-1. Paraboloide hiperbólico. Izquierda: Arquitectos como Félix Candela conocían las ventajas resistivas de este tipo de superficies. La prueba está en que estructuras de Candela han sobrevivido a devastadores terremotos. Centro-Derecha: Los sistemas físicos siempre tienden a su estado de mínima energía, de ahí que la forma de paraboloide hiperbólico minimice la deformación de la patata cuando, como consecuencia de los cambios de temperatura en el aceite caliente, sufre esfuerzos de presión-tensión. Por si fuera poco, también resistirá los esfuerzos durante el transporte y es apilable.

Diseño de la superficie compleja

Como ya hemos adelantado al inicio de este punto, la geometría de la superficie de la piel va a consistir en un paraboloide hiperbólico. Aunque matemáticamente se trata de una superficie infinita, nosotros, evidentemente, le vamos a dar forma, limitándola a una superficie cuadrada en su planta.

De tal forma, en su construcción podemos imaginar una superficie plana, en este caso cuadrada, definida por cuatro puntos (sus vértices) y delimitada por cuatro aristas trazadas a través de dichos puntos. Una vez esto, “elevamos” dos de los cuatro puntos que estén en la misma diagonal. El resultado es la superficie que buscamos.

Para su diseño en Grasshopper®, tal y como ilustra la Fig. A. IV-2, definimos cuatro puntos de tal forma que sus coordenadas en X e Y formen un cuadrado, en nuestra pieza de 50 mm de lado, a la vez que la coordenada en Z sea idéntica en dos puntos situados en la misma diagonal, pero distinta a los otros dos. (En el algoritmo no se define una relación entre los puntos, de tal forma, se tiene que parametrizar cada punto con sus tres coordenadas. Este hecho da flexibilidad al diseño, permitiendo variaciones en la geometría de la superficie en futuros proyectos más allá del trabajo actual).

En el presente trabajo, estableceremos que la coordenada en Z del punto 2 (Pt 2) y punto 3 (Pt 3) estén en Z0, dándole al punto 1 (Pt 1) y punto 4 (Pt 4) una altura de 10 mm. Esta diferencia de alturas de 10 mm es importante, según lo planteado en el modelo de colisión, y aunque no permite la impresión no planar con el *nozzle* que viene de serie en la Witbox 1, el mismo que ha sido utilizado para la impresión de las tres primeras piezas, sí la va a permitir con el *nozzle* de *nonplanar.xyz*. De ahí, la importancia de esta pieza en la didáctica del proyecto en su conjunto.

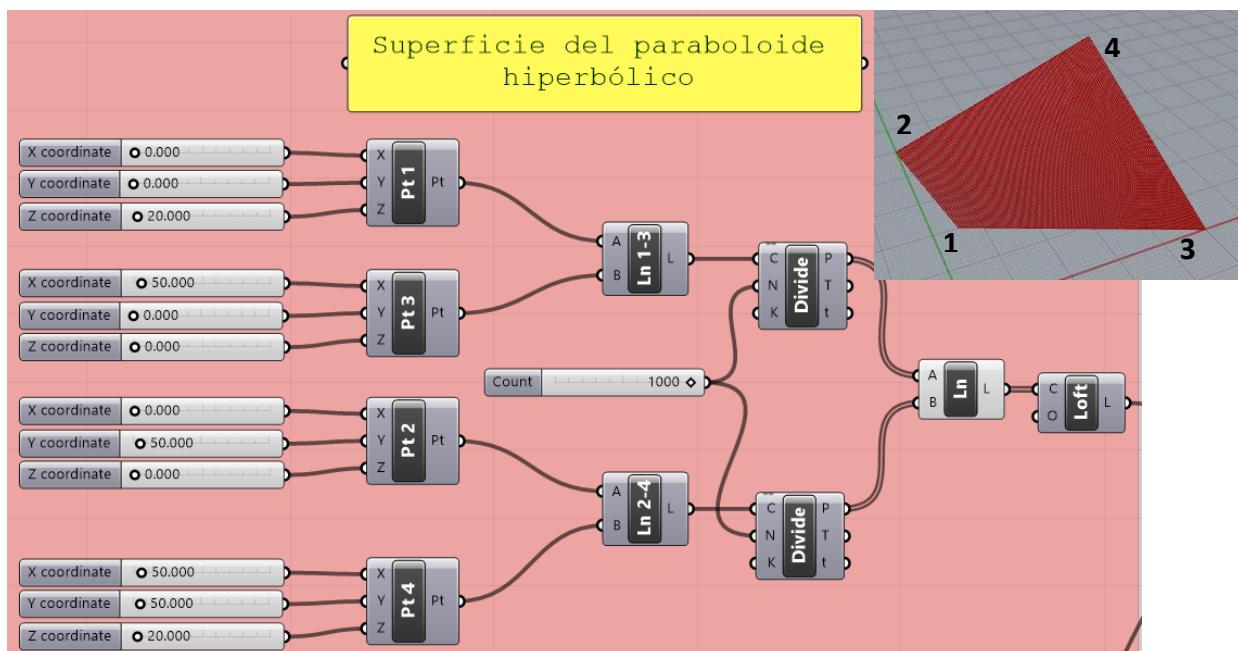


Fig. A.IV-2. Algoritmo en Grasshopper® para el diseño de la superficie del paraboloide hiperbólico.

Tras la creación de estos cuatro puntos trazamos dos líneas rectas paralelas entre sí a partir de estos puntos (Ln 1-3 y Ln 2-4). A continuación, dividimos estas líneas (*Divide*) en una serie de puntos (*count*) que tiene que ser el mismo en ambas líneas. Con estos puntos, creamos nuevas líneas, uniendo el primer punto de la Ln1-3 con el primer punto de la Ln 2-4, el segundo punto de la Ln 1-3 con el segundo punto de la Ln 2-4, y así sucesivamente (Fig. A. IV-3, Izq). El resultado obtenido es una serie de líneas, contenidas en planos paralelos entre sí, y de número (*count*) igual al utilizado para dividir las dos líneas. Por último, el componente *Loft* creará una superficie suavizada a partir del conjunto de líneas, lo que implica que la superficie será más “suave” cuantas más líneas tenga dicho conjunto (Fig. A. IV-3, Dcha).

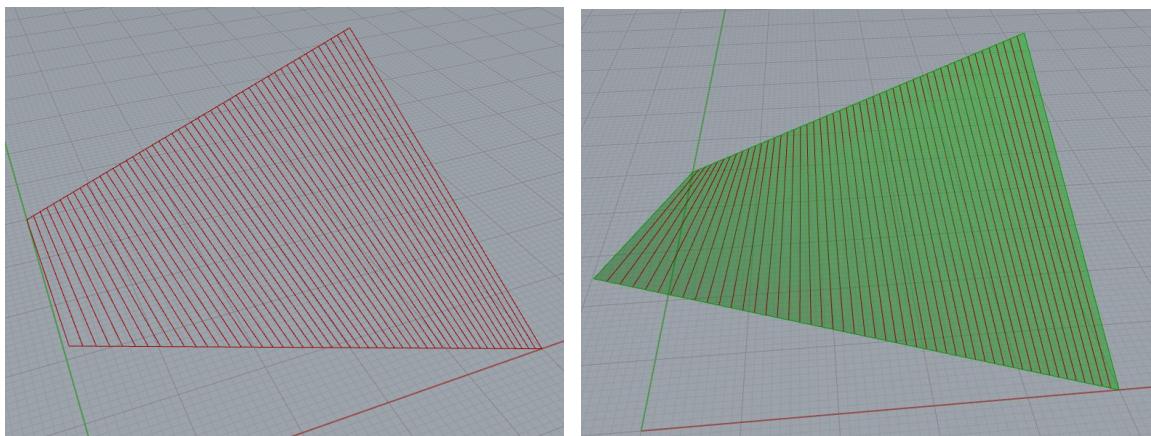


Fig. A.IV-3. Izquierda: Se crea un conjunto de líneas rectas a partir de la división de Ln1-3 y Ln2-4 en una serie de puntos. Derecha: El componente *Loft* crea una superficie suavizada a partir de un conjunto de líneas, en este caso rectas. (En estas dos figuras, para una mejor visualización del concepto, se ha tomado un conjunto de 50 líneas. En el diseño real se tomará un conjunto de 1000 líneas).

Creación de las trayectorias según su orientación

A lo largo del algoritmo que nos lleva a la obtención del programa final para la impresión de la piel, o pieles, es necesario destacar un aspecto relacionado con la creación de las trayectorias, más concretamente con la creación de la polilínea que las definirá. Recordemos del punto anterior que, obtenidas unas líneas, ya sean éstas rectas o curvas, nuestro objetivo era crear una polilínea lo más óptimamente ajustada a dichas líneas, a través de la discretización de éstas.

Ahora bien, como ya hemos introducido, y lo hemos podido comprobar al diseñar la superficie del paraboloide hiperbólico, dicha superficie es reglada, teóricamente es doblemente reglada, es decir, que se trata de una superficie que aun siendo curva se puede construir con líneas rectas. Esto implica que cuando queramos orientar las fibras a lo largo del eje X (Orientación 0,1,0 [0°]) o a lo largo del eje Y (Orientación 1,0,0 [90°]) será suficiente con discretizar la línea con un solo segmento, ya que se trata de una línea recta. Por lo tanto, en la entrada *count* del componente *Divide* pondremos 1.

En el caso de que la orientación del filamento sea a 45° (Orientación $-1,1,0$), ya no tenemos líneas rectas sino curvas y no se puede discretizar la curva con un solo segmento, ya que el ajuste va a ser muy deficiente. Aquí entramos nuevamente en la idea de que con cuantos más segmentos discreticemos la curva mayor será el ajuste entre la polilínea y la curva real, pero un número elevado de segmentos incrementará el costo computacional sin beneficio alguno. Por lo tanto, en este caso en concreto, y para nuestra tercera pieza, discretizaremos las curvas con 500 segmentos para formar la polilínea, dejando de lado la optimización del número de segmentos, que queda fuera del alcance del proyecto.

En la siguiente figura (Fig. A. IV-4) podemos visualizar lo analizado en este apartado.

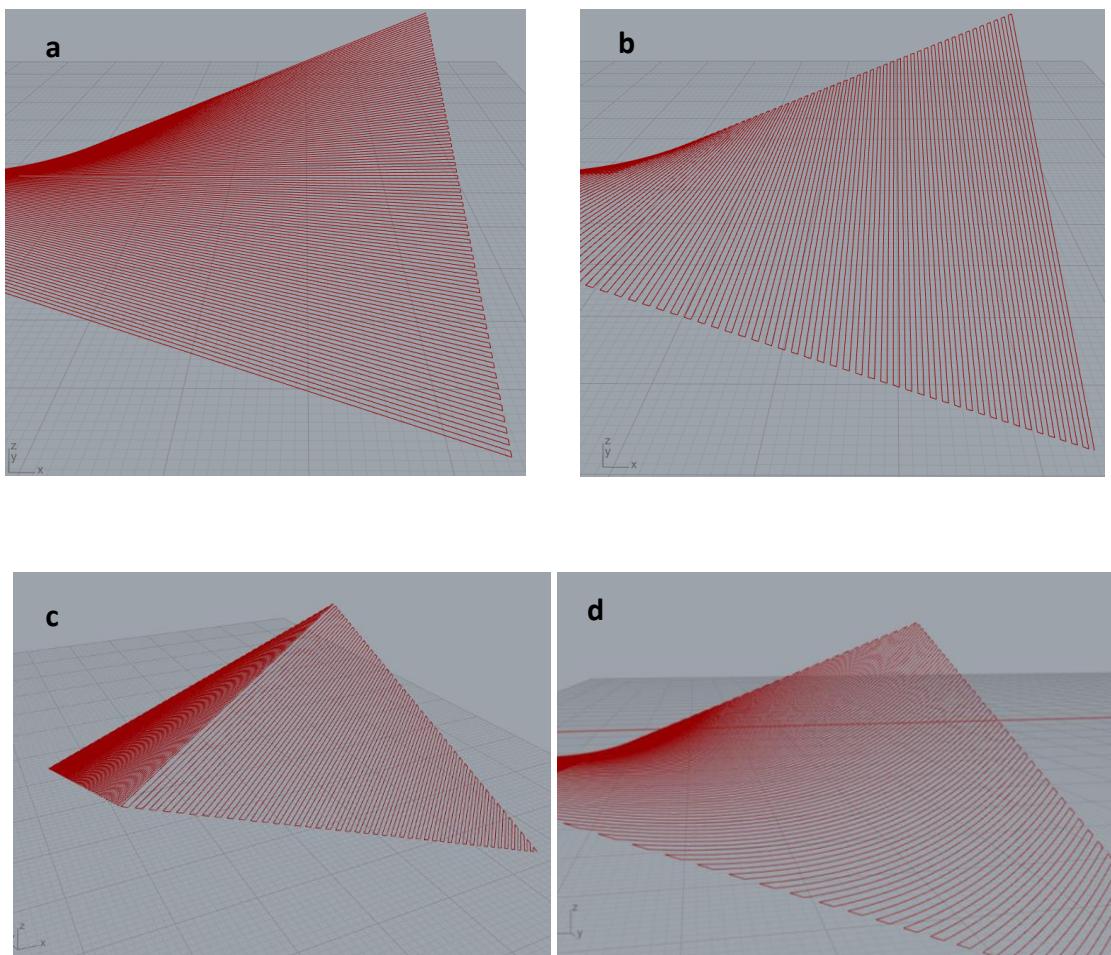


Fig. A.IV-4. (a) Orientación $0,1,0$ [0°], discretización correcta con un solo segmento. (b) Orientación $1,0,0$ [90°], discretización correcta con un solo segmento. (c) Orientación $-1,1,0$ [45°], la discretización no es correcta con un solo segmento ya que las líneas a discretizar son curvas. (d) La misma orientación que en el caso (c) pero la discretización se ha hecho con 500 segmentos. De tal forma, la polilínea obtenida se ajusta a la curva real.

Con excepción de lo ya analizado, todas las demás partes del algoritmo son semejantes a las utilizadas en la impresión de las anteriores piezas. Por lo tanto, dado que el algoritmo

visual está automatizado, si tenemos en cuenta los cambios que son necesarios realizar, obtendremos los programas finales para la impresión de la base y de la piel.

El algoritmo completo en Grasshopper® para el diseño e impresión del paraboloide hiperbólico se puede visualizar en el Anexo IV- A. (A.IV-A: Algoritmo Grasshopper® del paraboloide hiperbólico).

Impresión de la pieza

En la tabla A.IV-1 se muestra el valor asignado a cada uno de los parámetros necesarios para la impresión 3D de la pieza tratada en este anexo. Evidentemente, estos parámetros hacen referencia a los utilizados en el algoritmo Grasshopper® para el diseño e impresión de esta pieza desde un punto de vista no planar, dejando de lado otros parámetros que podemos considerar generales en una impresión 3D tradicional.

PARABOLOIDE HIPERBÓLICO	
<i>Parámetros generales importantes</i>	
Temperatura BASE [°C]	210
Temperatura PIEL [°C]	210
Velocidad impresión (pieles) [mm/s]	20
Multiplicador de extrusión	1
Nozzle largo <i>nonplanar.xyz</i>	0.6

ESTRUCTURA						
	Coordenadas Grasshopper®	Orientación (Contour)	Distancia (Distance)	Segmentos curva (Divide)	Altura capa (Height layer)	Ancho extrusión (Width extrusion)
BASE	123.5, 80, 2.2 [*]	-	-	-	0.2	-
1ª PIEL	123.5, 80, 2.3 [**]	0 ° [0,1,0]	0.55	1	0.2	0.65
2ª PIEL	123.5, 80, 2.6 [**]	45° [-1,1,0]	0.55	250	0.2	0.65

[*] Según el algoritmo desarrollado, y con los datos expuestos, la última capa de la base (superficie de ésta) se situará 0.3 mm por debajo de la primera piel.

[**] Tal y como indica Jin et al. [21], la altura entre capas sucesivas en una impresión no plana va a depender del sentido de la trayectoria de impresión, así como de la inclinación de la superficie, según unas fórmulas matemáticas establecidas. En este sentido se ha utilizado una altura en las pieles de 0.3 mm, aunque el cálculo de E se haya realizado con 0.2 mm.

Tabla A.IV-1. Parámetros de impresión.

Dada la geometría compleja de esta superficie “paraboloide hiperbólico”, es imposible establecer una correspondencia directa y precisa entre la orientación de las fibras en la pieza “plana” y la pieza no-plana. Esto implica, como se ve en el capítulo 5: “Evaluación de la calidad superficial”, que en estos casos no se debe medir la rugosidad en una línea sino en un área.

A la hora de llevar la impresión de este diseño a la práctica se probó, en primer lugar, con una altura máxima de 20 mm. Aunque, tal y como se observa en la Fig. A. IV-5.Izquierda, el resultado fue satisfactorio, se pensó en disminuir dicha altura en 10 mm (Fig. A. IV-5.Centro) con el fin de incrementar la visibilidad de los efectos positivos de la impresión no-plana.

Por otra parte, la superficie a verificar con el microscopio óptico es muy inferior a la pieza impresa. Por este motivo, y dada la simetría de la pieza, se decidió imprimir la mitad de la pieza. Este hecho, reducía los tiempos de impresión sin influir en los resultados de la medición de la superficie (Fig. A. IV-5.Derecha).

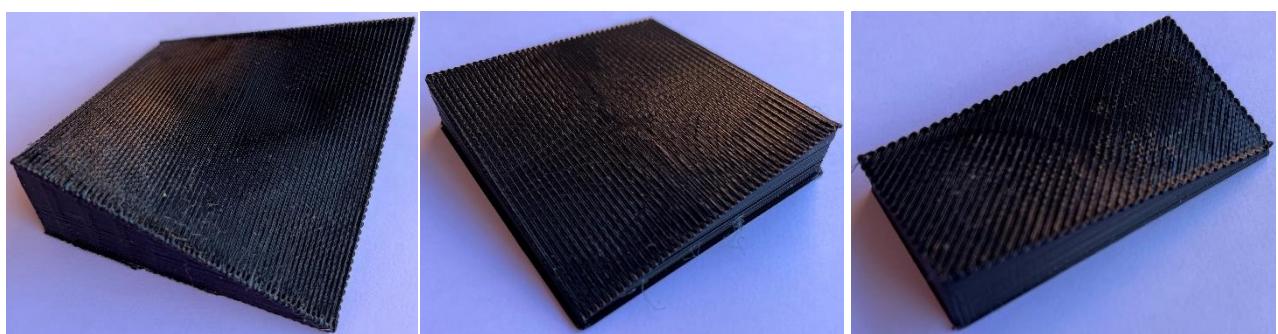


Fig. A.IV-5. (Izquierda) Paraboloide hiperbólico de 20 mm de altura máxima. (Centro) Mismo diseño con reducción de altura en 10 mm. (Derecha) Pieza utilizada en la medición de la calidad superficial. La reducción de las dimensiones de esta supone una mejora en los tiempos de impresión sin influir en los resultados de la medición.

En la siguiente figura (Fig. A. IV-6) se puede observar el proceso de impresión, así como la orientación de cada una de las pieles, en la pieza que servirá de medición.

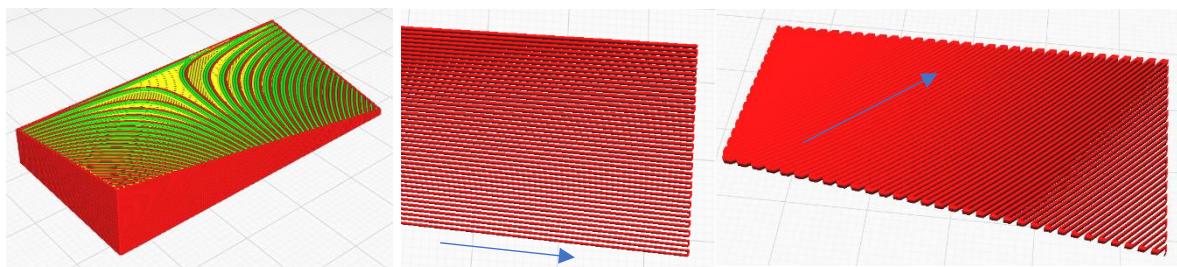


Fig. A.IV-6. (Izquierda) Disposición de las capas en la base. Hay que tener en cuenta que la base, además de ser el soporte para las capas no planas, es también idéntica a la pieza que servirá de comparación superficial entre las dos filosofías de impresión tratadas en este trabajo, plana-no plana. (Centro) La primera piel, con una orientación a 0°, amortiguará el escalonamiento. De tal forma, la segunda piel, que se apoyará sobre ésta, tendrá una mejor calidad superficial. (Derecha) Proceso de impresión en la segunda piel. Esta tendrá una orientación a 45°.

Por último, se muestran dos fotografías correspondientes a las piezas finales, una vez impresas, utilizando los dos métodos (Fig. A. IV-7).



Fig. A.IV-7. (Izquierda) Pieza impresa por el método FDM tradicional. (Derecha) La misma pieza impresa por un método FDM no-planar en sus últimas capas.

Experimentación con el *nozzle* de nonplanar.xyz

Uno de los objetivos buscados en este trabajo era la experimentación en la impresión no-plana con un *nozzle* que había sido desarrollado para este fin. Por este motivo, y tal y como hemos explicado en la memoria, se dotó a la impresora 3D de una boquilla de la casa *nonplanar.xyz* para la impresión del paraboloide hiperbólico.

Durante la experimentación con esta boquilla se ha observado que, si bien la geometría de ésta permite impresiones que con el *nozzle* de serie de la Witbox 1 no se podrían realizar, debido sobretodo a colisiones de las partes de la impresora o del mismo *nozzle* con la pieza, éste también presenta dificultades.

Dichas dificultades derivan principalmente de la resistencia al flujo del material plástico dentro de la boquilla, provocada por un gradiente de temperatura inadecuado debido principalmente a la longitud de la boquilla. Aunque ésta se cubre con una “manga aislante”, esta manga no es suficiente ya que el bloque calefactor encargado de fundir el plástico queda lejos de la punta de la boquilla.

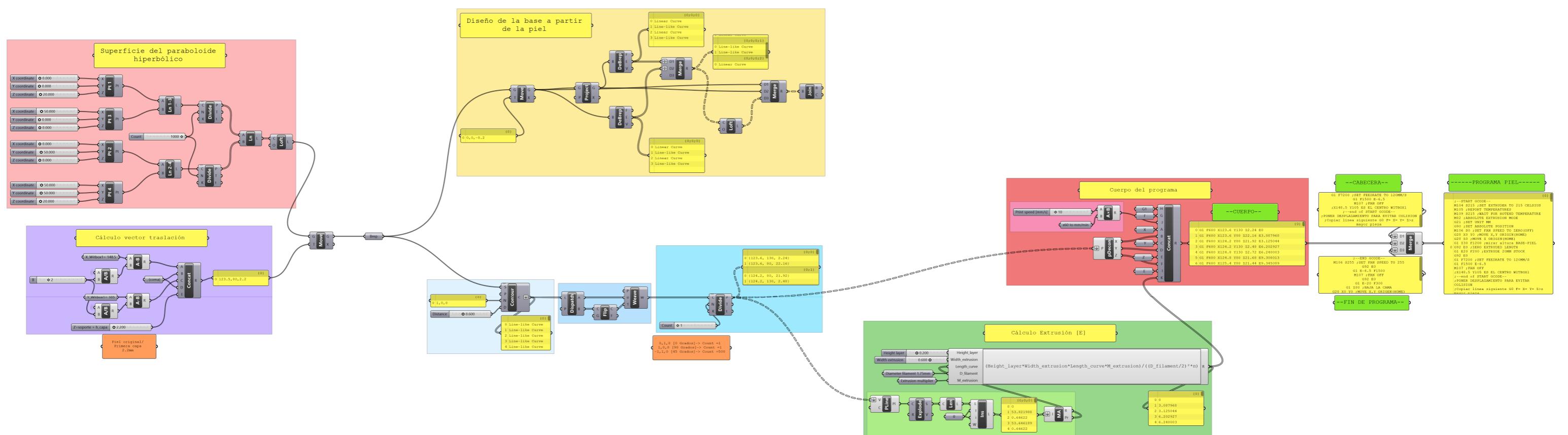
Por este motivo, se tuvo dificultad a la hora de imprimir con el *nozzle* de 0.4 mm, y fue sustituido por uno de 0.6 mm que ayudase en el flujo de material.

Por otra parte, es especialmente difícil regular el ventilador de capa. Una altura de este adecuada implica que este aireador se posicione cerca de la mesa de impresión, por lo cual, se convierte en una pieza que dificulta la impresión no-plana al colisionar durante el proceso de fabricación. En contra, si el aireador es corto, o no cumple su misión o provoca un flujo de aire contra la boquilla, lo que supone el corte instantáneo de la extrusión.

En cuanto a la impresión propiamente dicha, si bien la falta de flujo durante la fabricación de las pieles era subsanable, durante la impresión de la base los cortes en el flujo eran frecuentes, debido al incremento en el tiempo de fabricación que conllevaba esta respecto a las pieles. Esto implica que muchas piezas eran desechadas.

**Es necesario anotar que todas estas dificultades son las que hemos observado al imprimir con la Witbox 1. El análisis anterior no implica que este *nozzle* carezca de beneficios, o que no merezca la pena su compra, sino que con la máquina de trabajo utilizada y con los parámetros establecidos se han encontrado los citados inconvenientes.

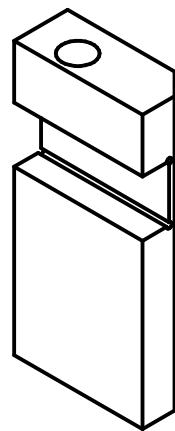
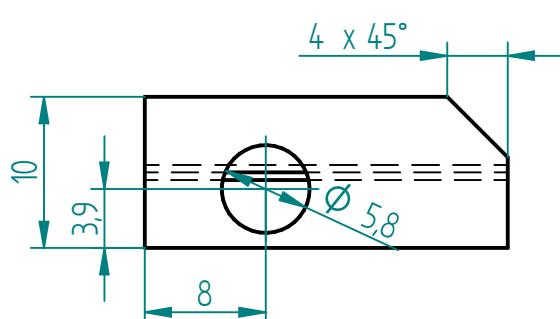
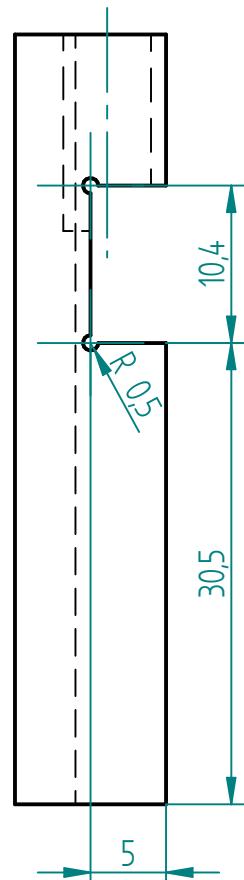
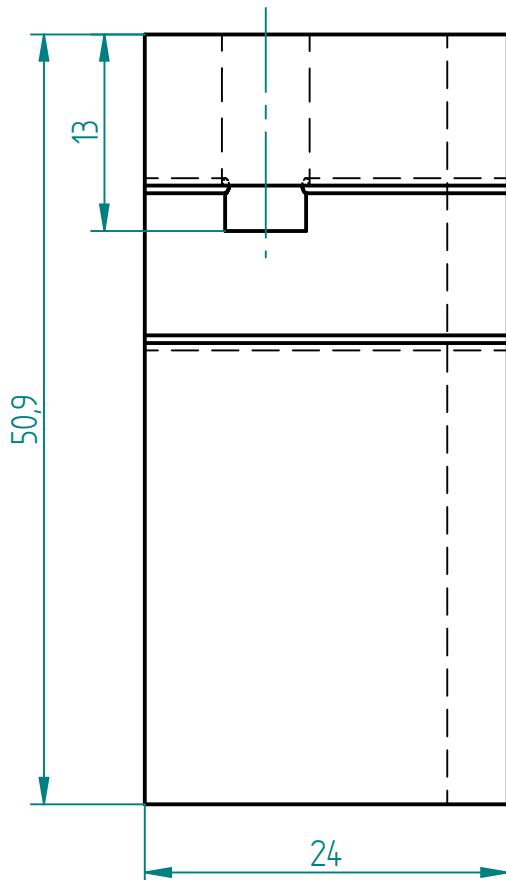
A.IV-A: Algoritmo Grasshopper del paraboloide hiperbólico



Pieza impresa en 3D. Material PLA 3D850.

En el agujero se roscará directamente el tornillo prisionero, haciendo este la rosca.

Para una mayor duración ver insertos roscados.



Dibujado	Fecha	Nombre				Universidad Zaragoza		
Comprobado	22/01/2021	A.Martínez						
Comprobado	03/02/2021	J.A.Albajez						
Escala								
2:1	EXTENSIÓN FINAL CARRERA Z - Witbox 1							
A4								
						Medidas sin tolerancias según DIN 7168 (precisión media)		
						Revisión: 01 / 01		

A.VI- Medición de la rugosidad

1. Rugosidad en la cuña

Pieza referencia/ Impresión FDM planar

Como el valor de la rugosidad Ra estimado es superior a 10 micrómetros, la norma ISO 4288 recomienda usar un valor de Lc de 8000 micrómetros. Dicho dato supone que como se tienen que medir cinco segmentos necesitamos escanear una longitud de pieza de al menos 40 mm.



Fig. A.VI-1. Zona escaneada de la cuña de referencia (FDM tradicional).

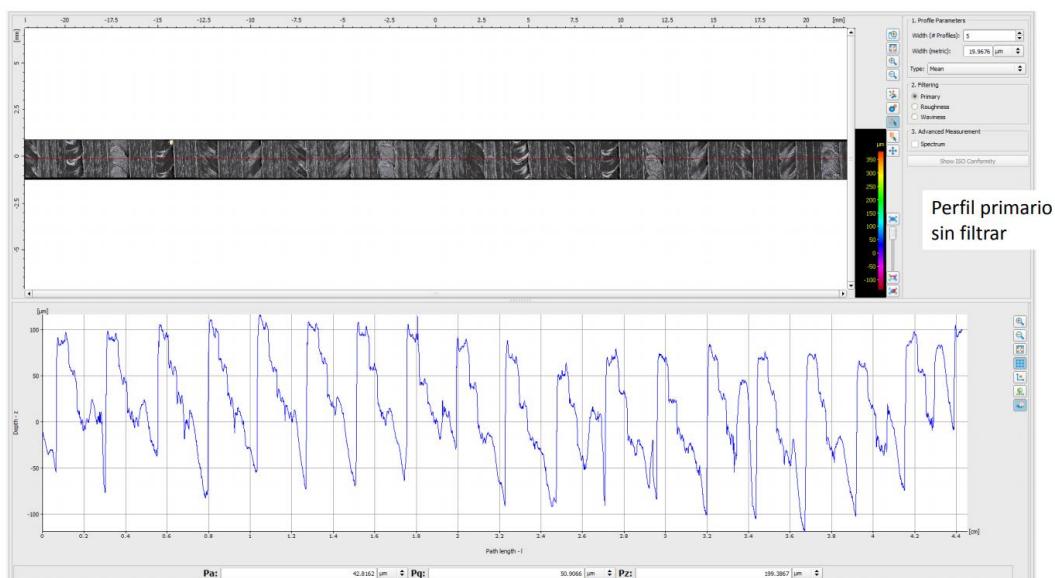


Fig. A.VI-2. Perfil primario sin filtrar.

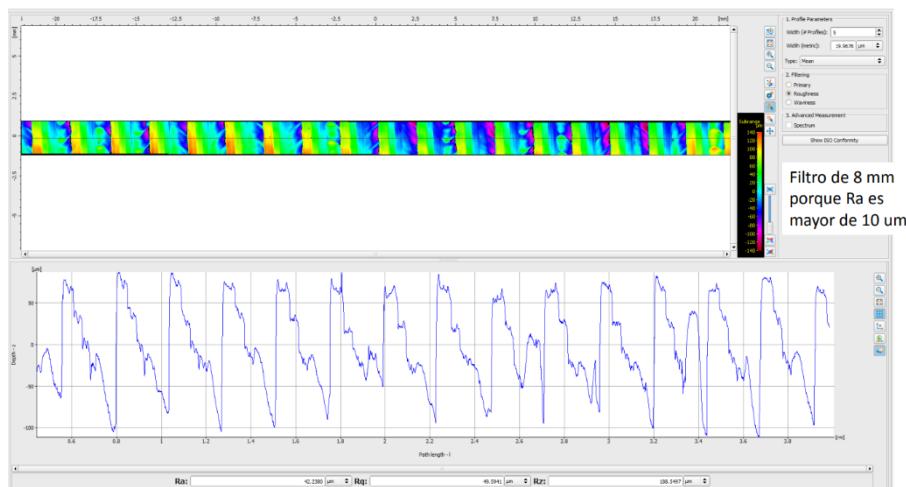


Fig. A.VI-3. Se utiliza un filtro de 8000 micrómetros dado que la Ra estimada es mayor a 10 μm .

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

CUÑA			
	Ra [μm]	Rq [μm]	Rz [μm]
FDM planar (Lc=8000 μm)	42.2	49.5	188.5

Pieza Impresión FDM no-planar

En este caso, como el valor de Ra estimado es inferior a 10 micrómetros, podemos usar una Lc de 2500 micrómetros. Por lo tanto, la longitud mínima a escanear es menor ($5*2.5=12.5$ mm), reduciéndose considerablemente el tiempo de escaneo. Aquí, hay que advertir que si analizáramos esta pieza con la Lc de 8000 μm nos saldrían prácticamente los mismos resultados.

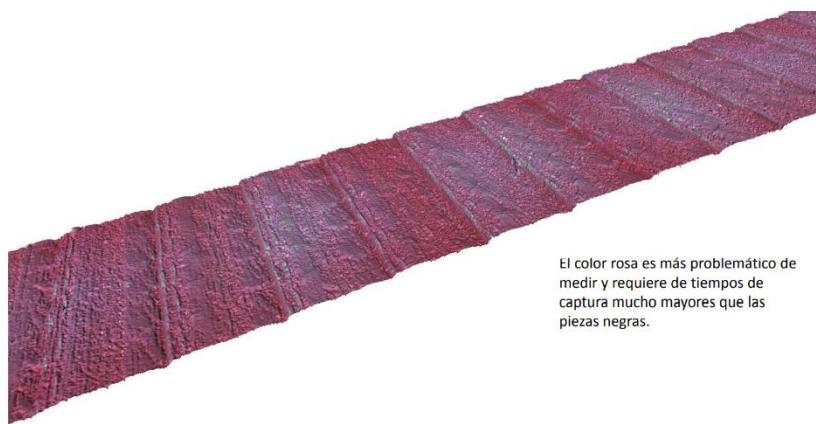


Fig. A.VI-4. Detalle de la zona escaneada. En un principio se pensó imprimir la piel con un color rosa, destacando de esta manera las dos fases que componen la pieza (la base/soporte y la piel). El problema de este tipo de colores es que son más problemáticos de medir y, por lo tanto, requieren de tiempos de captura mucho mayores. De tal forma, todas las impresiones fueron realizadas con un filamento PLA de color negro de la marca Smart Materials 3D.

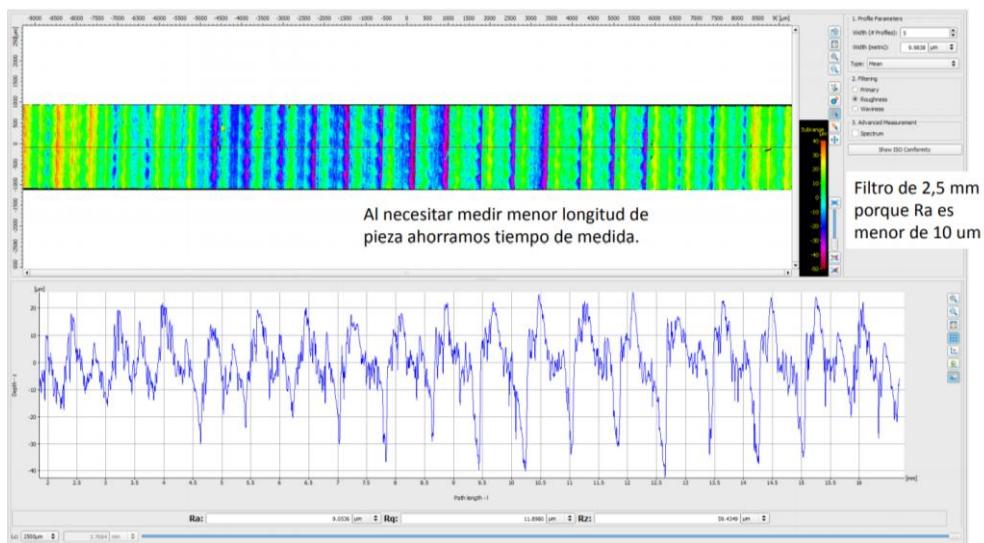


Fig. A.VI-5. Se utiliza un filtro de 2500 micrómetros porque se estima un Ra menor a 10 μm . Además, este hecho implica que la longitud a escanear sea menor, ahorrando tiempo en la medida.

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

CUÑA			
	Ra [μm]	Rq [μm]	Rz [μm]
FDM no-planar (Lc=2500 μm)	9.0	11.8	59.4

Como se puede apreciar, la mejora en la rugosidad es más que notable en esta última pieza, reduciéndose casi por cinco todos los valores.

Un detalle que anotar es que parece que la superficie presente ahora ciertas micro-marcas. Si se comparan las gráficas de ambos perfiles de rugosidad se puede notar que la nueva superficie es más “ruidosa”.

2. Rugosidad en la curva extruida

Pieza referencia/ Impresión FDM planar

En este caso, como la pendiente es variable, el perfil de rugosidad no será homogéneo, sino que irá disminuyendo según nos aproximemos a la parte superior donde la superficie se aplana.



Fig. A.VI-6. En la parte superior de la pieza se aprecia una superficie plana como consecuencia de la discretización de la pieza en el laminado.

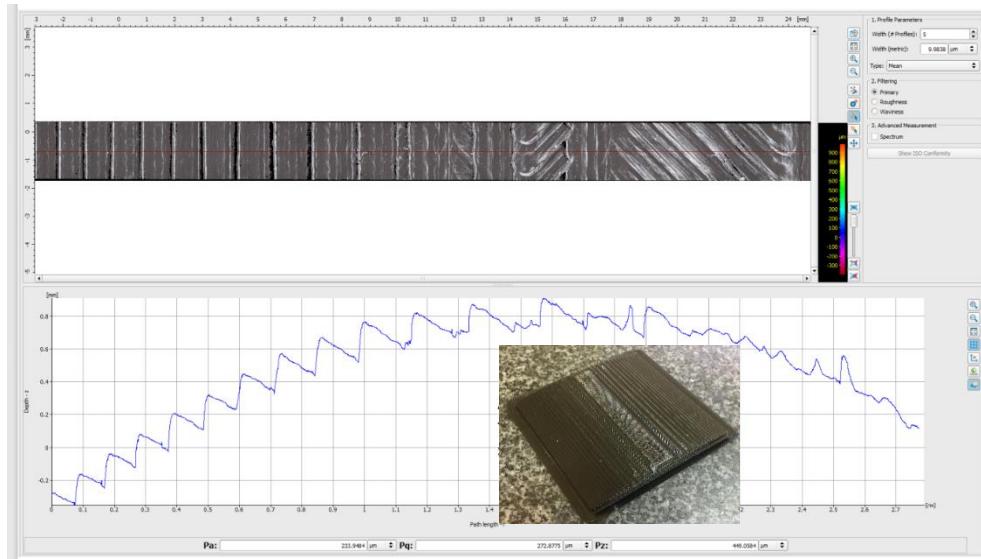


Fig. A.VI-7. Perfil primario sin filtrar donde se aprecia la curvatura de la superficie y la parte plana superior. Cuando se analiza la rugosidad de una superficie se tiene que filtrar el perfil primario que es el que contiene la curvatura de la superficie.

En la gráfica se aprecia perfectamente como los escalones tienen una altura más o menos constante en torno a 0.2 mm (relacionado con la altura de capa), pero su anchura va cambiando para adaptarse al cambio de curvatura de la superficie. En la parte final se ve como la superficie se ha aplanoado y ya no hay saltos de capa.

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

CURVA EXTRUIDA			
	Ra [μm]	Rq [μm]	Rz [μm]
FDM planar (Lc=8000 μm)	45.0	54.7	194.4

Pieza Impresión FDM no-planar

Procediendo de manera similar al caso anterior obtenemos los siguientes resultados:

CURVA EXTRUIDA			
	Ra [μm]	Rq [μm]	Rz [μm]
FDM no-planar (Lc=8000 μm)	17.4	21.1	72.4

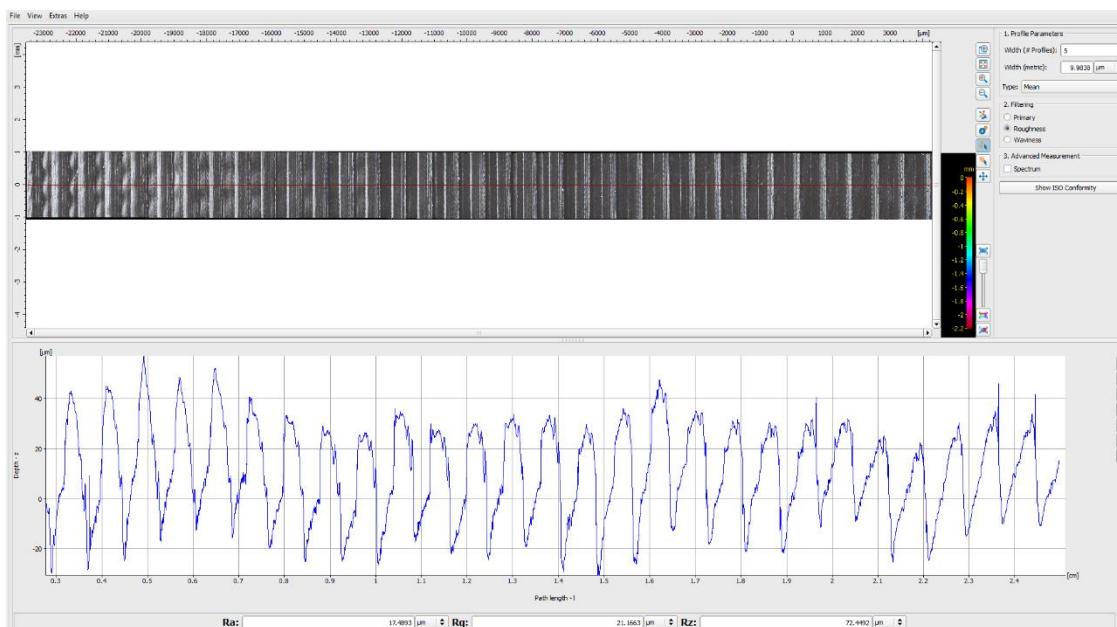


Fig. A.VI-8. Perfil de rugosidad una vez realizado el filtrado.

En este caso, aunque no es tan grande la reducción en el valor de la rugosidad como en el caso anterior, si analizamos el perfil completo, es muy evidente la homogeneidad de los escalones generados por el proceso de impresión no plano.

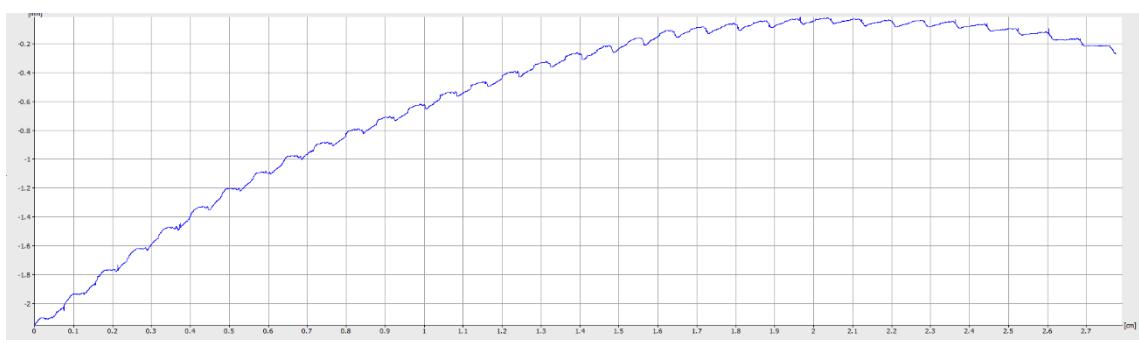


Fig. A.VI-9. Perfil sin filtrar donde se aprecia la homogeneidad de los escalones.

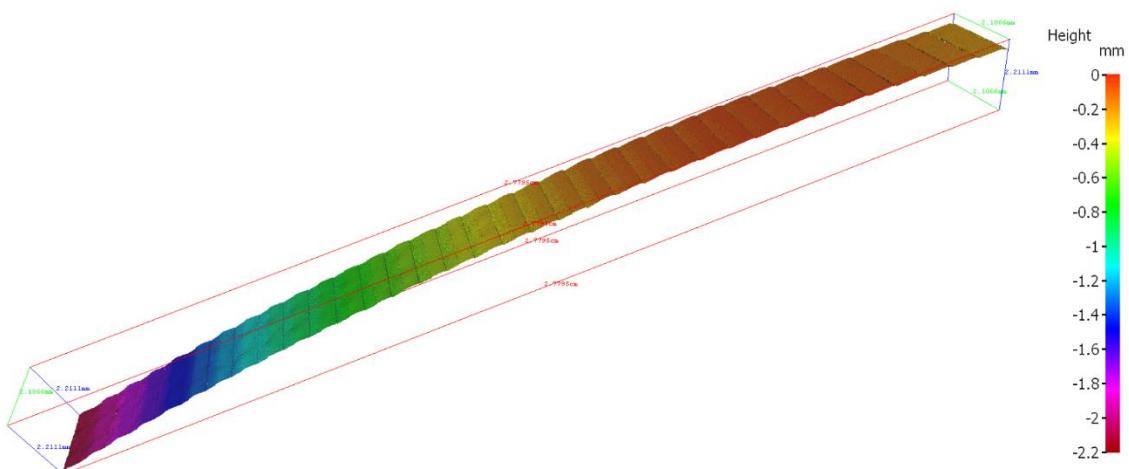


Fig. A.VI-10. Escaneado en color del perfil medido. En este se aprecia con un barrido de color las diferentes alturas (Z) donde se ha extruido el filamento.

3. Rugosidad en la superficie compleja "PIVA"

Pieza referencia/ Impresión FDM planar

A diferencia de los casos anteriores, en las siguientes dos piezas a medir ya no es evidente la existencia de una dirección preferente para evaluar la rugosidad. En estos casos, es más razonable abordar el análisis mediante parámetros de rugosidad superficial, frente a los anteriores parámetros lineales.

Los resultados de la rugosidad así obtenidos se reflejan en la siguiente tabla, donde hemos considerado dos longitudes de corte:

SUPERFICIE COMPLEJA "PIVA"			
	Sa [μm]	Sq [μm]	Sz [μm]
FDM planar (Lc=7500 μm)	51.8	66.0	1760.3
FDM planar (Lc=2500 μm)	42.8	57.2	1741.2

Por otra parte, se muestran dos fotografías del escaneo.

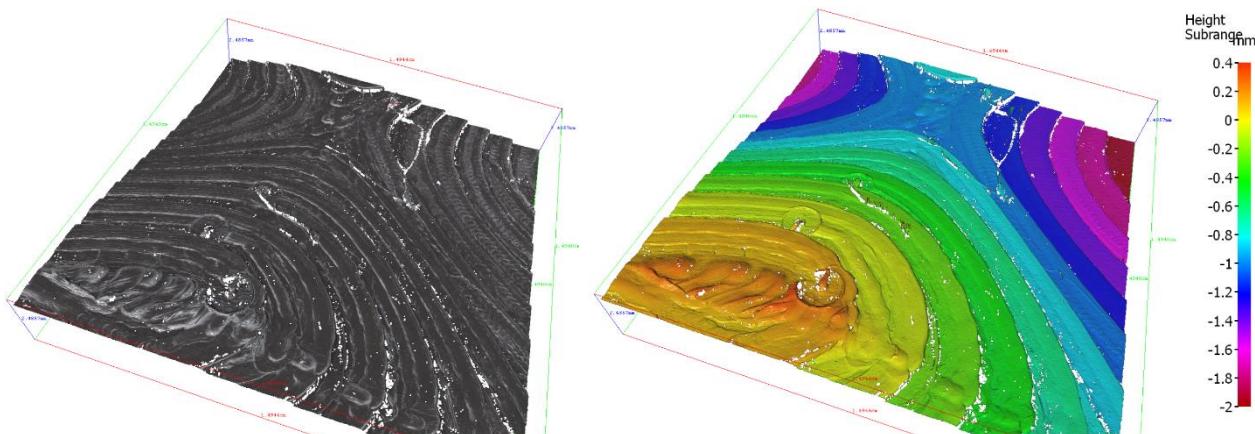


Fig. A.VI-11. Escaneado original y contrastado en color. En este se aprecian pequeños desperfectos en la superficie, ya enumerados en la memoria, que hacen que los resultados obtenidos no sean del todo precisos.

Pieza Impresión FDM no-planar

Para esta pieza no-planar se realizaron/imprimieron dos variantes cuya única diferencia radica en la orientación de la piel más exterior. En una pieza dicha piel se orienta de forma horizontal (0°) y en la otra de forma diagonal (45°).

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

SUPERFICIE COMPLEJA "PIVA"			
	Sa [μm]	Sq [μm]	Sz [μm]
FDM no-planar piel 45° (Lc=7500 μm)	32.5	42.1	668.4
FDM no-planar piel 45° (Lc=2500 μm)	21.8	30.5	661.8

SUPERFICIE COMPLEJA "PIVA"			
	Sa [μm]	Sq [μm]	Sz [μm]
FDM no-planar piel 0° (Lc=7500 μm)	30.7	38.9	592.7
FDM no-planar piel 0° (Lc=2500 μm)	17.3	23.6	570.3

A continuación, se recogen algunos escaneos del proceso de medición:

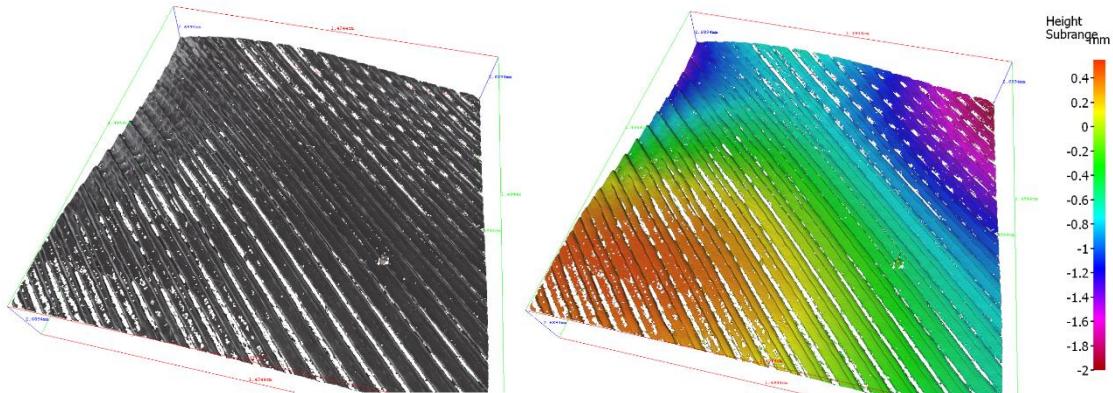


Fig. A.VI-12. Escaneado original y degradado en color de la piel impresa a 45°. En ambas imágenes se observan los huecos existentes entre los filamentos contiguos.

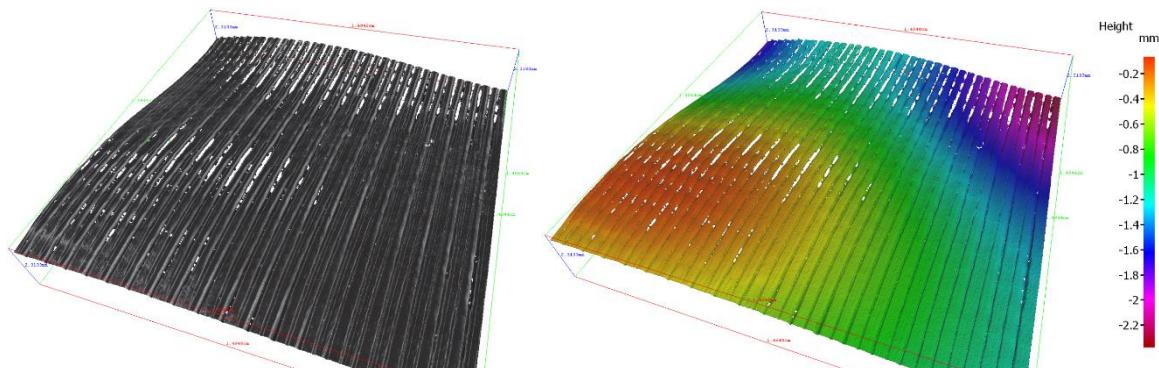


Fig. A.VI-13. Escaneado original y degradado en color de la piel impresa a 0°. En este caso se aprecia la existencia de menos huecos entre filamentos adyacentes.

4. Rugosidad en el paraboloide hiperbólico

Pieza referencia/ Impresión FDM planar

Al igual que en el caso anterior, la medición de la rugosidad del paraboloide hiperbólico se aborda mediante parámetros de rugosidad superficial, no lineales.

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

PARABOLOIDE HIPERBÓLICO			
	Sa [μm]	Sq [μm]	Sz [μm]
FDM planar (Lc=7500 μm)	48.6	58.6	506.1
FDM planar (Lc=2500 μm)	37.0	46.4	451.7

En este caso, por economía de medición no hemos medido un área de 40x40 mm. Tan solo hemos medido 15x15 mm resultado de combinar 8x8 imágenes.

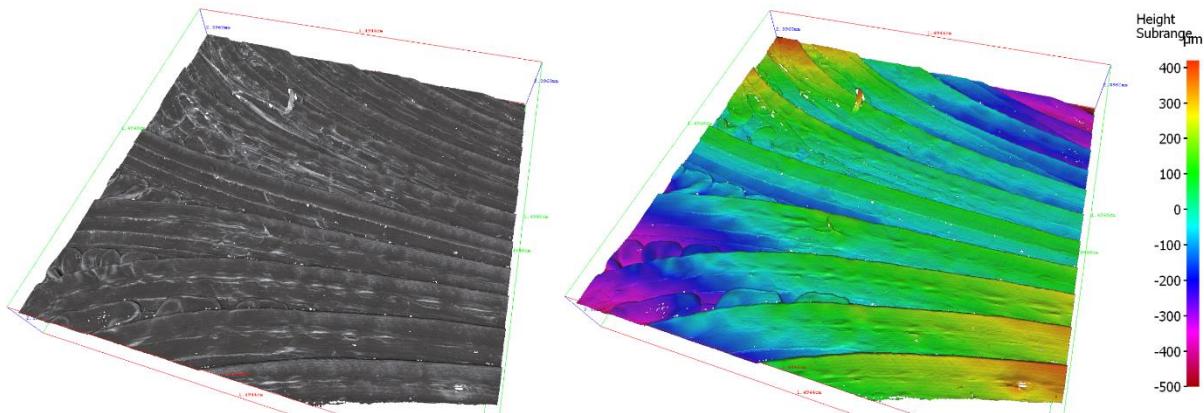


Fig. A.VI-14. Escaneado original y degradado en color de la medición.

Pieza Impresión FDM no-planar

Los resultados de rugosidad obtenidos se reflejan en la siguiente tabla:

PARABOLOIDE HIPERBÓLICO			
	Sa [μm]	Sq [μm]	Sz [μm]
FDM no-planar (Lc=7500 μm)	13.7	17.6	255.9
FDM no-planar (Lc=2500 μm)	13.4	17.1	254.9

En este caso, el hecho de variar la longitud de corte, Lc, apenas afecta a los resultados obtenidos. Esto es así porque la rugosidad es mucho menor que en el caso anterior.

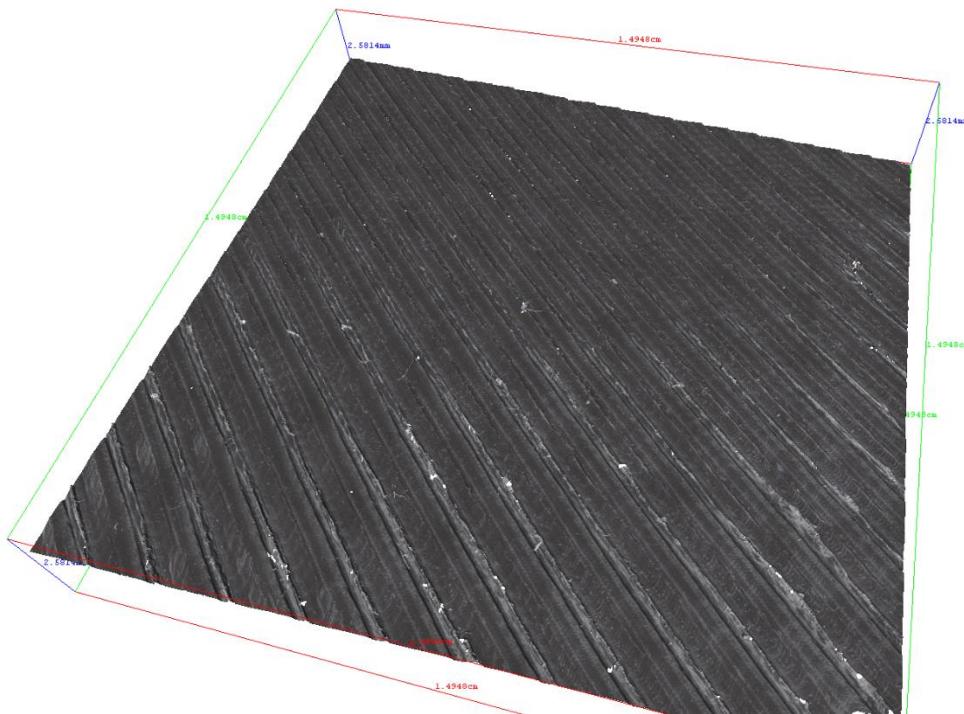


Fig. A.VI-15. Escaneado de la superficie del paraboloide hiperbólico impreso mediante el método FDM no planar. En la imagen se muestra la homogeneidad de la superficie, libre de los escalonamientos provocados por la impresión FDM tradicional.