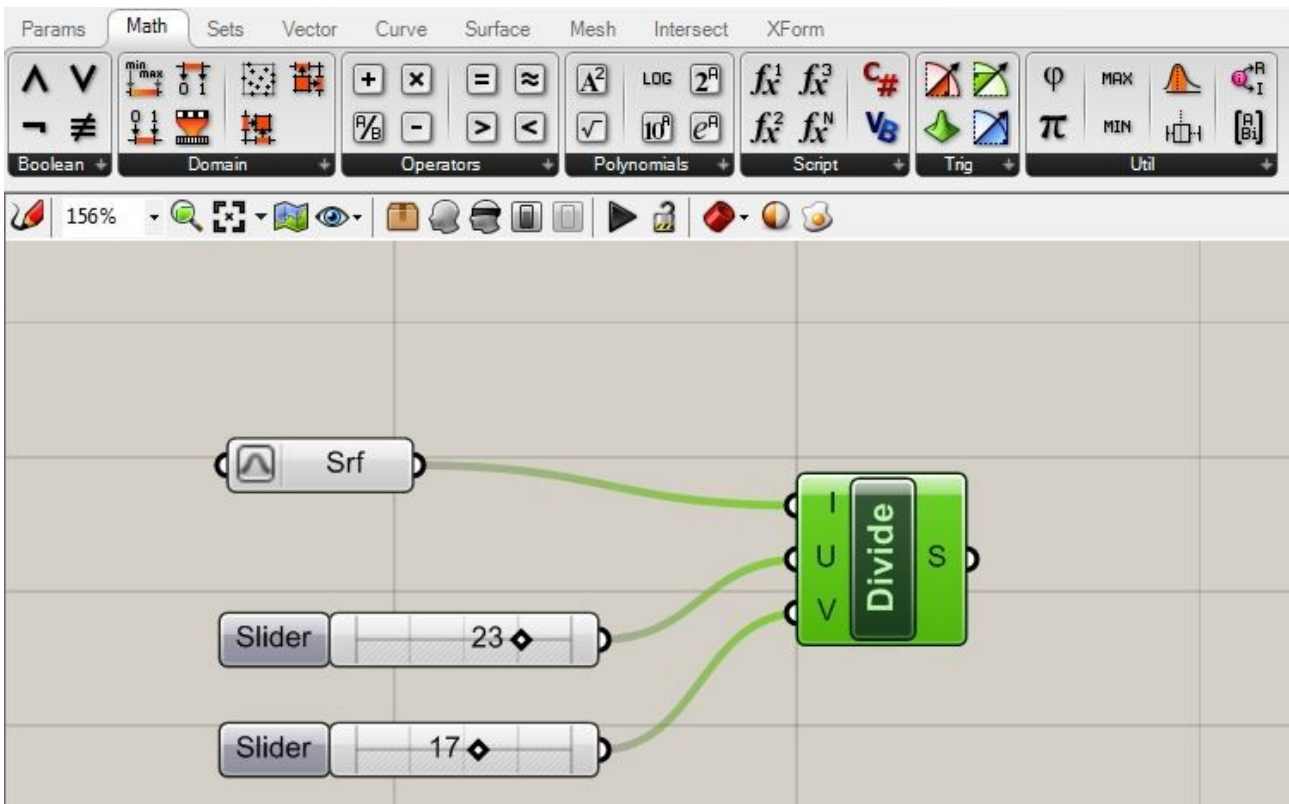


Vediamo ancora come lavorare sulle superfici attraverso la loro parametrizzazione, e suddivisione progressiva.

Alcune volte in Grasshopper ci sono diversi modi per fare le stesse cose.

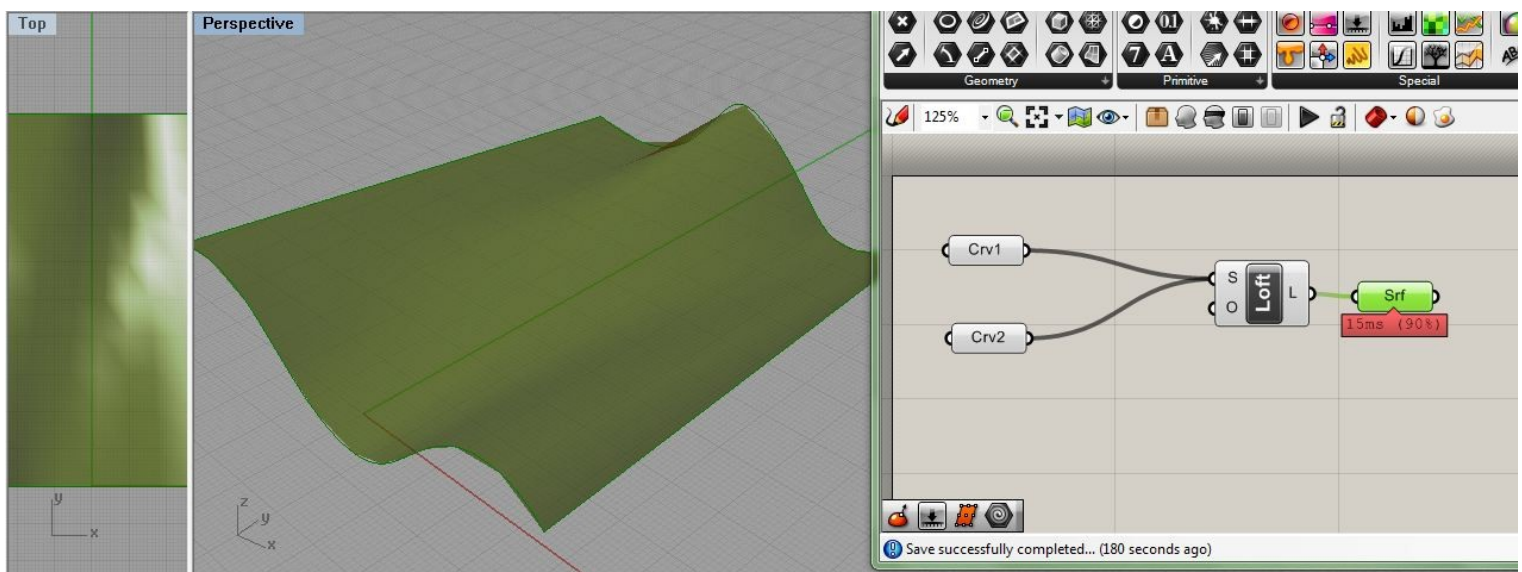
Abbiamo visto il component *SDivide*; in realtà posso sempre pensare che la superficie è un dominio bidimensionale di coordinate u e v , questo vuol dire che posso, considerando le operazioni da questo punto di vista, utilizzare alcuni component del pannello *Math*, come abbiamo visto

Quindi tornando al caso della superficie precedente posso decomporla utilizzando il component *Divide Domain*, che prendo dal tab *Math*, ed in particolare da *Domain*.



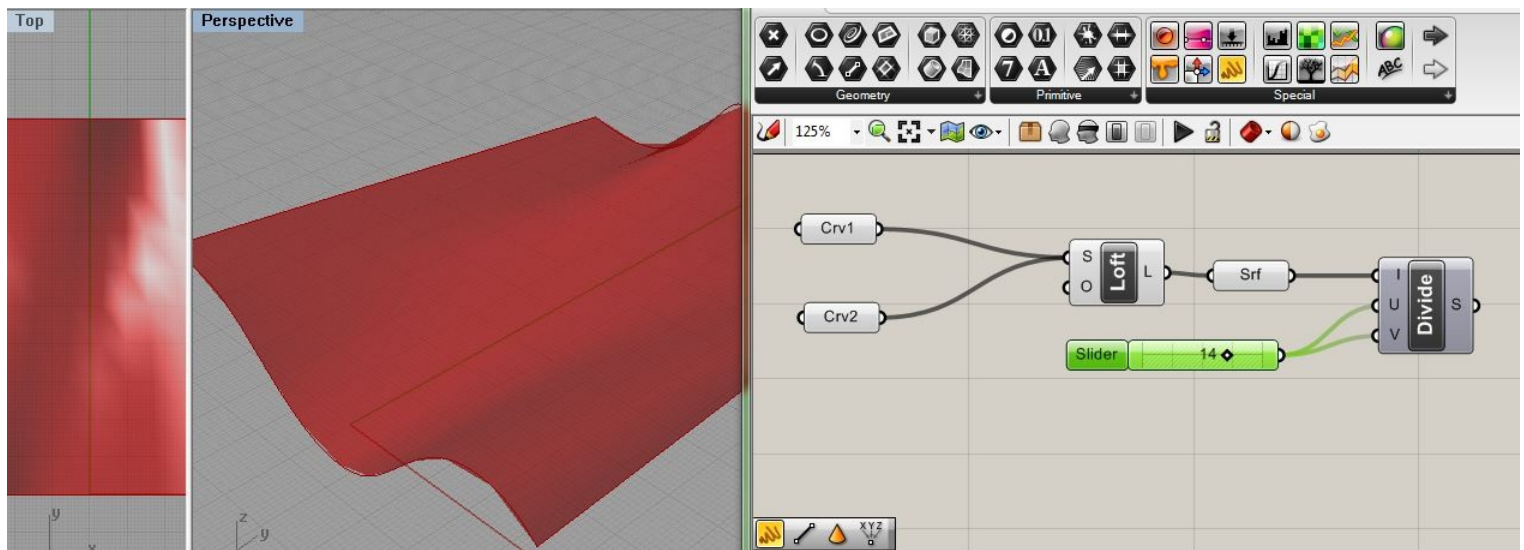
Connetto ora la superficie (che ho già ri-parametrizzato, con la *I* del component, e associo 2 slider, dopo che li ho definiti come *integer* (ovviamente).

Per fare però le cose più interessanti, questa volta partiamo da 2 curve sulla scena e generiamo una superficie con il component *Loft*. Abbiamo già visto questa particolare procedura e dovrebbe esservi chiara:

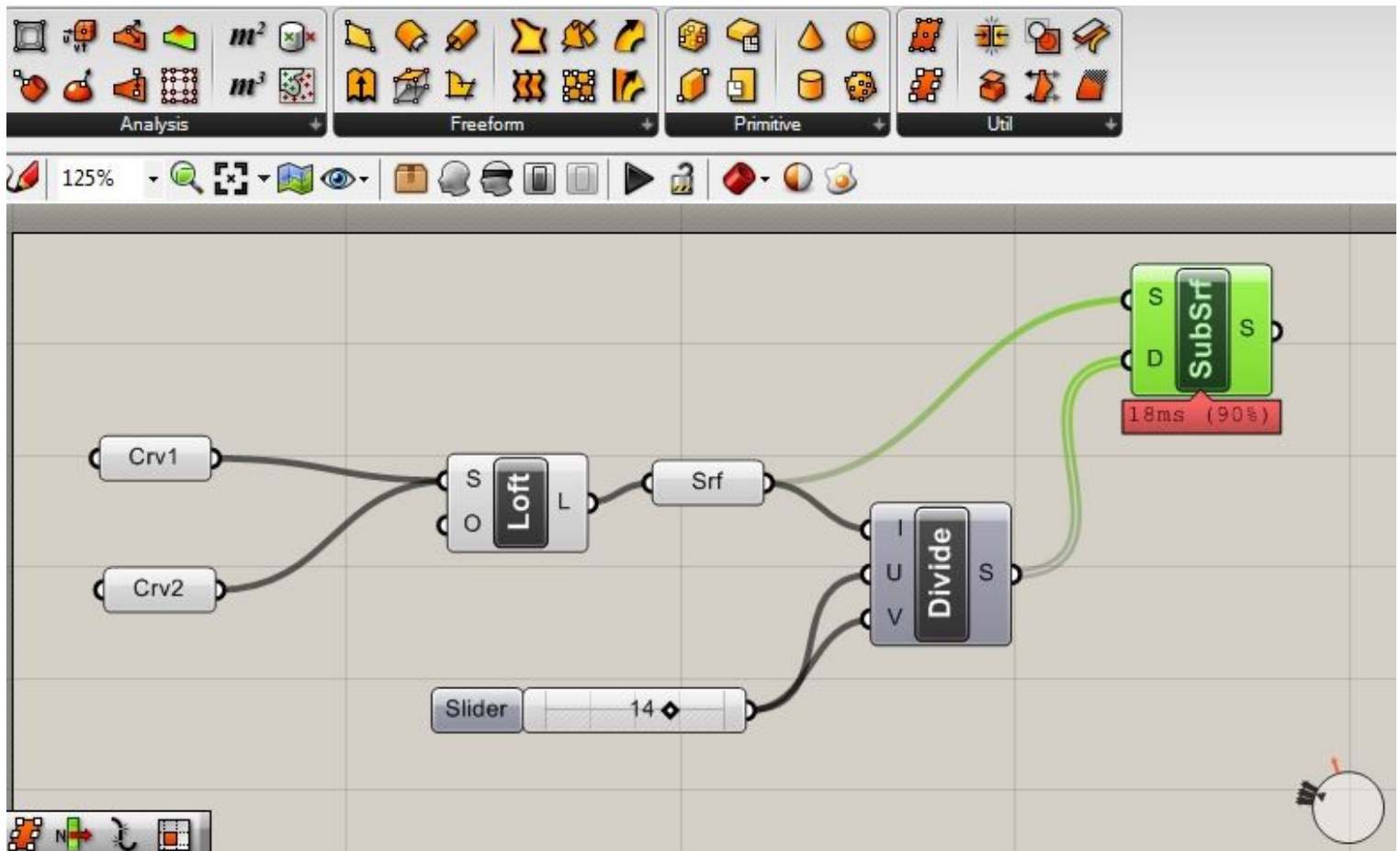


A questo punto come già fatto connettiamo l'output del *Loft*, con un component *Surface* da *Parameter*, che ci permette intanto di 'raccogliere' la nostra superficie.

A questo punto prendiamo il component *Divide2*, da Math, come ormai sappiamo e connettiamo la nostra superficie al dominio I. Per semplificarci la vita utilizziamo infine un unico slider, per i valori di u e v , scegliendo un numero di parti di suddivisione (nel caso abbiamo messo 20).

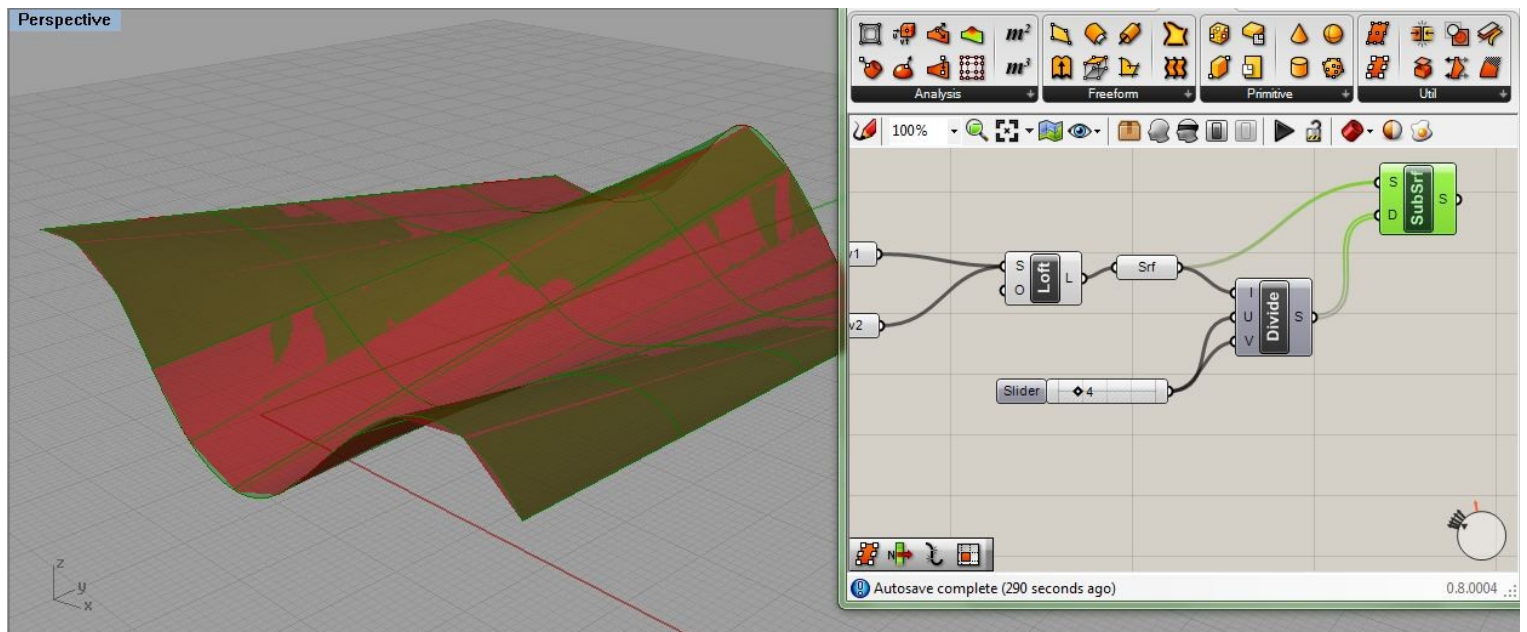
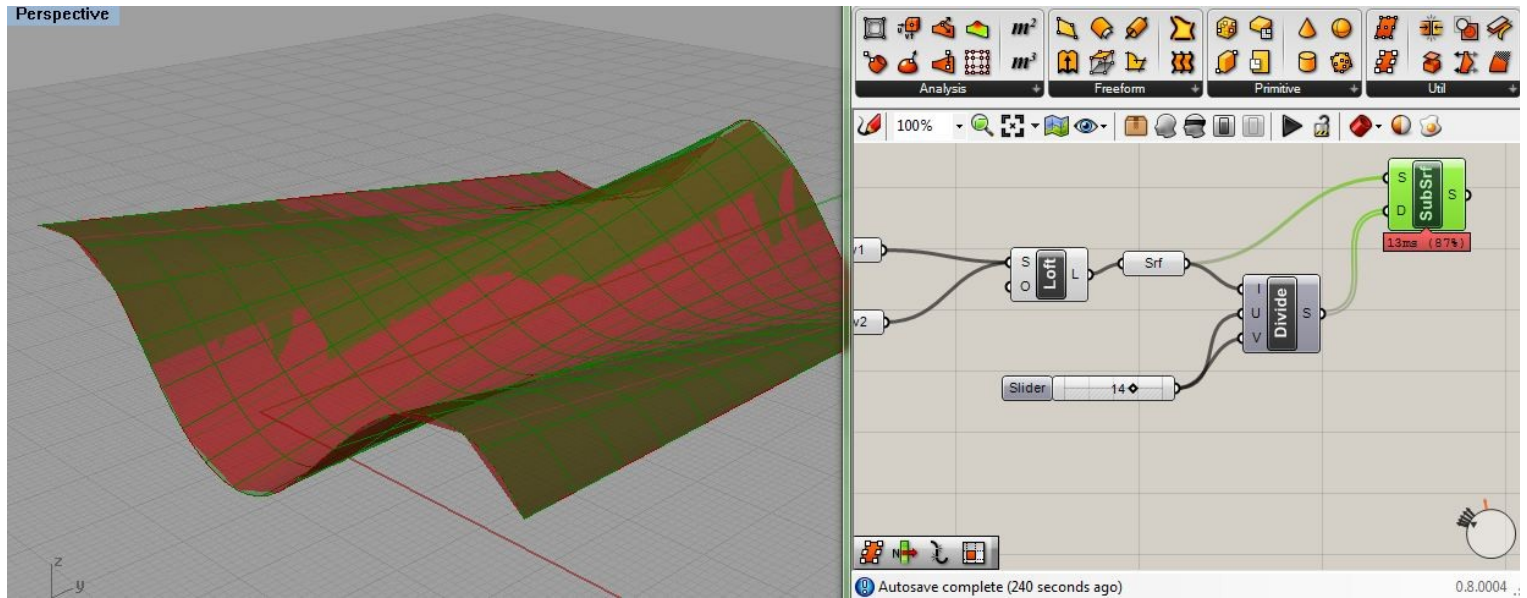


Ora, per poter operare sulle suddivisioni ottenute, andiamo nel tab Surface sul pannello Util, e selezioniamo il component *Isotrim*, che ci permette di estrarre un *subset* dalla superficie. Connettiamo l'input S con la nostra superficie e l'input D, con l'output S di Divide:



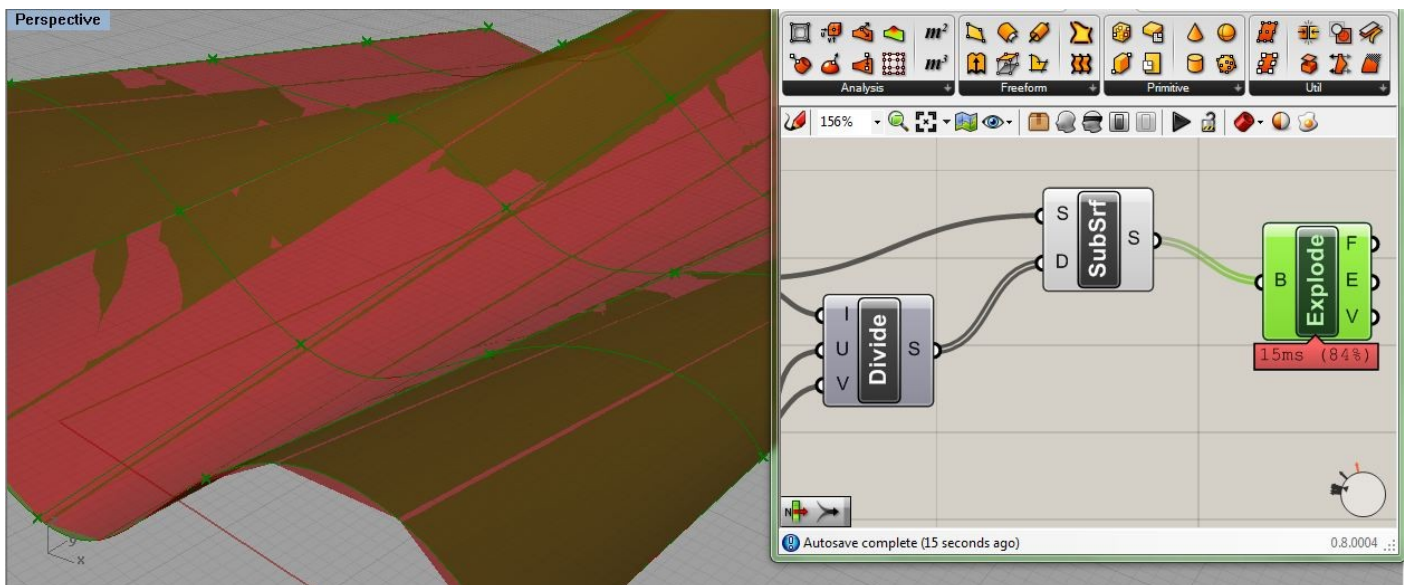
Dovremmo a questo punto incominciare a vedere la suddivisione della nostra superficie.

Questa ora viene divisa in sotto-superfici permettendoci così di lavorare a livello locale.



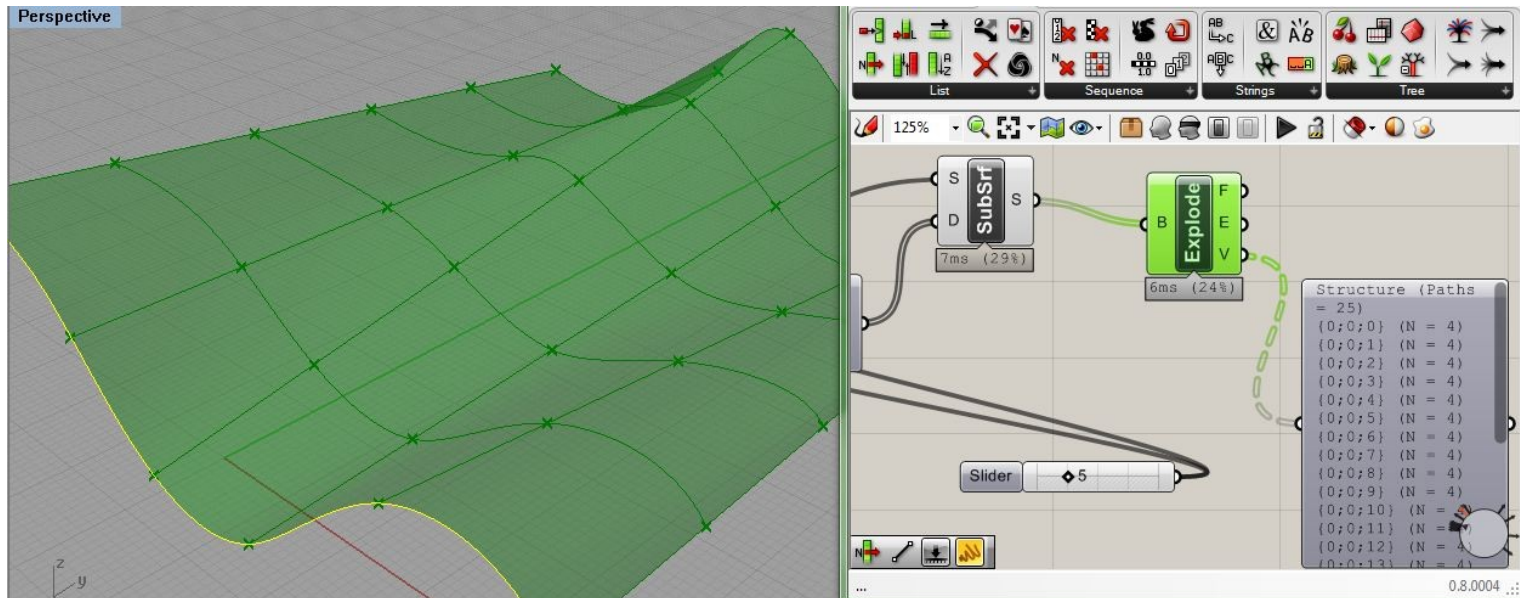
Per poter lavorare su tali elementi locali, dobbiamo pensare intanto a cosa significa manipolare tali *subset* a livello locale. Ogni elemento infatti è formato da bordi e vertici. Per accedere ad essi abbiamo bisogno dunque di *esplodere* i nostri elementi. Di averli cioè accessibili in quanto tali.

A questo scopo usiamo allora il componente *Explode (Brep Components)* sotto il tab Analysis nella sezione delle superfici; quando lo connettiamo infatti tutti i sub-elementi diventano disponibili a livello locale.



Gli output del component sono infatti: F, E, V, che corrispondono rispettivamente alle facce, ai bordi e ai vertici dei nostri elementi.

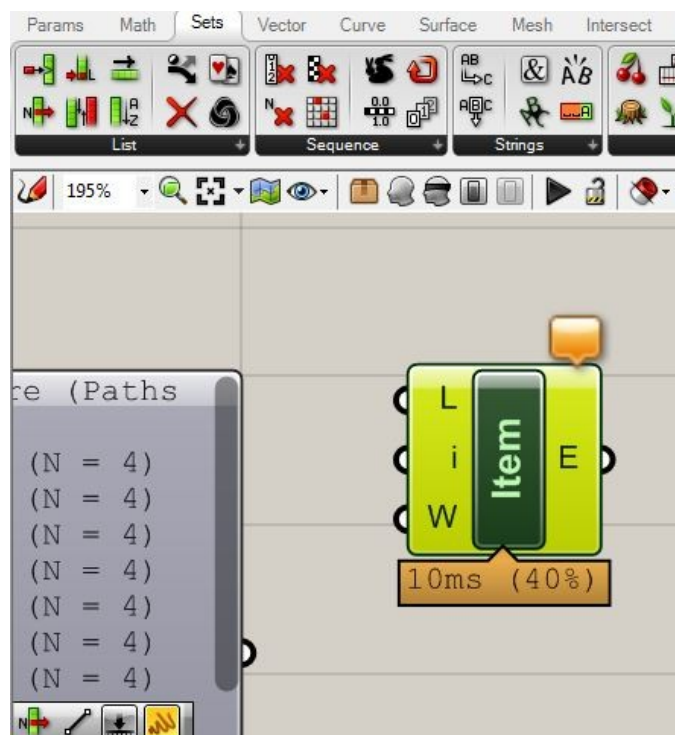
Se connettiamo l'output del component *SubSurface* ad un component che prendiamo dal tab Params/Special, in particolare *Param Viewer*, possiamo fare delle osservazioni interessanti sui dati che stiamo manipolando. In particolare i *Paths* variano col variare dello slider, (se è a 4 ne ottengo naturalmente 16, se è a 5 ovviamente ne ho 25, e così via). Inoltre ogni path ha 4 elementi su cui lavorare. Lo sforzo è quello di immaginare delle strutture di dati ad albero, e dunque ramificate. Ora, quello che ci preme è come recuperare questi dati.



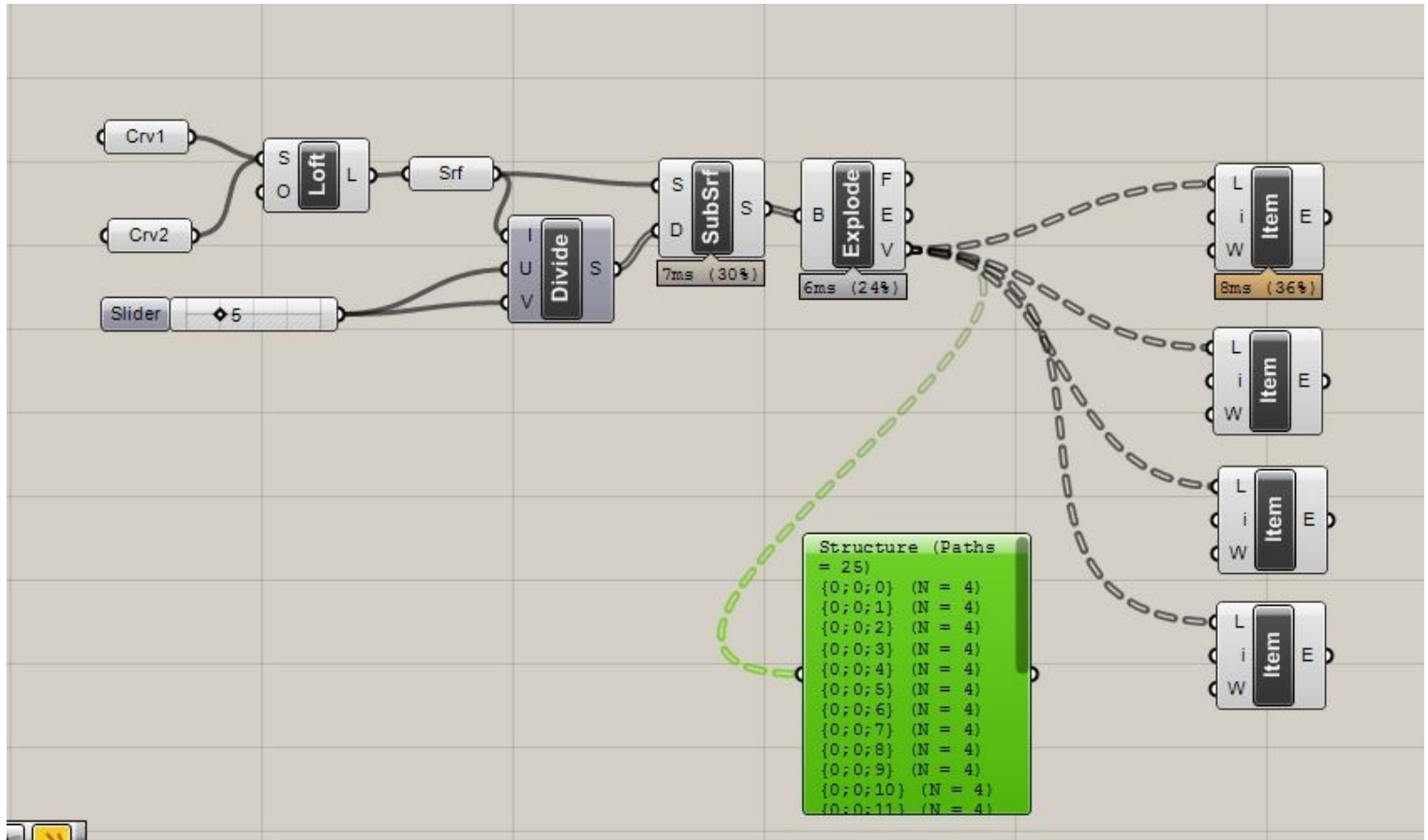
Abbiamo già esaminato il tab Seths, con in particolar modo il component *List*. Abbiamo anche visto che il component *List Item*, ci permette di recuperare un *Item* (ovvero una voce, o un articolo), da una certa Lista o elenco. In questo caso noi vogliamo recuperare i vertici dai nostri subset si superfici per connetterli e suddividere ancora la nostra superficie.

Trasciniamo sul canvas 4 component del tipo *List Item*, sempre dal tab List.

Ora esaminiamone di nuovo: a destra, in ingresso abbiamo la L che rappresenta l'elenco dei dati, e la i che rappresenta l'indice (i dati cioè, come abbiamo chiarito ieri, sono indicizzati). La w è una booleana che per ora non ci interessa.



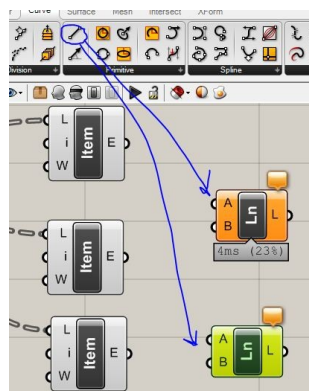
Ora di questi 4 component, dobbiamo fare le seguenti operazioni:
anzitutto ogni component dovrà avere in ingresso sulla L, proprio la lista di dati su cui operare, e dunque nel nostro caso i V, che provengono dal component *Explode*.



Il prossimo passaggio invece richiede più attenzione. Abbiamo visto che ogni paths ha 4 elementi su cui operare. Questi sono indicizzati nel modo che segue: 0,1,2,3. Sono 4 è vero, ma la numerazione comincia con lo 0. Quindi quello che dobbiamo fare è dare ad ogni i dei component *Item* ognuno di questi valori da 0 a 3. Per fare questo potremmo utilizzare uno slider. Ma andiamo direttamente col tasto destro sulla i del component, e cliccando sul menù su Set Integer diamo i numeri in questo modo:
alla prima i del primo *Item* diamo il valore 0;
alla seconda i del secondo il valore 1,
alla terza il valore 2;
alla quarta e ultima i diamo il valore 3.
In questo modo ognuno di questi component sa su quale elemento indicizzato della lista deve operare. In altre parole il primo *Item* opererà su tutti gli elementi della lista che hanno indice 0; il secondo su quelli che hanno indice 1, ... e così via.

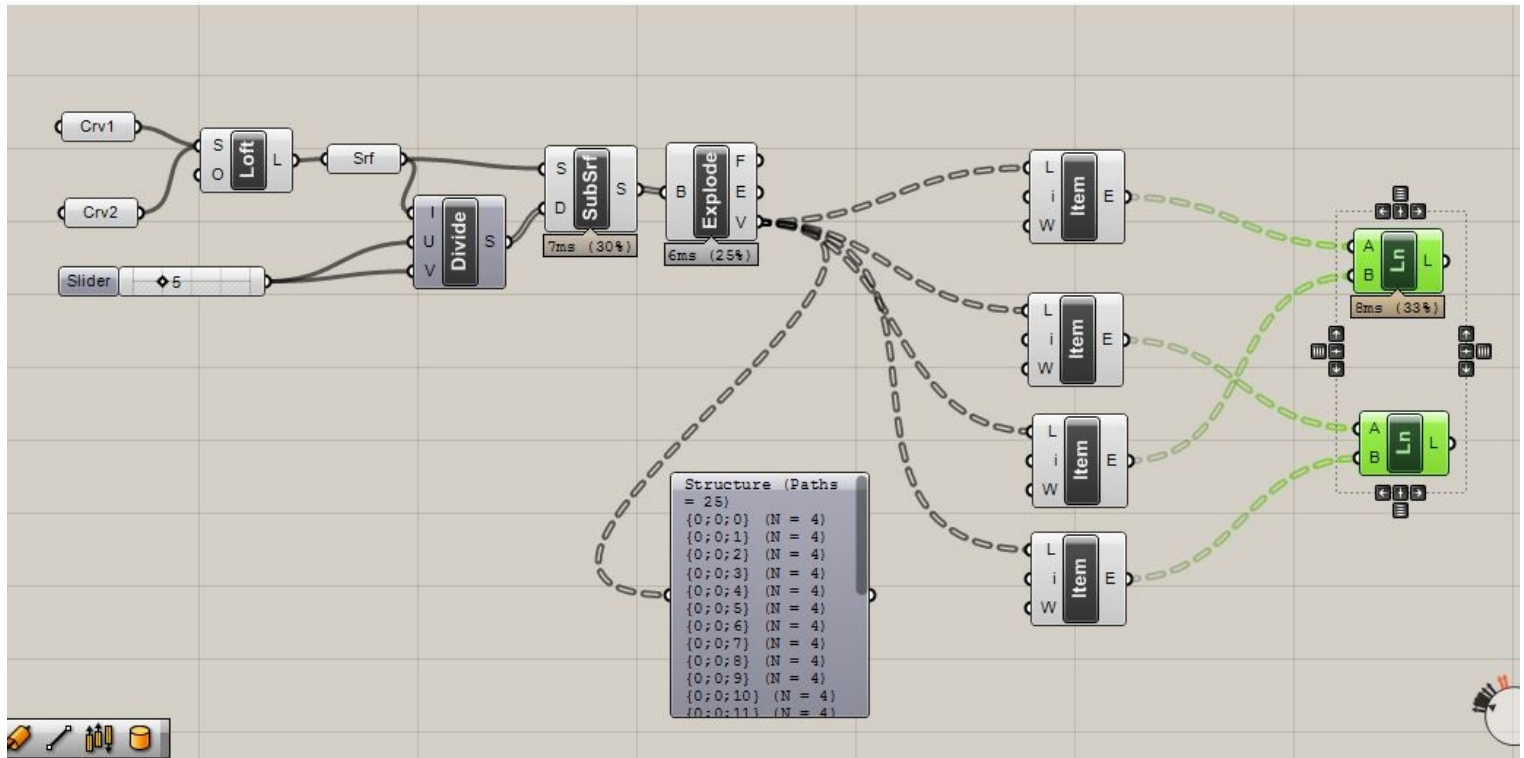
Fatta questa operazione, torniamo al nostro obiettivo che è quello di triangolare, per così dire i nostri subset di superficie. Per farlo io devo tracciare delle diagonali fra i vertici di ogni settore. Quindi prendo 2 component *Line*, dal tab Curve.

Ne prendo 2 naturalmente, poiché devo tracciare due diagonali.
Ma come li connettiamo?

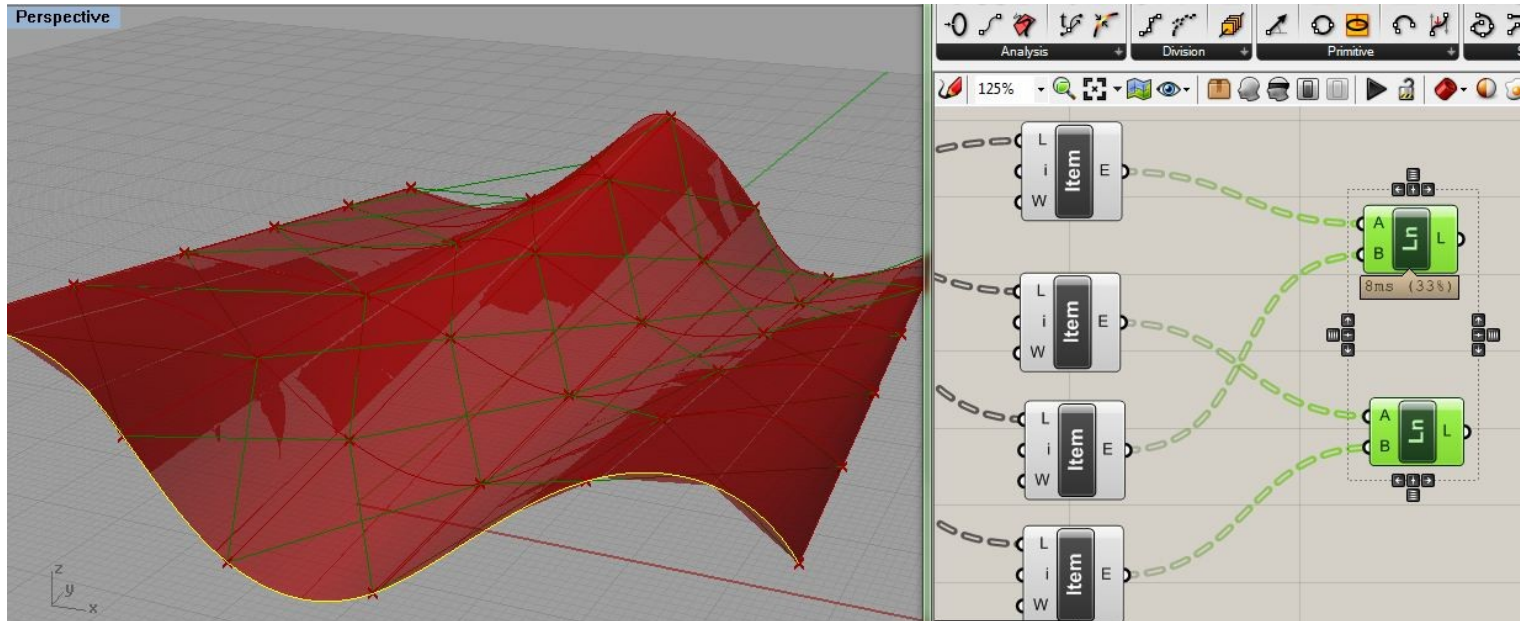


In sostanza, ogni component *Line* ha 2 estremi A e B.

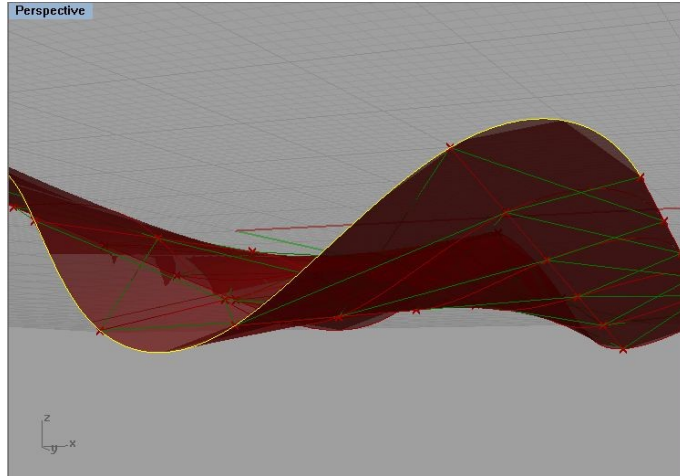
Le A vanno connesse con le i relative ai primi 2 *Item*, le B con i rimanenti, secondo lo schema visibile qui sotto:



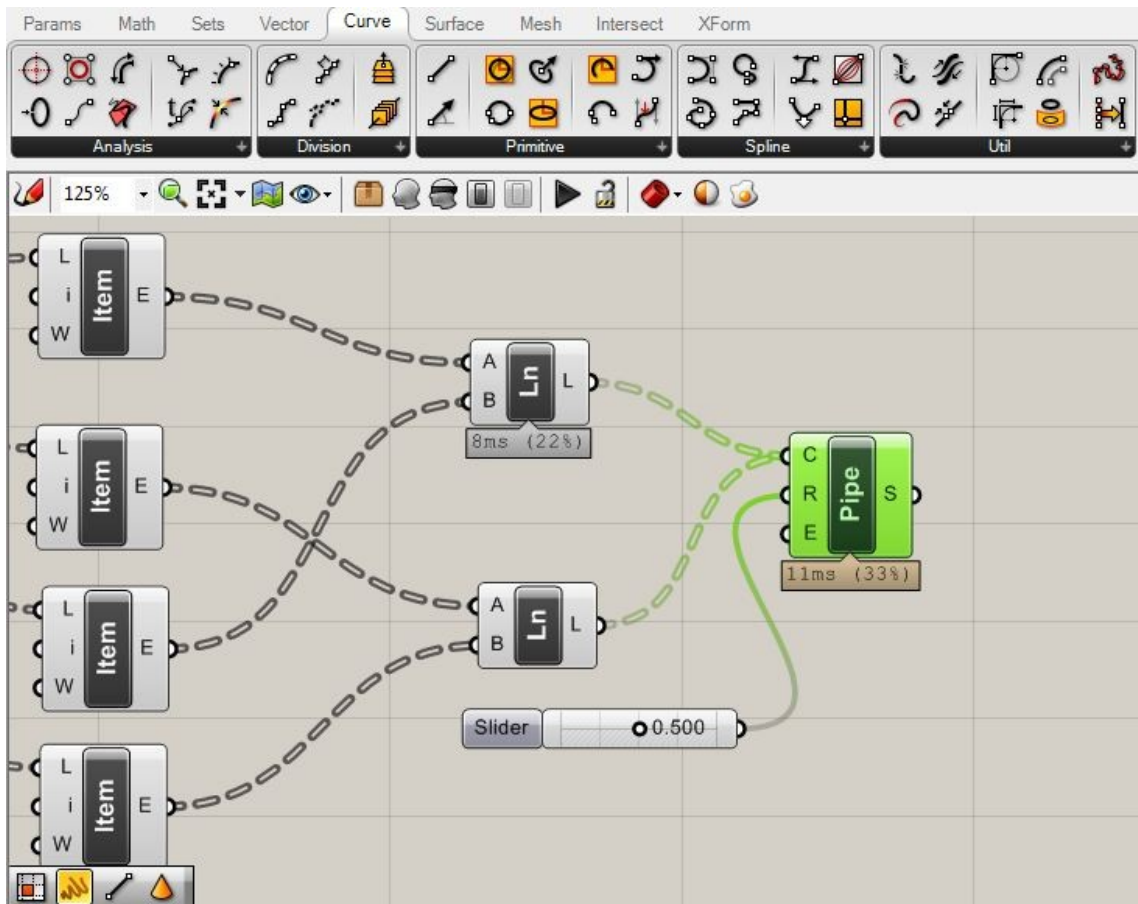
Cosa succede sulla scena? Possiamo vedere le diagonali di connessione lungo la superficie.

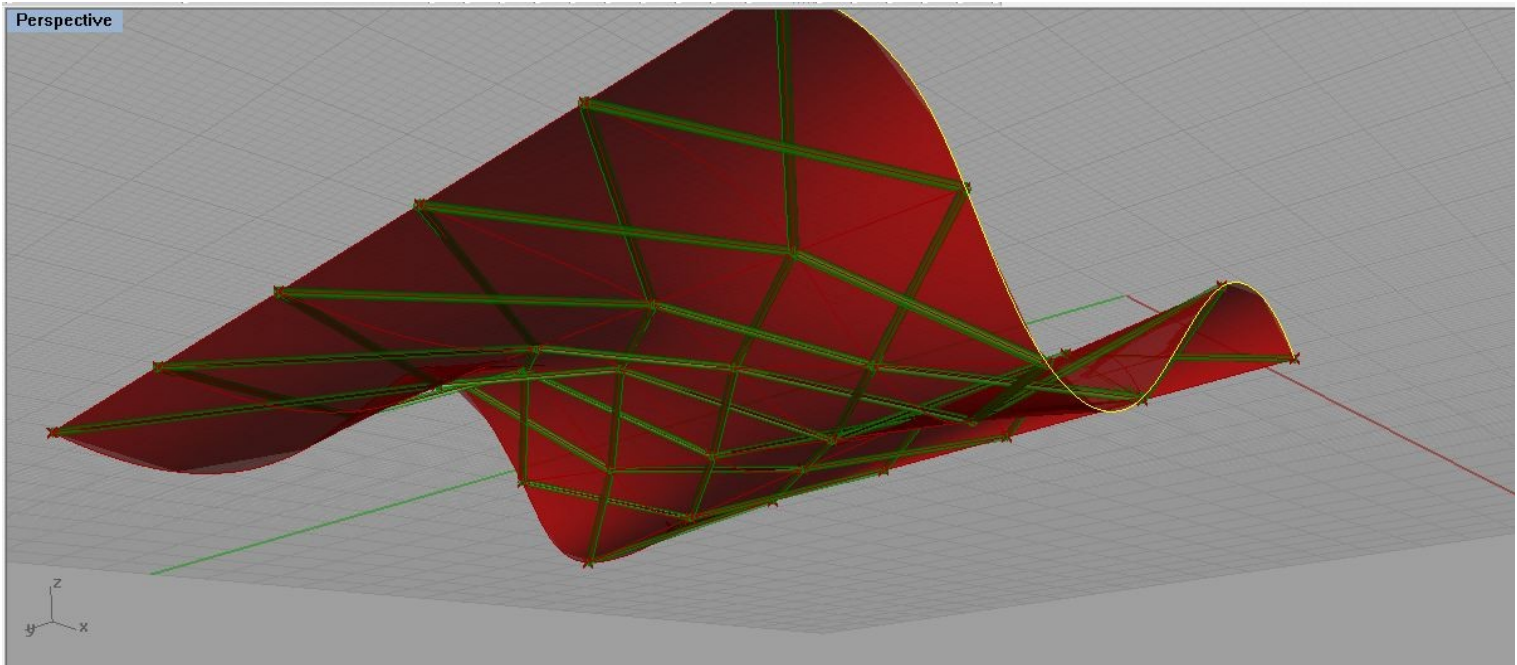


Ora, facciamo un ulteriore passo. Trasformiamo queste linee in dei solidi tubolari.

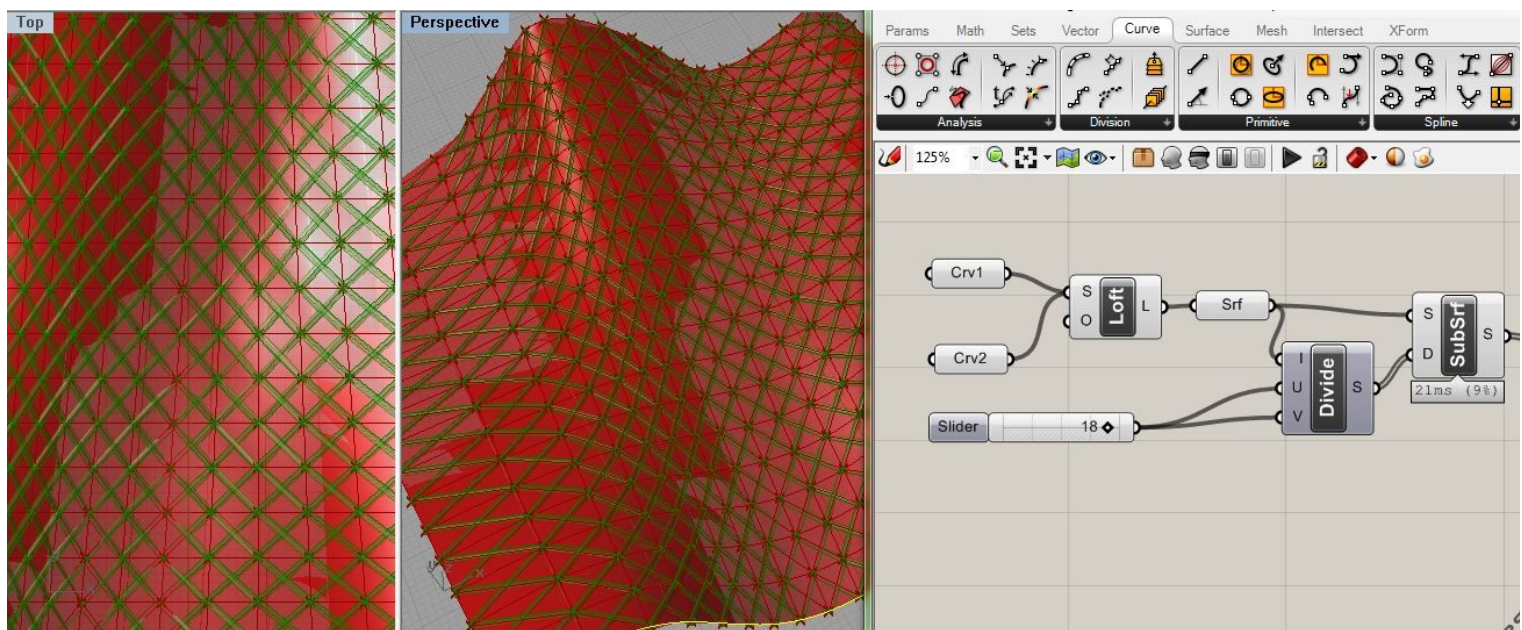


Per fare questo prendiamo il component *Pipe* dal tab Surface/ Freeform. Questo component, ci permette di trasformare delle curve in degli elementi tubolari. Connettiamo il valore in entrata C, con gli output dei component *Line*, e mettiamo uno slider per regolare il raggio dei tubi:

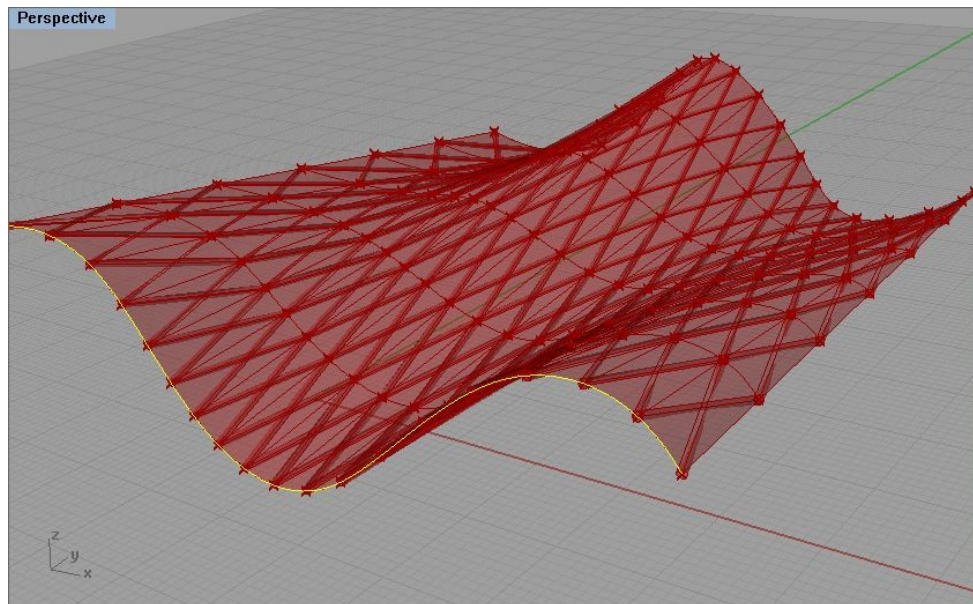
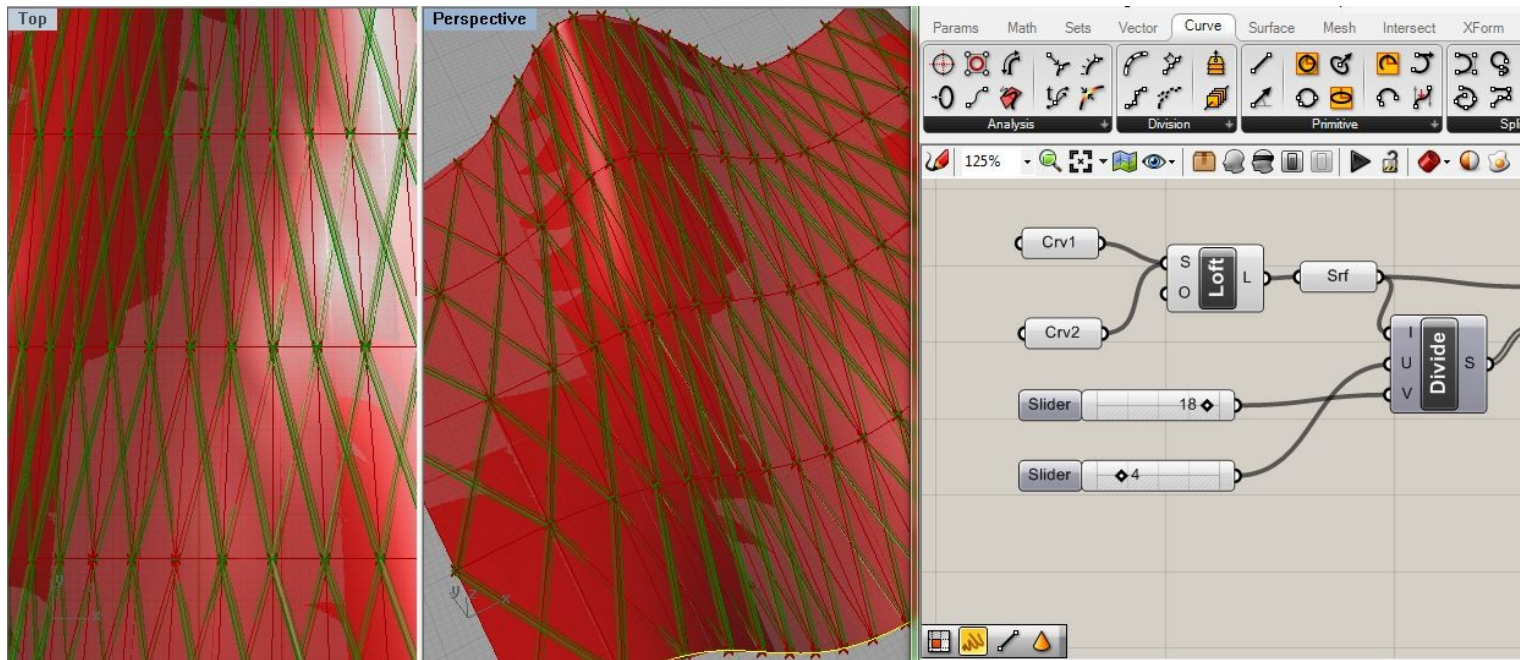




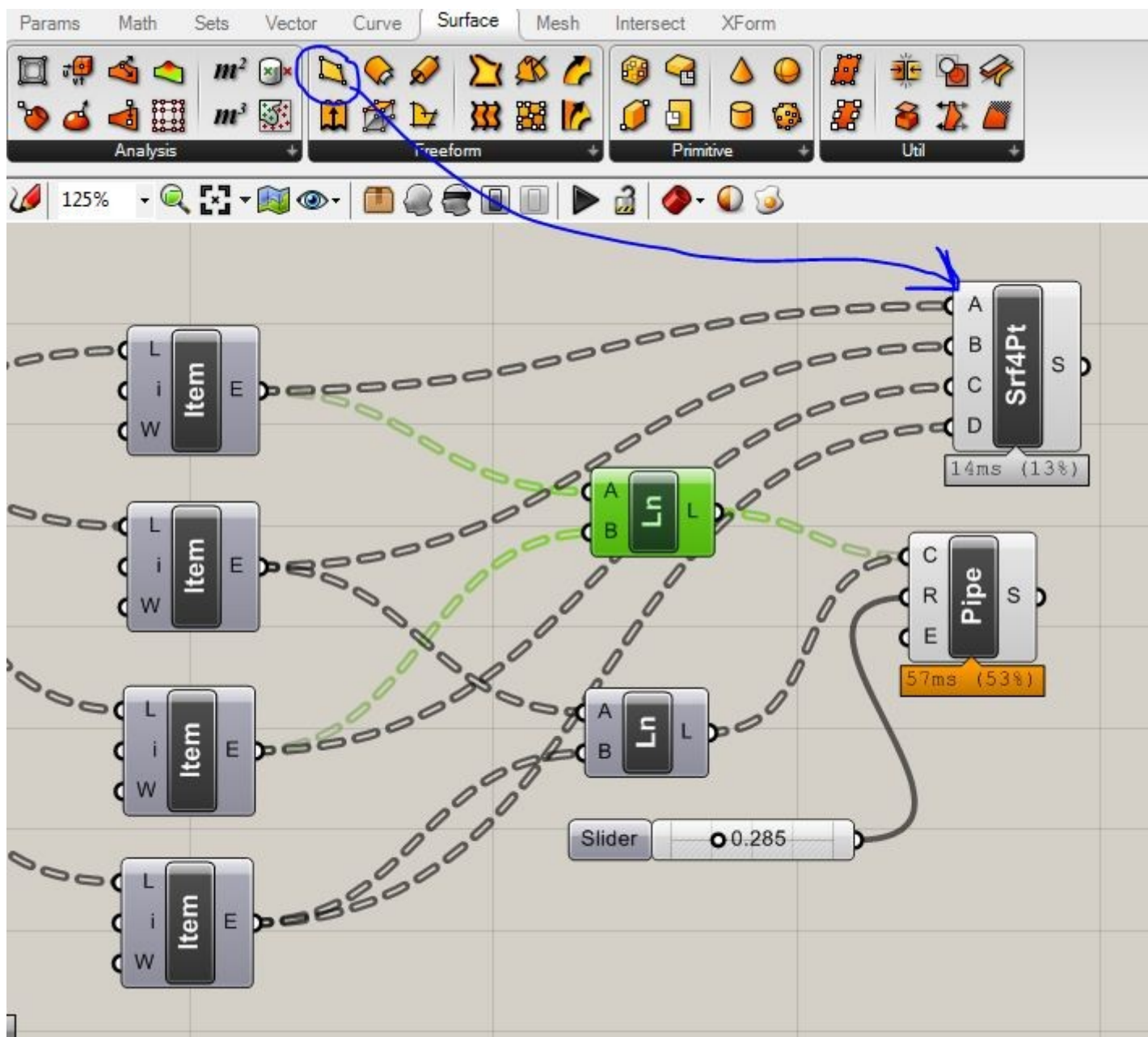
Questo è quello che dovremmo vedere sulla scena. Ogni linea è stata trasformata in un tubo. Ma ciò che è più interessante è che se torniamo allo slider che ci divide la superficie, possiamo cambiare la suddivisione di quest'ultima, ed ottenere così una densità maggiore o minore:



O ancora se mettiamo 2 slider diversi all'entrata di u e v , in entrata possiamo cambiare la tessitura:

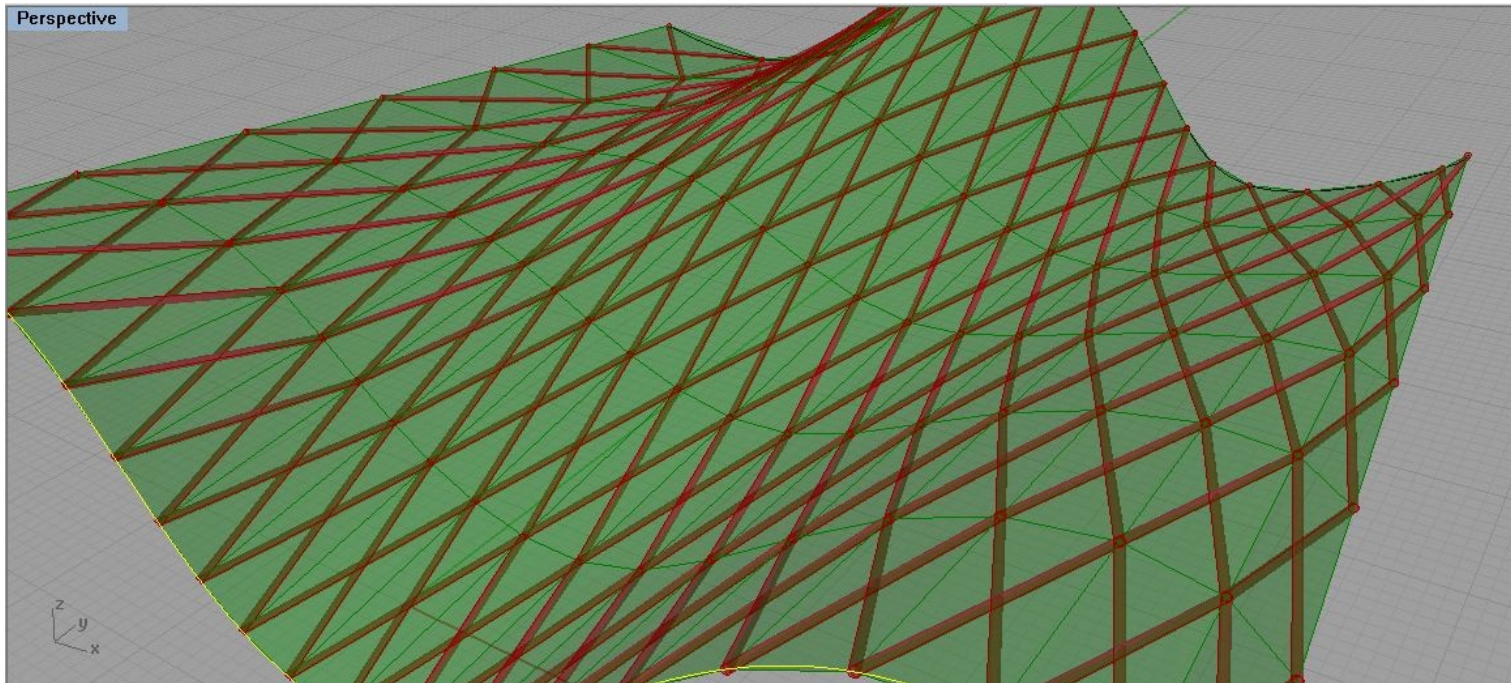


Una considerazione che però possiamo fare, è che osservando le nostre linee e i nostri tubi, è che sono rettilinei. Quindi forse potrebbe essere interessante ricavare dei pannelli piani fra i tubi così ottenuti. Per fare questo utilizziamo un component molto agevole; si tratta del component *4Point Surface*, che prendiamo dal pannello Surface/Freeform. Come dice il nome questo component ci permette di ottenere una superficie da 4 punti. Naturalmente i punti sono tutti gli output che ho già ottenuto dai component *List Item*. Quindi li associo uno per uno ai 4 input (A,B,C,D) di *4Point Surface*:



Qui non abbiamo fatto altro che fornire i 4 punti richiesti, suddivisione per suddivisione e poi darli al component il quale va a costruire le superfici relative.

Possiamo comunque ancora una volta tornare indietro e cambiare alcuni parametri, e dando il tempo al programma di generare il file, riuscire ad ottenere la nostra rete di tubi sulla superficie.



Questo tutorial è tratto dalla guida ufficiale di Grasshopper scaricabile come PDF da:
<http://www.grasshopper3d.com/>