**INGEGNERIA PARAMETRICA**

**L’ingegneria dal passato al presente…**

La figura dell’ingegnere non è poi così cambiata nel corso degli anni, quello che è cambiato e si è modificato è piuttosto il modo di pensare e di vivere l’ingegneria. Ciò che differenzia il presente, il passato e il futuro nel nostro lavoro sono la ricerca, lo sviluppo, la scoperta e la conoscenza di nuove tecnologie e nuovi materiali da impiegare nel settore.

Il primo approccio al *FEA*, acronimo di *Finite Element Analysis* avviene nel 1956 con un articolo sui metodi computazionali che viene concretizzato nel mondo dell’ingegneria strutturale nei primi anni ‘60. Il *FEA* è un processo mentale specifico che volta pagina ai calcoli a mano e ai lenti e macchinosi modelli per dare spazio ad una massimizzazione del lavoro ed una minimizzazione del tempo impiegato. Non dimentichiamoci che la matematica e la scienza delle costruzioni sono sempre alla base della logica dell’ingegneria e ciò che cambia negli anni e che cambierà nel futuro sono la forma e l’approccio al problema e come risolverlo, ma non la sostanza. Prima dell’avvento del calcolatore il professionista non utilizzava il computer e né tantomeno software per risolvere modelli meccanici ed è per questo che serviva molto più tempo e molta più manodopera per produrre risultati.



L’ingegneria computazionale con l’uso dei computer ha aiutato a sviluppare il design strutturale diminuendo i tempi di calcolo e migliorando prestazioni, velocità e qualità dei risultati. L’idea di utilizzare ragionamenti parametrici permette flessibilità nell’impostare svariati modelli e ottenere di conseguenza risultati in poco tempo. Il movimento del *Parametricismo* è nato per la prima volta con *il Manifesto* presentato da Patrick Shumacher, attuale direttore dello studio di *Zaha Hadid Architects,* alla Biennale di Venezia del 2008 dove concretizza l’inizio di un’Era in quello che è il modo attuale di vivere l’architettura Postmoderna. L’ingegneria in quanto strettamente connessa all’architettura e ai suoi movimenti si è quindi dovuta adattare al design, alle metodologie e allo stile di questo movimento. L’architettura odierna manifesta interesse in *pattern* sempre più organici e fluidi e l’ingegneria deve essere presente con lo stesso dinamismo e complessità.

**Aspetti positivi e negativi**

L’utilizzo intensivo della tecnica di modellazione parametrica ha aspetti positivi e negativi. Di seguito ne sono elencati alcuni di questi.

Lati positivi:

* procedura snella e flessibile
* ottimizzazioni
* migliori performance degli edifici
* modellazioni avanzate ( CFD, Physics based solvers )
* abilità di manipolare geometrie complesse
* espansione degli strumenti di analisi
* progettazione integrata con la produzione

Lati negativi:

* Generazione di vincoli involontari nella creazione dello spazio di ricerca
* Utilizzo forzato di script creati per progetti precedenti con conseguente propagazione degli errori
* Eccessivo affidamento alla tecnologia
* “Black box”

**Automatizzazione dei flussi di lavoro**

La pratica professionale ha la necessità di produrre risultati validi e allo stesso tempo speditivi. Allo stato attuale, un ingegnere non si può permettere di creare modelli complessi che potrebbero portare a risultati più accurati. I modelli meccanici strutturali, con lo sviluppo e ricerca, si sono ulteriormente raffinati e le numerose tecniche numeriche di risoluzione delle equazioni li rendono risolvibili.

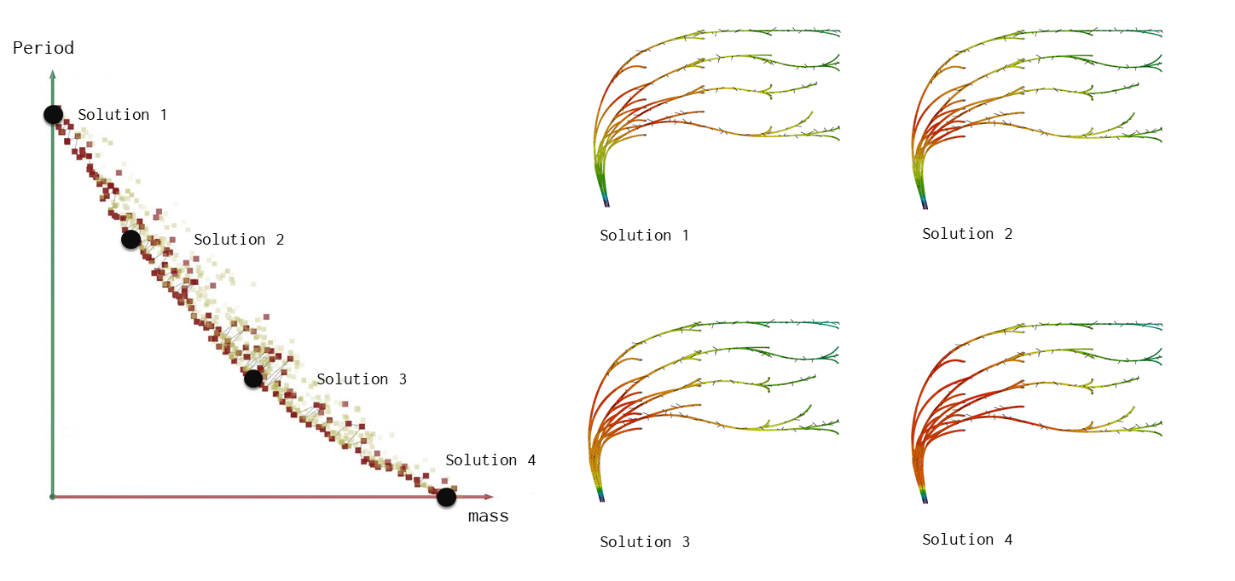
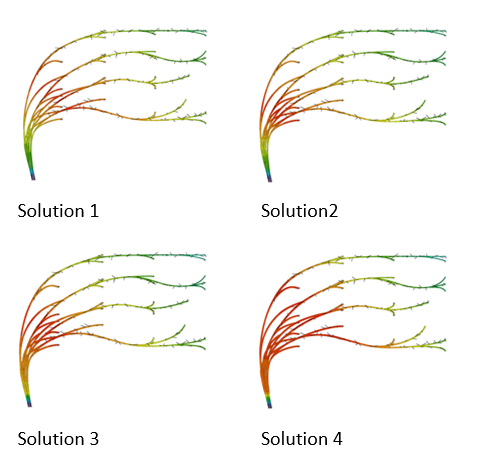
Il limite nello studio delle strutture sembra quindi essere dettato dal tempo materiale necessario per la creazione del modello piuttosto che alla sua risoluzione e validazione. L’industria dell’architettura e ingegneria ha quindi provato a risolvere il problema con la creazione di software per l’automazione dei flussi di lavoro (*workflow*).

Possiamo immaginare un *workflow* come un algoritmo che guida passo passo il progettista nelle fasi di studio e restituzione del progetto. L’obiettivo principale è concentrarsi sulla fase mentale-creativa a discapito della fase di “manodopera”.  
La fase mentale-creativa potrebbe essere impiegata con lo studio di diverse geometrie, materiali e modellazioni più raffinate. Non doversi preoccupare di elaborare grafici o report scritti libera spazio al lato creativo della progettazione. Un immediato vantaggio consiste nell’avere più tempo per studiare un determinato problema e questo è sinonimo di ottizzazione.

Le ottimizzazioni strutturali possono essere alla portata di ogni ingegnere e un’impostazione parametrica di un progetto ha il pregio di rendere flessibile il progettista.

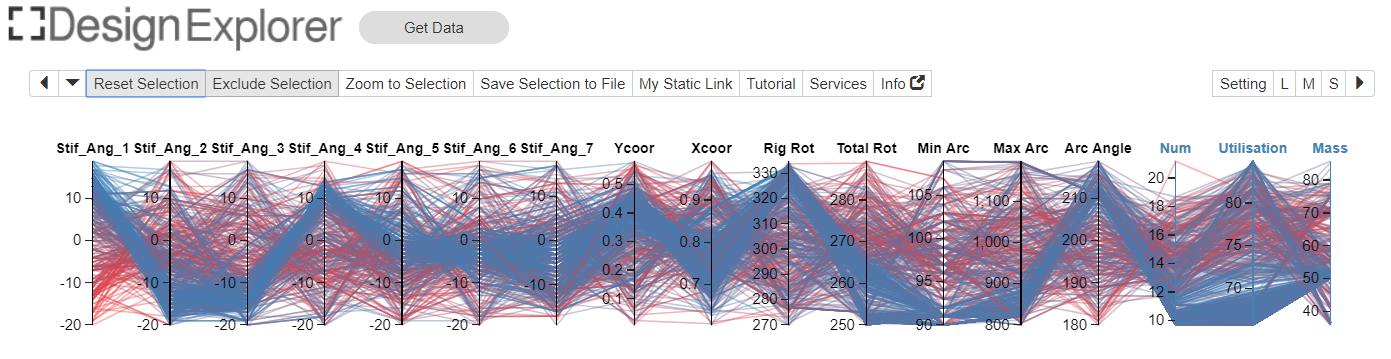
Le ottimizzazioni strutturali hanno solitamente l’obiettivo di massimizzare l’utilizzo del materiale per rendere la struttura più efficiente. La riduzione della massa strutturale potrebbe essere, ad esempio, un dato che il progettista potrebbe voler minimizzare. Quando la funzione da minimizzare è unica, si parla di *single objective optimization.*

Le *Multi-objectivive optimization,* invece, si hanno quando le funzioni da ottimizzare sono molteplici. Minimizzare la massa di una struttura potrebbe portare ad avere un comportamento strutturale meno rigidido e quindi spostamenti maggiori o frequenze proprie della struttura in un campo non favorevole. Le ottimizzazione “multi-oggetto” entrano in gioco là dove non vi è una unica soluzione ottimale.

Gli algoritmi genetici, il cui aggettivo si ispira al principio della selzione naturale ed evoluzione biologica, sono stati teorizzati intorno agli anni 1960 ma il loro utilizzo in campo strutturale è una materia abbastanza moderna. Gli algoritmi genetici rientrano nella branchia dell’intelligenza artificiale e sono di facile utilizzo. Il loro impiego è nei problemi per i quali non si conoscono altri algoritmi efficienti di complessità lineare o polinomiale. L’agoritmo, attraverso procedure di codifica, *cross-over* e mutuazione, tenta di trovare la soluzione ottimale imparando dalle soluzioni precedentemente individuate.

Ottimizzazione meno elegante che sfrutta a pieno la potenza di calcolo è la tecnica *Brute-Force*. Tutte le soluzioni possibili generabili dai parametri di input del problema vengono analizzate e catalogate. Il progettista deve successivamente andare ad individuare la soluzione ottimale.



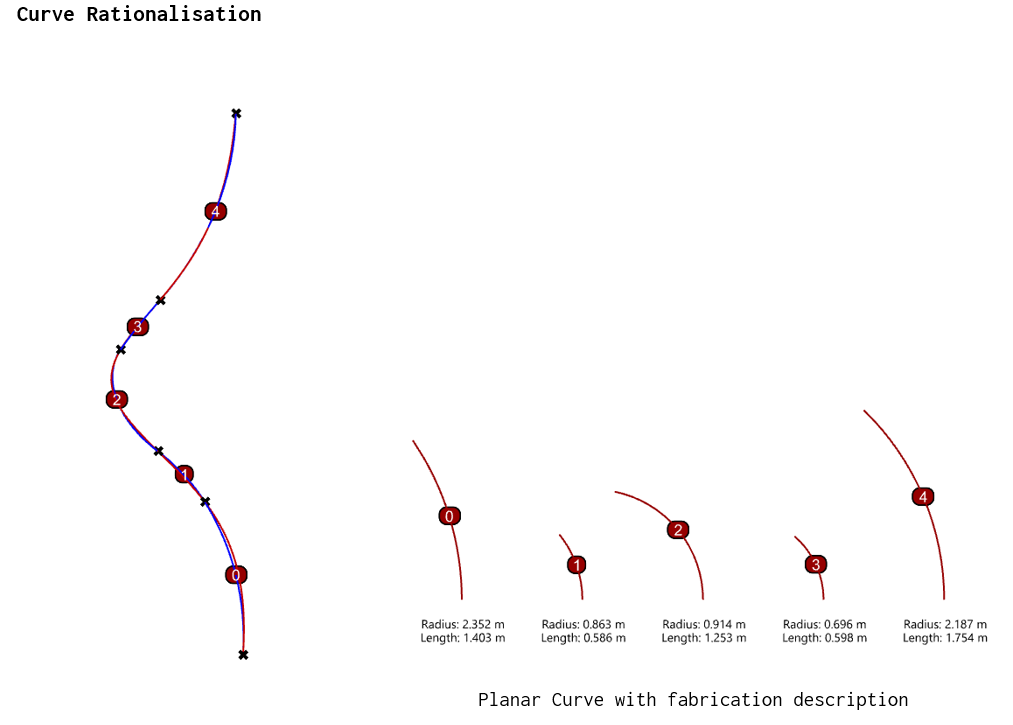
Il numero di soluzioni che definiscono lo spazio di ricerca è funzione esponenziale dei parametri di input. Saper analizzare i dati è fondamentale ( *Data scientist* è stata nominato come il lavoro più *sexy* del XXI secolo ). L’analisi avviene tramite concetti statistici, probabilistici presi in prestito dalla scienza che studia i dati. L’obiettivo è naturalmente quello di saper interpretare in maniera quantitativa il *database* creato dai risultati per poi individuare la soluzione ottimale del nostro problema.

**Fabbricabilità**

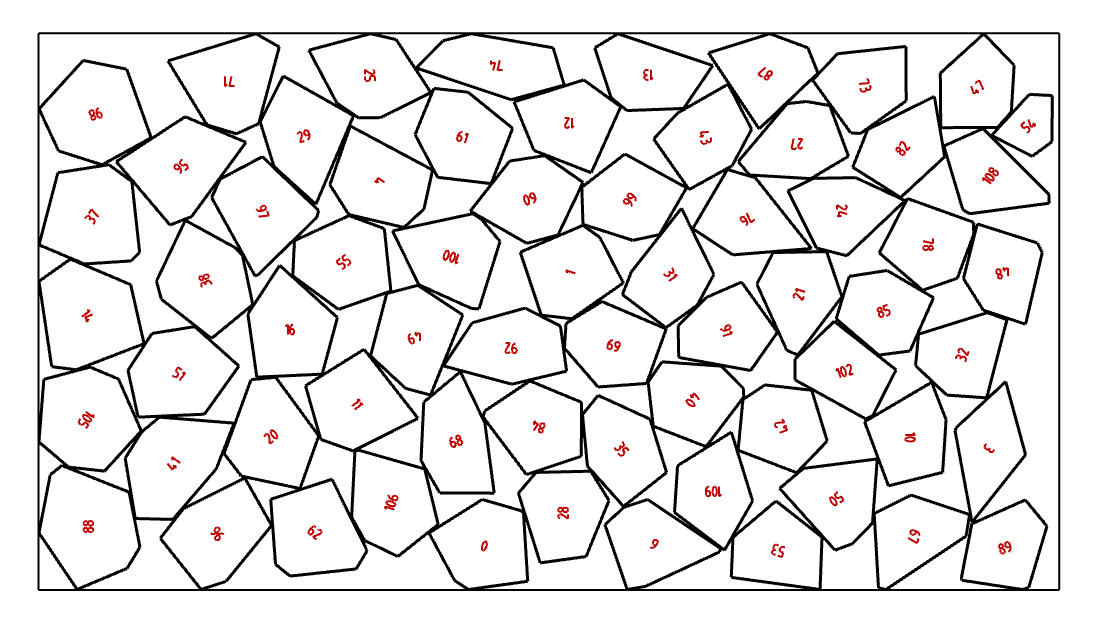
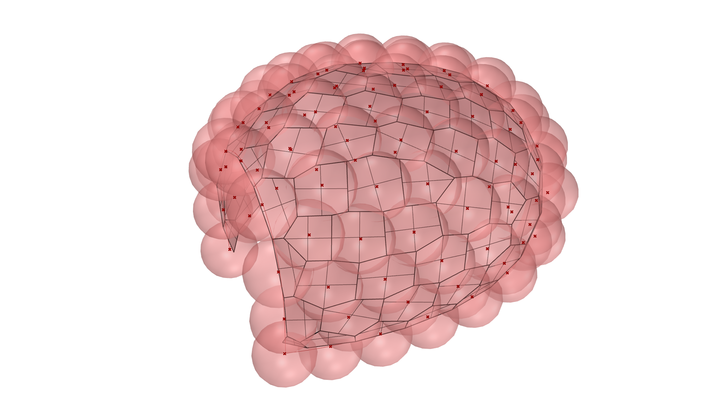
Nell’approccio dell’architettura e dell’ingegneria parametrica, uno dei problemi più frequenti da risolvere sono costruibilità e fabbricabilità. In quanto ingegnerizzazione di strutture complesse o *free form* è molto importante razionalizzare la forma seguendo una logica matematica in modo da ottimizzare il problema e quindi i costi e i tempi di fabbricabilità. La maggior parte dei progetti per i quali è necessario l’utilizzo di un processo computazionale prevede script e metodi di razionalizzazione unici e realizzati per un determinato scopo. Ogni geometria complessa è spesso unica ed è necessario un approccio pensato e individuato esattamente per quella determinata struttura. Affinché una struttura unica possa essere standardizzata con prodotti esistenti in commercio è importante capire come razionalizzare il problema per diminuire i tempi e i costi di realizzazione.

Alcuni dei progetti ai quali abbiamo lavorato prevedevano uno studio di razionalizzazione basato su *k-mean clustering* e *machine learning* che consiste nell’individuare una qualità o una proprietà di un determinato elemento e raggruppare oggetti che abbiano proprietà simili. Concretizzando con un esempio, uno dei progetti più interessanti nel quale abbiamo utilizzato questi tipi di algoritmi prevedeva la suddivisione in blocchi di un’istallazione d’arte di 40 metri di altezza realizzata con aggregati di tubi con forma ripetuta. L’algoritmo in questione è stato utilizzato considerando tubi raggruppati secondo la loro posizione e il valore del momento flettente che comandava la grandezza e lo spessore dei tubi. Blocchi con lo stesso diametro, materiale e spessore sono stati selezionati per essere assemblati in fabbrica ed essere trasportati facilmente in cantiere.

Nell’ingegneria computazionale sono frequenti algoritmi basati interamente su approcci matematici ed applicati in casi reali per semplificare problemi e massimizzare la costruibilità di un oggetto diminuendone i costi. Altri esempi di razionalizzazione che utilizziamo nel nostro lavoro spesso riguardano una modifica della geometria iniziale per favorirne la fabbricabilità. Un metodo di razionalizzazione utilizzato in svariati progetti con elementi tubolari curvi, consiste nel trasformare l’asse centrale dei tubi da una curva *Spline* o curva libera, in un susseguirsi di curve ad arco di cerchio che seguano l’andamento della curva iniziale. In questo modo è possibile utilizzare tubolari con dimensioni standard assemblabili tra loro nei punti di flesso o di raccordo senza dover utilizzare stampi fatti su misura. La stessa logica di standardizzazione può essere usata per ottenere superfici sviluppabili o pannelli di una facciata o di una superficie curva che siano comunque piani.



Queste procedure permettono di concentrarsi sugli aspetti più importanti della progettazione e rimandano al calcolatore quello che è possibile automatizzare. Un esempio concreto riguarda l’elaborazione dei disegni di pannelli in legno necessari per essere tagliati con un CNC in quanto la logica del processo si ripete per tutti i pezzi. Modellarli a mano è sicuramente possibile ma non conveniente mentre utilizzare un metodo computazionale per minimizzare il *workflow* garantisce un lavoro preciso con un risparmio notevole del tempo dell’utente.



**Algoritmi pensati su misura**

La potenza della *computational design* può essere iniduata nella possibilità di creare *workflow* progettati ad *hoc* a seconda del problema che deve essere affrontato. I software presenti sul mercato sono difficilmente flessibili. Un software FEM, per esempio, difficilmente restituisce immagini renderizzate. In questo caso sarà necessario esportare la geometria o addirittura ricrearla in un altro software. La *computational design* non prevede questo passaggio in quanto la metodologia punta a un flusso di lavoro continuo senza interruzioni su quella che puo’ essere visualizzata come “catena di montaggio”. Un esempio al quale diverse software house si sono sensibilizzate è il calcolo strutturale delle connessioni. Il calcolo delle connessioni tra i vari elementi è sempre stato effettuato in un *software* ( o foglio di calcolo ) che difficilmente dialoga in maniera fluida con il solutore principale. Questo problema è oggi evitabile in quanto le Software house hanno colmato il vuoto creando procedure automatiche di calcolo. Questi software sono altamente specializzati nel loro settore di competenza ma sono difficilmente flessibili e l’utente si ritrova molto spesso di fronte a quella che in gergo viene chiamata “black box”.  
Ovviare a questo problema è possibile scrivendo il proprio codice e utilizzare librerie open-source. Il concetto di open source è nato intorno al 1984 con Richard Stallman.

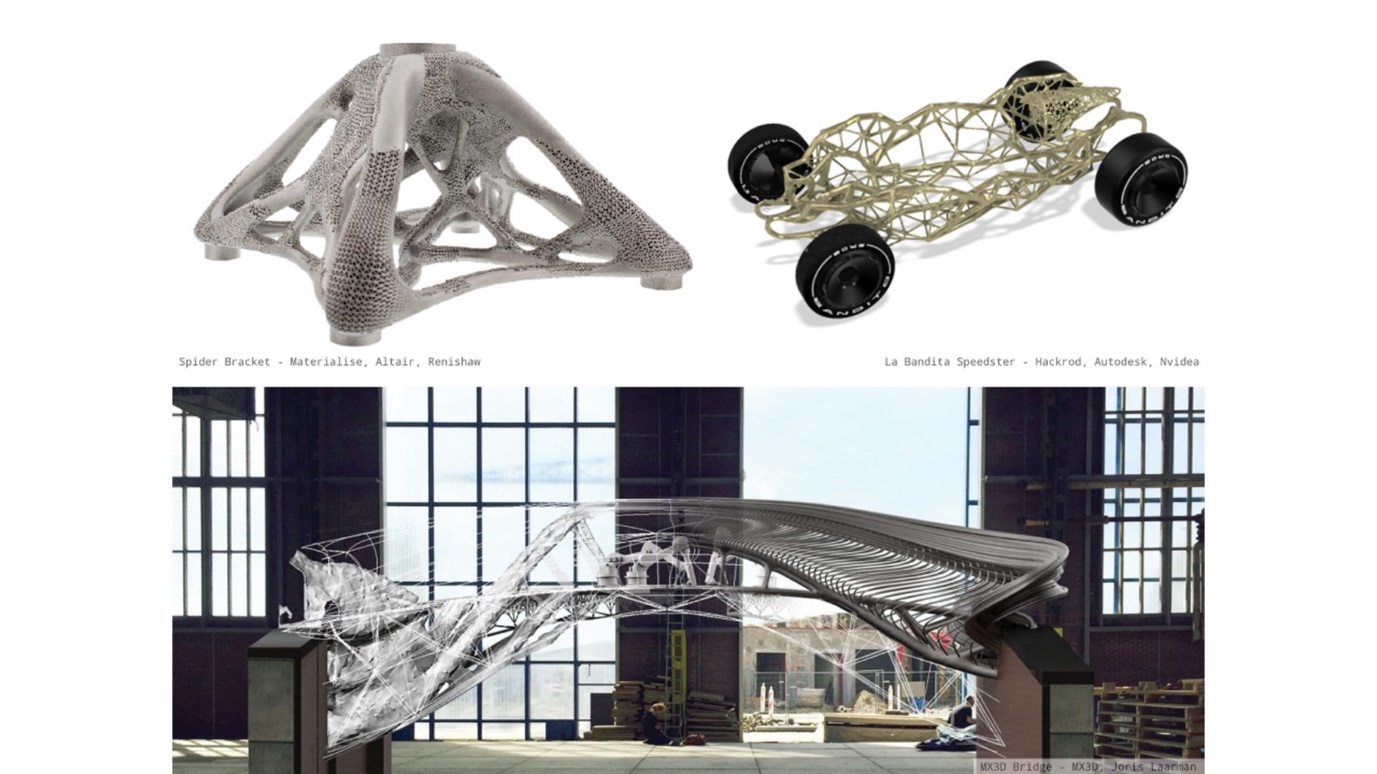
*“Il mio lavoro sul software libero è motivato da un obiettivo idealistico: diffondere libertà e cooperazione. Voglio incoraggiare la diffusione del software libero, rimpiazzando i programmi proprietari che proibiscono la cooperazione, e quindi rendere la nostra società migliore”*

Il web è uno spazio potentissimo. Avete mai provato a chiedere a un *software developer* chi è la persona a cui chiede consiglio quando è in difficoltà? La risposta è solitamente la solita: “Google”.

La *computational design* ha la flessibilità di poter attingere strumenti che non sono nativi dell’ingegneria strutturale. Le *mesh* sono sempre state utilizzate nella modellazione di oggetti tri-dimensionali e il campo scientifico della *computer graphics* sta continuando tutt’ora a creare nuovi algoritmi per la creazione di discretizzazioni ottimizzate. Questo tipo di algoritmi ha un impatto molto forte anche nello studio delle strutture in quanto è frequente la modellazione di lastre/piastre. Saper attingere ai codici sorgenti permette un guadagno di tempo, energie e efficienza.

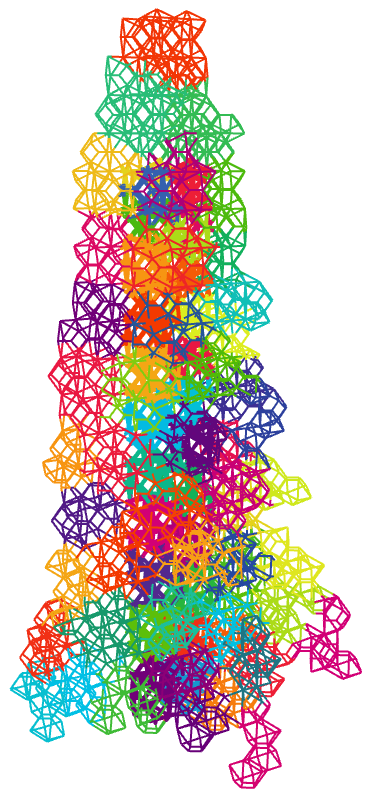
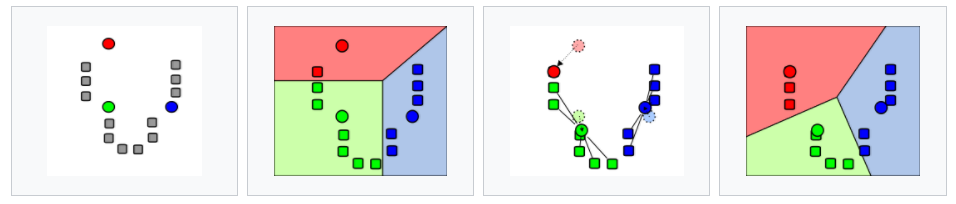
**… e futuro**

Il computazionale sarà sempre più importante nello sviluppo e nella ricerca futura. Già al giorno d’oggi è possibile creare geometrie ottimizzate che minimizzino la struttura e il materiale è utilizzato solo dove è necessario. L’architettura diventa struttura e la struttura diventa la forma. La scelta del materiale con le sue proprietà meccaniche e fisiche avrà sempre più peso nel risultato delle ottimizzazioni strutturali e oggetti sempre più interessanti avranno la possibilità di essere studiati e realizzati. Il computer e i robot della generazione del *Generative Design* assisteranno l’uomo in qualcosa che non sarebbe assolutamente in grado di costruire. Il futuro nell’ingegneria è rappresentato da robot, droni e realtà aumentata che potranno essere programmati in funzione dei limiti fisici dell’uomo ed andranno solo ad integrare la parte fisica della mente dell’ingegnere.



IMMAGINI DA INSERIRE:

Progetti futuri che mostreremo



Marco Pellegrino è un ingegnere civile, programmatore e aspirante eremita che si sente a disagio a scrivere di se stesso in terza persona. Lavora nel settore AEC come specialista di progettazione computazionale, sviluppando strumenti digitali utilizzati nella progettazione e produzione delle costruzioni. Attualmente è membro del gruppo “Computation and Design” di [Format Engineers](http://formatengineers.com/), dove guida la ricerca nell’elaborazione della geometria per l’esplorazione e l’ottimizzazione della forma.

La ricerca personale di Marco si concentra sull’applicazione di metodi numerici e di simulazione all’interno della modellazione di progettazione ingegneristica. Nello specifico, è interessato a come quanto sopra può facilitare la scoperta di nuove soluzioni di design aumentando la capacità del progettista di negoziare grandi serie di criteri di progettazione interdipendenti.