



1. Museum of The Future.

Generatief ontwerp van een constructief diagrid

Het Museum van de Toekomst in Dubai wordt gezien als één van de meest complexe gebouwen ooit ontworpen. Het project leverde nog voor het werd opgeleverd de AEC Excellence Award 2017 op en de algoritmische benadering van het stalen diagrid leidde ertoe dat de BBC sprak over 'het gebouw ontworpen door een algoritme'. Het gebouw is dan ook de belichaming van zijn missie: het exposeren van innovatie en incubatie van vernieuwing.

Eén van de moeilijkste problemen was het constructieve diagrid. Centraal in dit artikel staan de praktische toepassing van generatief ontwerpen, zoals modelintegriteit, de voortdurende wijzigingen in randvoorwaarden, databeheer en gegevensuitwisseling. Ook wordt stilgestaan bij de functionaliteit van Computational Design-strategieën als leermiddel, van ontwerp tot met de documentatie.

M. Henriksen en D.R. Visser MSc CEng RC

Dirk Rinze Visser is Associate Director bij Buro Happold in Den Haag en Berlijn. Martin Henriksen is Senior Computational Design Specialist bij Buro Happold in Kopenhagen.

Museum van de Toekomst in getallen.

onderdeel	omschrijving
bouwplot	15,700 m ² tussen Sheikh Zayed Road en de Jumeriah Emirates Towers
podium	19,200 m ² bovengronds vloeroppervlak, wat overeenkomt met een 16 m hoog en drie verdiepingen tellende betonconstructie; het podiumdak is voorzien van een groendak; ook is er een ondergrondse bouwlaag voor installaties
gebouw	11,300 m ² vloeroppervlak resulterend in een hoogte van 61 m boven het podiumniveau en verdeeld over zeven niveaus
verbindingsbruggen	overspanningen van 212 en 69 m die een verbinding vormen tussen het museum, de Jumeriah Emirates Towers en het naastgelegen metrostation

Het Museum van de Toekomst (afb. 1) is in de eerste plaats een tentoonstellingsruimte voor innovatieve en futuristische concepten, diensten en producten, maar het beschikt ook over een auditorium en ruimtes voor workshops en horeca.

Het project bevindt zich op een A-locatie en in de directe omgeving van het oude centrum van Dubai, iets noordelijk van de Jumeriah Emirates Towers, en langs de belangrijkste snelweg van Dubai (Sheikh Zayed Road), waardoor het project goed *in the picture* ligt. De visie voor dit project is aangereikt door de opdrachtgever, Dubai Future Foundation, als onderdeel van een uniek programma – in samenwerking met de regering van Dubai – om de stad te gebruiken als een proeftuin voor het oplossen van huidige en toekomstige wereldwijde vraagstukken.

Het architectonische ontwerp is van Shaun Killa (Killa Design) als het winnende concept van een internationale ontwerpwedstrijd in 2014. North25 is als Development Manager ingeschakeld, waarin Buro Happold de rol van hoofdconstructeur op zich heeft genomen. Verder is MSW de Project Manager en AECOM de kostenadviseur.

Het architectonisch concept representeert een toekomstvisie die in een vorm gevat is waarvoor de premier van het land de inspiratie bood. Het fysieke gebouw (met vloeren en ruimtes) laat de nabije 'toekomst' zien, zoals we die nu kennen. De opening of 'void' – het gebouw heeft de vorm van een open torus – daarentegen vertegenwoordigt datgene wat we niet weten. Het staat symbool voor mensen die naar het onbekende zoeken, zullen innoveren en blijven ontdekken, wat vervolgens naar een betere toekomst zal leiden (afb. 2).

De gevel bestaat uit 890 roestvast stalen en glasvezel-gefuseerde panelen, naadloos gemonteerd, waardoor het als perfect reflecterend metalen oppervlak met Arabische kalligrafie als beeldmerk de toekomst in zal gaan. De kalligrafie zelf fungeert als raamopening en bevat citaten uit de poëzie, geënt op de toekomst, van Sjeik Mohammed bin Rashid Al Maktoum.

Het probleem van het diagrid

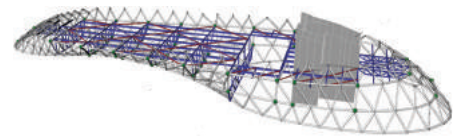
De grootste inspanning bij het diagrid is in eerste instantie het vinden van een adequate geometrie die de vorm volgt van de façade. Niet alleen moet het diagrid in de bepalende gevel passen, het doel is uiteraard ook om een ontwerp tot stand te brengen die de optimale balans vormt tussen de belangrijkste parameters, zoals de grootte en lengte van de constructiedelen, het aantal verbindingen, het gemak en de snelheid van fabricage, en uiteindelijk de bouwkosten. Vanzelfsprekend is het primaire doel van het diagrid niet alleen de gevel te ondersteunen, maar ook de binnenconstructie, zoals liggers en vloeren. Dit betekent dat het diagrid moet worden afgestemd op het interne bouwkundige systeem (afb. 3).

Tevens moet een esthetische overweging plaatsvinden, aangezien het diagrid gedeeltematig zichtbaar zal zijn door de raamopeningen, om het diagrid ook een zekere regelmaat te geven (afb. 4).

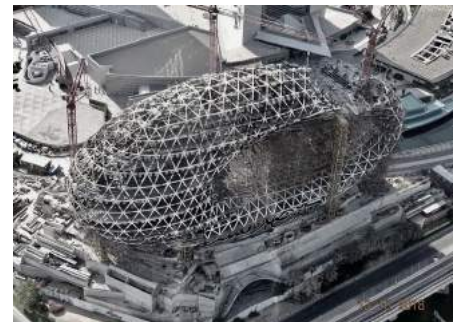
Bovendien moet het ontwerpproces met snelle iteraties plaatsvinden, aangezien de architectuur van het gebouw nog steeds in ontwikkeling is en het diagrid flexibel genoeg moet zijn om deze veranderingen op te kunnen vangen. Denk aan de binnenconstructie, de gevelpanelen, de installaties enzovoort, die ook nog niet volledig gedefinieerd zijn.



2. Conceptschets architect.



3. Voorbeeld van binnenconstructie die door het diagrid moet worden ondersteund.

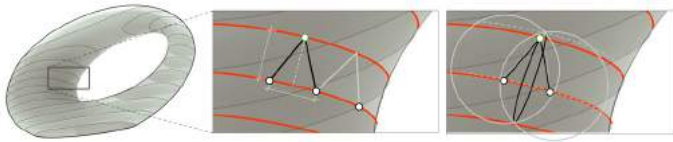


4. Diagrid in aanbouw.

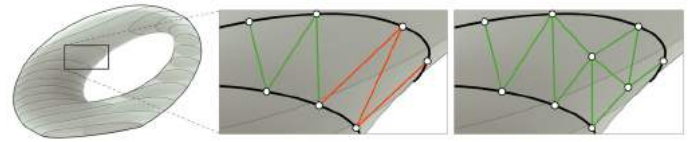
Het algoritme

Allereerst is getracht om het diagrid te ontwerpen op een meer traditionele manier, waaronder een op UV-gebaseerde topologie (projecteren van 2D op een 3D-vorm), een raster gebaseerd op basisspanningen, en een standaard *meshing*-algoritme met Kangaroo (een plug-in voor Grasshopper). Geen van deze algoritmen bleek geschikt te zijn voor deze specifieke taak, waardoor er een apart gridgeneratie-proces ontwikkeld is: een projectspecifiek algoritme.

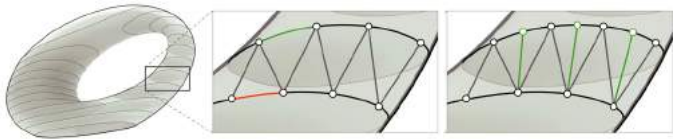
Het proces begint met de opzet van een serie driehoeken in het diagrid: vanuit één startpunt wordt het tweede punt van de driehoek op hetzelfde horizontale niveau op een bepaalde afstand verkregen. Het toppunt van de driehoek wordt gevonden door het vinden van het snijpunt van twee bollen op het referentievlak. Zo ontstaat een gelijkzijdige driehoek op het referentievlak.



5. Bepalen van de individuele driehoeken van het diagrid.



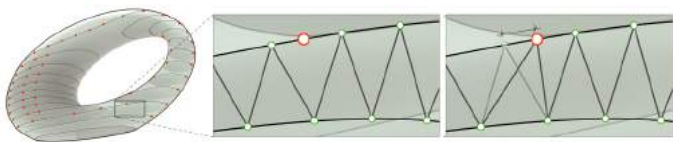
6. Voorkomen van te lange driehoeken.



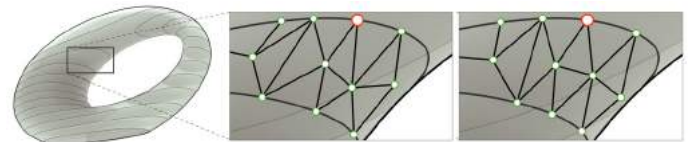
7. Driehoeken splitsen met te ver uit elkaar liggende toppunten.



8. Samenvoegen toppunten van meerdere driehoeken.



9. Nieuw toppunt valt samen met gewenste ondersteuning van de binnenconstructie.



10. Relaxatie-algoritme om regelmaat in elementlengte te verkrijgen.

De volgende driehoek wordt gecreëerd door dit proces te herhalen met het tweede punt van de vorige driehoek als startpunt voor de huidige driehoek (afb. 5). De toppunten van twee aangrenzende driehoeken worden met elkaar verbonden door een derde driehoek. Dit proces wordt herhaald totdat één horizontaal niveau volledig is gevuld met deze driehoeken. De bovenste punten van de driehoeken van het reeds voltooide niveau vormen het startpunt voor het volgende niveau van driehoeken.

Het is in deze fase niet nodig om de elementen van het diagrid te definiëren, aangezien de output van de gegenereerde driehoeken een *mesh* is. Een *mesh* kan gemakkelijker informatie aanreiken, zoals knooppuntentoe (het aantal aangesloten elementen per knoop), dat functioneel is bij het ontwerpen van de verbindingen in het (uiteindelijke) diagrid. Door de horizontale segmentatie van het oppervlak varieert de afstand langs het oppervlak van het ene naar het andere niveau aanzienlijk. Omdat de toppunten van de driehoeken ontstaan uit de onderste punten van een niveau hoger gelegen driehoeken, ontstaan er driehoeken tussen de horizontale niveaus waar de lengte van de lijnstukken te lang wordt. Om dit te voorkomen wordt er een maximale afstand bepaald voor nieuwe

toppunten resulterend in een aanvullend deelniveau (afb. 6).

Wanneer twee nieuwe toppunten te ver uit elkaar liggen, wordt de naar beneden wijzende driehoek ertussen in tweeën gedeeld, waardoor een extra driehoek ontstaat met slechts één extra knooppunt (afb. 7).

Op een vergelijkbare manier worden twee toppunten, die te dicht bij elkaar liggen, samengevoegd tot één toppunt, waardoor er één driehoek wordt geëlimineerd (afb. 8). Op sommige locaties is het noodzakelijk dat de knooppunten aansluiten op een specifiek punt om de binnenconstructie te ondersteunen. Dit wordt bereikt met een *distance-based attractor* – een voorziening in Grasshopper die werkt als een virtueel magneet om punten hetzij aan te trekken, hetzij af te stoten –, zodat nieuwe toppunten, die binnen een bepaalde afstand van het steunpunt zouden komen, juist naar het steunpunt verplaatst worden (afb. 9).

Hoewel deze aanpak voor een goede topologie zorgt, garandeert het nog geen goede algemene geometrie. Door knooppunten te koppelen aan opgelegde beperkingen, zoals het splitsen van driehoeken en het samenvoegen van toppunten, ontstaat er een diagrid met veel verschillende elementlengten. Om dit te compenseren wordt het diagrid door

een relaxatie-algoritme – een optimalisatie-algoritme om een lineaire doelfunctie te bereiken – gestuurd waarbij een minimale verplaatsing van knooppunten behaald moet worden (afb. 10).

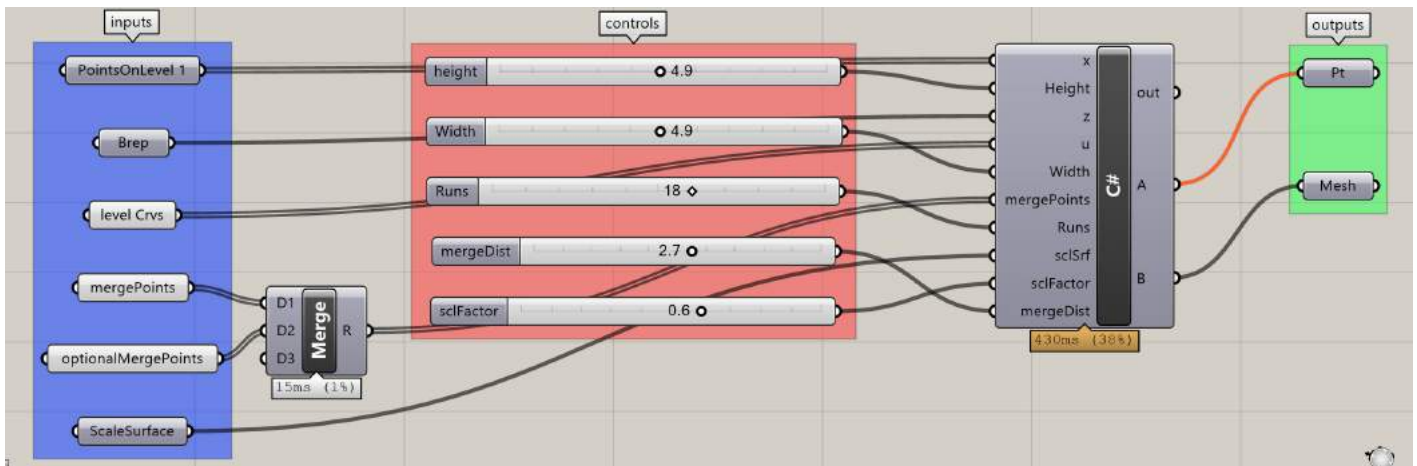
De individuele stappen zijn flexibel genoeg om handmatige interventies toe te kunnen laten. De relaxatiestap (afb. 10) laat zien dat de diagrid-topologie kan worden aangepast zonder regelmaat te verliezen.

Handmatige interventies

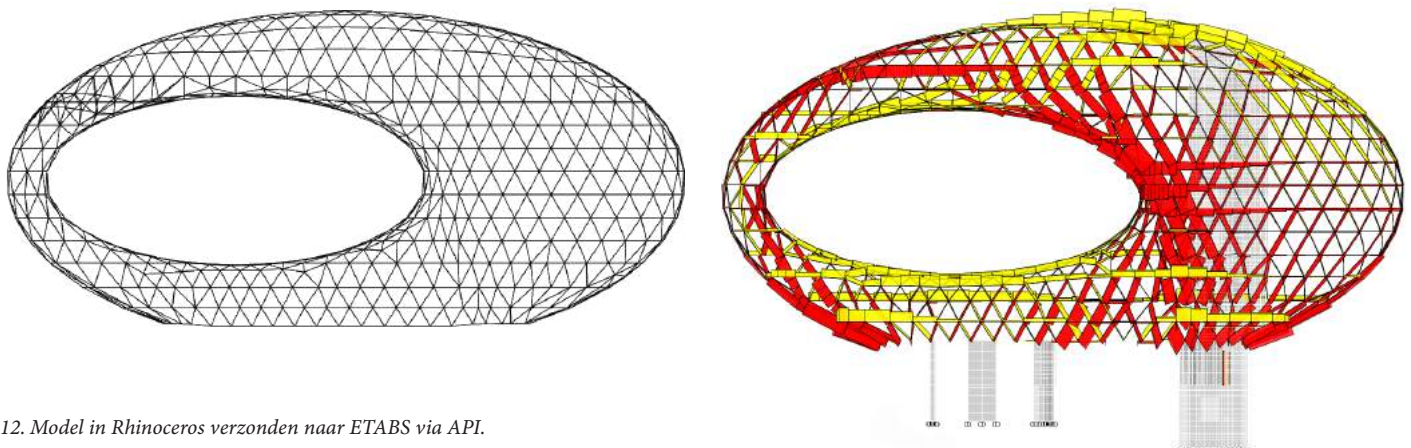
Elke stap binnen het algoritme dat de driehoeksmeting uitvoert, begint en eindigt met een rij punten. Dit betekent dat het kan worden gebruikt op individuele niveaus met verschillende instellingen of met (lokale) handmatige interventies. Het proces dat is ingesteld voor het genereren van het diagrid is geen ‘alles-of-niets’ benadering, oftewel het levert geen compleet diagrid op, maar slechts een deel.

Deze flexibiliteit is van belang, omdat het lokale, handmatige aanpassingen kan toelaten en waardoor het mogelijk is om kleine wijzigingen aan te brengen en sneller te itereren. Natuurlijk is de snelheid van de iteratie ook afhankelijk van de integratie van modelleren en analysesoftware.

Het diagrid is gegenereerd met een een-



11. Definiëren in Grasshopper.



12. Model in Rhinoceros verzonden naar ETABS via API.

voudige Grasshopper-definitie waarbij één C#-component – programmeertaal betreffende hoe het eerste ‘bouwsteentje’ is opgebouwd – het meeste werk doet. Deze Grasshopper-definitie fungeert louter als interface om parameters in te stellen en de variabelen van het algoritme te controleren (afb. 11). Het C#-script zelf is eenvoudig omdat het simpelweg een methode aanroept in een .dll-bestand (Dynamic Link Library). Dit betekent dat bijna al het codeerwerk in het pakket Visual Studio kan worden gedaan, hetgeen een groot voordeel is door een uitgebreider pakket productiviteitsgereedschappen (ook wel tools), die niet beschikbaar zijn in Grasshopper. *Auto-complete*, *method-suggestions*, *code highlighting* en *live debugging* verhogen de productiviteit.

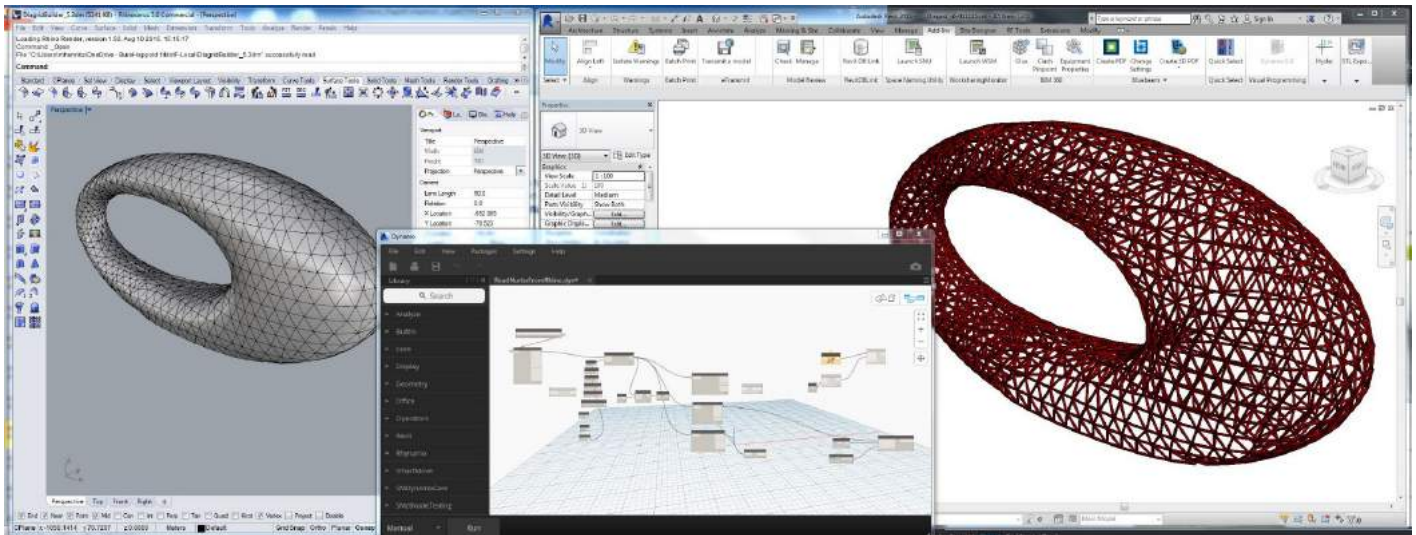
Gegevensbeheer: één object op drie platforms

Alle elementen die het diagrid vormen moeten worden gemodelleerd, geanalyseerd en gedocumenteerd. Dit gebeurt respectievelijk in Rhinoceros 3D, ETABS en Revit, wat impliceert dat er behoefte is aan een zekere mate van integratie tussen deze softwarepakketten, en wel zodanig dat niet elk element in elk softwarepakket en voor elke iteratie opnieuw gemodelleerd hoeft te worden. Het integreren van deze omgevingen, zodat de diagrid-elementen eenmalig gemodelleerd en overal gedeeld kunnen worden, betekent niet alleen dat er tijd bespaard kan worden, maar ook de kans op fouten reduceert. Interoperabiliteit van de model-analyse gebeurt door verbinding te maken met ETABS vanuit Grasshopper via de ETABS API (afb.

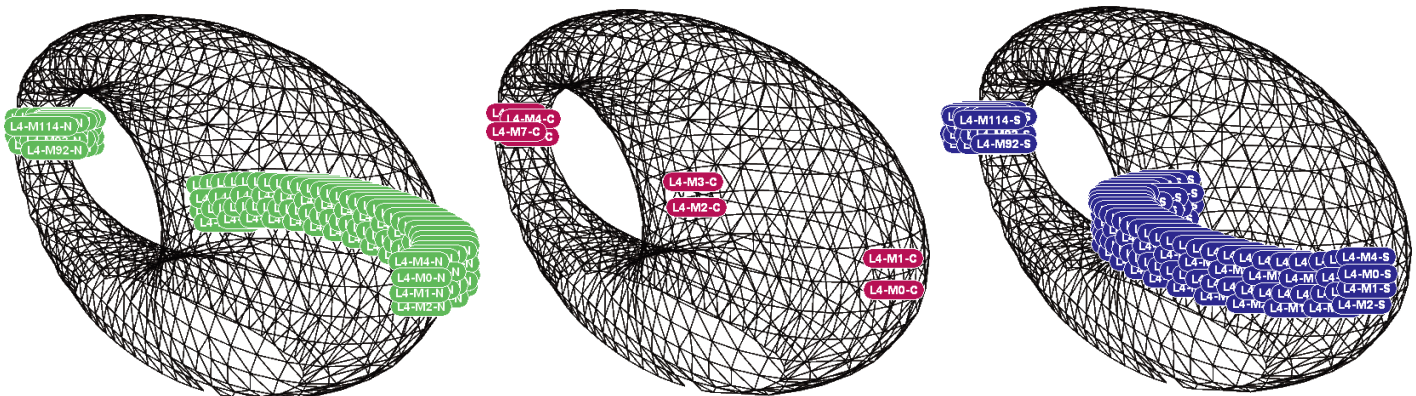
12). Methoden op de ETABS API worden aangeroepen via een .dll-bestand waaraan gerefereerd wordt in een Grasshopper C#-script component. De methoden worden gemaakt in hetzelfde .dll-bestand dat wordt gebruikt om de geometrie van het diagrid te genereren. De verbinding met Grasshopper is snel; letterlijk één druk op de knop. Dit is nodig omdat er veel model-analyse iteraties zijn, veel meer dan er model-document iteraties zijn. De koppeling met Revit verloopt via Excel en met Dynamo als *interface* (afb. 13). Ook kan in het Excel-bestand eenvoudig gegevens uit verschillende bronnen worden toegevoegd.

What's in a name?

Een consistente benaming voor alle elementen en knooppunten helpt bij de documenta-



13. Koppeling van Rhino en Revit met Dynamo voor inlezen gegevens uit Excel.



14. Elementlabels op niveau 4 aan respectievelijk de noord-, midden- en zuidzijde.

tie en het delen van gegevens. Zowel elementen als knooppunten worden volgens hetzelfde patroon benoemd: <Niveau nummer>-<Type><Nummer>-<O-riëntatie van het gebouw>.

Bijvoorbeeld L1-N03-S: knooppunt (N) op niveau 1 (L1) aan de zuidkant (S). Het lage nummer (03) geeft aan dat het knooppunt zich richting het oosten bevindt (afb. 14). Het benoemen van elementen is natuurlijk belangrijk bij verbale communicatie, zeker bij externe partijen. Een praktische naamgeving is belangrijk voor de documentatie, maar ook voor het kunnen koppelen van (constructieve) gegevens. Alle 2469 elementen van het stalen diagrid beschikken over specifieke constructie-

gegevens – vanuit ETABS – en geometrische gegevens zoals lengtes en buigstralen – vanuit Rhinoceros/Grasshopper (afb. 15 en 16). Met een duidelijke naamgeving per element is het eenvoudig om gegevens te combineren in Excel, en met het in de naam reeds gespecificeerde vloerniveau is het gemakkelijk om een overzicht te krijgen met alleen de gegevens die nodig zijn voor het betreffende niveau.

Toevoegen gegevens aan de workflow

De windbelasting wordt aangegeven door een externe consultant, BMT Fluid Dynamics, vanuit een windtunnelonderzoek (afb. 17). De resultaten zijn gegeven in een tabel met in de naam reeds piekdrukwaarden voor twaalf condities (terugkeertijd, % demping, wind-

richting, etc.) op locaties die overeenkomen met de bijbehorende knooppunten. Het constructieve ontwerp was nog volop in ontwikkeling tijdens het windtunnelonderzoek, waardoor de knooppuntlocaties van het diagrid al veranderd waren bij het ontvangen van de resultaten vanuit het windtunnelonderzoek. Er was een werkmethode nodig om de ontvangen windkrachten toe te passen in het aangepaste diagrid zonder de noodzaak voor een nieuwe windtunneltest. De fysieke vorm van het gebouw veranderde niet, maar wel de knooppuntlocaties aan het oppervlak. Het was mogelijk om de windkrachten op ieder willekeurig punt te berekenen door te kijken naar de drie dichtstbijzijnde punten zoals gebruikt ten tijde van het windtunnelonderzoek.

Member name	from node	to node	Member size	P (kN)	V2 (kN)	V3 (kN)	M2 (kNm)	M3 (kNm)	T (kNm)	Member length	Steel grade
L1-M0-N	L2-N2-N	L1-N0-C	CHS 457x25	-540/-3250	50/-10	50/10	170/-200	230/-60	70/20	8,768	S355
L1-M1-N	L2-N2-N	L1-N0-N	CHS 457x25	-1080/-4400	40/-10	20/10	40/-140	180/-10	10/-10	8,504	S355
L1-M2-N	L2-N2-N	L1-N1-N	CHS 457x25	-20/-1590	20/-30	10/-10	10/-90	60/-30	-20/-80	10,540	S355
L1-M3-N	L2-N4-N	L1-N1-N	CHS 457x40	-2500/-8060	100/20	-30/-100	440/-350	360/-280	-50/-200	8,626	S355
L1-M4-N	L1-N0-C	L1-N0-N	CHS 457x40	6590/2220	370/100	150/-50	230/-240	440/-650	50/-20	3,073	S355
L1-M5-N	L2-N8-N	L1-N1-N	CHS 457x16	-30/-1210	390/-120	50/-30	170/-20	430/-530	10/-50	5,388	S355

15. Voorbeeld schema met staafkrachten.

Node number	Coordinates			Node connected members						
	x	y	z	1	2	3	4	5	6	7
L1-N0-N	-1053,567871	116,722229	22,571501	L1-M1-N	L1-M4-N	L1-M6-N	L1-M10-N			
L1-N1-N	-1051,695679	119,619423	22,571501	L1-M2-N	L1-M3-N	L1-M5-N	L1-M6-N	L1-M8-N	L1-M11-N	L1-M12-N
L1-N2-N	-1048,558716	121,676689	22,571501	L1-M7-N	L1-M8-N	L1-M9-N	L1-M13-N	L1-M14-N	L1-M15-N	L1-M17-N
L1-N3-N	-1044,989136	116,582954	19,432013	L1-M10-N	L1-M11-N					
L1-N4-N	-1043,839478	118,567078	19,432695	L1-M12-N	L1-M14-N					
L1-N5-N	-1043,592041	123,360008	22,571501	L1-M15-N	L1-M16-N	L1-M18-N	L1-M19-N	L1-M20-N	L1-M21-N	

16. Voorbeeld schema met de locatie van knooppunten.

Vanuit de drukwaarden p_A , p_B , p_C , respectievelijk op de punten A, B en C, werd de drukwaarde p_X op punt X bepaald met een geometrisch gewogen gemiddelde:

$$p_X = \frac{\frac{1}{d_A} \cdot p_A + \frac{1}{d_B} \cdot p_B + \frac{1}{d_C} \cdot p_C}{d_A + d_B + d_C}$$

Gezien de geschatte druk p_X en de normaalvector n_X in punt X (vector loodrecht op het oppervlak met een magnitude van 1) worden de knooppunten berekend door:

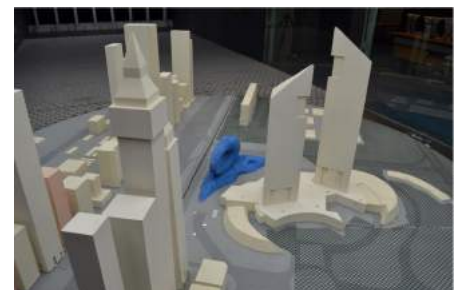
$$-n_X \cdot p_X \cdot t_A$$

Daarin is: t_A het invloedsgebied van het knooppunt is (afb. 18).

Het invloedsgebied van een knooppunt is het oppervlak dat dichtbij het beschouwde knooppunt ligt dan bij ieder ander knooppunt. Het berekenen van dit (vlakke) oppervlak is eenvoudig te bepalen met 2D voronoi-tesselatie. Voor het oppervlak van dit gebouw wordt een 3D voronoi-tesselatie



17. Schaalmodel voor windtunneltests.

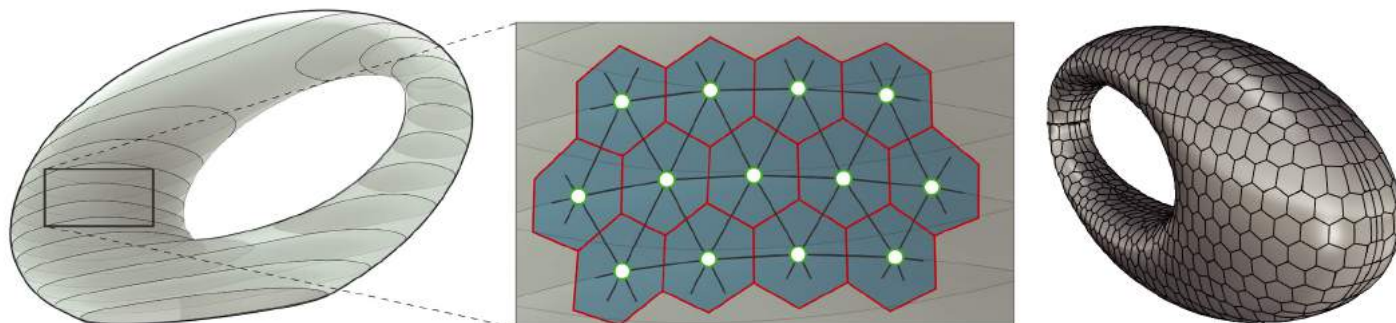


gebruikt om het relevante oppervlak voor elk punt te verkrijgen. Dit is echter slechts een benadering van de werkelijkheid, gezien de standaard 3D voronoi-tesselatie de lineaire afstand tussen de punten gebruikt en niet de afstand langs het (gekrumde) oppervlak (afb. 19). Dit verlies aan nauwkeurigheid is acceptabel, aangezien er in het algemeen minder dan 2% verschil bleek te zitten tussen de basisreacties in het constructiemodel van het oorspronkelijke model, en de modellen die met deze methode zijn gemaakt.

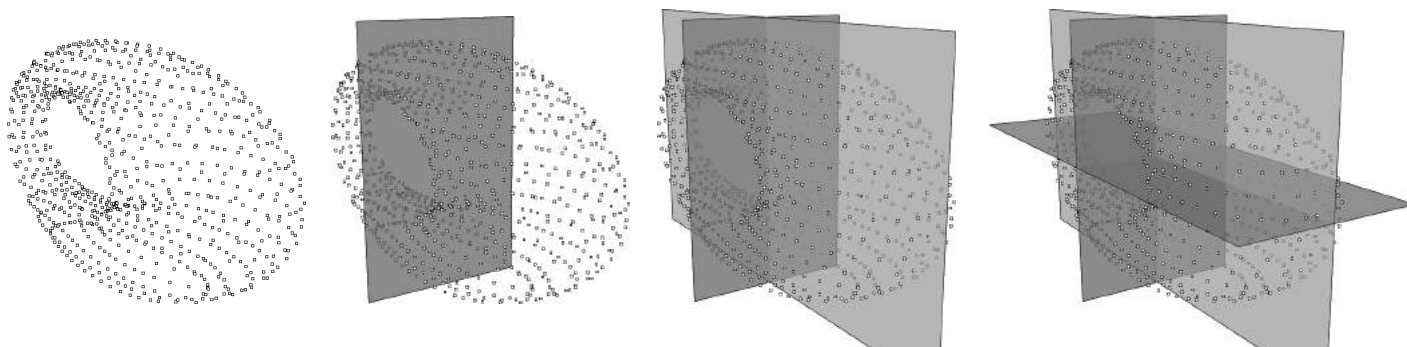
Zoeken multidimensionale gegevens

Om de gegevens vanuit de windtunneltests opnieuw te kunnen gebruiken met de reeds beschreven geometrisch gewogen gemiddelde

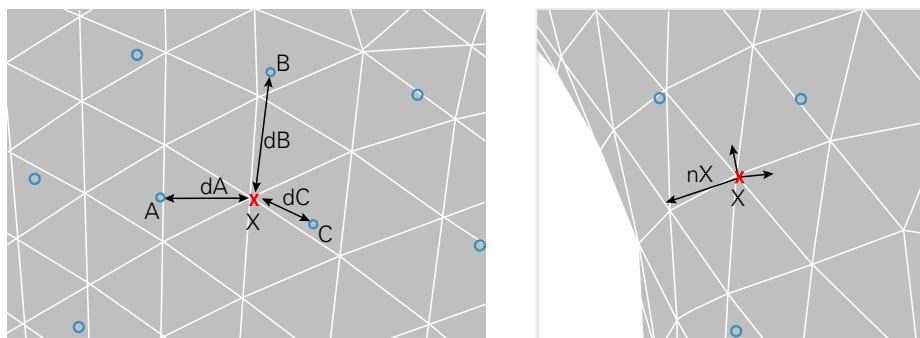
methode, moet elk knooppunt van het stalen diagrid de drie dichtstbijzijnde punten van alle punten in de windtunnelgegevens vinden. Dat is een verzameling van 1174 punten die gezocht moeten worden voor elk van de 841 knooppunten in het diagrid. Samenvattend betekent dit iets minder dan 1 miljoen punt-tot-punt afstandsrekeningen voor één van de drie dichtstbijzijnde punten. Het aantal berekeningen kan aanzienlijk verminderd worden met een datastructuur-methode die bekend staat als een k-dimensionale boom, of k-d-boom. Een k-d-boom is een datastructuur-methode die een ruimte segmenteert door herhaaldelijk een punt in de dataset te kiezen om die betreffende partitie vervolgens in tweeën te splitsen en



20. Segmenteren 3D-knooppunten in X-, Y- en Z-afmetingen.



21. De k-d-boom in twee dimensies, en het zoekpunt (paars).



19. Het invloedsgebied voor elk knooppunt wordt berekend met 3D Voronoi-tessellatie.

dit voor elke dimensie in een wisselende volgorde te doen (afb. 20).

Door de dataset op te splitsen in kleinere partities, is het mogelijk de data efficiënter te doorzoeken. Dit proces kan het beste worden geïllustreerd met behulp van 2D-punten: beschouw een gegeven set van X,Y-coördinaten – (7,2), (5,4), (9,6), (2,3), (4,7), (8,1) – en het zoekpunt: (2,7). De boom kan dan geconstrueerd worden door eerst te splitsen op $X = 7$, dan door $Y = 4$ aan de ene kant en $Y = 6$ aan de andere kant van de boom. Daarna wordt dit herhaald voor de rest van de punten in de verzameling (afb. 21).

Vanaf de wortel van de boom wordt de afstand tot de volgende knooppunten in de boom berekend en vervolgens vergeleken. Het zoekpunt moet links of rechts van de partitie liggen en die keuze wordt bepaald door het dichtst bijliggende punt, (5,4) of (9,6) (afb. 22). Aangezien (5,4) dichterbij is dan (9,6), kan die tak van de boom worden uitgesloten van de zoekopdracht. Dit wordt herhaald voor het volgende niveau in de boom totdat de laatste tak van de boom kan worden uitgesloten van de zoekopdracht. Het punt (4,7) wordt getoond als het dichtstbijzijnde punt in deze

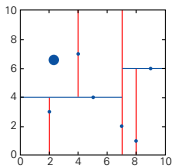
zeer kleine set van 2D-punten (afb. 23). Hoewel een k-d-boom een slimme datastructuur-methode is voor het zoeken naar de dichtstbijzijnde punten, wordt de methode steeds nuttiger als er grotere datasets gebruikt moeten worden. De verwachte zoektijd voor k dichtstbijzijnde punten is $O(k \cdot \log(n))$, terwijl dit met de brute kracht-methode een lineaire $O(k \cdot n)$ is, waarbij n het aantal te zoeken punten is.

In het geval van het stalen diagrid zijn er 1174 punten in de zoekruimte waar 841 knooppunten de drie dichtstbijzijnde punten moeten vinden. Dat betekent dat de brute-kracht methode $1174 \cdot 841 \cdot 3 = 2.962.002$ bewerkingen moet uitvoeren – in dit geval zijn dat afstandsberoeeningen. Met de k-d-boom methode is het aantal verwachte bewerkingen ongeveer 7.745.

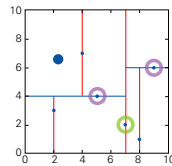
Dit is een aanzienlijke vermindering, maar moet tevens gerelateerd worden aan de benodigde tijd per operatie en de tijd die wordt geïnvesteerd in de uitvoering van de bewerking. Als een computer 10.000 afstandsberoeeningen per seconde kan maken, dan neemt de brute-kracht methode ongeveer 5 minuten in beslag, terwijl dezelfde bewerking met de k-d-

Projectgegevens

Opdracht Dubai Future Foundation (VAE) • Development Manager North 25, Dubai • Architectuur Killa Design, Dubai • Constructief ontwerp, brandveiligheid, installatie (etc.) Buro Happold (div. locaties) • Staalconstructie Eversendai, Sharjah (VAE) • Stalen buigwerk Kersten Middle East, Ras Al Khaimah (VAE)

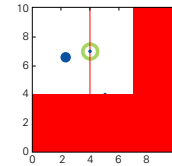
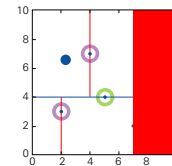


22. Eerste partitie in de k-d-boom.



23. Elimineren van zones om het dichtstbijzijnde punt in de dataset te vinden.

24. Koppelen van software: ad hoc versus gemeenschappelijk kader (BHoM).



boom methode ongeveer 8 seconden duurt. Als het zoeken naar de dichtstbijzijnde punten slechts een paar keer wordt gebruikt tijdens een project, dan is het misschien niet de moeite waard om de tijd te besteden aan het implementeren van een efficiëntere zoekmethode. Alleen als het zoeken naar het dichtstbijzijnde punt een groot aantal keren toegepast moet worden, zal de k-d-boom methode een positief rendement op de tijdsinvestering hebben.

Meer dan één project

Met een rekenkundige aanpak bij het ontwerpen van een project, zoals het Museum van de Toekomst, moet het rendement van de investering in de ontwikkeling van deze processen worden beschouwd in termen van het opnieuw doen versus opnieuw toepassen – wat kan worden gebruikt nadat het project is voltooid en wat kan hergebruikt worden uit eerdere projecten?

Alle projecten zijn leerzaam en er zijn altijd lessen te leren. Er zijn echter wel gradaties in hoe nuttig een geleerde les is als het gaat om de mate waarin deze in de toekomst hergebruikt kan worden. Is het een persoonlijke ervaring die persoonlijk blijft of wordt er gecommuniceerd en gedeeld met anderen? Bij deze vragen wordt alleen gekeken naar het nut in kwantitatieve termen: hoe vaak wordt een oplossing gebruikt. Het nut van een oplossing vereist echter één essentiële eigenschap om relevant te zijn buiten één specifiek project; het moet generiek zijn. Als een oplossing te projectspecifiek is, is het moeilijk om deze elders toe te passen. Soms kan een specifieke oplossing worden

opgesplitst in onderdelen en zullen sommige onderdelen algemeen genoeg zijn om elders toe te kunnen passen. Hoe meer korrelig/modulair de code is, hoe groter de kans is dat deze nuttig is voor andere projecten. Het nut van de oplossing vereist tevens een andere essentiële kwaliteit: documentatie. Als de oplossing niet gedocumenteerd is, blijft het een persoonlijke ervaring. Wanneer de oplossing als code is geschreven, is documentatie niet (noodzakelijkerwijs) nodig om de oplossing te (her)gebruiken, hoewel het altijd nuttig is om de toegepaste oplossing te begrijpen. Oplossingen geschreven als code biedt ook het voordeel van automatisering en code ontwikkeld voor het ene project hoeft niet opnieuw te worden ontwikkeld om te worden gebruikt op een ander project. Het is echt een kwestie van opnieuw doen versus opnieuw toepassen.

De beste manier om een code opnieuw toe te kunnen passen, is om deze niet in de eerste plaats als projectspecifieke functionaliteit te creëren, maar om het in een gedeeld raamwerk te maken, en waarbij de functionaliteit van de ontwikkelingen simpelweg de functionaliteit van het gedeelde raamwerk uitbreidt. Het Museum van de Toekomst vereiste specifieke functionaliteit die geïmplementeerd werd met ETABS API, zoals het creëren en uitlezen van staafelementen, het uitlezen van staafkrachten, het bepalen van puntlasten, et cetera. Deze functionaliteit is nuttig voor andere projecten die gebruik maken van ETABS als rekensoftware, en is ontwikkeld als onderdeel van een gedeeld raamwerk dat de naam BHoM (Buildings and Habitats object Model) heeft gekregen.

Meer dan één organisatie

Met een gedeeld raamwerk, dat het mogelijk maakt om projectspecifieke oplossingen te ontwikkelen als uitbreiding op dit raamwerk, worden projectoplossingen raamwerkoplossingen. Aangezien BHoM open source is wordt de oplossing beschikbaar voor iedereen die het framework gebruikt en dit ongeacht tot welke organisatie je behoort.

Het BHoM (afb. 24) is een samenwerkingsproject voor computationele ontwikkeling; een collectieve inspanning om code te delen en de gegevens, die we gebruiken om te ontwerpen, te standaardiseren. Onafhankelijk van de ingenieursdiscipline.

Het is overigens geen poging om exacte processen te standaardiseren of om de software die we gebruiken te standaardiseren. Echter, bij het standaardiseren van data biedt het grote mogelijkheden voor efficiëntie en samenwerking.

Het gehele BHoM-project (w) maakt gebruik van een open source model voor projectarchitectuur, co-creatie en planning. Iedereen kan de broncode verkennen en kan er tevens een bijdrage aan leveren (<https://github.com/BHoM>), ongeacht zijn of haar organisatie. Het delen en bouwen van code in deze open source-benadering betekent dat uiteenlopende kennis, ervaring en expertise kunnen worden gebundeld met het oog op een gemeenschappelijk doel – een beter ontwerp. En juist dat betere ontwerp is zichtbaar gemaakt met het Museum van de Toekomst. •