

Best practices voor parametrisch ontwerpen

Taakgroep Digitale Ontwikkelingen

Versie d.d. 16 februari 2021

Introductie

Tegenwoordig wordt parametrisch ontwerp in allerlei vormen steeds meer toegepast door constructeurs, ontwerpers en adviseurs. Omdat in parametrisch ontwerp regels worden vastgelegd die automatisch worden “afgespeeld”, heeft dit invloed op de borging van de constructieve veiligheid. Hieronder gaan we in op wat parametrisch ontwerp is en welke vormen we kunnen onderscheiden. Ook doen we aanbevelingen voor best practices.

Software

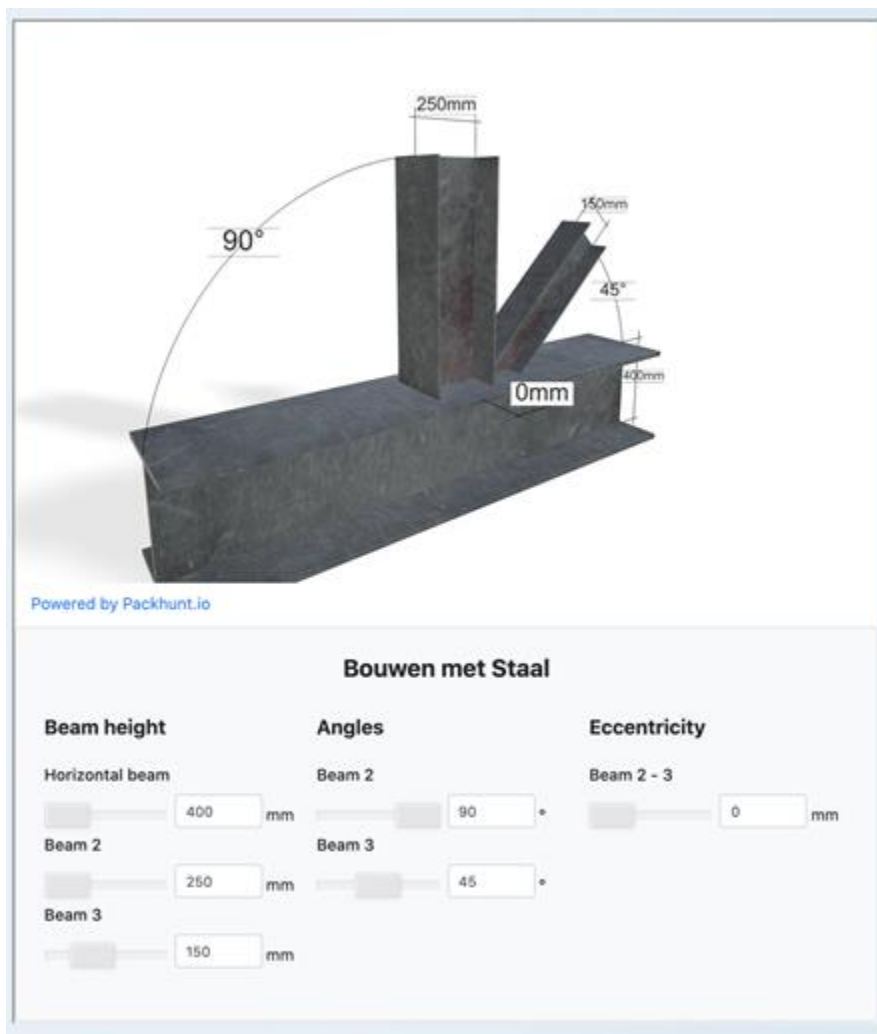
Voor parametrisch modelleren worden meerdere vormen van software gebruikt:

1. Scripting: de eerste parametrisch ontwerpers gebruikten programmeer- en/of scripttalen om parametrisch mee te ontwerpen. Een mooi voorbeeld hiervan is de Guangzhou TV & Sightseeing Tower, ontworpen door Information Based Architecture en Arup. Het constructieontwerp werd vanuit een AutoCAD VB applicatie gegenereerd en automatisch geanalyseerd.
2. Visueel programmeren: tegenwoordig wordt steeds meer de visuele wijze van parametrisch ontwerpen gebruikt, omdat dit laagdrempeliger en toegankelijker is. Door visueel programmeren ontstaat de mogelijkheid om een logisch model op te zetten door middel van visueel programmeren (dus zonder tekstuele code).

Parametrisch modelleren biedt belangrijke nieuwe mogelijkheden om logica en redenties vast te leggen in de computer en geautomatiseerd ‘af te spelen’. Bij verandering van een parameter (input) wordt het logisch model opnieuw berekend langs de gedefinieerde relaties.

Voorbeelden van veelgebruikte visuele software pakketten met veel gebruikers en een internationale community zijn Grasshopper, Dynamo en GenerativeComponents.

Het is tegenwoordig ook mogelijk om parametrische modellen in een online of gedeelde omgeving te gebruiken, zodat alleen een webbrowser nodig is om toegang te verkrijgen tot het model. Op de volgende pagina is een eenvoudig model weergegeven van een parametrische stalen verbinding.

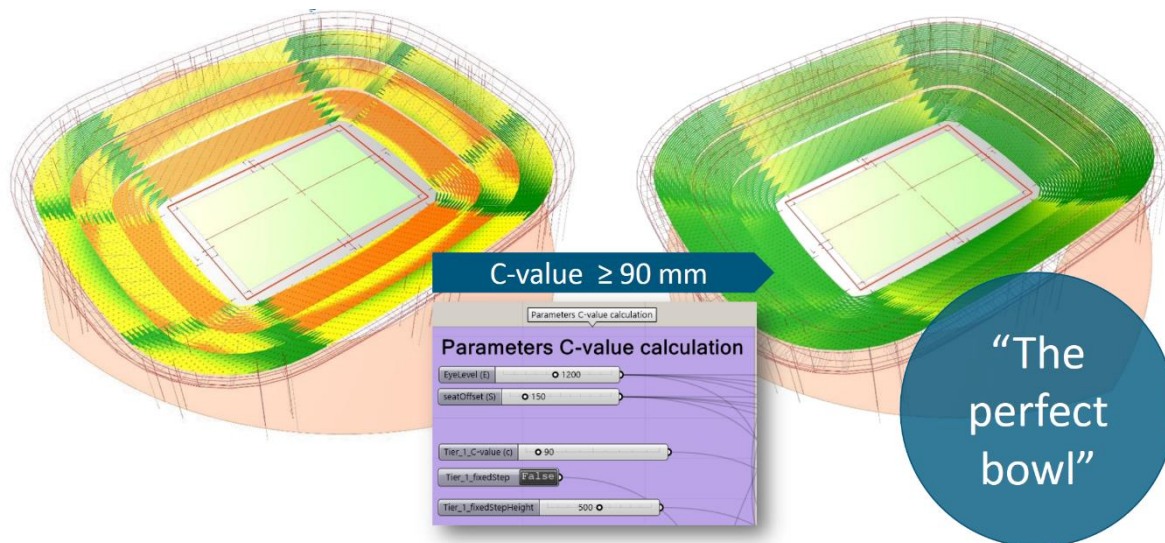


Voorbeeld van een parametrisch opgebouwde staalverbinding. Als de parameters worden gewijzigd, past het model zich ‘automatisch’ aan. (Bron: Rayaan Ajouz, Bouwen met Staal)

Ook door middel van (tekstueel) programmeren of ‘scripting’ is het mogelijk om modellen te maken die zich op dezelfde manier. Hiervoor wordt tegenwoordig bijvoorbeeld de Python programmeertaal gebruikt, maar er al een lange geschiedenis met scriptingtalen als LISP, VBA en VB.NET. Maar omdat het visuele en interactieve aspect ontbreekt, willen wij hierbij wel de kanttekening plaatsen dat deze wijze van programmeren verder van de ingenieur afstaat dan visueel programmeren (gevaar van de ‘black box’). Om de constructieve veiligheid te waarborgen, is een goed beheerd programmeerproces noodzakelijk. Daarom kiezen bedrijven er vaak voor om een samenwerking te organiseren tussen programmeurs en ingenieurs met de domeinkennis. Dit is een goede beheersmaatregel bij scripting, maar vergt veel communicatie om het wederzijds begrip te waarborgen. Dit is minder van belang bij visuele programmeeromgevingen, waarmee de ingenieur vaak wat directer zelf uit de voeten kan.

In het geval van tekstueel programmeren wordt ook wel gesproken over een ‘code-benadering’, bij scripting van een ‘low codebenadering’ (scripting biedt een tekstuele programmeerwijze om voorgeprogrammeerde bouwblokken te combineren) en bij

parametrisch modelleren van een ‘no codebenadering’ (omdat er visueel geprogrammeerd wordt).



Voorbeeld van een parametrisch ontwerp voor een stadion met behulp van ‘visueel programmeren’ (bron: Royal Haskoning DHV)

Spreadsheets

Spreadsheets zijn strikt genomen geen parametrische software, maar gedragen zich hetzelfde. We kunnen parameters (getallen in een cel) definiëren en relaties leggen (formules toevoegen). Bij veranderingen in de invoer worden wijzigingen door de gedefinieerde logica automatisch verwerkt. Daarom gelden veel van de aanbevelingen voor het gebruik van spreadsheets en zelf ontwikkelde software ook voor parametrische software; zie ook de notitie “Gebruik van rekensoftware”: <https://kpcv.nl/wp-content/uploads/2020/07/20200706-Gebruik-van-rekensoftware.pdf>

Koppelingen naar simulatie- en analysesoftware

Het gebruik van parametrische software is sterk in opkomst in aanvulling op BIM, maar biedt ook steeds meer koppelingen naar andere simulatie- en analyse software. De verwachting is dat dit gebruik nog verder zal toenemen de komende jaren. Dezelfde aanbevelingen gelden hiervoor als voor de rekensoftware (zie bovenstaande link); verhoogde waakzaamheid is geboden. De (constructief) ingenieur bevelen we aan om zich verder in deze software te verdiepen en de best practices te kennen, om klaar te zijn voor de toekomst.

Gebruik van zelf ontwikkelde software

Hergebruik van zelf ontwikkelde software (zoals spreadsheets) leidt tot belangrijke risico's waar de ingenieur rekening mee dient te houden. Als de software een fout bevat, zal deze mogelijk in meerdere constructies sluipen. Zelf ontwikkelde software dient daarom op systeemfouten te worden gecontroleerd en met verhoogde waakzaamheid in ogenschouw te worden genomen. Daarnaast moet bij voorkeur een test- en validatieroutine worden

gevolgd wanneer meerdere mensen de software gebruiken (zoals dit ook bij andere vormen van softwareontwikkeling gebruikelijk is).

Aanbevelingen best practices

De volgende zaken worden aanbevolen als best practices:

- Maak gebruik van de mogelijkheden van (geautomatiseerde) software testing.
- Documenteer modellen door middel van annotaties (toelichtingen in het model en 'comments') en visuele middelen ter structurering (bv. clusters, grouping).
- Geef parameters, operaties en groepen duidelijke en betekenisvolle namen (zodat het voor anderen duidelijk is wat er gebeurt).
- Pas 'grouping' toe om logica die een geheel vormt ook logisch te verbinden en 'encapsulation' te bereiken (heldere taak voor een groep).
- Pas 'clustering' toe om veelvoorkomende logica die een geheel vormt tot een herbruikbare component om te vormen.
- Pas plugins toe die (grotere) modellen opschonen of structureren.
- Maak gebruik van versiebeheer.
- Pas bij overdracht van gegevens tussen applicaties extra controles toe op de verschillende modellen om de overdracht en conversie te controleren.
- Pas plugins toe die data-uitwisseling standaardiseren, zodat de kans op fouten in zelf ontwikkelde data uitwisseling wordt geëlimineerd.
- Controle van tussenresultaten voorkomt het black box probleem. Bouw mogelijkheden in het model in om dit te toetsen.
- Bij validatie is het belangrijk om het toepassingsgebied te beschouwen.