SimplyRhino
sales, training and support

This is the full transcript of the recording from the Grasshopper User Group Meeting organised by Simply Rhino, featuring engineers, Eckersley O'Callaghan, and Clemens Preisinger from Karamba3D. This meeting took place live, online, on Thursday 22nd April 2021 and the video recording can be found here on the Rhino3D.co.uk website.

Paul - Good evening, welcome to the Grasshopper User Group Meeting. We have a group here for you. I am Paul from Simply Rhino. I think you all know that Simply Rhino supply Rhino support and associated training services here in the UK. I just want to say a few things.

Now, we're going to have presentations from Sam Gregson from Eckersley O'Callaghan, thanks for the wave, which will be followed by some questions and some discussion and then we're going to here from Clemens who is the developer of Karamba 3D, known I think to quite a few of you in the audience. Also, we are happy to have a guest panellist with us also, Jeroen Coenders. Jeroen is from Thornton Tomasetti, the core studio team that is a growing team in London. Jeroen has an announcement for us that he is going to make in between the two presentations about an interesting event that I'm sure some of you are going to be interested in hearing about. So, we'll hear more from him later on.

A couple of other notes, it's being recorded of course, as usual and you'll find all the recordings of the previous events on our YouTube channel and the Rhino User Group Meeting event pages. We are holding these events and there tends to be three streams for these events. There's the general Rhino stream, there's a Grasshopper stream and also AR/VR, which we've held a few of those meetings recently. But I think we're going to hold a few more with the Grasshopper and some Rhino focused events over the next few iterations of these meetings.

So, just a note on questions, I'm going to hand over to Sam in a moment. please type questions in. The more questions the merrier, so please, questions are very welcome. We do also have the possibility of turning people's mics on. So, if there is a burning question that you'd really like to have a discussion with Sam or Clemens, or any of us, you're very welcome to do so and we'd like to encourage that. So, please make it known that you'd like to have a discussion with anybody and we'll make that possible.

Other than that, I should have also introduced Steph who is here and helps organise these events, if you were wondering who that was also. Other than that, I'm going to hand over to Sam now, so Sam your presentation is up and we'll all be back with you after your presentation for the questions.

Thank you.

Sam - Okay, so EOC are Structures and Façade engineers they dabble a little bit in both, although my background mainly is in Structures. We have eight offices around the world, done projects everywhere pretty much. In fact, I've only had one project in the UK in the last six years, which is the one on the bottom left which is the Sky Pool in London which will be opening later this year hopefully. So, I'm very much excited about that. We're best known for our Apple Stores, especially the glass, so top left, bottom right, Central World and Marina Base Sands at Singapore, and we've started to do a little bit of super yachts and cruise ships, so on the top right, so glass staircase that we worked on, on a super yacht by Benetti in Italy.

I'm going to talk about favourite algorithms. My favourite algorithms, starting with K-means. So, K-means is a clustering algorithm, of which there are many. So, there's a list of a few there and what I'm mainly using K-means for is geometry or rationalisation, so I'm going to touch on that.

So, here is an image of some panel clustering. You might have seen this kind of things before in relation to K-means. So, basically, the idea is to try and rationalise the number of panels, reduce the number of panels and thereby hopefully save cost and logistics on site and that kind of thing.

So, just going to go through the problem which sounds very simple but it's actually quite hard to do. So, the problem is, how do you group a number of objects into K groups or the number of groups. So, this is how the algorithm works. So, we want four groups. This is the basic vanilla version. You pick four items, you assign the groups based on the closest item to each one and then you move that original point in to the mean of those groups and repeat. So, you'll see as we progress through here, the points get reassigned to their closest centroid centre, mean, and then the points moved to the centre of those objects and we iterate until we converge. So, it's a nice simple algorithm, which is kind of why I like it. I'm a simple person. It seems to work quite well.

So, when it comes to panels, the first thing we've got to figure out is how do we measure the distance between two panels. It's quite easy to understand how you measure distance between points in space, but with panels, how do we measure the similarity or difference?

So, two panels will be the same, triangular panels if their edge lengths are the same. So, one easy way to measure the distance is by this formula here. One thing we've got to be careful about is the orientation, what happens if it is rotated and therefore there is not an alignment with the sides, while we've got to measure it three times and take the minimum. Some other things to consider are, are we allowed to flip panels and then there are some other considerations when we go to the higher polygons, quadrilaterals etc.

So, this is what it looks like in our Grasshopper environment. We've got our edge lengths here. We stick it into our clustering environment, tell it whether we're allowed to flip or not and it spits out our clustered objects and our panel shift and panel flip parameters. So, it's a little bit more complicated than that as you could well imagine. We've got our groupings, but that doesn't mean they're more identical. So, we create an average and then replace our panels in their original position.

So, here's a little study we did for a Star Pavilion with looking at how the K, the number of groups, affects the deviation of each of these panels, basically the gaps we'd need between them. In the end, the client decided that he was very much keen on having as few panels as possible and didn't really care about the gap as it was a rain screen system and we were fine with one in the end. But it was interesting to do this study looking at how that changed.

Now in this study that I did recently for a project, the architecture was a lot more freeform, which meant we were more able to play with the geometry. So, here I'm using Kangaroo in the background and clustering and then smoothing the structure to fit the clusters and repeating. So, there's a bit of Kangaroo play there. You can see the deviations coming down to that 3mm for ten different panels.

So, a few things to consider on the algorithm side of things, what is K? How many panels do we want? As you can see, you can kind of play about and see what the deviation is. Another thing to consider is how you measure distance. I've explained about triangles but you might be comparing other things, not just panels, but any sort of geometry and there might be some normalisation required.

But other things that are really important that I want to stress is the other side of things to consider, which are like, what is the tangible benefit you are getting? It's nice to think that if we make many panels the same then everybody will be the happy, but you might just be simplifying one part of the process and then adding complexity in other parts. So, what tolerances are acceptable, and these two questions, you really need to think about the details. How is it manufactured? How is it installed? It really depends on the system. Have we got an open joint rain screen cladding system where the joints are allowed to be large, or have we got some toggle fittings or plant detail where really we're only talking about a handful of millimetres that is acceptable. Some interesting thought that needs to go behind these as well.

This was a cruise ship project. What was interesting here is other ways of tackling this problem, using normal geometry like a torus here, Sir Martin Francis chose a torus layout to get a high level of repetition. So, not using clustering panels. But the reason I bring this up is because at the end of the day, the contractor was like, I don't mind, I'm cutting these panels with machines and it's all going to be automated, and he smoothed things, changed things around as he wanted. So, that was an interesting thing to note that it's not always of benefit to do these things. So, that needs to be considered. And here, I just want to show properties on the left. I've got a nicely optimised structure, this pavilion structure, which is glazed with glass. I've optimised it, every beam size is as optimal as it can be for stresses and deflections, but it's not very buildable. So, one nice way of using K-means is reducing the number of properties here and then we've got the rationalised model on the right where we've only got five properties down from 30. I find this quite a nice visual to demonstrate what happens when you reduce the number of sections, which is on the X axis, what happens to the weight of the structure. So, you can see here it really starts to spike, when we reduce it further below five. So, that was chosen as our nice stopping point. So, a nice balance between steel savings but making it easier to fabricate.

My next one is SIMP, meaning Solid Isotopic Microstructure with Penalization. So, you might not know the name but this is a topology optimisation algorithm. Clemens might talk about BESO, I know that's in his Karamba software, but SIMP is another popular method. You will have seen things like this. So, it's topology optimisation, putting the structure where it's needed basically.

One thing that I wanted to just highlight is, you will have seen this before, Gaudi's hanging chain models. That inspired this. It's got nothing to do with SIMP, but my point is, we ended up with this inspired from the hanging chain models, not this. It's inspirational. I think people take these things too literally and I think what I want to get across is that you can use them to inspire and also to understand. So, here's a dome for a different cruise ship project and here we can see, we're minimising strain energy, which is a measure of how well it's working fixed with some constraints there, and what is quite nice is that it gives us a little bit of understanding of the structure. So, at the top here, we've got a two way spanning structure because it's quite flat. Then we've got more or less columns at the side, and where it bows around, we've got a tie and that's to contain the thrust that this happens from the arching action. So, it's quite a nice tool to understand how the structure wants to behave and therefore, if we do deviate from this, we understand what the consequences might be. Here's another one, it's very similar, but in this case, we've got structure spanning diagonally, and the reason for this is that we've got relatively flat sides which like to bend a lot and we've got curvy corners, which are really stiff. So, in this instance, the structures really are wanting to go to the stiff support of the corners and so it's added columns there as well.

So, you really get a lot of insight that helps you to think about where you might, even if you don't build something like this, where you might put your biggest columns, your biggest members, and then we've got a ring beam again, very similar to the last one, to contain the thrust from the roof. There are some different methods. They add to optimisation. Peregrine is a Grasshopper plug in that does that, SIMP in the middle and then automated beam sizing gives you similar ideas as well.

Then just my final one is Nelder-Mead method, also known as Downhill Simplex Method. So, this is gradient free numerical optimisation method. So, you will all be familiar with Galapagos which is a genetic algorithm. Nelder-Mead is similar but different. I'll just go through that. The reason I looked in to the Nelder Mead method is because genetic algorithms are great, but every problem has a different requirement. So, in some cases, what you really want to minimise is the amount of iterations. If I'm doing a finite element analysis that takes a minute to solve, I can't afford to have 800 iterations to optimise the structure, because it's going to take me 800 minutes. So, maybe if something like 63, Nelder-Mead on the right, 63 minutes is maybe more acceptable for an optimisation. But obviously this is a very simple set up here where we're trying to get from the high bits down to the low bits and structure is a lot more complicated than that.

So, just very quickly, how this works is we have a starting point and we want to get to a optimum up there. So, the way Nelder-Mead works is that this is in 2D space. So, we start with a simplex which is made of three points, because we're 2D, we go with three points. We just go with one more and that gives the algorithm an idea of which direction to go, because that's got arms in different directions. So, you know which way to go. We know the worst point and the best point and it moves in the direction based on the worst and best points using different methods, either reflection or expansion if reflection looks like a good direction, tries the expansion to go a bit further. Maybe it wasn't quite as good as these different methods that it moves about to explore depending on how good the values it gets from moving around in space. So, this is how it moves in 2D, a way to understand how it gets from the blue to the red where X and Y are the input parameters. Obviously, this isn't the problem we're looking to solve but you'll see some examples in a sec. So, we implemented this and we've got a plug in in Grasshopper for this.

Here is a quick example. This is a structure that I showed briefly before. In this case, we input parameters instead of X and Y coordinates in space. Input parameters are the vertical position of the control points of our surface here, so we're moving the surface and we send this into a structural analysis programme from Grasshopper. It spits out our stresses, our deflections and we see how good a structure it is by moving these points up and down. We move the coloured points together, so we don't get weird asymmetry.

So, here are some of the solutions that came out. We've got two local optima. We find those by re-starting a new position. So, we've got our base line in the middle where you need a very big fabricated hollow section and then for the local optima either side, we've got some different aesthetics with some different sizes of much smaller steel work. So, that was one example.

A second example relates to glass. So, here's the two methods of shaping glass. Every architect wants funky shapes of glass. The top one is slumping, which is quite expensive, requires a lot of energy and has limitations but you can get all sorts of freeform geometry. The bottom one a lot cheaper, lower energy, and also we can use coatings with a better performance which is better for the environment as well. Limitation is, it's cylindrical or flat only. So, our other use for Nelder-Mead was to take objects, glass panels and see what cylinder they might fit best to. So, we've got a script on the left, but with a cylinder fitting algorithm. This time, our Nelder-Mead is embedded inside our C# component, so there's no Galapagos interface. We've got our panels on the right and they spit out the optimal cylinders moving around these cylinders and finding an optimum. So, we can look at the deviations, the original curvatures and the rationalised curvatures and check whether these deviations are acceptable or not.

We did this on a project Two Taikoo Place in Hong Kong. The architects originally were going for conical which isn't the same as cylindrical, so we optimised and found our best cylindrical representation of those conical panels and replaced them with conical panels with minimal visual affect. But now making it a lot cheaper and allowing the coatings that were required to be applied to the façade with a better energy performance.

Here is Sberbank in Moscow where we extended these capabilities, now looking at organising the panels in to flat, cylindrical, cold bent, slumped, and this required a little bit more analysis of the stresses. So, we are taking our models, going into final element analysis, looking at the stresses and taking them back again.

Then finally, I just wanted to touch on NURBS, which is a work in progress, something I'm interested in at the moment, but want to spend some more time looking at. Obviously Rhino started out, it was all about NURBS. That was how it started as an AutoCAD plug in to deal with NURBS. But I only realised how cool NURBS were more recently with isogeometric analysis that is a new thing that has come in the last few years, at least from what I've seen. Kiwi is an example. But I think there is more potential for NURBS. What I've been looking at specifically is… so NURBS is basically the mathematics of the structure and you can derive all sorts of things like developability and stuff, and this will give potential to optimise freeform structures in a lot better way than we can do with meshes. So, I've kind of dabbled in this a little bit. Really, what I'd like to see, hopefully, maybe I'll end up doing it or someone else could do it, is having a goals NURBS stuck in to Kangaroo here. That would be great. I know that Daniel Piker has now enabled us to programme our own goals which is a great idea. If we can do this with NURBS and free us from the meshes that would be great. So, that's something I'm looking into a little bit further. That's me. Thank you.

Paul - Great. thank you very much Sam. There's a nice note from Daniel Piker who is watching your presentation Sam, who said, definitely, get in touch please. So, instant feedback there. So, please, anyone, questions for Sam. Please type away those questions but perhaps Clemens, please? Your mic.

Clemens - Thank you very much for the great presentation. That was super interesting. I have a few questions actually. Did you apply it to connection design because this has such huge potential to limit the number of different connections and so on?

Sam - The physical nodes themselves?

Clemens - If you think of steel connections, these are normally quite complicated. Is there a certain limitation on the number of dimensions that it makes sense to use?

Sam - No, in theory, there's no limit on how many dimensions you can go. I don't know if practically it starts to break down at some point, I'm not really sure, I've not looked into that. I haven't personally looked at node geometry but I’ve read a little bit about it, and thought a bit about it, looked at presentations and that kind of thing. It’s quite a complicated thing. Again, it’s about how do you measure distances between two nodes. You’ve got to formulate the maths to compare two things. Again, it’s a question of why as well because we did start thinking about this for the cruise ship project but then the contractor was like, I’m just going to mill all of my nodes and so it doesn’t matter if they are all unique or complicated and twisty. I’m just going to get a lump of steel and mill it. So again, it comes down to how these things are going to be fabricated, what are they going to look like. But there’s definitely potential to do that with this kind or algorithm, yes.

Paul - Okay, a question from Alexander for you Sam. Does your clustering node only work for triangular panels?

Sam - No, I touched on triangles because they were the easiest to explain but you can do it with quadrilaterals, it just gets a bit more complicated because even if all four sides are the same, one could be skewed. So, you’ve got to start looking at the diagonals and stuff like that, but it works for quadrilaterals as well.

Paul - Okay, from Alexander again, can you please tell, in an example from Kangaroo, with clustering, which Kangaroo nodes were used?

Sam - In this case, the definition of the similarities of the panels was about edge lengths. So, we’re basically using the Kangaroo edge lengths to try and make them more similar to each other. So, it’s the edge length component, the Kangaroo component that we’re trying to get closer to each other. So, that’s basically the gist of that.

Paul - Another question for you, could you please explain a little more about layout optimisation?

Sam - You’ll have to get the Peregrine guys on a UGM, because I know very little. But I do know it sounds very clever. It’s basically, rather than, so the topology optimisation SIMP is mesh based, whereas this is element based. So, that’s one difference and it’s a little bit cleverer than throwing Nelder-Mead or Galapagos at it because it uses the stiffness matrix and things like that. So, it’s a much more mathematically based approach. But ask the Peregrine ideas.

Paul - That's a good idea. Maybe we will have them present. Now from Marcello, how do you consider the seismic load and to add to that, do you use another programme, or only with Karamba?

Sam - In the SIMP topology optimisation I'm guessing, yes, for the SIMP topology optimisation, I'm using Strand Seven as a structural solver in the background. So, I've built everything up inside of Grasshopper and then feeding it out to Strand Seven, feeding the results out and doing that iteratively. But there's no reason you couldn't use Karamba. Karamba can deal with gravity loads, static loads going sideways and stuff, which I think for this iterative process, you want to keep it simple. You don't want to do crazy dynamic stuff, because this is more early stage early optimisation.

Paul - Thank you. Again, question from Adalina. Did you write your own K-means algorithm or use one available from elsewhere, and also, what other applications do you see this used for?

Sam - So, I kind of nabbed the starting point on a Grasshopper forum like many moons ago. Someone put something there and then I adapted it. That's the great thing about the community, everyone is sharing things. There was a conversation on there years and years ago and I adapted it because I needed to rotate my panels, I needed to be able to flip my panels and compare them that way. So, yes. Half completed by myself

Paul - Something else here from Ronaldo, how did you originally get in to computational design and what recommendations would you have for people that have joined us who are new to these subjects?

Sam - I did a tiny bit of coding at university, that was MATLAB. I actually started off studying aeronautical engineering and then made the switch to structural engineering. So, I got a tiny bit of coding in there, but other than that, it's just self-taught, diving in to things and Grasshopper is a really great gateway drug to more hardcore coding, because it's so easy to get in to and it gives you that logic that you need to build on to get in to things like Python and C# and that kind of thing. I think the best thing you can do is just go for it and just get stuck in. There's no wrong project for doing things like this I don't think. No project is too simple, so just get stuck in.

Paul - Good advice. Okay, now another question. I start to see this trend of structural engineers knowing how to code. Do you see this as a necessary skill to pick up? If so, would you recommend starting off with C# or Python?

Sam - I picked C# because I had to pick something and that was quite good for building components for Grasshopper for example. I do believe you can do that with Python now, or you used to be able to at least. But I don't think it matters to be honest, because the skills are transferrable. Things like machine learning and those kind of things, there's a lot of libraries in Python. So, if machine learning, AI, that kind of thing is what you want to get in to, then maybe Python is better. But just get stuck in to one of them and you can always learn the other one.

Paul - Jeroen or Clemens, would you have a view on that?

Jeroen - Actually, quite similar. This is an ongoing discussion and a never ending discussion. It's people's favourites. But I tend to agree with Sam actually. Once you get in to it, it doesn't really matter anymore. Python is a great one often to start with because the syntax is a little bit easier to write, but personally, I'm a bigger fan of C# I would say, for the same reason Sam was saying. That's where we started. It was much more integrated within Grasshopper. So, it allows you to script within Grasshopper, but then if you want to make the next step as Sam says, to actually compile components. You can do that as well. It's very much transferable in C#.

Clemens - I would also say C# is my favourite, because there's one thing you have to find your variables, which you don't have to do in Python and this sometimes causes the code to get quite messy if you're not very disciplined. But Python, is very mighty, because there are many libraries available and that machine learning side of the matrix and computations.

I would also say both languages are good. But for the beginner though, C# is even easier. I don't know.

Paul - Okay, thanks. So, one other question. Have you got any other GH Plug Ins or components for running algorithms, or are you developing your own?

So, the Nelder-Mead one is on Food4Rhino, we put that one out there, but there is other optimisation algorithms by other people that work well. We write a lot of code in house but I think it's one thing to be yourself and do something that does the job and another to make it user friendly and UI being nice and stuff, and I'm sure Clemens knows all about that, and writing documentation and stuff. So, I think you really have to put a good reason to delve into that and as the consultants, it's not something you have time to do. I applaud those that do stick their stuff on Food4Rhino, it's great, because there's a lot of work that goes in to it.

Clemens - One question, the SIMP algorithm, did you code it yourself?

Sam - Yes, SIMP is an algorithm that there is many papers on. I think Millepede is potentially a SIMP implementation but I coded my own so it works with the Strand7 software that we have, that we like to use in house.

Clemens - Very cool patents that we sell from this.

Paul - Okay, so is there any final questions within the group here? What we'll do now is just switch to Jeroen, you can make your announcement and following that we'll move straight on to Clemens presentation. So, I think now, you will have control so we can see your screen.

Jeroen - That sounds great, thank you very much. Actually, just a very short announcement. We're really excited to host our new next AEC tech workshop and set of presentations around it. Really, that is today's announcement. So, July 8th to 12th 2021, we'll host this. A lot of details need to be finalised and figured out but together with Grimshaw, KPF and Simply Rhino, we're getting together and doing this. In the first instance it will be virtual, but let's see how things develop from now until then let's say, if we can actually meet in person yes or no, then there might be some events even in our own studio or with Grimshaw. But that's it, July 8th to 12th. So, please stay in touch and we'll let you know more in advance.

Paul - Excellent, thanks Jeroen. So, now we can go switch to Clemens and we'll get in to Clemens presentation and then we'll all meet again at the end. So, see you in a while.

Clemens - I have to get the right screen. So, now you should see my screen, I hope this is the correct one. First of all, thank you for having me, for the invitation. I would like to present today a little bit about Karamba 3D and about how it is applied in the office of Bollinger+Grohmann where I also work. We have used it for several projects now. I will show three of those projects.

Karamba was actually a product of a university research back in 2008. There was a research project at the University of Applied Arts in Vienna which was about optimisation of space trusses and from there on it developed, also in cooperation with Bollinger+Grohmann which is great, because from there I get constant feedback of what could be improved. However, it is now a think on its own also, and there is a special thing about Karamba. It is a finite element programme like many others which are now around since the 1970s approximately, so since computers have gained access to offices.

The main difference to other element programmes is that Karamba lives inside Grasshopper. So, finite element calculations are usually connected with huge amounts of data that need to be passed around and this makes this process quite sluggish sometimes. You don't have to export geometry from Grasshopper to some finite element package. Since Karamba lives inside Grasshopper, this bottleneck doesn't exist and this is also one of the things which we wanted to achieve, an interactive use interface so that if you change something on the geometry.

This was the slide I talked about. This is Karamba inside Grasshopper, and for some reason it is out of sync. If you change something in the geometry, you can change something in the geometry and Karamba, you see the impact on the changed geometry and the loading of the structure on the structure itself. Since Grasshopper is an environment for doing geometry calculations, all the stuff that comes with structures, like loads, supports, cross-sections, needs to be supplied by Karamba. Since it's implemented in Grasshopper, you can also make use of the whole eco-system. Like here for example, there is an optimisation done using Galapagos on the simple truss.

With Karamba you can calculate beam structures and also shell structures. There are several algorithms like buckling analysis, natural vibrations, cross section optimization, BESO (Bi-Directional Evolutionary Structural Optimization) to optimise structures. Because this was the first focus to use Karamba for structural optimisation in the early design stage. This was where it was used first.

From a Karamba calculation, you get graphical results like from any other programme, and you also get out the corresponding numbers which is quite nice because this can then be used to improve design and develop a feedback loop.

In the meantime, there have been plug-ins around for Karamba that can be used to export the data to other finite elements software, and this is how it is done in the office, that Karamba is used up to a certain stage and for producing documentation or reports or so on, for doing approved calculation. It's often the case that we move towards RFEM / RSTAB or some other finite element programme to do the final documentation.

One thing that gets more and more is Scripting with Karamba, and that is one of our aims to open up the software so that you can use Karamba in your own Python scripts or C# scripts and in the meantime there's a Unit Test project available on github that you can download and API references.

When it comes to future development, one thing which is new, that's also available on ShapeDiver and of course, I want to implement more and more things to make it useable in different areas, is more things like Line Joints which has already been implemented physically and Result Superposition and Spectral Analysis.

At the moment, since Karamba is mainly used for structural optimisation, there are some features missing in the normal finite element code, because when it comes to optimisation, it's often not necessary to calculate a very detailed mechanical response for the structure, but it's rather important to have a relative measurement between different alternatives of geometry or structure, to make optimisation work.

Of course, there is also a github page where you can constantly download the latest builds and it's an open development in so far as feature requests can be posted or back reports.

Now, I would like to turn to three projects. The first one is a dwelling house. It's called Marxer Active Energy Building. It was done by falkeis architects here in Vienna, therefore the building site is in Vaduz, Lichtenstein, a small country near Austria, and the special thing about this project is that it has all the latest energy saving technology and solar panels and everything, so that it actually creates energy instead of consuming it and Bollinger+Grohmann was asked to do the optimisation of the structure. This is how the structure basically looks like. There are several floor slabs, there is a large wall at the back and an elevator shaft which is there for the horizontal stiffening. Since this is not quite sufficient, there are also these V shaped columns which also provide some horizontal stiffness and of course can also provide for vertical loads take down. The question was, where to place these columns, and for this, my colleague Robert Vierlinger used Karamba, and the question was, how many columns do we need and where shall we place them? Here the BESO procedure was used. There are regions in the floors where columns could be basically placed. You can see them here. These are the dark grey areas. So, he set up a structural model, filled it with columns, placed the driving loads on it, dead weight and wind in this case, and then the BESO procedure removes gradually elements which are not so much utilised, so in case of columns, a certain percentage of the columns with the smallest normal force get removed in each step. Then step by step, the number of columns is reduced and what remains is then a pattern which shows how the forces flow through the structure. In this case, you can see there are column trees which remain and these are then associated with the first approach to optimising the columns and Robert Vierlinger also used a second approach, there's a second step, namely Galapagos. He used Galapagos here to orientate the columns and you can see that there is this large gain at the beginning of the optimisation that gets done less and less.   This is how the structure looks like. Our columns are prefabricated, so they are nice and smooth surface. However, the concrete in this case is rather covering for fire protection, so actually it is more steel than concrete inside those columns. This is what it looks like in real life, so it comes quite close to the vision of the architect.

Here, a second project. It's by Delugan Meissl Architects it's the so-called Hyundai Goyang Centrea . It's basically the flagship store of Hyundai Motors. So, it's a big shop for selling cars. The idea was to have a large open space underneath the roof that flows above this space. The roof is not only a roof but also houses programme. There are offices. There is a little garden there, cinemas. So, quite some programme.

So, the first step here was to design the steel structure which conforms to the programme. You can see here there are four reinforced concrete cores which are there for the horizontal stiffness of the structure and the span between those cores is approximately 50 metres, so quite large spans. The question was, how many additional columns are necessary, and of course the architect wanted an open space, so no columns. But it's a question of cost also.

So, my colleague set up this geometry here as a Karamba structural model including the governing loads, did a cross section of the optimisation clustered also the cross section, because it's not practical to have hundreds of different cross-sections on such a project. And what he also did, he used Octopus to do a mighty objective optimisation.

The columns are actually not columns but as you can see column clusters which are small bits of bamboo and the precision of the number of these column clusters was parametrised and optimised using Octopus and there were different goals, like the maximum displacement and total mass of the structure. The mass of the structure is a very good indicator of the cost and the diagram here that you can see was done in a very laborious calculation… there were lots of optimisation calculations. It took a very long time to compute this diagram, and what you can see is if the displacements need to be small, then we need to invest in a lot of mass for example here, for five clusters. There's a maximum here and the more displacement that is allowed, the smaller the mass that needs to be invested. So, for five clusters, you have the other extreme. Mass for six clusters is getting better and seven, it's quite relaxed actually. So, this diagram was the guide for the architect and they from this diagram chose one alternative that was then built, this one here.

The other nice thing about having this model in Grasshopper was that the drawing of sketches could be optimised. My work colleagues they set up quite a large Grasshopper definition at that time and so the changes which the architects want to have are quite easy to accommodate and so the sketches could be updated automatically. Today we would rather use Revit for this task but at the time it was done using Grasshopper.

You can see here how the building site looked like and it's now been opened two years ago. It was a very nice project where Karamba was used up to the detailed design.

This one here is a rather smaller project in comparison. It's a pavilion which was done together with SOMA Architects from Vienna. It's called White Noise and this is the image that won the competition for the architects. What they wanted to achieve was the impression of randomness, that's also the reason why it's called White Noise. The dimensions are 20 metres by 50 metres approximately and it consists of a large number of aluminium rods of 2 metres and it looks very random, but actually it's not, because the system consists of many layers of these rods. The deposition is defined by the membrane which is on the inside, and what gives the randomness is the inclination of these rods. So, this is one layer. This is the next and the next and the structural system works like this, where there are intersections between two layers, here connection starts, these are the black circles. The inclination actually governs the way the forces flow through the structure. So, what the architects wanted to have was a structure that looked random, but at the same time stands up of course and has a good structural performance.

So my colleague, he set up a Grasshopper definition. He used the geometry of the internal hull. There was a membrane that was for sun protection and rain protection and he parameterized the model with the surface and split up the whole pavilion in to five parts to make it work with Galapagos, because that's quite essential to reduce the number of free parameters, because the more parameters you have, the longer it takes to get to a reasonably good result and the governing load cases, or in this case the pre-stress of the membrane underneath the structure at the same time weight and of course dead weight. Using Galapagos which allows to take account of only one objective, he had to put together a fitness number which took account of the maximum displacement, bending energy which are both indicators of the stiffness of the structure, then the number of joints which is an indicator of the cost. From all of this, he set up a global fitness number which was then used to do an optimisation which took approximately one night and at the beginning since the starting point is a completely random orientation of the rods, at the beginning, the performance of the structure is very poor. But it gets better and better over time and we certainly haven't found the global optimum but it was a solution that was reasonably good and this was fine for us and in order for the architects.

Since all of the parts were different, it was quite essential to have a digital workflow. So, at that time we used extra sheets to transfer the data to the workshop and also it's very important to have certain order so that the right parts are there at the right time. So, we can see how the structure was set up layer by layer. This is how it looked on the day of the opening. I didn't mention this but it is the main structure of the Salzburg Biennale, which is a festival that takes place every two years. So, every two years this pavilion is set up for three weeks. At the beginning people didn't like it because they are quite conservative, but maybe over time they will get used to it.

Karamba is a team effort and you can see the team members. Thank you for your attention and I will be glad to answer your questions.

Paul - Thank you so much Clemens. Right, I'll just stick my screen. So, a few questions for you and for the panel. First question from Andrew Phillips. Clemens, are you aware of any universities using Karamba as part of their core structural mechanics teaching?

Clemens - Yes, there are a few. University of Virginia, I would not start to count them but there are 20 to 30. Most of the time people write to me and then I send over licenses for workshops and so on. So, I would say over the past year maybe 50 or so. It's hard to say actually.

Paul - Andrew is at Imperial College and perhaps we can get a list to Andrew separately if he'd like. We can come back to you on that with a more complete list.

Just a question for the panel for their thoughts on this. Do you think there is a risk that students, we're talking about university students, the question has come from Andrew again. Is there any risk that students will fail or struggle to learn core theory if digital design is given more prominence? This isn't Andrew's thinking but it is something that we do hear from traditionalists. It would be interesting to hear your views on that.

Clemens - I would say, there is a risk that people start to use these tools as black boxes, so they are not in the position of control but rather get controlled by the tools. But I think they are, on the other hand a great way to make things easier to grasp, because if you think of interactive structure design, you see how the structure reacts and thereby it's maybe easier to develop a certain kind of feeling for the structure I would say. Also, one learns to give hierarchical structure to design maybe. I would say it's sometimes helpful for architects.

Jeroen - I tend to agree actually. It actually opens up more possibilities in a way of more options, quicker options let's say. So, the basics, they still need to come from the university curriculum, the base curriculum. But they just add another layer to it I feel, like Clemens says. But also, I really liked how Sam was talking about the dome and actually the optimisation algorithms and how you can derive principles from it but not directly use that as the answer I'd say. I'd say that's probably the critical line where one needs to be anyway.

Sam - Yes, I'd agree. I think, I guess there is the danger but the same with using finite element software rather than by hand, it's a similar analogy. But it obviously also offers opportunities as well. I'd say that I've had to revisit some of my university and school stuff by getting in to coding SIMP and remembering what strain energy is and trying to come up with the optimisation algorithm and remembering digging out my notes and understanding what virtual work is. So, there's principles you do have to understand but you've always got to question what you're doing and why you're doing it, like the clustering optimisation. Do we actually need to do it or are they going to laser cut everything and it will be fine? So, always questioning what you're doing and I think you'll be alright.

Paul - Thanks for your answers. Question from Mark for you Clemens, how would you compare Karamba to the popular standalone FEA products generally and how would you position it?

Clemens - The unique thing about Karamba is the connection of geometry and structure I would say which you don't get in normal finite element software. Usually, they have their own geometry use interface and also ways to interact with those. But this is something which was added late on. The other thing is that most of the finite packages that you can buy that wants to have more functionality. You can do print out reports so that they are more geared towards the production steps that you have in the structural engineering office. On the other-hand Karamba is more geared towards structural optimisation, working at the interface of engineering and architecture.

Sam - I would say that because Karamba is embedded so tightly with Grasshopper and Rhino, what most packages are really bad at doing is dealing with geometry and it's such a pain to set up a model in most finite element software packages and Rhino and Grasshopper do that really well. So, you get that for free with Karamba and that's what I think makes Karamba really powerful.

Paul - Clemens, have you seen evidence of this now that we're seeing the popularity of Rhino running inside Revit, having an impact in more and more studios? Have you seen, particularly with what Sam was saying there, the impact of that, not just in a Rhino environment but within products like Revit now, that is possible? Could you say anything about that?

We use it at the office of Bollinger+Grohmann and it's a great way to interact with Revit. I'm personally not a specialist in this data exchange side of things but I think it's a great way to use Grasshopper to control an IFC model so that people don't have to use Dynamo anymore. I don't know.

Paul - Thank you. A question from Sebastien. Is it possible to perform concrete design? Does it assist at all with this according to Euro codes with Karamba? Does Karamba help in that area at all?

Clemens – For shells there is a reinforcement design procedure there which is based on procedure by Kaufman. So, you get reinforcement quantities based on a linear calculation of the cross section forces. For beams, not yet, but I'm planning on having it as a feature so that you can at least do a linear calculation then to cross section design based on correct state. Maybe if non-linear calculations, but that's still a few months in the future.

Paul - Okay, and a question I hope I can make sense of. This relates to the White Noise structure. How did you analyse such a random structure with the described nodal joints?

The model was controlled by 15 or 20 parameters. These were then used to create a spline and from the spline, the angular values were derived for the individual elements because they were much more than these 14. And this was how it was done. The intersection was done using Grasshopper geometry calculations. So, for each set of parameters there was always a feasible structure. This was the important thing.

Sam - You didn't have any collapsing flying structures or anything?

Clemens - No, because… maybe there were some but that's always a problem with optimisation engines like Galapagos. If you have models that do not return the solution then they cannot be optimised. If there are too many of those then the optimisation normally doesn't work.

Sam - And you just did a set of 14 and that as duplicated?

Clemens - That was blown up, because if there are too many parameters, these algorithms are very time consuming. Apart from of course having something like Nelder-Mead, but I think it depends a little bit on the kind of problem that fits best. So, for totally random… it wasn't actually totally random, but for structural optimisation of this type, it's usually good to do it on a generative algorithm to avoid this local optima.

Paul - Okay, well I'm going to begin to wrap up. I just want to of course thank everybody for joining us whatever time it is where you are and please do for future meetings, for keeping up to date with that subscribe to our newsletters or follow us on any social media platform. I want to say thank you of course to our presenters. Thank you Sam, thank you Clemens, thank you Jeroen for joining us and helping us out. Good presentations, lively questions, so all very good.

Just to remind everybody that it is being recorded. We hope to have the recording up within a few days, and these meetings are all transcribed as well so you see the full text there too if anyone needs that. We do have sponsors for these events so I must just quickly thank them, the two sponsors that I mentioned, BEAM and PNY. BEAM isa very nice elegant tool for communicating between Rhino and Revit and vice versa and PNY who you might know, they are the distributors of some pretty powerful graphics cards from NVIDIA. So, they help us with these events.

Finally, I want to mention that our next meeting is on the 20th May and this will have a slightly different focus. We're going back to our roots you could say with more of a pure Rhino story with an industrial and automotive designer who has been a user of Rhino for many years. So, hopefully that's of interest to some of you. You'll be welcome to join us again on 20th May, and it's the date after that in June we'll be returning to another Grasshopper story, perhaps with an architectural flavour next time.