DAVID WEINBERG: Welcome, everybody. I hope everybody's recovered from yesterday, last night. Generative Design with Autodesk Nastran Topology Optimization. I'm David Weinberg, Senior Software Architect. And I'm going to be doing this with Mike Smell.

MIKE SMELL: Hello, I'm Mike Smell. I'm the Product Manager for Simulation Workspace and Fusion 360.

DAVID WEINBERG: All right, so class summary, key learning objectives, they're online. There's going to be a lot of stuff here. And you can download it and reference it. I can't really go through all the items and everything because it's just too much content, even for 90 minutes.

But if you have any questions, please stop me. And there will be a period towards the middle when I can answer a lot of questions because I'm going to run a video. And while that's going, you guys are free to chime in and say anything you want.

So here's the basic theory. This may sound a little confusing, but there is a difference between topology optimization and shape optimization and parameter optimization. With shape optimization, you're dealing with dimensions. You have a part and you have a radius someplace and you want to vary that radius to come up with an optimum design or dimension or thickness.

With topology optimization, you're dealing with a design space. And people say, well, is it subtractive? No, the answer is it's actually both. You can add elements, you can subtract elements. It all depends on what you start with.

But you are working within a volume. And that volume has to be defined. So the way this is done with finite element analysis-- in the old days, it would be done with trusses. And, with trusses, you can see every node is connected to every other node. And then you simply go in and remove elements by changing their area. If an area goes to zero, the element gets removed. And, in the end, you end up with that design over there, which we all know makes sense from an engineering standpoint.

The way topology optimization works is it's based on density. Now, it's not density in the true sense of the word. It's density because it's this variable Xe here. And it's either zero, when you have a void, all materials removed, or it's one, when the material is there, present. And, in actuality, it can be somewhere in between.

And sometimes people see that and say, well, that doesn't make sense. Well, If you're dealing with a curve, a curved shape, you may need something because the elements are voxel, let's say, and they're cubes. It needs to define that. And we'll see that later. But the volume of the model is the sum of the volumes of the elements times that density.

And the stiffness, however, is not. It's the density raised to a power p. And this is called SIMP, solid isotropic material penalization. And the p in that is that exponent there.

And what that does is it makes the final design more black and white, which means you see, obviously, where elements need to be removed or material removed and you see where material needs to be kept. And the curve to the right there shows you because it's an exponent. That's how the stiffness is varying.

So this is the design evolution of a structure that we all are familiar with. If we start with this simply supported beam, apply a point load at the middle, most designs, if you were to do it intuitively, you would say, well, that's one method there. All of us who have seen train bridges know that it should look like the one down there.

But the way this thing starts is, initially, the red is material that's being kept. The blue is material removed. And the colors in between are something in between 0 and 1. And if we look at the curve to the right, that's a curve. And I'm going explain compliance here in a minute a little bit better.

But, basically, compliance is the inverse of stiffness. And normalized compliances, you start with that right there, a full model, full stiffness. And as the compliance goes down, you're getting stiffer, relative to the initial stiffness. If it goes up, you're getting less stiff.

So we noticed that, in the beginning, the structure starts off adding material and it gets stiffer. But, why? Well, we started with an initial volume fraction, in this case, of 0.5. We said, hey, I want this design to be half the weight of what it is currently. So it starts with that.

And then it starts adding stiffness. It increases the stiffness in those elements up there. And then it starts removing it again. So you can see it getting stiffer with each iteration and then getting less stiff as it removes material and then stiffer in the end. And the final design is a 22% increase in stiffness over the full model and it's half the mass. And that's why everybody loves this because this is a great way to get better designs.

Now, here's some definitions. Objective, that's the goal. Usually, it's something like minimizing mass or minimizing compliance. Design constraint, that's what you have to worry about. You don't want, for example, a stress to exceed a certain value or a temperature to exceed a certain value. Those are the limits on what you can do with the model.

Manufacturing constraint specifies how you're going to manufacture it. Does it need to be symmetric? Is a part limited to a certain dimension? Does it need to be extruded or 3D printed?

Compliance, the inverse of stiffness. Volume fraction is just the ratio of the full volume of the part to whatever the volume is or whatever you want it to be in the end. And the design sensitivity is the change of the objective with respect to the design variable, which, in our case, is element density. And it can also be the change in the constraint. And we look at both. We measure the gradient based on objective and on the constraints themselves.

So global versus local is a very important point because a lot of times people will see designs and they'll go, well, you know, that's not as light as I expected. Or, well, why, when I made this change, I got a lighter design, a better design? And the reason is that this is like if you go up to the top of a mountain in a mountain range, you can see pretty much where all the peaks are.

But if you're in a valley, you do not know if you're in the lowest valley by just sitting in a valley. You know you're at a low point in the valley because you can see the walls around you. But you don't know if that's the lowest with respect to all the other valleys around you.

And that's what happens here. So, based on the mesh density, based on the initial starting fraction, volume fraction, actually, based on other things, other parameters that we can change, we may end up there or there or there. And when we do that, if we do a bunch of these, the one that's the lightest is actually your global, most likely. But you'll never know if you actually have achieved a real global minimum.

So the way all this works is, if we just look at the objective, in other words, I start at some point and I do an iteration and I look at a sensitivity, a response like compliance, and I see my change in compliance, and as I make changes to this model, my compliance, let's say, is going down. And it keeps going down, it keeps going down. And then all of a sudden, I reach a 0 slope and then I see it going back up again. I know that I have achieved a minimum. And I can stop.

But the problem is that we have these things called constraints. Because if we just keep removing material, we're going to end up with nothing. And, yeah, it's the smallest mass. We've minimized mass. We've minimized it all the way to zero. But it's not a structure that is usable.

So what will happen here is we now move and we have this constraint. And as soon as that constraint is violated, we have achieved, let's say, the lowest mass but with a stress that's at a point where, if it goes any further, the stress is going to exceed a limit.

Now, that's simple for one constraint. But when you have two constraints, for example,-- and, by the way, we deal with hundreds sometimes-- then it's a little bit more complicated. And that curve there is a little tough to understand. But, basically, what's happening is you're moving along a curve. And then you hit this other curve and you hit its constraint and you have to stop.

So some examples. This building, I believe, is in Shanghai. Excuse me. It's in Shanghai. And it possibly is done right now. I'm not sure. It said that construction will be done in 2016. But they use topology optimization to actually come up with the framing concept for this building.

For a wheel, for example, you can start with a solid wheel. That's obviously not going to work too well because it's too heavy. But topology optimization will actually give you an optimum design in an aircraft where it's really important to keep weight down. You can see how a complicated rib was designed and what it evolved to. And then also in the automotive industry for making doors lighter and stiffer.

Now, the big challenge with this is that you end up with something that looks like that in the end. And now we have to turn that into geometry. Well, one approach is to use beams.

You can go in and connect all these up and then add an area or some dimension to the beam. And then you end up with that over there. But that's not what we always have. So it's actually a very big challenge for us. And we're going to talk more about this later. Mike's going to cover that.

So Nastran and topology optimization specifics. Now, this is the list of objectives that we can solve Right now, currently in the version that we have that we're developing, it's going to be the same as the constraints, for the most part.

So, basically, what you normally would think of as a objective can be a constraint and vice versa. You can switch the two around if you want to. And the codes down there tell you what it's for. And you guys can look all this up, download the PowerPoint and have all this information. But you can see we do quite a bit, including stress, including temperature changes-- so you can do a steady state heat transfer and look at a gradient-- including a frequency or buckling. I'm going to demo those.

And then here are the constraints. And then you can see here, again, instead of minimizing or maximizing in this case, you have a range, OK? Either it's greater than and less than or it's less than or greater than. And the reason why some of those are less than is because greater than wouldn't make any sense.

And then these are the manufacturing constraints that we currently have. Non-designable regions, symmetry, minimal member size and design for extrusion. We're going to be adding to this milling and ALM so you can do 3D printing and probably a few more.

So what is minimum member size and how does it work? Well, if we look at this structure right here and we apply a shear load on it, we're going to get something that looks like that. And we totally expect that. I mean, most people would agree that that makes sense.

But what happens is, as the mesh gets finer and finer, you can get very small, thin members created because it thinks those are more efficient. But it's not factoring in things like buckling, necessarily, when it's doing this. So most people say, well, you know, I don't want to manufacture something that's super thin. And they limit the dimension. And it actually is built into the code right now that we have a certain minimum member size built in so you don't get ridiculous things like one element in a really fine mesh or something like that.

All right, so this is just looking at volume fraction and compliance, OK? So before I showed you a whole laundry list of different objectives that you could minimize or maximize and constraints. This is going to be we're going to minimize compliance and we're going to set the volume fraction. In other words, I want it to be this much lighter and this is what happens.

So what we're showing here is the effect of mesh density on these designs. Obviously, as we increase the mesh, we have more options for it. And you can see how the designs get more complicated and probably stiffer, too.

And also you can notice that, over here, for the volume fraction of 0.8, it's going to use a lot more material than 0.3. It's removing a lot of material. But we haven't looked at anything like stress or buckling or anything like that yet. This is just simply the volume fraction compliance.

Now, if we start looking at other types of constraints, this is a pretty good problem. It shows you all three happening here. So we've got our preserved boundary, and that's that red area. That means we're not going to touch that. We're not going to change it.

It's usually a place where loads are applied or some type of boundary condition. It could be an interface in a complicated part where you have fasteners. And in this particular case, we're saying that it cannot move more than 0.3 inches in one direction and it can't move 0.3 in the other direction.

And it has two load cases. And each of those load cases have those loads that you see there. And what the code does is it just simply goes in and it finds the maximum displacement at the end. It picks that point and it says, I'm going to watch this now and track it.

And you can see the designs. If we say, hey, this has to be extrudable, we get that design there. And that makes sense. It's not symmetric. One part of it actually is bigger. If you look here closely, this is bigger. And then that's a little bit less. And then that's a different size. And then that's the smallest right there.

But if we say symmetry, now we get this over here, which obviously looks a lot heavier. And it's removing material here because, for symmetric, it can. And then if you say both, then you get a design that looks like this over here.

All right, so here's some examples of how we would use this today. For example, you could go in and say, I want to minimize my mass. And I'm going to say, OK, I want to make sure that the displacements at a point don't exceed a certain value. And I'm also going to make sure the stresses are below some limit.

And people will talk about factor of safety. Well, we don't support that directly. But we don't have to because all you need to do is change your allowable to a lower value and then that'll accommodate the factor of safety. Or you could say reaction force.

At my bolted connection, I know that my bolt can only handle, say, 2,000 pounds. Don't exceed that. Sorry. Or you can have something like this where you minimize the mass and you have compliance index.

Now what that does for you is it allows you to go in and say, my model can only be three times less stiff than it was originally. And that's helpful because then it won't remove all the material.

Or, for example, frequency. This is pretty common here, where you'll say I need my frequency to be above a certain value, or the buckling load factor, for example.

So what happens traditionally is a lot of optimization programs out there, like HEEDS, for example, will take an existing general purpose finite element code and it'll run it. And basically what it does is it says, OK, we work with all these codes. We're going to run the code and we're going to get some results and we're going to run it again and run it again and run it again. And it's inefficient because all the stuff that you're doing in the beginning, reading in the model, writing out all the data that it needs, slows everything down.

So a better way of doing this is to do it based on the FEA solver doing it internally. And for parameter optimization and topology optimization, it works really, really well. And that's what we've basically done. We're not running the solver over and over again externally. We're running it inside.

And by doing that, we can make it very efficient. For example, we don't have to calculate the element stiffness and mass matrices repeatedly. We just calculate it once and then we scale them back and forth, which allows us to go very fast.

So this is the framework that we currently have. And I know this is an eye chart and everything. Again, it's really good for reference so that when the code's running and you're wondering what it's doing, this is basically the operations that are taking place. And all this is going to be-- I'm going to demo it here shortly and you'll see what it's doing.

This entire presentation, up to the point where Mike's going to talk, just focuses on the engine, the solver, the Nastran solver. When that gets embedded into our other products, it'll be behind the scenes and you won't notice it. You'll see information externally. But this is actually what it's doing, so people can appreciate that.

All right, so we have two methods that we use. The traditional method here is the optimality criterion method. And that method is actually very efficient. It's scalable. It works really well.

But it only works for volume fraction compliance. So all the stuff you've seen a lot of in topology optimization where people, you know, I want to reduce it by half the mass, whatever, They're going to use this method. It's going to work really well.

The other method that we use is this moving asymptotes method. And we use this because this is the only way-- or one of several ways, I should say-- of doing constraints, multiple constraints, and having different constraints, for example, like displacement. And the reason why this works is because we use what are called adjunct loads. And I won't get into that, really, in this class. But just suffice it to say that it's going to involve a lot of extra information that it needs to actually do this analysis.

And we looked at the scalability. So the OCM scales really well. And, by the way, this engine, this is going to be the bottleneck when you're running topology optimization with lots and lots of constraints. It's not going to be the finite element stuff that it's doing. It's going to be it computing the design because that can take a lot of iterations.

So we want to paralyze this. And right now we've seen as much as 11 times faster speed ups and scalability with OCM. But MAM is about four and a half. And it's not that that's going to stay that way. But that's what it is currently. And we're going to improve that and make it faster.

Now, one thing that's very important to understand is you'll see a single part. And like the models I've showed you up to now, you'll have the non-designable region and then you'll have the design region. You can actually have many, many design regions.

So you can have an assembly, you can have different materials, whatever you want. There's no limit on these. Actually, I think the internal limit's 1,000. But, basically, that's controlled by this bulk data entry called TOPVAR.

And what that allows us to do is specify all the manufacturing constraints and different parameters. So you can say, on one part of the model, I want it to be extrudable and I want it to be symmetric. But then another part, you can say, I don't need that and I want it to start with a different volume fraction.

And another important point here, this last bullet, is that, again, if you have a part that's made of, let's say, aluminum and steel, whether it's fused together, welded together, or it's an assembly, you would be able to, with the current code, have two separate design regions and have two separate stress allowables. So we actually pay a lot of attention to this. And when we built the code, we made sure to account for things that people are going to want to do in the future.

All right, so this is an example of design space. And, basically, what we're saying here is, let's say that's my design region and I go in and I design this simple thing. All it can really do in this case-- and, by the way, I am using an extruded manufacturing constraint. And you'll see that here in a second. But it's fixed at one end. It's loaded with a sheer load at the other end.

I'm going to limit the displacement in this to 0.25 in the direction that the load's applied. And I want to minimize my mass. And that's my goal. So I have a constraint that's displacement. My objective is to minimize the mass.

And I have two models. I have one, that's this one here. That's my existing part. But then I have another one where I said, well, let me expand the design space. I'm going to add more elements over there-- those are the yellow ones-- to see if I get a better design. And indeed we will.

So you can see here, there's the original model. And then you can see the new design. And it's a 24.6% reduction in mass. But it really can't do much.

Now, on the other design, it comes up with a way more efficient design, which is obvious to engineering because it's able to take advantage of the moment of inertia. And there we get a 69.7% reduction by just expanding the design space. So, often, we'll have an existing part and we can easily remove material for an existing part. But what's great about this is that we can expand the design space around that existing part and maybe get an even lighter design.

Now, I included this slide only because I think it's important to understand that the basic model that we're dealing with is an Nastran bulk data file. And what this shows is what's actually getting added to it to turn it into a topology optimization, OK? And, basically, the way it works is you have an objective, you have a constraint, and then it's referenced.

And then there's that TOPVAR bulk data entry that I mentioned before. I didn't want to get into this too much. But I thought it would be interesting to show what's under the hood. And that's it there.

So I'm going to show some examples now. And this is going to be, really, the important part of this presentation. So I came up with this model because it's a model that we all can relate to as engineers.

We have a non-design region at the bottom that we can't touch, OK? And it's got a constraint on each end. It's not all the way on the bottom. It's just at the corners, OK? And we have two loads at the top in those directions there.

And what we're going to do is we're going to do static loads and we're going to do thermal gradient. So we basically have a temperature gradient that's going on this thing. And you'll see why here in a second.

And our constraints are going to be stress, compliance index, buckling, the load factor, lowest frequency, and maximum temperature. We're going to do all of these individually and then we're do them all at the same time. And we're going to use a symmetry constraint.

So if I go in and I do these separately, OK,-- and this is the really cool thing here-- we know that that design at the top there makes a lot of sense, OK? You can just see it. Yeah, OK, that's what it should be. And that's with a constraint less than 400, which is allowable for the material.

And I reduced the allowable to factor in the thermal effects that we're about to see. And I've also said that the compliance can't be any less stiff than one fifth the initial stiffness, all right? And that's very helpful because then that way it won't remove all the material.

And, for buckling, we get something that looks like that. It looks a little odd but it does make sense because it's going to want to buckle out of plane. It's going to want to bend over, not buckle like you would think, like one of these things buckling. It's going to buckle out of plane.

For normal modes, I first looked at them and I go, that's kind of weird. But, no, it makes sense because the frequency is 20 Hertz. It has to be above that. Well, it's already above it. It does a normal modes analysis, it removes all the material, and it's way above 20 Hertz. It's at 44 Hertz.

So it does what you told it to do. You said, remove all the mass and it did. And then what's left is above 44 Hertz. And then for linear steady state heat transfer, the requirement is that, at this point right here, there and there, we don't want the temperature to go above 200 Celsius. So you end up with this heat sink. It's quite interesting.

And, by the way, this is, for the steady state heat transfer, its volume heat addition. So basically it's like a-- hold on-- volume heat addition. So what it is is we're dumping power into this like a CPU, OK? And then this is the heat sink. And it's got convection. And so it's convecting away the heat up here and also conduction. But then that's the heat generation down there.

So I'm going to start off by not including the steady state heat transfer. And the reason why I don't want to include that is because it's going to create a totally different design. Basically, when you look at linear statics, normal modes, buckling, they kind of all want to work together. But heat transfer doesn't because it's a totally different animal. And this is really important because, without connecting these all together, the final design is not intuitive.

So I want to go back to the mass thing because I was so curious about this. I said, well, wait. What if I put a point mass here and now run the analysis again? And now it designs a structure.

And it needs to because what's happened now is it has to connect up to that. Because if it doesn't create rigidity, the frequency is going to be really low. So it just shows you.

And I don't want to include that point mass because you'll see, in the end, it'll only reduce the mass of the part by seven and a half percent because so much mass is already in the top. There's not much it can do to reduce it. But it just shows you the effect of point mass and that the normal mode is working and it does what we expect it to do.

So now, if we run this and we run it in a compliant design, we're basically saying, well, run all these together and give me a design that satisfies the fact that my stress can't go above 400 megapascals. And I don't want it to buckle. So that means the buckling load, the eigenvalue, has to be greater than one, which it is. I need the frequency above 20 Hertz, and it is. It's at 234. And we've met all of that.

And that's our design. And, again, it kind of looks like the original design. Well, it added some more material at the base and a few changes here. And we have a mass reduction of 76.5. And everything is great.

Now, if this was my requirements and I didn't have to deal with the heat transfer, I'd be in great shape. Boy, I'm happy. But the problem is I have to deal with heat transfer.

So now I'm going to run this thing with heat transfer. And this is going to show you how it evolves rather than animating it right now. Basically, what I did is this is where it starts off. That's iteration 15. That's after 25 iterations. This is after 35 iterations, after 45, 69, and converged.

And you can see here that the part basically has those fins, like you would expect in a heat sink, to radiate out or convect out the heat. What we end up with now is 57.5 versus 76.5. So it's definitely heavier. But it satisfies all the requirements. And to figure that out, I never would have guessed that.

So this slide here is the demo I was talking about. And what this is is showing all four runs actually running. And you can see there the linear statics is on the left. And this is what I call the independent solution. And these three over here are the dependent.

So what this thing's going to do is it's waiting, OK, for these three over here to finish and give it their information. And basically that's their constraint and-- excuse me, their constraint and objective data. And then once it gets that, it's going to crank those out and come up with a new design and feed that new design back to these other runs. And they're all basically communicating back and forth with each other, sharing information.

But the actual design is being created by this master process or independent process right here. And I know it's scrolling by really quick and it's an eye chart and you can't see it, but basically it's telling me right now whether the constraints are passing or failing, where the objective is. This thing's doing normal mode, so it's taking longer. And buckling also takes longer.

So occasionally this thing has to wait. But the beauty of this that makes it so awesome is there's really no reason why this can't be on separate computers or on a computer with, let's say, 100 CPUs. And you don't have to limit it to four. You could have 40 or 100. And they could be doing different solutions or different constraints. But the requirement is the model has to be the same and the objective has to be the same.

Now, does anybody have any questions up to now? No, good. Excellent.

All right, so these are the results. And basically what we want to do is I went back and I made sure we saw the model. But I wanted to make sure that the stress indeed was below what it should be. And this shows that we end up with an eigenvalue of 1.2 for buckling. And that's important because if it was less than 1, it means it's going to buckle, So it's not.

And then the frequency's 255. And the stresses are actually below 10 megapascal. So it's not really being driven by stress in this model at all.

One of the things that Nastran does, it will actually write out a Nastran bulk data file of the design, the optimized design. And you can just take that and run it. And what I did is I ran it from the different solutions to make sure that, indeed, the topology optimization wasn't lying to me.

And, yeah. OK, the buckling mode's 1.35. And this could be your design, basically, if you wanted to use it. It's jagged. But it does, in fact, meet all of the constraints. And it's all up there to see.

But if you go back now and you create geometry and then re-mesh it,-- and the reason why you want to do this is because, for stress, we have these little radii in here. And those usually are stress concentrations. So I went in and I did a fine mesh-- you can see how fine this thing is-- and re-ran it.

And now this is our final design. The eigenvalue for buckling is 1.5, still above one. Frequency's still around 230. Volume heat addition, I'm at 184 degrees right there, where I can't exceed 200. And my stresses are all below the 400 limit for the two cases.

This is the thermal load. By the way, the thermal load does drive the stress pretty high and it gets really high and very close to the limit right in here and here because the thermal expansion is going to want to bend out those radii.

Now, let me explain global stress. People think that when you look at topology optimization that, well, why aren't you looking at the highest stress element? Why aren't we driving the constraint on that?

The reason why is because you can't do that. You have to have a global measure that represents the entire model. So we have this concept of global stress. And I know that the equation looks really nasty and hairy, but it's really very simple.

What we do is we take the density of the element, we take the square root of it, we multiply it by the von Mise of stress, OK? We divide it by the limit that you told me. You said, well, I can't go below 400 megapascals. So we put that there.

And then we raise it to this huge number. It's 10 or something. Or, actually, the exponent's 10. And then we sum all those up for every element. And then we divide by that and we take it to the power of one tenth, OK? And that's a global stress.

Now, if you think about that, what's going to happen is that if you have a few elements that have really high stress, that's going to be represented very nicely in this value here, OK? And that way, when it looks at the constraint, it knows that stress is a problem in some part of the model. Maximum stress is discontinuous and oscillating. And this would really be a problem because it would drive the optimization algorithm nuts. And that's why we use this, OK, because it's continuous.

Now, to make this better, what we do is we have this concept of buckets. And essentially what we do is let's say we have 2,000 elements. Well, we're going to divide them up like this, based on the highest stresses.

So we're going to take, in these subdivisions,-- and these all actually end up being separate constraints-- we take one stress constraint and now we're going to make it into five. And this is something the user can change. And I'll show you that here in a second. And we put the five highest elements in this bucket and then the next 20 in that and 87 and on and on.

And we update that on and on. Every iteration looks at that and it makes a change. And eventually what ends up happening is that we're able to zone in on that region where the high stresses are so that, in the end, when you have your final design, it indeed doesn't have a stress above your limit. And I'll show that here in a second.

So this L bracket is very popular. It's a standard benchmark in topology optimization for stress. And you see it come up all the time. And the big challenge here is this notch. Because what the topology optimization engine has to do is it has to work around that.

And you can see, in this case here, we say that max is 1.2. And I'm going to minimize mass in this model but my constraint is stress. So I can't go above a certain stress, That's my constraint, minimize mass.

As we add more sub regions in here, we get closer to that limit. So it's failed, really, here. The peak stress ends up being higher. Now, you're going to say, well, how come the topology optimization kept going? Why did it converge?

Well, it converged because the global stress, it was right there where it needed to be. But the problem is that there's still elements that are above where they need to be because the global doesn't look at one element specifically. It looks at a whole bunch of them together. So this is a way to make the model represent what you want it to represent.

So then, again, another case here, if we raise the limit to two, we get these designs. And you can see they all look so similar. But basically what this shows is that you need to have around

10 subdivisions for each stress constraint.

And that gets back to why it's so nice to be able to break this up into lots and lots of processors. Because you can imagine, if I have, let's say, 100 load cases and I use stress, that means 100 times 10. And that's how many constraints, 1,000, that you're going to be dealing with. So it's going to get really busy.

So if we look at the error here, just in those plots, you can see that by doing that technique, we're actually able to get pretty good results here. If we didn't, if we use a constant, then we have this error. So we've proved that our technique works and the whole concept of buckets works really well for stresses.

Now, here's some common parameters that are useful. I included this more for reference. But these are the kinds of things that, if you're running topology optimization in Nastran, you want to possibly mess with. Everything else I wouldn't touch.

But things like the maximum number of iterations, it rarely goes to 200. But if it does, you can extend it. What will happen is, when it gets to the limit, it basically says, I'm done. And it will give you results but it didn't converge. And then those are the divisions there, right there, that you can change.

And then these are some of the things for convergence tolerance. And you can see here it's usually a trade-off between performance and accuracy on all of these parameters. All right, so now we're ready. Does anybody have any questions at this point?

AUDIENCE: What kind of memory and hardware are you talking about?

DAVID WEINBERG: For, let's say, a big model, let's say it has a million elements, you're going to want a machine that's got 64 gigs or at least maybe half that, and four CPUs. Yeah, definitely. But more is better, all right?

And that's just the reality of it, the number of computations. Now, I'm talking about something with lots of subcases. You've got, let's say, ten subcases and 100 constraints and all of that, yeah. OK, thanks.

MIKE SMELL: Thanks, Dave. Great. Great stuff there. Can everybody hear me all right with this where it's at?

OK, so, as I said, my name's Mike Smell. I am the product manager for the simulation functionality inside of Fusion 360. I'm going to talk to you all about shape optimization. That's the study name inside of Fusion 360 Ultimate. And I'll put one of these on each side to pass around because it'll be relevant to where we go with the conversation.

So one of the things I'm just going to point out real quick, we came up with the study name of shape optimization, really, for two reasons. One is that we have some longer term visions around what we're doing inside of Fusion. It's in tech preview now, so that doesn't mean that that's its ultimate end state. So what you see is more akin to true topology optimization as Dave defined it. But we've got some longer term plans to go beyond that.

The second is we got a lot of feedback as we were starting to go down the path of doing topology optimization. Many of the Fusion users, even many users in the simulation space that we were looking at, really struggled with the concept of what that actually is. So that's why you see that name. So I wanted to put that out there so that there's not a whole lot of confusion as we go through this.

So the first thing that I want to talk about is the workflow. And how many folks in here use Inventor? Good many. How many folks who use Inventor have used the shape generator tool inside of Inventor? OK, good many.

So this recipe is very much similar to what you're going to do in Inventor. A lot of what you're going to see is similar in nature to what you can do in Inventor as well with a few subtle details that I'll try to point out along the way. So there's going to be six bulk steps that I will talk about today and you'll see as we go through the demonstration in Fusion.

So the first is, really, around developing the design space. And the design space can be, really, a couple of different things. The first is, in some of Dave's examples, we started out with a very generic design envelope. We told it where it was constrained, where it was loaded, and we let the solver come up with what the shape should be. And that's really good for the initial conceptual engineering phase, if you're trying to start with a blank sheet of paper, to do a design.

The alternative is you can start with an existing design that you have, that you've been making for years, and use this as an easy process to look at, what's the best strategy to lightweight that part further without having to start from a blank sheet of paper? So we we'll look more at the design envelope example and the things that you'll see here today.

The next thing that you will do is define your simulation settings. So behind the scenes, as Dave talked about, its linear stress, its buckling, its motile, its heat transfer. All those things are things that the solver is working with. We have only a small subset of that available in Fusion and Inventor, for those that have used it. But, generally speaking, you're still doing a traditional FEA simulation.

The next thing that we'll do then on top of that is define the optimization settings. And Dave talked a lot about the objectives and constraints. And that's ultimately what optimization settings are. And, again, in Fusion and Inventor, what we have today is a small subset of all the things that Dave's been developing in the solver.

Once we define the settings, we're going to go ahead and run the study and we'll review the results. And that's going to let us look at [AUDIO OUT], how much mass have we removed. In Fusion, we can move the slider to look at the isosurface. And then we'll make a decision about what mass target that we've set, based on the setup that we have, is what we want to proceed forward with.

And then we go into the workflows that we have built into both in Inventor and Fusion today. And that's where we're promoting that mesh and essentially using that as a template to guide us in the design or the model workspace to improve our design.

And then we want to then go through the modeling process. You'll see in the examples that I have. We'll update the design region to what we might call our final design or near final design and run it through a stress analysis, model analysis, whatever it is that we need to validate that design to ensure that what we started with meets all of our operational requirements, OK?

So I'll be jumping back and forth between PowerPoint and the software. But I want to start by setting up what we're doing. So as I've been working with the simulation technology that Dave provided us for Fusion. I've been engaged with a number of different companies, looking at how our technology works, how it fits into users' workflows who are doing this for production purposes.

This is an example that I've been working on with GKN Aerospace. This was brought to us by my colleague, Chris, from the Delcam team. And what we're looking at is optimizing the shape for an aircraft elevator hinge. And this is more akin to the design envelope example.

We had another design that I can't show here. But it was taking the original design and using the topology optimization to shape optimization solution to remove material from the existing design. So we're going to look at just this one.

And what we're talking about is we want to optimize the design based on the envelope that we were given. We know what the material is. We know what the factor and safety requirements are. We know what the displacement requirements are. And we've got a number of different load cases.

And multi-load case in Fusion is something that's going to be different than what you're able to do in Inventor. And we'll see that when we go into the software. So this is what I started out with.

So what we'll do is jump into the software now and I'll take you through the process of how I set this model up. I'll talk about specifically some of the characteristics of the design space and the functionality that we have inside of Fusion. And then we'll jump back out. We'll talk about some different result things that we want to pay attention to and then we'll look at those in the software.

All right, so here we are inside of Fusion. And the shape optimization study type, for those of you who are not familiar with Fusion, is new as of last Wednesday. So if you want to check out Fusion or you want to see this in Fusion, it's brand new. So it's pretty cool stuff.

So when we look at this design envelope, I want to point out one thing that's quite interesting. So what typically happens when folks do topology optimization studies and, in this case, shape optimization, in Fusion is there is this concept there, as Dave mentioned, of design regions and then non-design regions. And the non-design regions are the places that you typically want to have your loads applied or your boundary conditions applied such that you're not going to remove material from those key locations.

So, in this model, if I look at bodies, you'll see here that we've got a number of non-design regions that we've built in, OK? So those will be our non-design regions. And you'll notice here in the browser, marked in red, to guide you as to what to do next, we say, what is our target? And the target is going to be the design region. So I'm going to go ahead and edit this. And I will pick the main design envelope space.

Now, one thing that's very unique about the implementations that we did in both Fusion and Inventor, when we look at some of the competitive tools in this space, like Inspire or OptiStruct, they require this workflow where you're bringing the model in as an assembly, at least as of two or three months ago when I was last looking at it.

What we do inside of Fusion and Inventor is we give you this ability to generate virtual preserved regions. And this makes it a lot easier to work with the model and not have to go through all these cut CAD modeling steps, especially in the context of if you're starting with this blank design envelope.

So I can pick a centroid on the design. If I was trying to mimic what they did here, I might pick that circular hole. I can adjust the size of this, and create that same thing without having to have made these modeling setups in my design space. So this is something pretty unique to our workflows.

I wanted to point that out because it may save you some time in the CAD setup process or the CAD prep process for these problems. But I got this file from the GKN team and they prepped it for exactly the purpose of what we were doing.

All right, so once we have our typology settings defined, or our design space defined, we can then go through the standard process of setting up an FEA model. So we can define constraints. And for this scenario, we were using an axial direction, only pin constraints. So I can simply just go ahead in and pick the surfaces and constrain these. And I will save you the time of watching me do it on the bottom, but you get the concept.

So we can work through putting on our constraints. And, very similarly, we can do the same for load. So, in this scenario, we were using a bearing load,-- actually, we were using vector force with this model-- where I can pick the two surfaces of interest and I can say my vectors. And I have the magnitudes here that I'll just type in quickly so that we can see what happens.

All right. All right, so I've applied the load. And you can notice here in the browser, we have the information about where our loads are applied, where our constraints are applied. And one of the things that I mentioned in the setup process was that this was a multi-load case exercise.

So what we do in Fusion is we give you the ability to simply clone the load case. And what I would do is I would do this for the number of load cases that I am looking to solve. And in this case, the only thing changing between the load cases are the directions and magnitudes in my load. So this is making the setup process quite easy.

And then I can come in to load case two, double click on this, expand the load heading, and simply edit my forces, OK? Now, once you have your forces defined, we will come into the optimization criteria. And, as I mentioned when we started, what we have in Fusion today in the tech preview is a small subset of what Dave has built out in the solver. So we have the ability to define a target mass, which is that volume fraction that Dave talked about.

So here I have a value of 30. And I set the objective of maximizing stiffness. Now, as Dave said, they have some intelligence built into the solver around minimum member size, where, automatically, we're trying to keep three elements together so that we avoid these spinier sections that are super thin as the mesh density goes down. We can define a minimum member size, if you want to do something outside of that.

But be aware that, in its current implementation, and logically, we don't want to define a minimum number size that is smaller than our smallest element. Because then it can't create that if the elements are not small enough to make a member size that's smaller than that. So that's just one thing to pay attention to.

So after we do this, I'll point out just a few more things just so that you can see some of what Dave talked about as far as constraints go. We talked about the preserved regions. That was the non-design regions that Dave talked about in the solver. We looked at the objectives and constraints in the optimization criteria. We also have the ability to define a symmetry plane.

Now, in the case of this model, while it looks symmetric probably from the distance that you're looking at it, there is a slight growth from the right to the left side of the geometry. So it's not terribly applicable here.

The other thing that we can do inside of Fusion is clone studies. And this is important because one of the things that I will show everyone is, as I went through this exercise to learn more about our tool and to come up with a better design for GKN, I wanted to understand what the mass targets were going to do for me and see what kind of shapes I would get.

So in this set up here, we have a target mass defined as 30. So I can simply clone this study just like I cloned the load cases. All the load cases that I've defined under that study will come along with me. And I can simply come in and edit the mass target.

So in this case, maybe I'll look at something like 10. And I can do this for a number, as many as you're willing to look at. And then I think everybody heard about cloud simulation. In Fusion, it was part of mainstage.

At that rate, we go off to solve. I don't have these completely set up. But I can select all these studies, send them up to our cloud. They'll all be solved in parallel and you'll have all of these studies come back such that you're able to post-process these results side-by-side. So let's take a look at some of those considerations now.

OK, so when we start talking about results, this is an example of multiple mass targets with the multi-load cases set up to find exactly the way the problem should be. And you'll see here I went through a couple of different ones. And the way that the product is set up now without stress or displacement constraints built in, you're going to use a little bit of engineering judgment and gut feel on which one of these is the best to move forward with. Because then you're going to need to do some modeling adjustments, which we'll talk about in a bit.

The other one that I want to point out is the notion of multi-load case. And this question comes up a lot internally when we talk about, what is the importance of this? Well, in the case of this elevator hinge, or many designs, there is potentially different load scenarios in its use case, so take-off, landing, mid-flight, or if you have any other type of bracket that has a tensile load, a compressive load, a moment load. These are things that we want to consider.

So if you notice here in the upper, to your right, quadrant, this is looking at an x direction load, a y direction load, and z direction load. And the only difference between these two objects is, when I had the load case definition, I had to look at the same x and y loads but both direction z loads. So in and out of the page, positive z, minus z.

And if we look at that, as you would expect, when the z direction load is positive, we're going to carry the load on one side. When the z direction load is negative, it's going to be on the other side. But during operation, this bracket's going to see both of those.

So this design may not be adequate for the positive when it was designed for the negative and vice versa for the bottom. So that's where we want to use multi-load case. And you'll see that this result is quite a bit different than these two here, OK?

So let's take a look at this in just a little more detail inside of Fusion. And I'll point out some of the results options for post-processing this. So here we are inside of my main design study. This is where I did multiple mass targets just so I could learn about the model. It has all of the load cases that I talked about in the set up.

You'll see here, this was a six load case model. And I'll jump over to the results. And what we're going to give you is a load path criticality result. And this is basically the load path of the forces that you've applied moving to the boundary conditions with a critical density, as Dave explained, that's required to carry that loading direction.

Now, what we can do is look at, with the slider, how material is removed from this model. So we can see what elements get removed first. As we start to hone in, can we come up to a potentially easier shape to make, understanding what we want to use as our template. So we can move this around and learn about the model.

The other thing that is a new addition to Fusion but is super handy in cases like this, and, really, any simulation cases, is our compare workspace. So I am able to set up a number of views. I like the vertical one a little better.

And here I can look at 5%. I'll set this one to 50, 30, and 10. These are the experiments that I did. And now I'm able to interact with the model and see how all of these designs differ and what characteristics go away as we reduce the mass. And, again, this is just learning that's going to help me make some engineering judgments as I go through and do some of my model updates, OK?

And very similarly, just to make it more visible, this model here, I've done the multiverse single load case. So in studies one and two, I only have a single load case active. You'll see here load case two. If I come down to study three, I've got load case one and two.

And if I bring these back over to compare in a similar fashion, I'll go ahead and take a closer look at the results. And you can see we looked at a single-direction z load. It gives us a different shape based on plus or minus. And then, in this case, where we've combined those, we have a different shape, OK?

All right, so let's talk about the next step in this process. So now that we understand the results, we have to do something with them. Because you don't want to just get a rainbow picture. You want to make a design improvement.

So the workflow that we've established now-- and this is common in Inventor and Fusion-- is we have the ability to promote the mesh back to the modeling workspace such that you can use that as a template to guide your design process. So let's take a look at what that looks like. So I'll come back to the exercise that I did with my full set up.

In the model that's passed around, you'll notice that it looks very similar, to the best of my modeling abilities, to what is number five. And I'll talk about a couple of things to pay attention to in the modeling process when we get there. But here I am in the results environment.

You'll notice that there is a button on the toolbar called Promote. And that's going to let me send this model to the modeling workspace. And it automatically comes over as a mesh body. So when I look at my design geometry, it gets nested under that. And I can toggle that on and off.

And at this stage, I would just begin my modeling process using all the intuition and knowledge that I have about Fusion and inventor. And I would start sketching out some planes. And we'll look at how I did some of that.

So for this model, like I said, there was some uniqueness in that, if I zoom in here, you can see that, into the page, there was a slight taper. And just that slight taper made some of the modeling steps a little tricky. But in any case, I started out building a number of construction planes that defined essentially the outside of that envelope. And that was helpful in building some of my cuts and extrusions.

And you can see I've got a long timeline here of where I started removing material. And if we use the play feature, you can critique my modeling and see the process that I went through to start using sketches, using extrusions, doing some booleans, making some fillets, and adjustments. But, at the end of the day, I came up with something that's reasonably OK.

The one thing that I do want to point out is-- and this is something that I overlooked because I've not designed an elevator hinge before. I've been an FEA guy my entire career-- we talk about preserve regions in the context of the design envelope of places where we don't want to remove material. But there's also another consideration that you need to think about where you can't have material.

So one of the things that I overlooked was, in this model, on the top,-- probably because it was earlier and I felt a little bit better about my modeling skills-- there's plenty of eyesight room to get a tool into those mounting locations. And when I got to the feet, I wasn't quite as careful about, how do I get a tool in there, right? So when you're developing that initial design space, this may be something that you want to consider as well, is to give yourself room if you have mounting locations that you know are going to be critical or that can't change.

Because one of the things in the design space is we have the option to allow material to move away from boundary conditions. I didn't show that in this example, but we can do that. So something to keep in mind when you're using this tool, that there has to be some forethought that goes into how you might do that to handle your non-design regions and to make sure that you give adequate room for that.

Now, at this stage, we've updated our design. We can use all the functionality at this point that's inside of Fusion. We can go into the rendering environment, create a rendering. If this was something that we wanted to make with subtractive manufacturing processes, we could use the CAM environment to generate the g-code for this. I didn't do all of that. I actually talked about 3D printing, which we'll see in just a second.

So based on this design, I applied my loads and constraints as such, my six load cases. We said that the factor of safety couldn't be less than 1.2. So I've adjusted my legend here. And you see there's only just a few very, very small regions around my boundary conditions.

I do have some mesh anomalies there. But, generally speaking, we did a fairly good job designing down to the targets that we had. There's the displacements as well. We said that was limited to 0.55 millimeters and I think it's something like 0.53 millimeters in this design, OK?

So you guys all have the plastic model getting passed around. Working with our colleagues on the Delcam team, we are going to print this in metal for GKN as an example. I might have to do some design updates to that bottom section first to make it a little bit more actually usable. But, in any case, this is part of the project that we have.

So I'm going to pause here for just one second, see if there's any questions about what we've talked about so far. OK, in that case, I'm going to give you a glimpse into the future.

Some of the things that we're working on inside of Fusion. And Dave has essentially already given you a glimpse into this because the solver's further ahead of what we've implemented inside of the tool. So we do have working prototypes of stressing displacement constraints. And you can see here how that does, in fact, change the geometry. We've got mass target only. This is essentially the set up that we had in the example that we just looked at and what we used to develop the bracket.

When we add a displacement constraint, you'll notice this rear section changes a bit. And then when we add stress constraint to that as well, that rear section changes a little bit more. We have a pocket built out here and we've got two spars carrying a load instead of just one. So these constraints will definitely drive differences in your designs, as Dave demonstrated with some of his solver work.

Now, this is probably the most exciting piece. And this is another working prototype. And you see that I have it working here. So as you saw, I had a very long timeline inside of Fusion because I'm not an expert and I had to really think my way through how to model that thing, cutting three different directions, not getting really funky features on the model. It took me a pretty good amount of time to get to something that I was happy with.

We are working on the ability to convert that mesh, smooth it in one click, convert it to BREP solid. So in this scenario, I ran my topology. Similar to the Promote button, rather than promoted as a mesh, I promoted it to a BREP. This was one click, took it back into my simulation environment, meshed it, loaded constraints, and I was able to look at stresses and displacements or any other simulations that can be done inside of Fusion. Yeah?

AUDIENCE: Will you identify the constraints of that?

MIKE SMELL: What's that?

AUDIENCE: Will you identify the constraints of that?

MIKE SMELL: So David already has extruded, working in the solver. They're working on additional technology. It will just be a matter of time until we, as a consumer of the solver, hook up some of those things, OK?

And then the last one that I will talk about is another prototype that we've been working on. You saw some of this concept around generative in mainstage. And one of the things that we're working on there is the concept of lattice.

So we took this design. And in this example, I won't go too much into the details of the subtleties of the design process. But, at the end of the day, I've done a topology optimization workflow, or shape optimization workflow, inside of Fusion. I went through the modeling exercise to take it down to that more optimized shape.

And then I said, well, what if I wanted to lighten that part further? Well, in this case, because I have already went to a near-optimized design, but if my requirements were different or if I would have designed this to the 10% mass target or the 15% mass target, I could design there and then add lattice to further lightweight. And in many scenarios, if this model was designed down to 1.2, if I were to design this to maybe two and then latticed down to 1.2, I would most likely end up with an even lighter structure than this and it's different structure.

So this is something else that we're working on. And if anybody is interested in this, we're trying to collect some feedback on what this workflow should look like as far as-- we're in the design workspace. We do a shape optimization to do the topology workflows that Dave is supporting us with, further than latticing that to get to an end-state optimized design. So if there's anybody that's interested in talking about this, I'll be around for a little bit longer today. I'd be glad to engage with you on this, OK?

We can skip that slide because I put it in accidentally. But that's all I have. So I hope that what we just talked about for the last few minutes gives you an idea of how to take advantage of Dave's typology technology inside of Fusion as the shape optimization study. We definitely want to hear your feedback about it.

And, again, most of what you saw is applicable to Inventor. So all the folks who raised their hands as Inventor users, these same strategies and that workflow of design space, simulation set up, optimization set up, understand results, promote and modify design, that recipe holds true for Inventor or Fusion, OK? So I think we have time for-- any questions? Or anything to add, Dave?

DAVID WEINBERG: Yeah, I have some questions. I have two of these. And the first question is, who can name two constraints that we typically use or talked about up here, design constraints, one of these?

AUDIENCE: Heat transferance.

AUDIENCE: [INAUDIBLE] [INTERPOSING VOICES]

DAVID WEINBERG: I heard these two here. Heat transfer? What did you say?

AUDIENCE: Heat transfer and load.

DAVID WEINBERG: OK, is it-- here you go. And you said, too, I think. There you go.

AUDIENCE: Thank you, sir.

DAVID WEINBERG: And now we have some of these. I don't know if you guys have enough of these yet. But name one manufacturing constraint that we talked about.

AUDIENCE: Extruding.

DAVID WEINBERG: Who said that? Here you go. All right, and, let's see, you have any questions?

MIKE SMELL: No, but I do have two more points to cover on this slide. So has anybody been down to the Idea Exchange yet? It's to the left side of the answer bar. OK, I would ask everybody please go down-- please join us down at the Idea Exchange.

My UX partner, Tyler Henderson, is down there doing research around Fusion simulation. We also have some questions we'd like to ask about the generative stuff with the lattice, which was the last slide that you saw there. So if you have time, we'd love it if you'd come down and spend some time with us.

AUDIENCE: I have one question.

MIKE SMELL: Yeah?

AUDIENCE: How are documentation [INAUDIBLE]? Generally, the Fusion 360 documentation is a bit lacking, right, bits and pieces everywhere. If you're fairly new to that subject, [INAUDIBLE] and other stuff that you just taught that I remember from [INAUDIBLE], but it's so far back. So how do you refresh it if you haven't had some sort of documentation to get back into that?

MIKE SMELL: Yeah so our strategy around that, with the 11/9 release, we've done a complete solver overhaul to Nastran. So everything inside of Fusion is using Dave's Nastran code. We're putting a heavy effort in getting started in usability.

So, in the past, Fusion had just a couple of getting started exercises. We've got tutorials for every single study type that we've added, including the tech previews. So that's there for you. The event simulation solver that-- I think Lee was here-- we added five new sample models in there. So they're set up, ready to go. You can see all the things that the event simulation solver can do.

But that is a very strong focus for us going forward, is to build out the documentation, have a lot of strong samples so that our user community can become more proficient with the software. And I would encourage you to go down and see Tyler.

Because there are a couple of specific projects that we're working on around usability and coaching for not only choosing the study type but interpreting the results. And these are some things that we think could really help us differentiate our technology as well as help users be successful with simulation. So we'd love to hear feedback on that.

AUDIENCE: Is this a separate product? Or does Inventor 2017 [INAUDIBLE]?

MIKE SMELL: Is what part of it a separate--

AUDIENCE: Nastran. Is Nastran something you have to buy?

MIKE SMELL: So Nastran comes a couple of different ways. So with Inventor specifically, you have access to add-on, which is called Nastran In-CAD. Fusion 360 has all the Nastran solver built in. So that's the standard solver for the Fusion 360 product.

And I know Vince is in the back. If you have any questions about the purchase or understanding what that make up is, he will be able to-- he's got his hand up. He's one of our business development guys. So he'll be able to answer any real specific questions about that process, OK?

And then last thing I think you had on here,-- yeah, we already talked about questions. Survey. So this is something that is taken very, very seriously.

I hope that everyone enjoyed the class. We learn and adjust based on these every year. So we'd love it if you would take the time to log on to AU website and give us feedback on the class. And closing comments, Dave?

DAVID WEINBERG: No, but we'll hang out here. So if anybody wants to ask questions, right up here we'll do that, too, all right?

MIKE SMELL: All right, thanks for your time and attention this morning.