[Slide 2]
Hello everyone and welcome to this introductory talk on flood modelling. My name is Chris Sampson, I am a flood scientist by background and we at Fathom have undertaken the flood modelling that forms part of the Meteor project.

[Slide 3]
The aim of this talk is to give you an introduction to flood models. What are they? How are they built? What kind of data do they require and why are they needed?

Fundamentally, flood models are computer-based simulations of flooding or flood inundation as it's more properly known. Floods move across the landscape as a wave, and more specifically as a shallow water wave. Shallow water waves can exist across a range of scales from short-lived metre scale waves that might characterise the kinds of processes that happen during urban flash flooding, all the way through to seasonal flood waves moving across the world's largest river systems that can be hundreds of kilometres in length and last several months. The annual flood wave on the Amazon river is a good example of this. All shallow water waves share the common trait of having a low slope and varying gradually. Flood waves represent a major control on wetland biogeochemistry and on the carbon cycle. In their more extreme forms that can pose a very significant risk and cause an awful lot of damage.

[Slide 4]
Shallow water waves can be used to describe the processes that control flooding from the relatively small urban scale such as that shown in the pictures here…

[Slide 5]
…all the way through to very large floods on continental scale river systems such as the Mississippi as seen from space here, in flood on the right-hand side.

[Slide 6]
Now I thought it would be useful to start by giving a little bit of history of flood science so that we can see how we have got to the kinds of models that we use today.

[Slide 7]
Classically, before the age of computing, scientists used analytical solutions and laboratory data to try and understand how water moved across space.

We've got an image here of a large-scale laboratory test setup. Now in these kinds of scenarios, the model parameters and geometries were very well known, and the validation data were either exact if you're undertaking a specific analytical solution, or very accurate because it's a small-scale constrained problem. As such, simulations were never limited by the power of compute because you're looking for an exact analytical solution. This led to the paradigm of incremental added complexity which was fundamentally to say that we were trying to solve very small simple problems, and if you wanted to solve more complex problems you had to add complexity and very quickly this became impossible to handle.

[Slide 8]
Now if we fast forward to around the year 2000, laboratories and analytical-type solutions had been replaced by computer models. These computer simulations were already very sophisticated, and they could solve the underlying equation that describe the movement of water in two dimensions or three dimensions - and they could do this very accurately. But because of the very high computational cost that was associated with solving these kinds of equations, they were only applied to relatively small-scale problems.

[Slide 9]
This was OK because the end-users, who were typically engineers looking at individual projects, only really needed this information at small scales, and the general consensus was that models could always be improved by adding more physics and by adding more complexity.

[Slide 10]
However, this approach started to run into problems as the scale at which we were applying models started to increase. This was really because of two main issues. The first was that the data needed to drive these models had to be high resolution, which meant that very rapidly they became limited by the power of computers. The second problem was actually the availability of the kinds of high-quality data that were needed to drive these models as we started to look at larger scales. It is relatively easy to collect high-quality data on a small scale, but as you start to look at larger and larger areas, this becomes a very big problem. Certainly 20-years ago, attempting to build models of entire cities or even entire regions and countries was completely impossible because of these two major issues: computer power and data availability.

[Slide 11]
The major breakthrough came with the advent of remotely sensed terrain data, or ‘topographical’ data, and LiDAR, or laser altimetry data, is the best kind of remotely sensed data that is available. It is collected by attaching a laser scanner to an aeroplane and it is extremely precise. It is accurate to within about 5cm vertically, you have a spatial resolution of a couple of metres, and you can collect something like 50 square kilometres of data an hour. So you can very rapidly build quite large terrain models using this technique.

[Slide 12]
Modellers, of course, became very interested in this because it opened up a whole world of new possibilities for the application of models. What was needed was an observed validation dataset where we also had this new terrain data in order to be able to start to do some testing. A breakthrough for this came in 1995 in Europe where we managed to obtain very good observations of flooding along quite a large reach of river. We had a lot of data; we had 86 measurements of maximum water levels; we had aerial photography of the inundation; we also had synthetic aperture radar observations of the flood extents. Those three things together gave us a very comprehensive view of what that flood had looked like.

[Slide 13]
We also had observations of how much water was entering and exiting this stretch of river during the flood event, so we had a very good benchmark dataset against which to test different kinds of flood model. This experiment yielded some quite surprising conclusions, and the first and probably the most profound was that simple models did as well as more complex models giving the error in the data. Fundamentally what this told us was the adding more physical complexity to the model didn't actually produce a better result because the model was limited by the quality of the terrain data and the quality of the flow data going into it; the physics was not the limiting factor. The most important way to improve model scale was to increase the resolution of the terrain data on which we were simulating.

[Slide 14]
This led to a new approach to modelling which was to realise that actually what we needed were faster models with simplified physics that could be deployed on high performance computers to allow us to run high-resolution models over large areas. At the same time, we needed to stop worrying so much about the physics in the model and think more about the quality of the data that we were driving the model with.

[Slide 15]
So with that background, how do flood models work today?

[Slide 16]
I'm going to talk about the most common form of dynamic flood model that is used day which is a grid based two-dimensional model. These kinds of models do two things: they conserve mass and they conserve momentum and they simulate the flow of water from one point to another in a fixed period of time.

[Slide 17]
I'm not going to go into the details of this, but over the last 20-years there have been refinements to the mathematics and the simplification of the underlying physical equations that we use in these models, to enable them to be run faster and faster and faster. The actual pure analytical solutions of the shallow water equations have been known for well over 100 years, but they have an awful lot of complexity in them and, as I mentioned earlier, that is not actually important when we're simulating floods at large scales. So they've been simplified down into the form shown on the screen here in order to allow us to run the models very quickly while still maintaining enough accuracy and precision.

[Slide 18]
…and this graph can just give you an indication of how much faster the modern formulations of these equations, known as the inertial formulations, are compared to the formulation we were using 20-years ago known as the diffusive wave formulations. We're talking about orders of magnitude speedup in our simulations.

[Slide 19]

So what this side tries to do is to show in the simplest possible way, what happens within a flood model. If you look on the left-hand panel at Time 1, we basically have the model putting some water into the middle pixel on the screen. Now the model then solves, between Time 1 and Time 2, how much water should flow from the central pixel into each of its four neighbouring pixels over the time between Time 1 and Time 2. As you can see, the total amount of water in the model has not changed; all that has happened is that the water has spread out, with the amount of water moving into each pixel being governed by the terrain (the slope) from each pixel to its neighbour and the equations that were shown on the previous screen. The same then happens from Time 2 to Time 3, and you can see that same volume of water spreading out just as you might intuitively expect it to do.

[Slide 20]
So what kind of data is needed? Well, at the most simple form, there is not an awful lot of data that is needed! You need the terrain model because you need to understand the slopes because that ultimately is what is going to control which way the water moves. If you're simulating river flow then clearly you need the river network. You need to know where the river is, and you also need to know the geometry of the river. Now the geometry can be simplified down into a rectangular channel, but you do still need to know where it is, how wide it is, and how deep it is. You also clearly need to know how much water you're putting into the model, whether it is going into that river channel or whether you're actually simulating a rainfall event. Those are the basic inputs that are required. Obviously, you can and often do, add more complexity if it is needed or if the data is available. These can be things like flood defences, friction maps, soil types, flow structures, coastal water levels etc. There's actually quite a long list of things that you can add these models if you want to.

[Slide 21]
If you put the necessary inputs into the model then what you will get is something that looks similar to what you can see on the screen right now. So this is a video of a simulation of a real event that happened in Carlisle in the UK.

[Slide 22]
Now an important question of course is to ask: Do these models work? Are they accurate? And fortunately, over the last 20-years, a lot of work has been done to validate the fact that these models do work. I'm going to give an example here, again in the UK, of a typical validation procedure that might take place. So this is a flood on the River Severn between Worcester and Gloucester.

[Slide 23]
We have a high quality, high resolution, LiDAR terrain model at 3m resolution.

[Slide 24]
We also have for that same stretch of river a very high-resolution synthetic aperture radar observation of the real flood, and we have river gauges the tell us how much water was going down the river during this flood event.

[Slide 25]
When we compare the model result to the SAR observations, we get a very good fit indeed. So, on the map that you can see on the screen here light blue is where the observation and the model agree; red is where the model overpredicts slightly and yellow is where the model underpredicts slightly. You can see on the right-hand panel the flow going into model, and at the point in time where the SAR observations were taken you can see we have a model fit of about ninety percent. So the models are indeed very skilful, they can be used to accurately predict and recreate flood events.

[Slide 26]
However, this does lead to the question of whether or not these models are actually necessary? You might quite reasonably ask ‘can't we just use a simpler GIS approach to try and estimate flood risk?’ The simple answer is not really, or only in certain circumstances. There are two alternative approaches that have been tried in recent years to proper dynamic models.

[Slide 27]
The first is machine learning techniques, to basically use imagery of floods combined with things like terrain data to try and develop a machine-learning based tool to predict flooding. Probably the most successful attempt at this has been undertaken in the US, where the very large US government Federal Emergency Management Agency (FEMA) open-source flood maps were used as the training dataset along with the US terrain dataset using a technique called random Forest classification, a machine learning technique.

[Slide 28]
The method did do quite a good job. It managed to recreate a flood map with about an 80% hit rate against the FEMA training data. And so what we can say is that yes, actually a machine learning technique can quite successfully emulate the input training dataset and it can do so very quickly. The issue is that it cannot extrapolate beyond the training dataset. This model is trained on the FEMA 100-year data; now it can fill in gaps in that FEMA 100-year data, but it cannot be used to simulate a 500-year flood or a 5-year flood; it can only predict within the range of the training data. It also obviously inherits any errors in the training data, and it can't be used to simulate things like land-use or climate change. So machine learning techniques are fundamentally limited to interpolation only; they can’t be used to extrapolate into new conditions.

[Slide 29]
The other approach that is commonly applied is a GIS approach which is to say that the physics are not represented, it is simply an elevation-based approach. Possibly the most well-known of these is called HAND, or height above nearest drainage, which basically just takes a water level off a river and then tries to spread it using some simple rules on to the floodplain around it.

[Slide 30]

Now there are two main types of flood spreading algorithm, either non mass-conserving or mass-conserving.

[Slide 31]
These work fundamentally by spreading water around. If they are mass-conserving, they maintain the volume. If they are non mass-conserving they are even simpler and they just extrapolate the water levels across, but don't worry about conserving mass.

[Slide 32]
The main issue with flood spreading algorithms is that because they don't represent the dynamics they can't represent any kind of transient behaviour and that means that very often they fail most of the benchmark test cases. Therefore, they aren't actually able to accurately recreate flood conditions in many scenarios, and certainly when you start to consider very large, flat, wide floodplains, flood spreading algorithms really struggle because they are just not representing the diffusive manner in which water spreads across large flat surfaces.

[Slide 33]
The conclusions of the last 20 or so years' worth of research is that actually many large-scale river flows can be represented very successfully by simplified shallow water physics, and that given the finite computing resources that we all always have, model skill is improved more by increasing resolution than it is by improving the physics within the model. Today we have reached the point where highly resolved accurate models are possible at really quite fine resolutions at very large scales. By that I mean resolutions of 1 to 2 m over whole cities, assuming that you've got something like LiDAR data for that city to permit a model at that resolution, and then at 30 to 100 m resolution over the entire planet using globally available data. Ultimately it is the combination of new types of remotely sensed data with these improved efficient flood simulation engines, that have yielded new insights into surface water dynamics and flood risk across the planet.