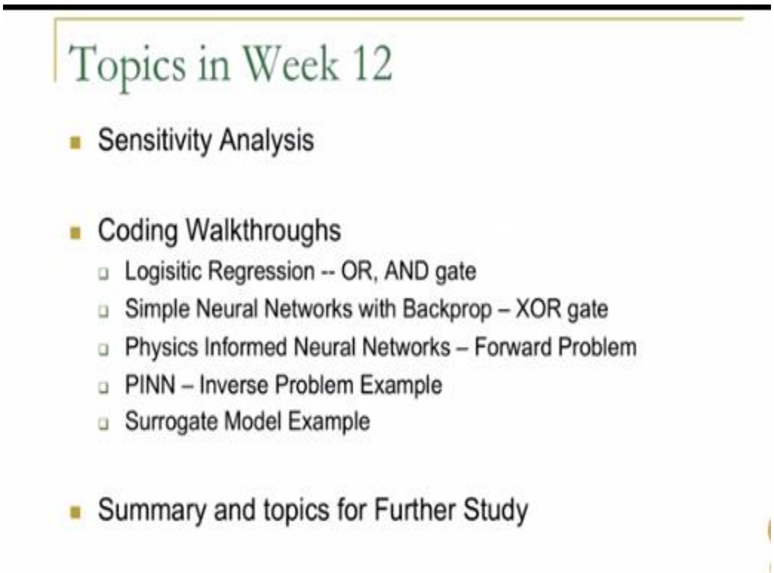
Inverse Methods in Heat Transfer
Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology-Madras

Lecture - 61
Introduction to Week 12 - Sensitivity Analysis

Welcome to week 12 of inverse methods in heat transfer. This is the final week of this course. This video gives you a short introduction to what we will be doing. We just have a shortlist of topics in week 12. And also, I will give you a brief introduction to something called sensitivity analysis that we did not discuss in detail in the previous videos.

(Refer Slide Time: 00:46)



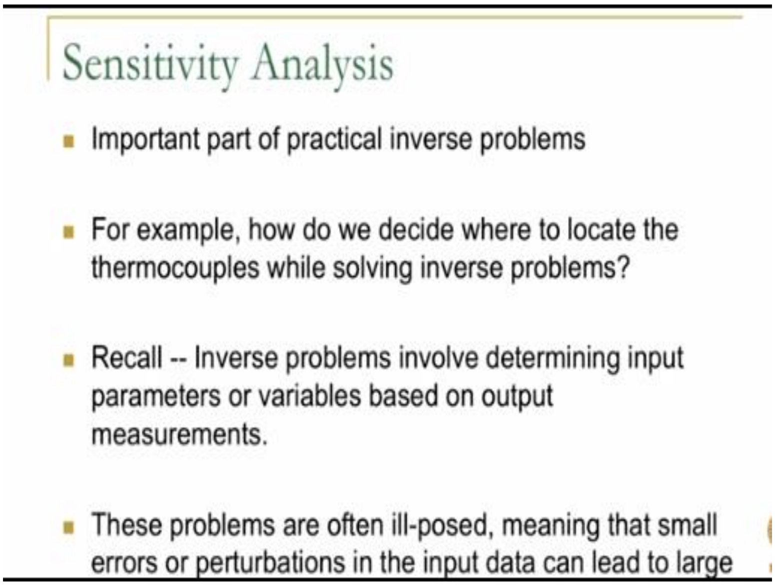
So, topics we will be looking at in week 12, of course, right now in this video sensitivity analysis, there is something that I had told you I will do a couple of weeks ago, which was coding walkthroughs. So, there are a lot of examples that we did that I just did in theory. I would like you to see the codes in action here. Of course, as has been my practice in this course, I will be just doing MATLAB coding walkthroughs.

So, we will look at I have written the spelling wrong here. So logistic regression. So coding walkthroughs of logistic regression for OR and AND gate. Then as we had seen earlier, XOR gate does not work with logistic regression. So, I will show you a simple neural network with a back propagation algorithm which we had seen earlier. We will see an XOR gate example there.

Then, the physics informed neural networks that we looked at in the last week, I will show you a simple forward problem that we can solve. Finally, I will show you an inverse problem example with physics informed neural networks. And the last example would be a surrogate model example, which is an alternate use of neural networks within inverse problems.

Finally, we will end with a summary and topics for further study. There are as I said earlier, this is just a very, very barely introductory course for the huge topic of inverse methods. So, there are several topics of further study. And I will just discuss what those topics are in the final video for this course.

(Refer Slide Time: 02:23)

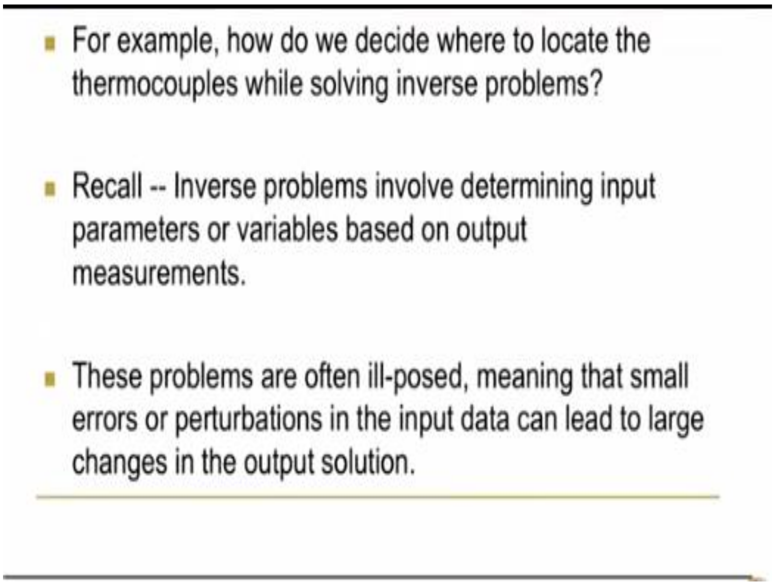


So let us come to sensitivity analysis, the main topic of this video. Sensitivity analysis is a very important part of practical inverse problems. Honestly, the problems that we have been doing so far have just been toy problems just to illustrate techniques. Remember, this is not an inverse heat transfer course. It is an inverse method in heat transfer. So really it is just an introduction to some of the methods that exist within heat transfer.

So where is sensitivity analysis used? Just to give you a simple example, suppose we want to decide where to locate the thermocouples. So, I had just said some standard locations. But suppose you want to do a more scientific, more analytic technique, or more rigorous way of finding out where would be the best locations to keep the thermocouple so that your inverse problem, inverse solution is less erroneous okay, has fewer errors, can you actually decide that?

And it turns out that sensitivity analysis would help you to do that. Again, I am not going to discuss how we do that in detail in this video, but just give you a preliminary idea and for a more detailed analysis, I would request you to go through a more rigorous complete course in inverse methods.

(Refer Slide Time: 03:41)

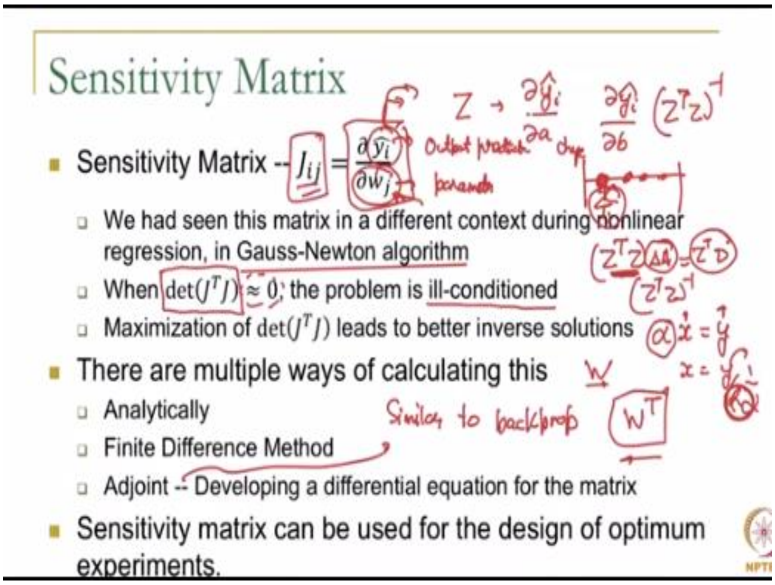


So, recollect that inverse problems basically involve determining input parameters or causes, given effects. So given on based on output measurements. For example, we put thermocouple measurements, we want to find out, let us say something like the thermal conductivity, okay. Of course, often these problems are ill-posed, which means that if you make small errors of perturbations in the input data, then this leads to large changes in the output solution.

What this means is, if I change the thermocouple location a little bit or the thermocouple measurement a little bit, in case it is an ill-posed problem, the problems that we did again in this example, as five problems were not particularly ill-posed. But it gets ill-posed as the number of data points actually increases. So that can actually or the number of parameters to be determined increases.

So, the whether the problem is ill-posed or not can actually be determined using sensitivity analysis.

(Refer Slide Time: 04:44)



Now what is the sensitivity matrix? The sensitivity matrix is something that we already saw as our Z matrix. If you remember, Z was $\frac{\partial \hat{y}_1}{\partial a}, \frac{\partial \hat{y}_1}{\partial b}$. You might remember these quantities that were sitting within the Gauss-Newton algorithm. If you do not, I request you to go back and take a look to find out how this Z arose. But let us look at what it physically means.

It physically means how much the output prediction changes, given a change in the parameter. So, this matrix is called a sensitivity matrix. So, in case you determine this matrix, so you are given all these places where you are measuring your outputs, or in this case I had only a single output. Many times, we simply had temperatures or we could have six, seven temperatures.

So y1 hat in our slab example would mean $\hat{T}_1$ the temperature at location one. So, the question is, suppose I change the location a little bit then how much does that, how sensitive is this change in location to the change in parameter? And typically, this can be answered using this simple question, find out this matrix $\frac{\partial \hat{y}}{\partial w_j}$. We had seen this, of course during our Gauss-Newton algorithm.

Now the way to determine whether a problem is well-posed or ill-posed is to determine this matrix y $J^T J$, you might remember we had these terms, $Z^T Z$, and we had $Z^T Z \Delta A =$

$Z^T D$. So, you remember that we actually effectively had to $(Z^T Z)^{-1}$ when we were doing Gauss-Newton, which is why the determinant of this matters.

In case the determinant is really small, then the problem obviously, is ill-conditioned because a small change here in this will cause a large change in the value of ΔA. Another way of saying it is, suppose I have something like $\alpha x = y$, let us take a scalar equation. If α is really small, then a small change in x will charge cause a large change, or a small change in y will cause a large change in x because $x = \frac{y}{\alpha}$.

So, in case α is really, really low, then we would have this extreme sensitivity to small changes in y because they get amplified by the factor $\frac{1}{\alpha}$. Similarly, small changes in the right-hand side or small errors can get amplified by $(Z^T Z)^{-1}$. So that is why we are interested in the sensitivity matrix J. I have called this J because that is the standard notation.

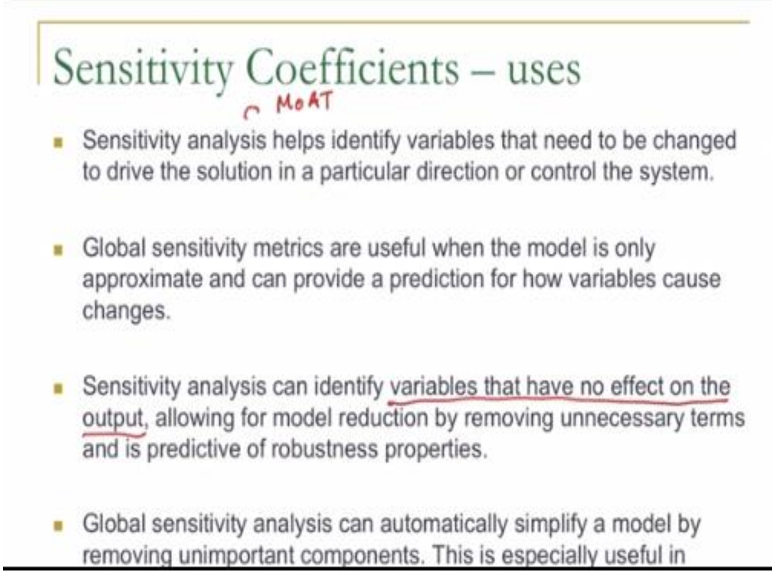
Z is in the particular context of Gauss-Newton. But effectively, whatever we calculated in Gauss-Newton was also effectively a sensitivity matrix or a Jacobean as we called it there. Now when we want to calculate this J, there are multiple methods to do so. We can calculate it analytically as we did when we were doing the Gauss-Newton algorithm. Another way is to use finite difference.

As I told you while doing backprop, this can be less effective. Remember, if you have a large number of parameters w, then this is actually a hard task. Finally, in case of differential equations, you can use something called the adjoint method, which is one of the topics we have not covered here, okay? There are clever ways of calculating an equation directly for J, just like we have a differential equation for y we can calculate a differential equation for J.

It turns out, you can actually develop an entire differential equation for this matrix also and that is called the adjoint method. Turns out this is similar to using backprop. So remember, I had the backprop algorithm during neural networks, turns out this adjoint approach is very, very similar to what happens in backprop. You might remember that in backprop, we got terms like w^T.

If the forward transformation had W, the inverse transformation or during the backprop had w^T and that is similar to using the adjoint approach. This matrix can then be used for the design of optimum experiments. How so?

(Refer Slide Time: 09:32)



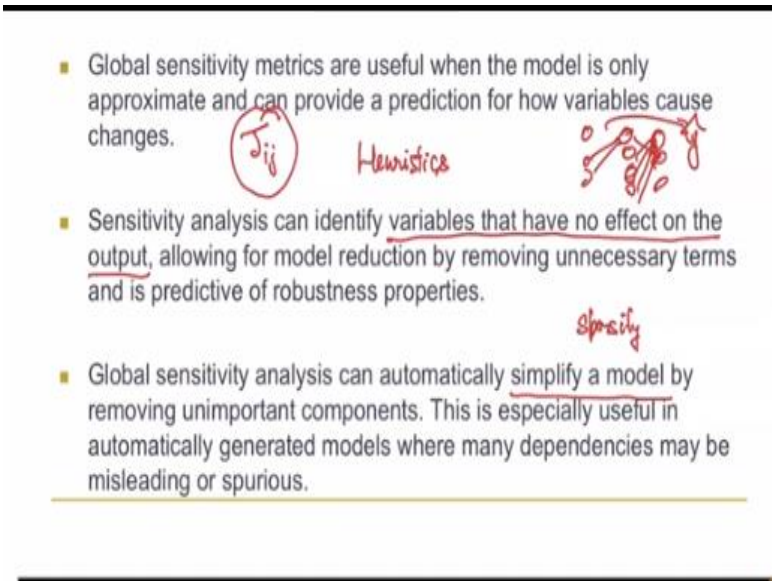
Basically, this sensitivity analysis will help you identify variables that need to be changed if you want to drive the solution in a particular direction or to control the system. Why is that? Once you calculate this, you know how sensitive each output parameter is to every input parameter. What will turn out is in some cases, you will have variables which have no effect on the output, okay?

So, suppose there is a $\hat{y}$, which is not affected by some particular w, you know that this w has very little effect on $\hat{y}$ on a specific $\hat{y}$. So, it is less sensitive to that. Or in case one w does not have almost any effect on $\hat{y}$. So, there are other methods for doing this appropriate sensitivity analysis. There are methods like Morris one at a time etc.

When you have a large number of parameters and a large number of inputs, these are topics incidentally that I have not covered in this course. So, a proper sensitivity analysis, which uses just these this matrix in various different clever ways. The idea overall however is very simple. In order to find out which parameters are important and which parameters are not important, okay?

Or which parameter calls very little effects, so in that case, those are robust parameter estimates, okay?

(Refer Slide Time: 11:05)



So finally, neural networks we often use this to simplify a model, okay? So, in case you have like a neural network with lots of weights, if it turns out that some weights, you cannot simply remove low weights, if it turns out that some weights have very little effect on this y hat, you can actually throw them away, okay? So, you can remove a few weights here and there and get lesser parameters.

So, these are called sparse models. So, you can sparsify your model in case you do sensitivity analysis. So, all I have talked about in this video is simply heuristics. Heuristics means rules of thumb, simple ideas that you can use for sensitivity analysis. We did some computation of sensitivity matrix, of course when we did Gauss-Newton algorithm.

We did it as part of that algorithm, but many times whether you use Gauss-Newton or not, we do calculate this J_{ij} because it helps us find out whether the problem is well-posed or ill-posed, whether we can remove some parameters or not. So, it is useful for all these cases. So, in more advanced and genuinely practical uses, and also for uncertainty analysis, when we use inverse methods, we actually you will see this playing a very important role.

So hopefully, if you continue your study of inverse methods, you will take a lot of look at J_{ij}. So, I will end this short introduction, heuristic introduction to sensitivity analysis here, and I will see in the next video where we will start doing our code walkthroughs Thank you.