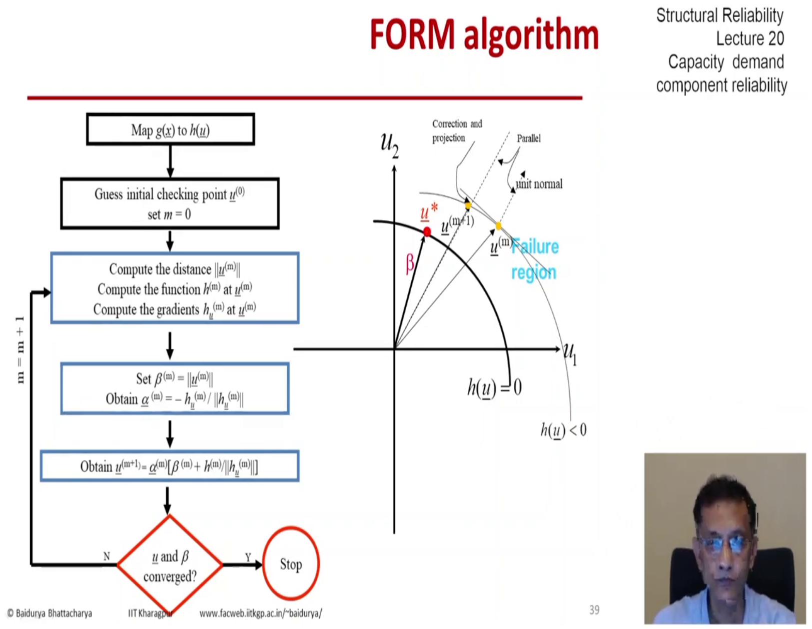
Structural Reliability
Prof. Baidurya Bhattacharya
Department of Civil Engineering
Indian Institute of Technology, Kharagpur

Lecture –165
Capacity Demand Component Reliability (Part 13)

(Refer Slide Time: 00:27)



FORM algorithm
Structural Reliability
Lecture 20
Capacity demand component reliability
Map $g(\underline{x})$ to $h(\underline{u})$
Guess initial checking point $\underline{u}^{(0)}$ set $m = 0$
Compute the distance $\|\underline{u}^{(m)}\|$
Compute the function $h^{(m)}$ at $\underline{u}^{(m)}$
Compute the gradients $h_{\underline{u}}^{(m)}$ at $\underline{u}^{(m)}$
Set $\beta^{(m)} = \|\underline{u}^{(m)}\|$
Obtain $\underline{\alpha}^{(m)} = -h_{\underline{u}}^{(m)} / \|h_{\underline{u}}^{(m)}\|$
Obtain $\underline{u}^{(m+1)} = \underline{\alpha}^{(m)}[\beta^{(m)} + h^{(m)}/\|h_{\underline{u}}^{(m)}\|]$
$\underline{u}$ and β converged?
N
Y
Stop
m = m + 1
u_2
u_1
Correction and projection
Parallel
unit normal
$\underline{u}^*$
$\underline{u}^{(m+1)}$
$\underline{u}^{(m)}$
β
Failure region
$h(\underline{u}) = 0$
$h(\underline{u}) < 0$
© Baidurya Bhattacharya
IIT Kharagpur
www.facweb.iitkgp.ac.in/~baidurya/
39

So in terms of a flowchart let us first introduce the gradient based algorithm for finding the minimum distance point. Here what you have is the u 1 u 2 spaces in 2 dimensions and we have h equals 0 the mapped limit state equation and then the failure region is also known. So, we need a method to find the minimum distance point. So, let us go step by step. So, we obtain h and then we always have to have an initial guess. So, let the initial guess is u and the superscript indicates the iteration number.

So, we are talking about the zeroth iteration and then we compute the distance from the origin to this initial guess and then we also compute the functional value at that point we compute the gradients of the function at that point and obviously the gradient vector is very useful because it points in the direction of the most rapid increase of that function. So, that is that is the basis and then we set we stored the distance we obtain the gradients alpha by normalizing the gradients of h I am sorry we obtain the direction cosines alpha by normalizing the gradients hu.

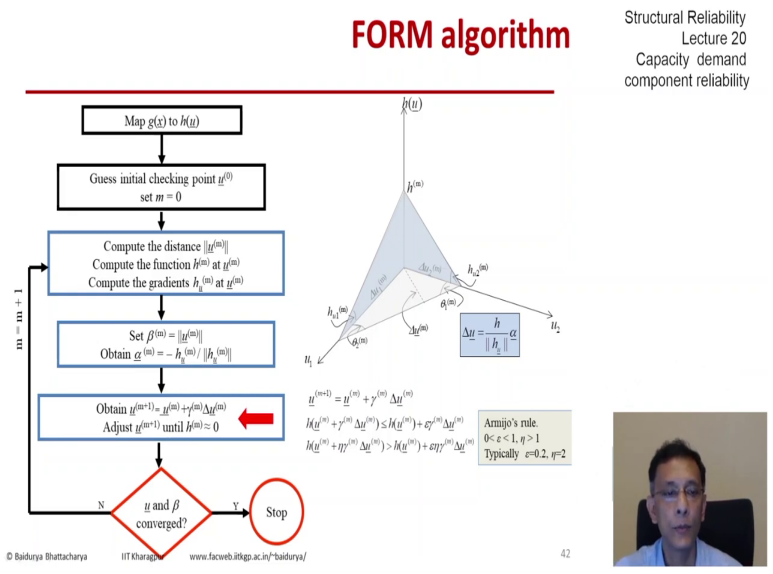
So, hu is the partial derivative of h with respect to u 1 and u 2 as the case may be and then we obtain a new point the next one in the iteration in terms of the direction cosines in terms of earlier stored distance beta and correction factor which depends on the current value of the function and it is the norm of its gradient. So, we do this until u and beta converge. So, if the answer is yes within whatever tolerance we have decided we stop otherwise we increase the counter by m by one.

So, m becomes m + 1 and we repeat the process until convergence happens and hopefully convergence will occur now on this plot let us identify the key steps. So, let us say there is as we are somewhere in the middle of this process and. So, that is our point in orange and we have the vector from the origin. So, we know what the function is at that point. So, we know the h of u going through that point which happens to be negative.

So, h of u is negative beyond the limit states towards the failure region as we know very well we can draw the tangent to this limit state at we can find the unit normal and we can draw a parallel line to that unit normal from the origin and that actually has the direction cosines alpha and then what we do is we make a correction that would be the last step on the flowchart on the left and project it on that direction alpha m and that would get us our new point u of m + 1.

And we keep doing this until we finally reach the optimal point which does not require any further correction and that is our u star and that distance is beta now let us probe a little deeper as to how this works. So, let us let us see how we get these alphas and the partials of h with respect to u and why we need to normalize the h sub u and so on.

(Refer Slide Time: 05:27)


Map $g(\underline{x})$ to $h(\underline{u})$
Guess initial checking point $\underline{u}^{(0)}$ set $m = 0$
Compute the distance $\|\underline{u}^{(m)}\|$
Compute the function $h^{(m)}$ at $\underline{u}^{(m)}$
Compute the gradients $h_{\underline{u}}^{(m)}$ at $\underline{u}^{(m)}$
Set $\beta^{(m)} = \|\underline{u}^{(m)}\|$
Obtain $\underline{\alpha}^{(m)} = -h_{\underline{u}}^{(m)} / \|h_{\underline{u}}^{(m)}\|$
Obtain $\underline{u}^{(m+1)} = \underline{u}^{(m)} + \gamma^{(m)} \Delta\underline{u}^{(m)}$
Adjust $\underline{u}^{(m+1)}$ until $h^{(m)} \approx 0$
$\underline{u}$ and β converged?
N
Y
Stop
m = m + 1
$h(\underline{u})$
$h^{(m)}$
$h_{u1}^{(m)}$
$h_{u2}^{(m)}$
$\Delta u_1^{(m)}$
$\Delta u_2^{(m)}$
$\theta_1^{(m)}$
$\theta_2^{(m)}$
$\Delta\underline{u}^{(m)}$
u_1
u_2

FORM algorithm

Structural Reliability
Lecture 20
Capacity demand
component reliability

$$\Delta\underline{u} = \frac{h}{\| h_{\underline{u}} \|}\underline{\alpha}$$

$$\underline{u}^{(m+1)} = \underline{u}^{(m)} + \gamma^{(m)} \Delta\underline{u}^{(m)}$$

$$h(\underline{u}^{(m)} + \gamma^{(m)} \Delta\underline{u}^{(m)}) \le h(\underline{u}^{(m)}) + \varepsilon\gamma^{(m)} \Delta\underline{u}^{(m)}$$

$$h(\underline{u}^{(m)} + \eta\gamma^{(m)} \Delta\underline{u}^{(m)}) > h(\underline{u}^{(m)}) + \varepsilon\eta\gamma^{(m)} \Delta\underline{u}^{(m)}$$

Armijo's rule.
$0 < \varepsilon < 1$, $\eta > 1$
Typically $\varepsilon = 0.2$, $\eta = 2$

© Baidurya Bhattacharya IIT Kharagpur www.facweb.iitkgp.ac.in/~baidurya/ 42

So, let us look at a three-dimensional plot u 1 and u 2 as we had before and the third dimension actually is the functional value. So, above the u 1 u 2 plane it is positive and below it is negative and that's what we identify as failure. So, this h surface wherever it cuts the u 1 into plane is actually my limit state equation the line separating the safe from the failed domains. So, we identify two triangles on the u 1 h and u 2 h plane in terms of the gradient of h with respect to u 1 the gradient of h with respect to u 2 and the height of the function at that point h.

So, that gives me the height along with the gradients h u 1 and h u 2 give me delta u1 and delta u 2 the distance traveled along u 1 and u 2 axis as the function comes down from h towards the u 1 u 2 plane. So, in terms of simple trigonometry this is the expression of delta u 1 and delta u 2 that you see on your screen which you're going to use next. And now we covered the three points of intersection on the three axes and that is the gives me the direction along which the search wants to move.

So, the projection of that is the dashed line that you see on the u 1 u 2 planes and that is actually the delta of u. So, that is the next incremental distance that I want to move to get to my next point. So, from I will go to u m plus one now we need to do that we need to know the angles and theta 1 is the angle that you see and theta 2 is the angle that you see. So, those will give me the direction cosines alpha it is easy to find out cosine of theta 2 in terms of delta u's and by plugging back the relation between h and delta u 1 and delta u 2.

I obtain the cosine of theta 2 as the partial of h with respect to u 2 over the norm of the gradients and there's a negative value there for obvious reasons as u 1 increases h actually reduces in value decreases in values. So, goes for u 2. So, that is why the negative sign is there continuing this way we can define alpha 1 also alpha 2 is cosine of theta 2 alpha 1 is cosine of theta 1. So, we can generalize this that the alpha vector is the gradient vector of h with respect to u normalized by the norm of the gradients.

So, that is that is the arrow that you see on the on the left that is the red arrow that I just pointed. So, that is the reason behind why we compute alpha that way the other correction that we make in terms of the distance the value of delta u the norm of delta u is what you see in terms of the second red arrow and in terms of h and the norm of hu it is given as you see we get it from simple geometry from on the u 1 u 2 plane.

So, together that gives me the vector delta u that i need to move in order to get to my next point now there is I can make an improvement to the search process to this line search process if I make sure that I do not travel very far from the h equals 0 lines. So, if every time I try to move away from the h equals 0 lines I would like to come back to that line itself because that is my constraint I want to find the minimum distance to this line.

So, I need to only look for points on this line h equals 0. So that is what you see a change from the previous algorithm that we presented. So, if our new point could be given in terms of the old point plus a fraction or a multiple of the incremental vector delta u and there would be ways of choosing that factor gamma and which could vary from iteration to iteration and then the u that I get the m + 1 iteration that I get I would like to adjust it until I am back at h m equals zero.

So, I am back at the limit state line itself uh. So, there are ways of doing that and um. So, there is this so-called army just rule which uses a sort of bounding action so we need to find the value of gamma that satisfies these two limits and once we do that we can keep adjusting u m plus 1 until we are back at h 0 and that is done in this iterative manner. So, once we are reasonably close then we move to again the upper block and we find a new a new hu and so on until we are we have converged.

So, this can sometimes help matters when the previous algorithm has some convergence problems or some oscillation type behaviour.