Economic Operation and Control of Power System

Dr. Gururaj Mirle Vishwanath

Department of Electrical Engineering

IIT Kanpur

Week - 02

Lecture – 10

Hello and good morning, welcome you all for the NPTEL online course on Economic Cooperation and Control of Power Systems. Just to summarize we have learnt about three different topics, the first one is piecewise linear approximation, where we find out the optimal schedule while we carry out different segments for each generators starting from the lowest λ minimum to the highest λ maximum. So, then we will choose we will formulate a table and in the ascending order we will make a entry to each and every row. So, that we will get the optimal schedule choosing from the lowest λ minimum to the highest λ maximum. And the second was about participation factors, if there is a optimal schedule which is already been done and there is a sudden change in the load and what is the share of the load with respect to each generator such that the new generation with respect to each generator will be also an optimal schedule or the most economical schedule. And the third aspect that we have learnt is let us say the N number of generators which are connected to a bus and to that bus there is also a load and a new generator is coming up which has some fuel constraint and how do we carry out this operation.

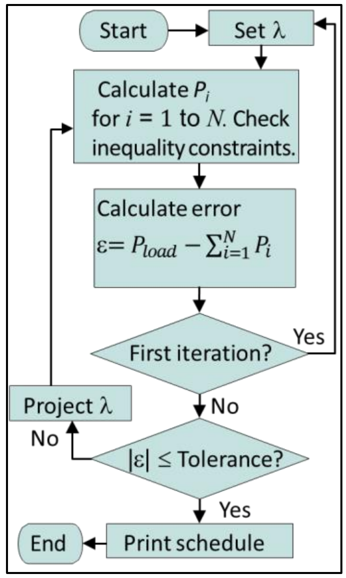
So, we have understood that the N number of generators can be represented with one single generator which we call it as a composite generator. So, then we started with the algorithm what we have designed helps us understand that the λ is starting from the λ minimum. Each generator has its own λ minimum and λ maximum. We will find out the most minimum λ minimum among all the λ minimum. The most maximum λ maximum among all the λ maximum and we will start from λ minimum the most minimum λ to the most maximum λ and while we vary in terms of step then we will find out the total generation and total fuel cost. And if the tolerance is very less let us say 1 percentage then then it is reasonably matching with the actual curve. So, we will obtain the smooth curve we will join the points we will obtain the overall fuel characteristics by correcting the points and the objective is to ensure that with the help of curve fitting formula we will obtain as smooth curve as possible. So, that we will not jump into the non convex problem. So, in today's class we will try to explore further topics the first topic is the lambda iteration method. See the solution to the optimal dispatch can be approached by graphical methods also. Plot the incremental cost characteristics of each generator. See you can see here this is the incremental cost characteristics that is $\frac{dF_1}{dP_1}$ of the first generator and incremental cost characteristics which is $\frac{dF_2}{dP_2}$ of the second generator and incremental cost characteristics for the third generator which is $\frac{dF_3}{dP_3}$. So, you can see here let us say this is the λ if this is the λ then this is the lambda corresponding to this lambda you would get a specific generation from each generator. So, sum up together, we have at a specific lambda let us say you have three specific generations coming up you see here that is a P_1 then P_2 and there is P_3 you see lambda is already been chosen and the lambda minimum of the third generator is higher than the lambda which is already chosen.

So, but still we need to stick on to that principle that each generation should have you know its own will have its own minimum and maximum generation. So, but in this case to be very honest you know the third generator will not be contributing any generation, but because you need to meet out that constraint you need to operate this particular generator at its minimum power. So, all of them put together you are getting some generation. Let us say your load demand is what load demand that you are expecting is exactly equal to the sum of the generation then you can consider that as a right lambda. Let us say if the generation is increased or if the demand is increased then you have to accordingly change the lambda such that for the new lambda the total generation will be matching to the load requirement. So, let us say the generation is increased from this lambda the total demand is increased from the original P_R to P_R plus delta P_R just for an example. Then, if the demand is increased with this specific generation you will not be able to meet out this demand right. So, what you have to do? You have to increase the lambda. Once you increase the lambda let us say you increase the lambda from lambda to lambda plus delta lambda that means you will have a new lambda and that will be something like this let us say for example. So, with respect to this incremental cost characteristics curve of each generator now you can easily identify that the generation will be of each generator will be higher than the previous generation. Because the generation is increasing now the sum of the generation put together will be meeting out to the new increase in demand such that the constraint equality constraint which is sum of the generation is equal to load will be met and it will inherently ensure that the frequency of operation will be constant. If let us say the demand is decreased from P_R to P_R minus delta P_R now what you have to do? You have to go in the opposite direction. So, decrease the lambda such that the total generation will be again equal to the new change in the load right. So, this is what this approach is all about. Now, you can see here with this error curve you can easily understand let us say you have chosen an initial lambda, lambda of 1 right corresponding to that there is an error it is not a exact lambda.

So, you will find out the generation with respect to this lambda. So, that sum of the generation put together is greater than the or whatever the difference between the sum of the generation and the load is not 0. Then if the error is positive then you have to get a new lambda. Now, the next lambda is lambda 2 and corresponding generation is a new generation. Now, here you can see here earlier the error was positive now the error is negative.

That means now again you change the lambda go in opposite direction now you get a new lambda which is lambda 3 and again generation is increasing or the error is positive now right. But this error is lower than the previous error the first error. So, similarly somehow you will reach the final point where error is 0. So, with so many iterations right. As you can see here if the total power output is too low, increase the lambda value or if too high, decrease the lambda value. Now, this is your algorithm flow chart set a lambda right and calculate total generation from for all the generations from i to 1, i to n check the inequality constraint. Now, for this lambda you should ensure that you know none of the generation should violate the constraint. If it violates then you assign if the generation is coming out to be lower than the minimum you assign it that to be operating at its minimum point. If it is maximum more than the maximum capacity you assign it to be p maximum right. Then you calculate the error and then you check the tolerance if the modulus of the error means the absolute value of the error is less than the tolerance.

If it is yes then that is the end that is an exact lambda otherwise you project the lambda increase the lambda or decrease the lambda accordingly based on the error. And then for the new lambda you again carry out the generation and the steps will be followed exactly as it is right. So, one thing you need to identify is as an iterative procedure stopping criteria must be established because if you keep exactly 0 it will keep on oscillating and it may not be able to reach and it may take so many iterations sometimes it may enter into the infinite loop also. So, somehow you should keep some lower threshold as close as possible to 0 need not be 0 exactly right. So, two types of approaches are you know followed the first one is total output power is within a specified tolerance of the load demand right. So, there is a total generation minus total load and that error can be within the specified band or you can fix the total number of iteration all together as shown below:



Let me carry out let us say 100 iteration whatever the value it gets converge. So, let me fix that. So, usually they follow the first approach right. Now, let us take an example to understand this in a better manner. So, you have a characteristics heat characteristics and here you can see here it is not you know quadratic expression it is a polynomial of 3 right. So, degree 3 now for 3 generating units find the optimum schedule for a 2500 megawatt load demand using the lambda iteration method right. So, you get the ABCD constants P minimum, P maximum everything is given to you and the fuel cost is also given. Now, set the value of lambda on the second iteration at 10 percentage above or below the starting value depending on the sign of the error for the remaining iterations lambda is projected in the figure. I will help you understand.

Let us say we are fixing to start with lambda is equal to 8 right.

Initial iteration: $\lambda_{start} = 8.0$

Incremental cost functions

$$l = dF_1/dP_1 = 6.95 + 2 * (9.68 * 10^{-4}) * P_1 + 3 * (1.27 * 10^{-7}) * P_1^2$$

$$l = dF_2/dP_2 = 7.051 + 2 * (7.357 * 10^{-4}) * P_2 + 3 * (6.453 * 10^{-8}) * P_2^2$$

$$l = dF_3/dP_3 = 6.531 + 2 * (1.04 * 10^{-3}) * P_1 + 3 * (9.98 * 10^{-8}) * P_3^2$$

find the roots of the three incremental cost functions at $l = 8.0$

$$P_1 = (-5575.6,\ 494.3),\ P_2 = (-8215.9,\ 596.7),\ P_3 = (-7593.4,\ 646.2)$$

Use only the positive values within the range of the generator upper and lower output limits

So, now already we have fixed lambda is equal to 8. So, obtain incremental characteristics then you because now the differentiation after differentiation will get a quadratic expression right. So, each quadratic equation will have 2 roots right. So, here one is negative another one is positive for each generation, but you should take out negative I mean you should ignore the negative value because ultimately the if demand is positive common sense will indicate that generation cannot be negative right.

So, you will have to pick up the positive values of each generation then ensure that this lies within the range of their limits P minimum to P maximum. Then calculate the error:

$$e = 2500 - (494.3) - (596.7) - (646.2) = 762.9\ MW/h$$

With a positive error, set second λ at 10% above start λ_{start} : $\lambda^{[2]} = 8.8$

Second iteration: $\lambda^{[2]} = 8.8$

Find the roots of the three incremental cost functions at l=8.8

$$P_1 = (-5904,\ 822.5),\ P_2 = (-8662,\ 1043.0),\ P_3 = (-7906,\ 958.6)$$

Calculate the error:

$$e = 2500 - (822.5) - (1043) - (958.6) = -324.0\ MW/h$$

Project λ:

$$\lambda^{[3]} = \frac{l^{[2]} - l^{[1]}}{e^{[1]} - e^{[2]}}\left(e^{[2]}\right) + \lambda^{[2]} = \frac{8.8-8.0}{762.9+324.0}(-324.0) + 8.8 = 8.5615$$

Continue with third iteration

Error is total demand which is 2500 megawatt minus sum of the generations right and still the error is positive. That means load is higher compared to the total generations for which we have calculated using a specific lambda which is not a optimal lambda for this specific combination of generation which should be equal to the total load right. Now you have to increase the lambda. Now let us say increase by 10 percentage it will be 8.

8. Let us see how it works now for the 8.8 you will get a specific P 1, P 2, P 3 and the error is coming out to be negative now. That means somewhere your common sense will clearly indicate λ will be between the range of 8 to 8.8 right. Now you project the λ again

you can go for 10 percentage minimum you know 8.

8 into 0.1. So, all you can do all these things but or 0.05 you can take some value but how to project it. So, it is a simple mathematics let us say. Now you have 2 set of values, 2 values that is you have first iteration you have taken λ is equal to 8 and you have corresponding error which is positive right. And in the iteration 2 you have taken 8.

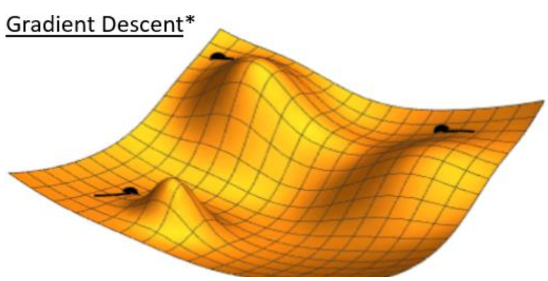
8 λ and you have a error which is negative. That means now you have an understanding with respect to change in λ how much is the variation in the error. So, with respect to that you can project a new λ that is the difference in λ with respect to the change in error with respect to change in λ indirectly we are doing d λ by d error right. So, then you add it to the previous λ you get a new λ which is 8.5615 and to for this the total generation is coming out to be closer here you can see here 2510.

2 and again you go for the fourth iteration now it is almost converging 2499.2 and the corresponding generation is P1, P2, P3. So, understand λ iteration method will first fix the λ, calculate the generations, find out the error and project the new λ and for the new λ again carry out the total generation P1, P2, P3 and the steps keeps on followed. So, that you will ultimately reach a point where error is as close as possible to 0 right. But there is a catch that if the finding out the initial point is very critical here.

Now, if you choose a λ which is not proper optimal λ you know then close to optimal λ then it will keep on oscillating that is one problem in the λ iteration. Let us say I will take an example where λ is equal to 10 instead of choosing lambda is equal to 8 if I have chosen λ is equal to 10 then you can see here there is a too much of oscillation. You can see here the total generation is decreasing and then increasing right and then again decreasing that means error is oscillating basically ultimately right and then it is finally, coming to the close point of 2500 megawatt, but it is taking 10 iteration the previous case it was just 4 iteration right. So, this is one of the challenges choosing how to choose a specific λ. Now, the next approach is economic dispatch using gradient method.

Now, if you choose a specific you know let us say if the polynomial is like you know whole number or degrees whole number in the earlier case. So, then it would have been you can follow *lambda* iteration method that is fine and, but if let us say the fuel cost characteristics of a specific generator has a function which is having you know decimal degrees right. So, a refractional power and it has even more complicated polynomial expression. So, the λ search technique requires the solution of the generator output for a given incremental cost and it is possible with a quadratic or piecewise linear function however hard for complicated functions the lambda iteration will be very tough. So, we need to have a more basic method to solve this problem.

The gradient search method uses the principle that the minimum is found by taking steps in downward direction. Now, the gradient search method is a better approach in kind of this in the case of this kind of problems. So, you can see here from any starting point let us say you have a starting point x of 0, one finds the direction of steepest descent by computing the negative gradient of f at x of 0. So, let us say you have total generators 3 that means you have 3 variables p 1, p 2, p 3 and you also have another variable which is λ because in a Lagrangian equation you have let us say n number of generators are there n plus 1 will be the total variables because 1 is also λ right. Now, you can see here now there is a function this is a general expression let us come to the Lagrangian function in the next slide. So, let us say there is a general expression function there is a function which is a dependent on x variable x is a vector here right. So, you obtain the gradient of this function with respect to each of the variable right gradient means what it is a slope actually right right it is a slope that means it is d y by d x basically. So, the partial differential equation of the function with respect to each variable you find out that means what I am doing here is you can see can you see in this picture it is a 3 D diagram and let us say you say there is a starting point here there is another starting point here right there is another variable here. Now, you find out the steepest slope starting at each point for this function that means if you place a let us say you place a ball here this is ball 1 you place another ball here ball 2 you place another ball here you know this is ball 3. So, in a hilly region if you place a ball, ball will come down based on the it will follow the lowest slope right the highest slope actually you know as shown below:



So, if as this slope is more steeper it would follow that path and finally, reach the point right. So, that is what we are finding out what we are doing is the function we have to find out the minimum value. So, we are finding the negative of the steepest descent such that all of the variables will converge to one specific point where the error is almost 0 close to 0 right. So, for this the path could be something like this and for this for this the path could be something like this and for the third variable the path could be something like this, but all of them would converge at this point. So, the principle of gradient search method in ED is that the minimum of function f of x can be found by a series of steps this is called as process of iteration going in the direction of maximum descent or the steepest

slope you can consider. Hence the search should be directed towards negative of the gradient of the function right. To move in the direction of maximum descent from x of 0 let us say you start from one point then for the next iteration you need to have a new starting point new point where it can start right. So, then you have to obtain the next variable, the next iteration value of the same variable. So, that is x 1 is equal to x naught minus alpha into gradient function which is calculated at the previous starting point x of 0 and here alpha represents the you know acceleration factor how fast you need to reach the destination you can choose this randomly let us say 20, 100. So, based on that you will there is a trade-off actually if you jump too fast you may miss out the actual the most maximum point or if it is too slow it may takes lot of time.
So, you can choose it optimally. Then where alpha is the scale or the acceleration factor that is used to process the convergence of the search method and the best value of the alpha is obtained by experimentation. It is like a empirical approach where you find out what is a suitable alpha for this specific problem statement. Now this is anyway known to you for economic dispatch problem what is objective function it is to minimize the total cost. So, which is a total cost is nothing but summation of the individual cost of the specific generator. So, and the constraint is summation of generation is equal to total load.

For ED Problem,

$F_T = \sum_{i=1}^{N} F_i(P_i)$ and the constraint $\sum_{i=1}^{N} P_i = P_{load}$

Obtaining the Lagrange function

$$\mathrm{L} = F_T + \lambda(P_{load} - \sum_{i=1}^{N} P_i)$$

$$= \sum_{i=1}^{N} F_i(P_i) + \lambda(P_{load} - \sum_{i=1}^{N} P_i)$$

Now obtain the Lagrangian function Lagrangian function is total fuel cost plus λ into the constraint. Now you obtain the gradient for this Lagrangian function. Now what is the gradient:

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial P_1} \\ \frac{\partial L}{\partial P_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial L}{\partial P_N} \\ \frac{\partial L}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \frac{\partial F1}{\partial P1} - \lambda \\ \frac{\partial F2}{\partial P2} - \lambda \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial FN}{\partial PN} - \lambda \\ (P_{load} - \sum_{i=1}^{N} P_i) \end{bmatrix}$$

So, that means you need to obtain the partial differential function of the Lagrangian function with respect to each generator as well as with respect to λ. Now I will explain you the steps how we can solve this problem with the help of gradient approach:

Step1: Select the starting values, P_1^0, P_2^0... P_N^0
Where, $P_1^0 + P_2^0 + ... + P_N^0 = P_{load}$
Step2: Compute the initial λ_i^0 for each generator $\lambda_i^0 = \frac{\partial F_i(P_i)}{\partial P_i}$ at P_i^0;
$\forall\ i = 1, 2, \ldots, N$
Step3: Compute initial average incremental cost
$\lambda^0 = \frac{1}{N}\sum_{i=1}^{N}\lambda_i^0$
Step4: Compute ∇L
Step5: If $|\nabla L| \leq k$ (tolerance), the solution converges (pre-chosen) otherwise go to step 6.
Step6: Update $x^i = [P_1^i, P_2^i, \ldots, P_N^i, \lambda^i]^T$
$= x^{i-1} - \alpha\ \nabla L$
Step7: Go to step 4

Now in the λ iteration approach what we did ? we chose a specific λ for which we calculated the generation. Here it is a opposite way what we are doing we are taking out we are you know assuming some generations some generation random generation value for each generator such that the total generation will be equal to total load. Now you have anyway find out the found out the generation obtain λ for each generator the for the specific generator that means obtain dou f i by of that specific generator with respect to dou P i right at P i 0. So, you obtain dou f i by 0 and you do this for all the generators that means obtain λ for the chosen power for all the generators right and compute the initial average incremental cost because this is not the optimal value the λ of each generator will be different. So, obtain the average now. So, ultimately we need to get a point where λ of each generator will be equal to the average value right. In the beginning, the λ of each generator will not may not be equal to the average. So, then compute gradient of Lagrangian function right. So, what is the gradient of Lagrangian function if you see here dou f 1 by dou P 1 minus λ that means what is this ? let us say this is the first iteration I am speaking λ 1 of 0 λ 1 of 0 that means λ 1 represents λ of the first generator at the 0th iteration minus λ average of 0 at the 0th iteration what is the total average that is what we are doing indirectly. Now if this is 0 then that is the final point now we have come to the average we have come to the optimal point. So, if that error is not 0 that means we need to do the next iteration you obtain the average and check out whether this error is less than the tolerance limit or not and if it is not the case then update the value of x i that means obtain new variables what are the new variables new generation basically. So, that means the generation value of each generator is updated and that is based on the gradient expression x i is equal to for the next generation the value is the previous generation minus the steepest descent alpha into gradient of L right and then go to step 4 then ultimately you will get the point where you will get the error as close as possible to 0. Now let us take a problem to understand it in a better way solve the economic dispatch problem for a total load of 800 megawatt right using these generator cost functions write a MATLAB program to also do the same and I can suggest you also can carry out a you know simulation or MATLAB coding C coding and to you know have an experience or exposure how this problem can be used using a computer program right. So, λ iteration also can be tried out and this can also right gradient approach. Now anyway you have the fuel characteristics for each generator choose a alpha then let us say it could be 100 and choose some generation for each initial generation value for each generator given by:

$$F_1(P_1) = 1683 + 23.76P_1 + 0.00468P_1^2$$

$$F_2(P_2) = 930 + 23.55P_2 + 0.00582P_2^2$$

$$F_3(P_3) = 234 + 23.70P_3 + 0.01446P_3^2$$

Use α =100 and starting from

$P_1^{[0]} = 300\ MW, P_2^{[0]} = 200\ MW,$ and , $P_3^{[0]} = 300\ MW.$

λ is initially set to the average of the incremental costs of the generators at their starting generation values:

$$\lambda^{[0]} = \frac{1}{3}\sum_{i=1}^{3} \frac{d}{dP_1} F\,i\left(P_i^{[0]}\right) = \frac{1}{3}\begin{bmatrix} 23.76 + 0.009372\ (300) + \\ 23.55 + 0.01164\ (200) + \\ 23.70 + 0.02892\ (300) \end{bmatrix} = 28.27$$

Now increase for the next iteration check whether the λ is same as the λ average if that is not the case then ultimately you get the new λ you follow the same steps which I have asked you to follow then ultimately you would get this set of results:

Iteration	λ	Total Generation	P_1	P_2	P_3	Cost
1	28.28	800	300	200	300	23,751
2	28.28	800	301.7	202.4	295.9	23,726
3	28.28	800.1	303.4	204.8	291.9	23,704
4	28.35	800.2	305.1	207.1	288.1	23,685
5	28.57	800.7	306.8	209.5	284.4	23,676
6	29.23	801.8	308.7	212.1	281	23,687
7	31.06	805	311.3	215.3	278.4	23,757
8	36.08	813.7	315.7	220.3	277.7	23,983
9	49.79	837.4	325.1	230.3	282.1	23,632
10	87.19	901.9	348	253.8	300	23,449

But you can see that there is no convergence to a solution because the fuel cost you can see here ultimately you get the generator values and corresponding fuel cost the fuel cost is not converging you can see here fuel cost is keep on changing right it is not constant because you have to choose the right acceleration factor otherwise I told then it could lead to wrong values. So, then and simple variation can give us a better solution. So, what is that realize that one of the generator is always dependent variable and remove it from the problem. Now earlier there were three generators now what we are doing is anyway at some of the generation of three generators should be equal to the total load or the third generator let us consider it to be dependent variable. So, P3 is nothing but total demand minus P1 minus P2 this is also same. So, now then total cost function becomes cost function is nothing but the cost function of the first generator, cost function of the second generator and the cost function of third generator which is again dependent on cost function of two generators. Now, this function stands by itself as a function of two variables with no load generation balance constraint. Now we are just removing that constraint another constraint which is summation of generation is equal to total load because that is been already taken care by this expression. Now obtain the cost now the cost can be minimized by gradient method as follows. Now dc by dP1 by dc by dP2 because you now have only two variables you have removed two other variables one is lambda and P3 you have removed them. Now this is nothing but df1 by dP1 minus df3 by dP1 and df2 by dP2 minus df3 by dP2. Note that the gradient goes to 0, when the incremental cost at generator 3 is equal to that of generators 1 and 2 obviously. Now a simple variation the gradient steps are performed similar to the previous case x of 1 the new update will be the previous minus the alpha into gradient of C where x is equal to just two variables now. Earlier x was P1, P2, P3 and lambda. Rework the previous problem with the reduced constraint reduced gradient now you will get gradient of C is nothing but these values and alpha is set to 20 now. Now if you solve this problem by using a MATLAB program then you can find out that earlier case the fuel cost were not converging you can see here now it is converging you get the same value that means the generators combination is almost in a optimal manner. Now there is another important method which is Newton's method which is even faster compared to the modified gradient method. Now the solution process can be taken one step further observe that the aim is to always drive the gradient to 0. Now the goal is to make the gradient as close as possible to 0. Now since this is just a vector function Newton's method finds the correction that exactly drives the gradient to 0. Ultimately what we are doing we are finding that correction minus alpha into gradient of x that is nothing but a correction. So we need to find out that optimum correction to which you know the new variable will be taking us or leading us to the 0 value. So that is what Newton method is trying to solve.

First we will take help of Taylor series of expansion. Let us say there is a function:

$$g(x + \Delta x) = g(x) + [g'(x)]\,\Delta x = 0$$

This is a basic Taylor series that you can learn from your basic mathematics. So and the objective function is given by:

$$g(x) = \begin{bmatrix} g_1(x_1, \ldots, x_n) \\ \vdots \\ g_n(x_1, \ldots, x_n) \end{bmatrix}$$

Now obtain the Jacobian:

$$g'(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial gn}{\partial x_1} & \cdots & \frac{\partial gn}{\partial x_n} \end{bmatrix}$$

The adjustment at each iteration step is given by:

$$\Delta x = -[g'(x)]^{-1} g\,(x)$$

If the function g is the gradient vector ∇L_x, then

$$\Delta x = -\left[\frac{d\nabla L_x}{dx}\right]^{-1} \Delta L$$

For economic dispatch problems:

$$L = \sum_{i=1}^{N} F_i(P_i) + \lambda\,(P_{\text{load}} - \sum_{i=1}^{N} (P_i))$$

Ultimately here you have:

$$\frac{\partial \nabla L}{\partial x} x = \begin{bmatrix} \frac{d^2L}{dx_1^2} & \frac{d^2L}{dx_1 dx_2} & \cdots \\ \frac{d^2L}{dx_2 dx_1} & \frac{d^2L}{dx_2^2} & \cdots \\ \vdots & \vdots & \ddots \\ \frac{d^2L}{d\lambda dx_1} & \frac{d^2L}{d\lambda dx_2} & \cdots \end{bmatrix}$$

Solve the previous economic dispatch problem example using Newton's method. Also write a Matlab program for the same. You can also try out Newton's method to have the solution by using Matlab program. The gradient function is the same as the previous example. Let the initial value of lambda be equal to 0 and obtain the Hessian matrix. Hessian matrix is nothing but the partial differential function of the gradient function. Then:

$$[H] = \begin{bmatrix} \frac{d^2F_1}{dP_1^2} & 0 & 0 & -1 \\ 0 & \frac{d^2F_2}{dP_2^2} & 0 & -1 \\ 0 & 0 & \frac{d^2F_3}{dP_3^2} & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

Similarly obtain for all the entries. Now the initial generation values are also the same as that of the example 1. Now with this approach you can see that starting from λ is equal to 0 in the next iteration itself, the second iteration itself you are getting the value which is almost converged and you are just verifying with the other few steps as shown in Table:

Iteration	**λ**	**Total Generation**	P_1	P_2	P_3	**Cost**
1	0	800	300	200	300	23,751
2	27.19	800	366.3	313	120.7	**23,192**
3	27.19	800	366.3	313	120.7	**23,192**
4	27.10	800	366.3	313	120.7	**23,192**

So Newton's method is the most you know fast method where you get the solution. Even in the case of those problems where you have complicated polynomial where there is fraction involved. So the advantage is the quick convergence to a solution as compared to the gradient method. That is it. Thank you very much.