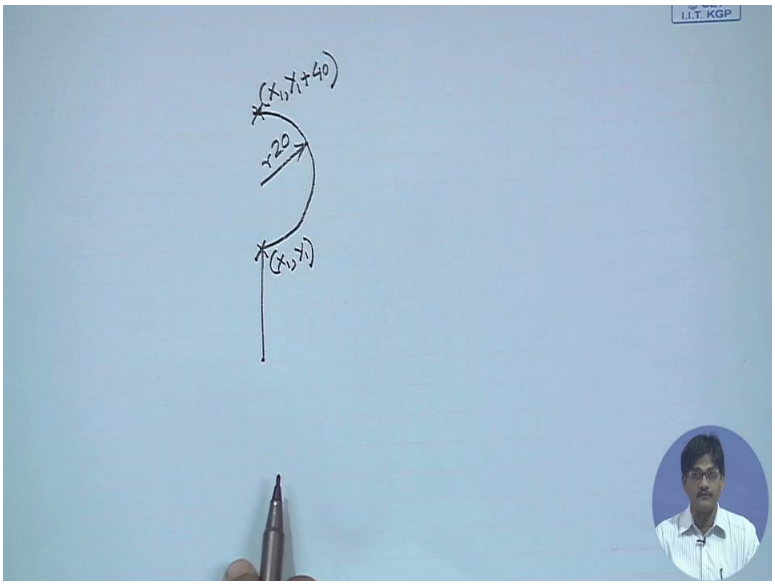
**Computer Numerical Control of Machine Tools and Processes**
**Professor A Roy Choudhury**
**Department of Mechanical Engineering**
**Indian Institute of Technology Kharagpur**
**Lecture 12**
**Computer Aided Offline Programming**

Welcome viewers to the 12th lecture of the online course "Computer numerical control of machine tools and processes". In this particular lecture we will learn something about computer aided off-line programming, what does this actually mean? This means that up till now we have studied about manual programming in which the operator was I mean the programmer was writing lines of the program by looking at the part geometry and that was being executed and the machine was successfully done. Problem is, wherever we have complex geometry, what do we mean by complex geometry?

Previously, all the geometries that we have studied, they are containing circles or parts of circles I mean circular segments which are taken, they are either quarter circles or half circles and straight lines are connecting them up in such a way is not difficult to find out or compute their coordinate, their end coordinate locations, let me give you an example on this piece of paper okay.

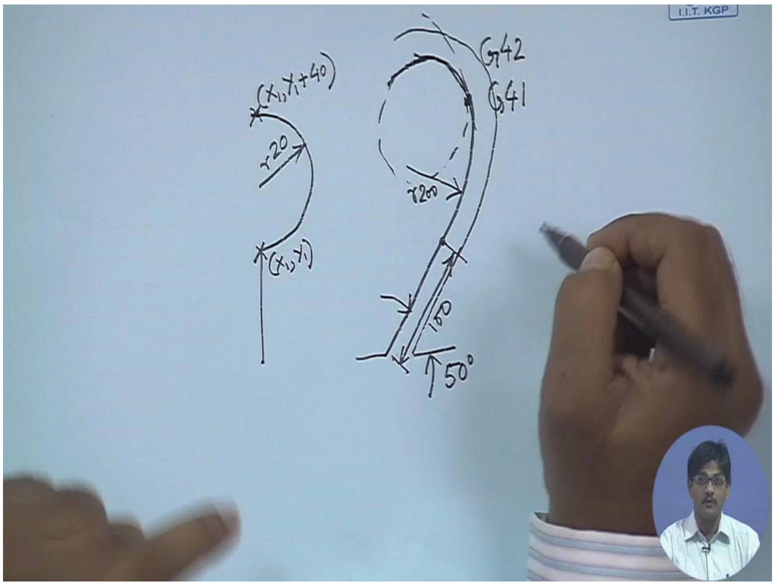
(Refer Slide Time: 01:50)




Suppose you are heading this particular body, from here I go up to this position and then I take a turn and come to this particular location. So if I know the radius of the circle like R 20 it is very, it is extremely simple for me to calculate this particular coordinate location, I will say this is X 1 and this is Y 1. If these are known to me, then I can definitely say this point must be X 1 once again and this must be y 1 + 40 that is it. So in the previous parts that we have discussed for manual programming, we have always selected those parts where computation of these coordinate locations where one geometry touches the other, the computation of these parts is very simple.

But there are parts in which it is extremely difficult because there will be very complex then tangency situations and we will be dealing with circular segments which are not exactly regular parts like they are not quadrants or they are not semicircles, et cetera; in those parts it is very difficult. Can we do away with those parts? But whenever we have some Engineering or aesthetic requirement, we have to incorporate those segments those geometrical segments into our part and there we will have a lot of difficulties, what sort of difficulties? First of all finding out those coordinate locations from the part data which is provided, let me give you another example.

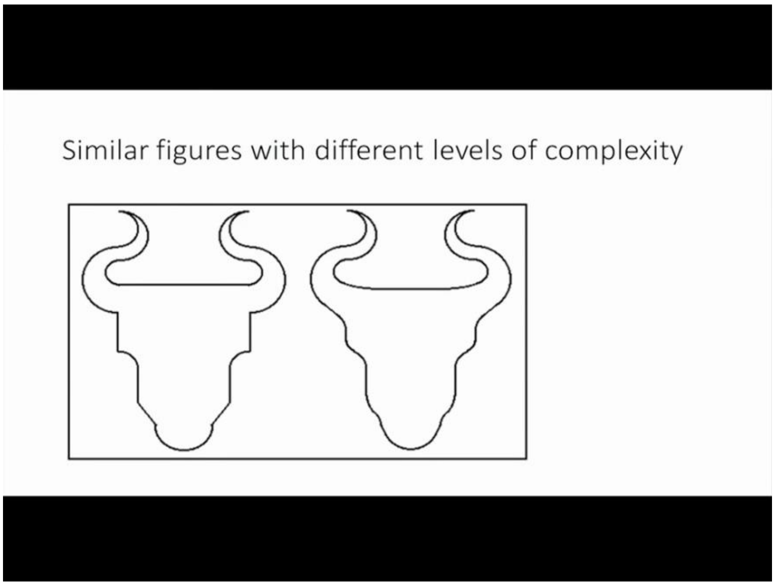
(Refer Slide Time: 03:50)




So if I have a straight line from here to here, some distance 100 and there is some angle provided here, this angle is say 50 degrees and then you are having a circular path say R 200 and this is again connecting up with another circular path here, large circle with small circles, in that case finding out these points is a bit difficult. Suppose your actual job profile is this, one circle connecting up with another circle and this circle connecting up with a straight line in these cases it is extremely difficult to find out 1st of all these points because you would have to use you would have to use a lot of coordinate geometry calculations. Over and above that, after that you have to calculate the path of the tool.

For the path of the tool we are employing G42 or G41 that is no problem. But whenever it comes to computation of all these geometry, there I mean the programmer has to do some calculation or there can be a software which can find out these coordinate points for us, how can that software do that? We have to intimate that information to the computer through some language and some instructional codes and the computer will be doing the calculations for us, so let us go into some examples, this is one example.

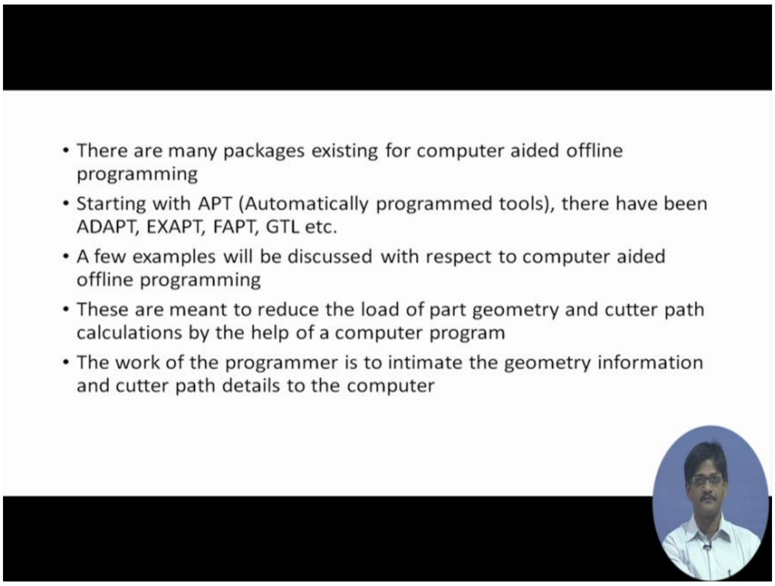
(Refer Slide Time: 05:50)




See this one, I have chosen all all of these segments to be either semicircles or quarter circles, et cetera et cetera. And this one though it looks formidably complex, once I give you this hint you will say oh give me those radii of those circles and I can find out any and every coordinate location, so we will start with 00 here. If this is radius, then this point must be r, r, if I give you this angle, you can find out this point, if I give you this segmental height, you can find out this point. Again it will be addition of radius to X and Y then rise up by this height along Y and then it is semicircle so Y increases by the diameter, X remains same Y increases by the diameter, X remains same so on and so forth, it is extremely simple.

But once you slightly raise the level of complexity and go for some round rounding and some rushing of all the sharp corners, you end up with something like this, it is much more difficult. For example, here you are having this circle is touching and this particular circle is this particular circle is touching this particular circle, it is much more difficult. So let us take an example to find out what kind of instructions have to be given to the computer to do it for us.

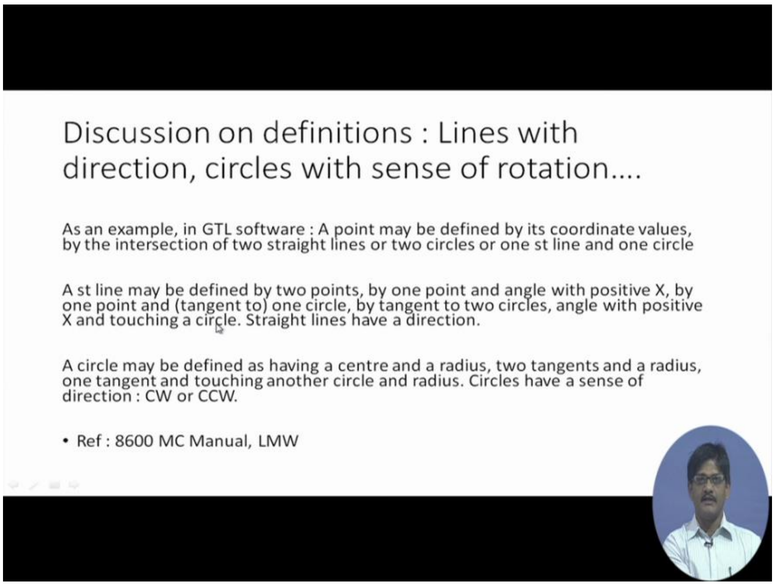
(Refer Slide Time: 07:22)




Lot of packages are existing for this for example, APT was the 1st package after this particular point was identified this problem APT (Automatically Programmed Tools) started out and it had a number of surfaces to be defined after which the computer could find out all these coordinate points and there have been extensions and further modifications and developments of this basic APT like ADAPT, EXAPT, FAPT, GTL, etc etc. So these programs will be reducing the load of the path geometry and the cutter path calculation by the help of computer program. So the programmer will have to intimate geometry information and cutter path details to the computer.

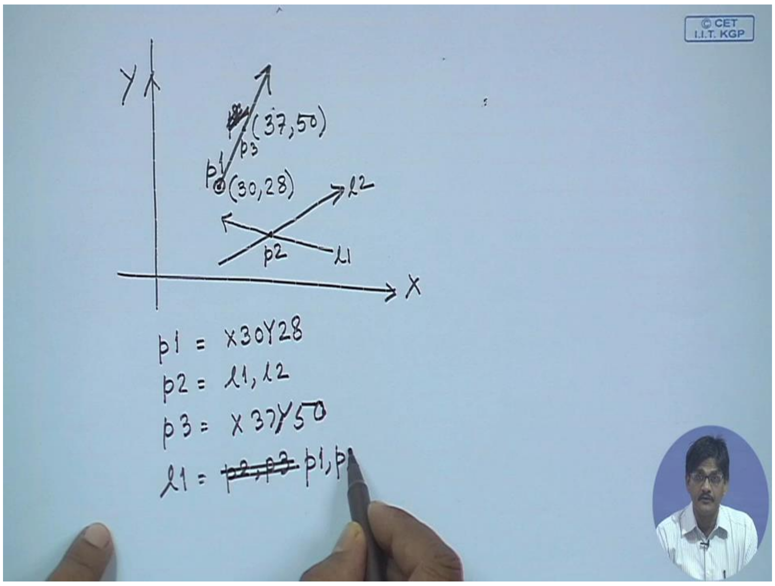
These software generally define coordinate systems, points, lines, circles surfaces etc and the cutter paths are defined with respect to these geometry elements it will be ultimately finding out G and M codes. So let us for example, a line might be defined as X20Y30, X220Y330, now what could it possibly mean? This means that the straight-line is being defined which is line between these particular points X20Y30 and X220Y330 and it is directed that means it has a sense now, it is directed from the first point to the next point, which means all lines and you will find all circles also in a the software that we are going to discuss, they are supposed to have a direction associated with them.

(Refer Slide Time: 09:17)




So let us look at some examples, so if we take up the GTL software from the reference manual 8600 MC of Lakshmi Machine Works, so there a point may be defined by its coordinate values like you can simply say P1 = X20Y30 it is a point okay and let me give you some examples here, say I want to define a point.

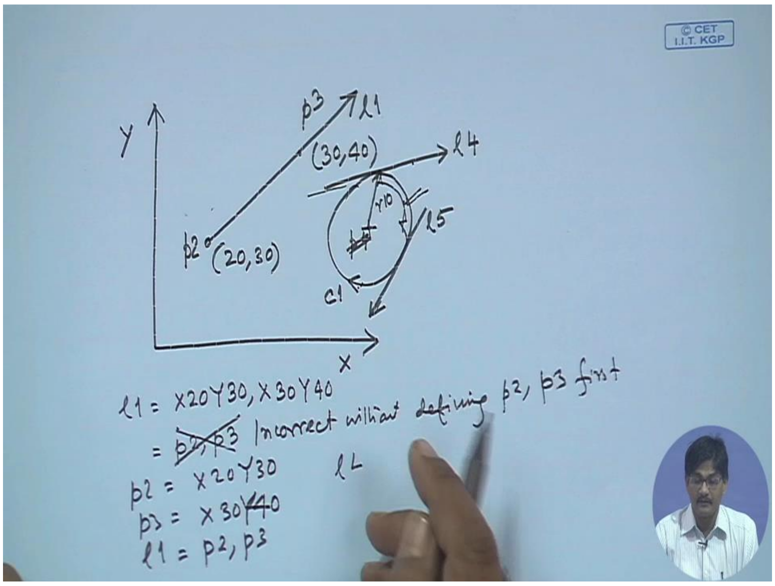
(Refer Slide Time: 09:53)




This is my coordinate system, this is X this is Y and I want to define this point which is say 30, 28, so I can simply say if this point be P1, P1 = X30Y28 that is it, it defines this particular point. Can I define points in some other manner? Yes we can, we can define points as intersection of two predefined straight lines like this might be L1, this might be L2 and we can simply right P2 = L1, L2 that is it, it defines P2 as this point. So in the same manner we can also define straight lines for example, suppose this is point P1 already define point P2 sorry P1 is already defined here, so say point P2 is say how much, 37 and say 50.

If this is point P2, we have to wait a minute point P2 is already there, so let this be P3. P3 can be defined once again that is simply as X37 Y50 and I can define a line as P2, P3, how would it look like? P2 to P3, this is the line that I am talking of, this is P 1 oh wait a minute I wrote P2, P3, I am cancelling it. P1, P2, let me make it a bit more clearer and cleaner, please bear with me.

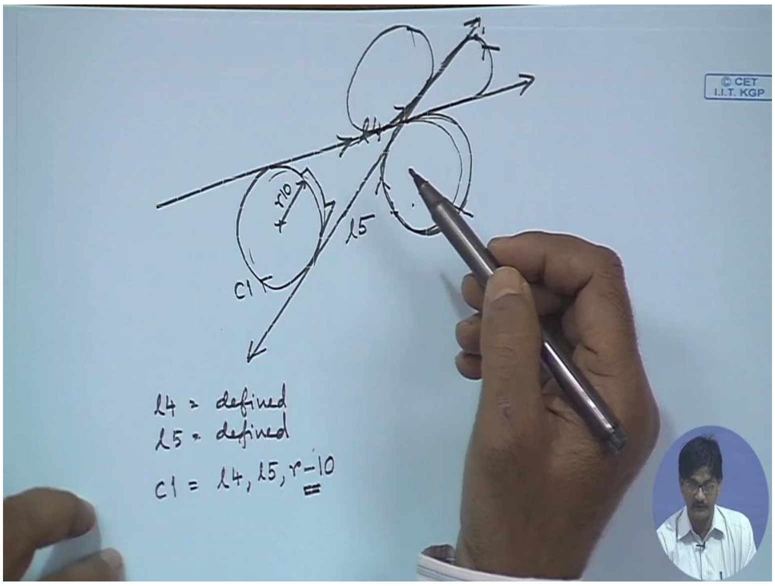
(Refer Slide Time: 12:12)




X Y P2 (20, 30) and P3 (30, 40) and this is the line L1, so I can write L1 = X20Y30, X30Y40, I can also write L1 = P2, P3, but in that case I need to first define P2 = X20Y30, P3 to be X30Y40 and then only I can define it as P2, P3. So this one would be key without definition of P2 and P3, this would be incorrect. Incorrect without defining P2, P3 first okay I hope this is understood. And therefore, we have ways of defining straight lines, we have ways of defining circles, say I want to define a circle here okay, how would you would you like to define it? Say I have already defined its centre to be P4 okay that that would be a bit easy.

Let us say I have already defined this line to be L4 and I have already defined that line to be L5 and I know that its radius is 10 millimeters, so how do you do that? What we do in this case is this that if the circle to be defined is to have this particular direction and if it is say C1, we will try to find out along C1 the shortcut from one geometry to another and it will have to be just like we have one-way roads, so it has to be a one-way route. So we will say that C1 will provide a one-way shortcut from L4 to L5 in this manner, so what does it do? So if you are driving along L4, you can take a shortcut along C1 and reach the correct direction of L5 and this direction has to match with the direction of defined C1.

(Refer Slide Time: 15:31)

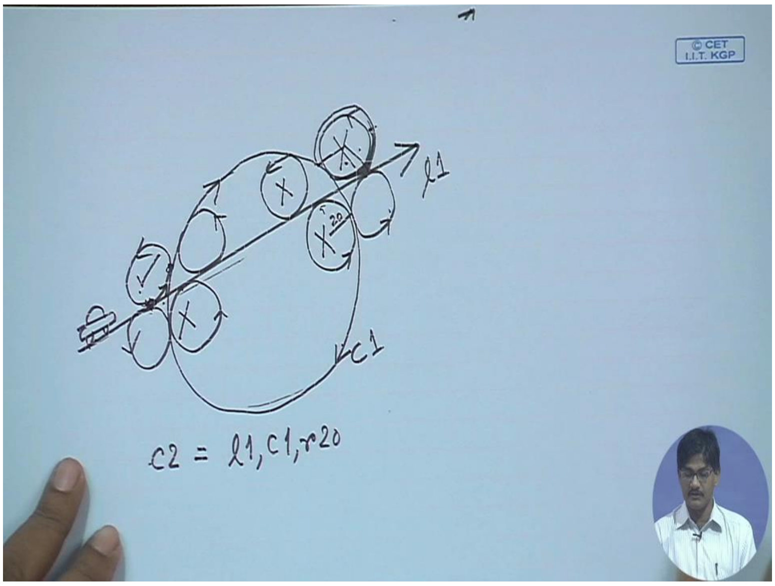



So if we have L4 let us take a fresh page, if you have L4 defined this way, if you have L5 defined this way and you want to define C1. L4 say is defined, L5 say is defined as shown in that case C1 can be defined as suppose C1 is having R10, so C1 can be defined as moving along L4 you can go to L5 in a direction clockwise this is the definition of C1. It reads this way, suppose you are moving along L4, if you go want to go to L5, using C1 as a shortcut or intermediate path, in that case if you take this particular direction, minus means clockwise and plus means counterclockwise, so along a radius of - 10 matches with the direction you intend to define C1, so this is the direction of the shortcut or intermediate path between L4 and L5 along radius 10.

So a circle and only one circle will satisfy all these conditions, why so? Can you have other paths from L4 to L5? We can definitely have, you can move this way and move to L5, but see in this case the problem is you are ending up in opposite direction of L5, so this path is not possible. What about this path? You can go this way and you can welcome back here, but problem is this is counterclockwise, this is not allowed, but this path is possible that is you go this way and you end up this way. This small path okay, in which direction does it move? It moves in the counterclockwise direction that is not allowed.

The moment in the definition you are giving minus, your path has to be negative I mean your path has to be clockwise, so this is not chosen because this is, wait a minute this is clockwise, this one is clockwise but it ends up in the opposite direction of L5, it is - L5, so this way we can quickly choose the particular circle which is being defined or the straight line which is being defined, let us quickly take just another problem and we will be proceeding without example.

(Refer Slide Time: 18:33)




I have this straight line and I have this circle, they are predefined. Suppose this is L1 and this is C1, I want to define a circle which is touching both of them and which is clockwise, so how many possibilities are there of a circle touching these 2? It can be here, it can be here sorry it can be here, it can be there, it can be here, so many. Oh my God, 8 possibilities and if you consider the sense, there are 16 possibilities so let us fix it up from the beginning itself, I want this circle to be counterclockwise so in that case only 8 possibilities are there 1 2 4 5 6 7 8.

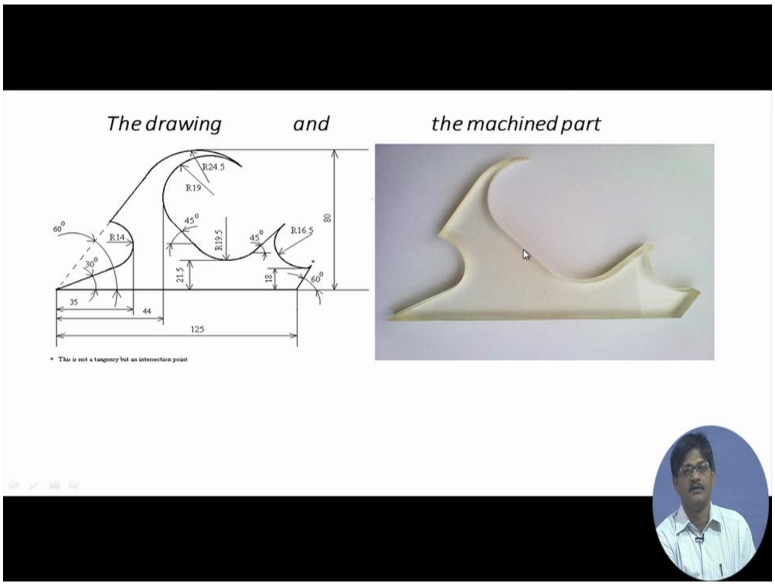
Now the circle suppose I define as C2 equal to and suppose I write simply say L1 C1 and then R20, so all these say they are of 20 millimetres radius, so which circle would I be talking about. So let us see, moving along Highway L1, you are moving along Highway L1, you want to enter into C1 with a radius of 20 millimetres shortcut and this has to be counterclockwise, so that is fine I can go this way and I can go along C1, but this part is clockwise, this has to be counterclockwise because it is positive, so this is not a possible choice. What about this, I can move this way, I can go all around and I can reach C1, but here is the problem is once again this path is clockwise, so this is not a possible choice.

I can move this way, I can move this way and I can come this way but 1st of all this is opposite of C1 and this is again clockwise okay, so in that case what about this path? This is your car, you are moving this way and you take this particular shortcut and you enter the correct direction of C1, everything seems to match, L1 C1 counterclockwise path, so this I will give a tick, but is it possible that there are other possibilities also? Let us take this one okay, okay I go this way, I take counterclockwise path, so far so good, but I go into - C1 this is not this is not possible.

What about this? Okay, I go this way I take a counterclockwise path and yes I have entered C1, that seems to be alright, but the problem is whenever there are 2 possibilities, the machine makes a convention that that particular one will be the default if it is describing the minor arc, this one is describing a major arc, so truly speaking it is not a shortcut at all. Okay those are just a verbal description, but generally whenever the major arc of the circle being defined is concerned in the in the definition, that is not accepted major arc. So though these two fully qualify in the definition, this one is dropped.

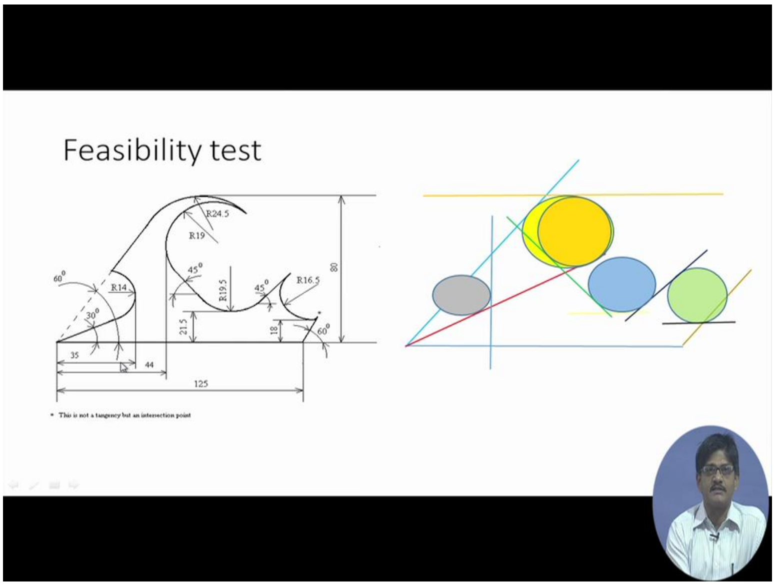
So we understand that by this shortcut method it essentially means that tat the points of tendency, both geometry elements would have the same direction. L1 comes here, the circle is rotating this way, so they have same direction of rotation at this point okay. So with this one let us move right back into our example.

(Refer Slide Time: 23:00)




This is the part that we have machined for this particular geometry, what does this geometry what does this geometry concerned with? This is made out of Perspex okay and it is done on a machining centre, first of all we have given wide number of geometry definitions and these are to start with, there is a straight line here, it is tangent to a circle that one there is another straight line making 60 degrees with the positive X that is touching a circle at the at the top and this circle is having a overall dimensions provided and it is touching another small circle internally that is having a 45 degree tangent so on and so forth.

(Refer Slide Time: 23:57)



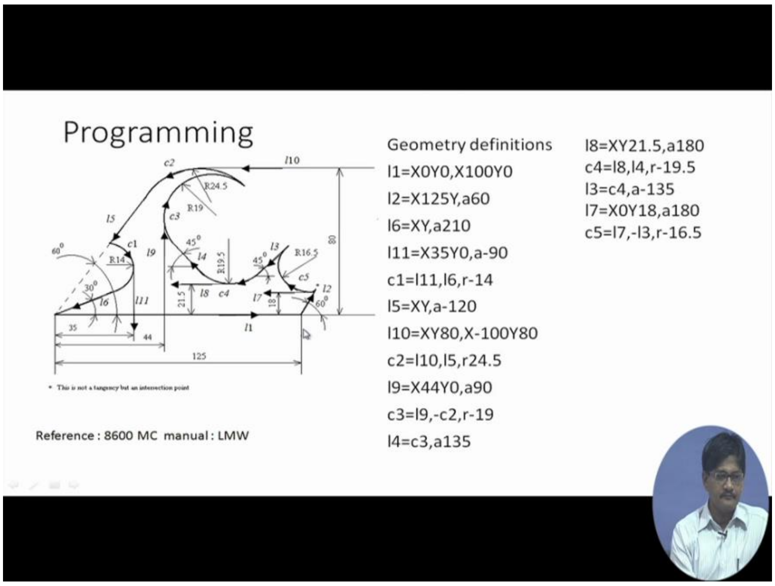

So whenever you come across such geometry elements which are extremely complex, so first of all there has to be a feasibility check. There is another example in which we have also machined it out using this particular software, let us see this feasibility test, what is this feasibility test concerned about? First of all we draw the geometry elements and find out whether we are able to draw it completely that means whether there is adequate dimensions and also whether there is redundant dimensions, in both of those cases there will be something wrong.

So first of all let us see we have a straight line here to start with at 30 degrees, so we reproduce it here, there is a line at 30 degrees. Next we have another straight line at 60 degrees starting from the origin, so we draw another line at 60 degrees so far so good, but there is a circular arc here, we do not know its position directly from the drawing. But from the drawing it is shown that it is having a vertical tangent and 35 millimetres away from the origin along X axis, so we draw that line 1$^{st}$ and find that this particular circle can only have one particular position which is just a moment.

Sorry yeah, this circle can have only one particular position if it is to be tangent to this straight line and that straight line, so this circle is fixed in position. In the same way, the upper yellow circle is fixed by the position of these 2 tangents okay, one horizontal tangent and one inclined tangent and its radius. The orange circle is fixed in its position by touching the yellow circle and it is also having a vertical tangent which I have not drawn here at 44 millimetres away from the origin along X axis. Next, this straight line is defined by being at 45 degrees to the orange circle.

So this way you will find, all these geometry elements can be checked for their feasibility whether they are properly define or not, whether adequate dimensions have been given, whether redundant dimensions has been given okay, so let us go for the programming.

(Refer Slide Time: 26:39)




We have given certain names to these geometry elements for this is L1, this is L2, this is C5, and this is L3 together with a particular direction. So let us see, L1 is being defined to be so starting from X0Y0 to X100Y0, so it is a straight line along the positive X direction. L2 is passing through this point, which is X125Y, Y means Y0 and at angle 60 that means the positive direction of L2 will be making an angle of 60 degree with the positive X, so this is L2 and after that L6 is defined. L6 is pointing downwards and therefore, it passes through the point 00, L6 is X0Y0 and angle is 210.

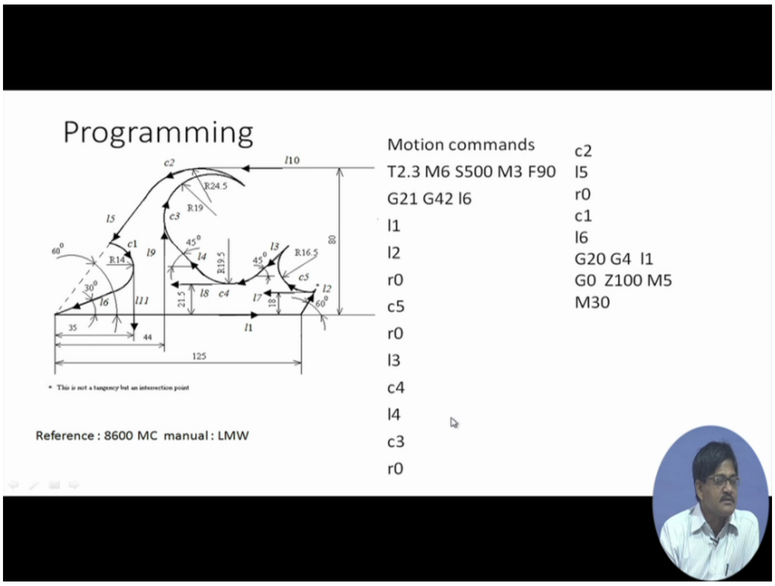
Angle is always measured counterclockwise to the X positive, so from X positive we move counterclockwise and end up with the positive direction with the defined line and here it is 180 + 30 = 210. In the same way we can define L11 which is directed downwards and it is defined as X35Y0 and angle - 90. If negative angle is given, it means from positive X we have to move clockwise okay till we reach the positive direction of the defined line, so L11 is directed downwards. C1 is defined to be L1, L6, R - 14, so following our convention that we had discussed L11, so driving along L11 if we move onto L6 okay and the radius of the circle being 14 and in the clockwise direction okay, we can define C1, so C1 is defined this way.

Likewise, we can define L5 that quickly see L5 at - 120 degrees, we can define L10 which is at the top, we can define C2 from L10, we move to L5 moving in the counterclockwise direction and the radius is 24.5, L9 can be defined as a vertical line upwards that is why it is A90, C3 can be defined to be driving along L9 if you come to C2 okay come to - C2 therefore, your rotation is negative and therefore it is clockwise and therefore negative, so R - 19 so this way all these circles and straight lines they have being drawn.

Once you define C3, you can define this line to be touching C3 and at 135 degrees, so there is L4, L4 is defined that way and this circle can be defined as moving along this line, you end up in L4, so this line has to be 1st defined as L8, so from the drawing we find that it has a horizontal tangent L8, from there you want to move to L4 and therefore, L8 is having 180 degrees angle with positive X and C3 is defined as L8, L4 are - 19.5, so this way all these geometries have been defined. Once the geometries have been defined, it is extremely simple to define all the cutter paths okay.

Let us just have a last look at the last geometries C5, C5 is defined as having a horizontal tangent L oh wait a minute, I think I have defined written yeah L7 right. L7 is defined previous to this, it is having a horizontal tangent L7 and from there we are moving to - L3 at clockwise 16.5 radius, so that way C5 is defined.

(Refer Slide Time: 31:10)



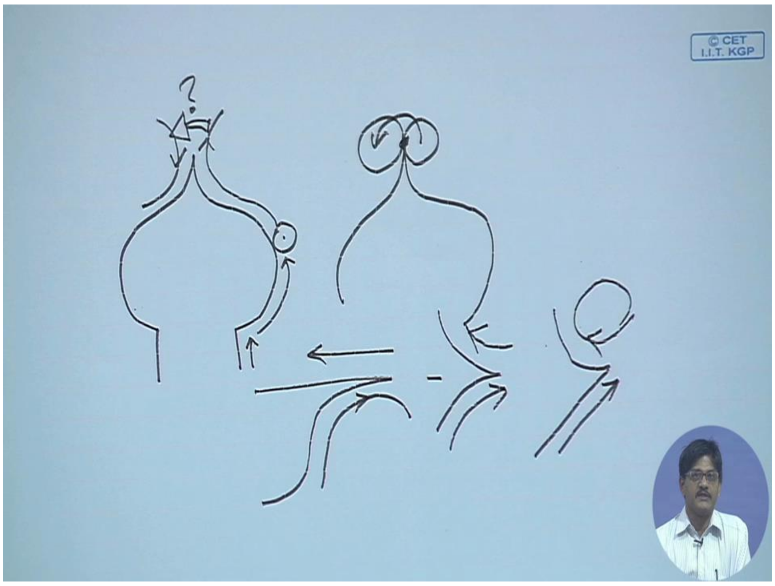

Once the all the dimensions I mean all the geometry is define, we chose the cutter, we change the cutter and put it in on the spindle, we define the speed and we define the feed then G21 starts or invokes or activates this particular GTL software, we declare that we are taking cutter right compensation and then we mention the first last geometry elements to be covered, in case of closed curves we give the last geometry element to be covered first and then we start giving the geometry in proper sequence. Why do we give that, to determine the starting point, the intersection of the last geometry and the first geometry will determine our starting point.

So, after that it is quite simple, we move along L1 if you if you ignore R0 for the moment, L1, L2, C5, okay L1, L2, C5 then L3, L3 is this then it must be C4, we have C4 then L4 then C3 then C2 then L5, C1 and L6 and at the end once again in order to determine where you stopped, the intersection of L6 and L1 will determine that, so L1 is mentioned at the end, L6 and L1 intersection and in the starting point L6 and L1 intersection. So G20 marks the end of the GTL cycle and this sorry this should be G40, one 0 has been missed, G40 L1, so this one stops the cycle and after that the the spindle goes up and the machine is put off.

What is R0? R0 means that whenever the tool compensation is being calculated, the machine will the machine control will draw offset lines and offset circles in order to find out the locus of the centre of the tool, but there are some situations in which these 2 loci will not have any intersection. For example, when the tool Centre is coming this way, it ends here and the tool Centre is moving along here along this offset line, these 2 lines do not have any intersection.

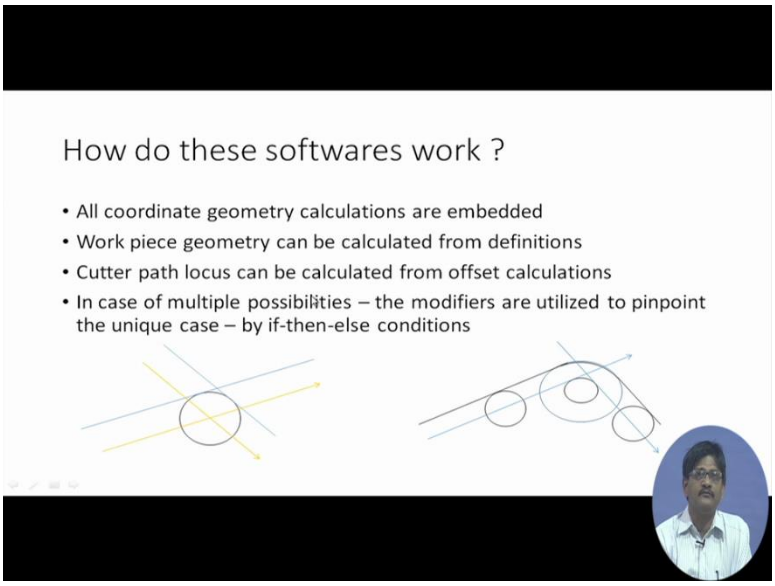
So if you have this, have a look at this plain paper, you will find how this is opening. For example, these cases definitely there will not be any intersection.

(Refer Slide Time: 33:48)




This one for example, if you are moving along this path so the tool which is coming, it does not have any intersection and this way, so the machine cannot find out how it is supposed to move from this path to that path. When it is machining this way, the cutter is coming here, it has a dilemma, and how do I know that I am supposed to move from this path to that path? So here the cutter is given an instructions to sort of roll about a 0 radius to come to this point, it rolls about this particular point in a radius of 0. In the same way, in all other cases either it is this or it is this or it is say this one, so in all these cases what happens is the cutter comes, it cannot move to this path.

(Refer Slide Time: 35:14)




The cutter comes here; it does not know how to move to this path, etc et cetera, so here in all these cases R0 has been mentioned so that the cutter moves from this side to that side okay, so this is an example of off-line programming. There are some other things which I think we will be putting in to the next lecture, so thank you very much thank you.