Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

EEE 212 : Numerical Technique Laboratory

Topic : MATLAB Project

Project Title:

Calculation of Duckworth-Lewis Par Score in One Day Cricket Match Using Matlab.

Submitted By-

Name : Rifat Hassan / 1506177

Department : EEE

Section : C-2

Level: 2, Term: 1

Date of Submission: 14.07.17

Introduction

Duckworth-Lewis is method a mathematical formula invented by two English statisticians Frank Duckworth and Tony Lewis. It is generally used in the ODI and T20 cricket matches when the match is interrupted due to weather or any other circumstances and the match officials are compelled to reduce the total allocated overs to the teams to finish the match within scheduled time. It is the most accepted suitable method to calculate the second innings target in those situations and is used worldwide highly in recent times.

There are two editions of this Duckworth-Lewis Method. One is Standard Edition which was mostly used from 1996-2004 and was replaced by Professional Edition in 2004. But still Standard Edition is available in some domestic cricket matches where Professional Edition is not available. Our project is on the Standard Edition of Duckworth-Lewis Method in One Day Cricket.

Mathematical Theory

Duckworth-Lewis method comprises of an essence called "Resources". Whenever a team comes to bat in a cricket game, they have to resources available which they use to score runs, one is overs allocated to them and the other is wickets available in their hand. The term "Resource" is a combination of these two factors. In an one day match, a team has 50 overs and 10 wickets in their hand, and according to D\L Method , they have 100% resources in their hand. If they can bat whole 50 overs without any interruption, they are considered to have used total 100% resources. But if there is any interruption and their allocated overs are reduced due to shortage of time, they are considered to have used resources less than 100%.

There is a table formed by Duckworth and Lewis which is the basis of D\L Method. This table exhibits the percentages of resources available to a batting team at all possible combinations of the two factors, overs left and wickets available. An over by over D\L table is presented below. ICC has published the ball by ball D\L table of Standard Edition, and we have used that ball by ball table in our program.

In the table we see that when a team has 50 overs and 10 wickets left, they have 100% resources left. When we gradually go down, like when they have 21 overs left, they have 58.7 % resources when they are no wickets down. 56.7% resources left when they are 1 wicket down etc….

OVERS LEFT	WICKETS LOST 0	1	2	3	4	5	6	7	8	9	OVERS LEFT
50	100.0	93.4	85.1	74.9	62.7	49.0	34.9	22.0	11.9	4.7	50
49	99.1	92.6	84.5	74.4	62.5	48.9	34.9	22.0	11.9	4.7	49
48	98.1	91.7	83.8	74.0	62.2	48.8	34.9	22.0	11.9	4.7	48
47	97.1	90.9	83.2	73.5	61.9	48.6	34.9	22.0	11.9	4.7	47
46	96.1	90.0	82.5	73.0	61.6	48.5	34.8	22.0	11.9	4.7	46
45	95.0	89.1	81.8	72.5	61.3	48.4	34.8	22.0	11.9	4.7	45
44	93.9	88.2	81.0	72.0	61.0	48.3	34.8	22.0	11.9	4.7	44
43	92.8	87.3	80.3	71.4	60.7	48.1	34.7	22.0	11.9	4.7	43
42	91.7	86.3	79.5	70.9	60.3	47.9	34.7	22.0	11.9	4.7	42
41	90.5	85.3	78.7	70.3	59.9	47.8	34.6	22.0	11.9	4.7	41
40	89.3	84.2	77.8	69.6	59.5	47.6	34.6	22.0	11.9	4.7	40
39	88.0	83.1	76.9	69.0	59.1	47.4	34.5	22.0	11.9	4.7	39
38	86.7	82.0	76.0	68.3	58.7	47.1	34.5	21.9	11.9	4.7	38
37	85.4	80.9	75.0	67.6	58.2	46.9	34.4	21.9	11.9	4.7	37
36	84.1	79.7	74.1	66.8	57.7	46.6	34.3	21.9	11.9	4.7	36
35	82.7	78.5	73.0	66.0	57.2	46.4	34.2	21.9	11.9	4.7	35
34	81.3	77.2	72.0	65.2	56.6	46.1	34.1	21.9	11.9	4.7	34
33	79.8	75.9	70.9	64.4	56.0	45.8	34.0	21.9	11.9	4.7	33
32	78.3	74.6	69.7	63.5	55.4	45.4	33.9	21.9	11.9	4.7	32
31	76.7	73.2	68.6	62.5	54.8	45.1	33.7	21.9	11.9	4.7	31
30	75.1	71.8	67.3	61.6	54.1	44.7	33.6	21.8	11.9	4.7	30
29	73.5	70.3	66.1	60.5	53.4	44.2	33.4	21.8	11.9	4.7	29
28	71.8	68.8	64.8	59.5	52.6	43.8	33.2	21.8	11.9	4.7	28
27	70.1	67.2	63.4	58.4	51.8	43.3	33.0	21.7	11.9	4.7	27
26	68.3	65.6	62.0	57.2	50.9	42.8	32.8	21.7	11.9	4.7	26
25	66.5	63.9	60.5	56.0	50.0	42.2	32.6	21.6	11.9	4.7	25
24	64.6	62.2	59.0	54.7	49.0	41.6	32.3	21.6	11.9	4.7	24
23	62.7	60.4	57.4	53.4	48.0	40.9	32.0	21.5	11.9	4.7	23
22	60.7	58.6	55.8	52.0	47.0	40.2	31.6	21.4	11.9	4.7	22
21	58.7	56.7	54.1	50.6	45.8	39.4	31.2	21.3	11.9	4.7	21
20	56.6	54.8	52.4	49.1	44.6	38.6	30.8	21.2	11.9	4.7	20
19	54.4	52.8	50.5	47.5	43.4	37.7	30.3	21.1	11.9	4.7	19
18	52.2	50.7	48.6	45.9	42.0	36.8	29.8	20.9	11.9	4.7	18
17	49.9	48.5	46.7	44.1	40.6	35.8	29.2	20.7	11.9	4.7	17
16	47.6	46.3	44.7	42.3	39.1	34.7	28.5	20.5	11.8	4.7	16
15	45.2	44.1	42.6	40.5	37.6	33.5	27.8	20.2	11.8	4.7	15
14	42.7	41.7	40.4	38.5	35.9	32.2	27.0	19.9	11.8	4.7	14
13	40.2	39.3	38.1	36.5	34.2	30.8	26.1	19.5	11.7	4.7	13
12	37.6	36.8	35.8	34.3	32.3	29.4	25.1	19.0	11.6	4.7	12
11	34.9	34.2	33.4	32.1	30.4	27.8	24.0	18.5	11.5	4.7	11
10	32.1	31.6	30.8	29.8	28.3	26.1	22.8	17.9	11.4	4.7	10
9	29.3	28.9	28.2	27.4	26.1	24.2	21.4	17.1	11.2	4.7	9
8	26.4	26.0	25.5	24.8	23.8	22.3	19.9	16.2	10.9	4.7	8
7	23.4	23.1	22.7	22.2	21.4	20.1	18.2	15.2	10.5	4.7	7
6	20.3	20.1	19.8	19.4	18.8	17.8	16.4	13.9	10.1	4.6	6
5	17.2	17.0	16.8	16.5	16.1	15.4	14.3	12.5	9.4	4.6	5
4	13.9	13.8	13.7	13.5	13.2	12.7	12.0	10.7	8.4	4.5	4
3	10.6	10.5	10.4	10.3	10.2	9.9	9.5	8.7	7.2	4.2	3
2	7.2	7.1	7.1	7.0	7.0	6.8	6.6	6.2	5.5	3.7	2
1	3.6	3.6	3.6	3.6	3.6	3.5	3.5	3.4	3.2	2.5	1
0	0	0	0	0	0	0	0	0	0	0	0

The most important task of calculating D\L score is to calculate how much resources the batting first team have used and how much resources the batting second team are going to use. For calculating this, it is to be taken into account in which stage the play has been interrupted and in which stage the play has been restarted. Then we have to subtract the available resources after the interruption from the available resources before the interruption. This is called “Resources Lost”. If there is more than one interruption, then we have add all the lost resources to get the total “Resources Lost”. For example, if a team is in 15 overs losing 2 wickets, it means they have 35 overs left losing 2 wickets, according to D\L table they have 73.0% resources left. Now if there is an interruption and the total over is reduced to 45 from 50, that means after the interruption they have 30 overs left losing 2 wickets, or 67.3% resources left. So the team loss 73.0%-67.3% resources or 5.7% resources. So the total resource used by them is 100%-5.7% or 9.3%. Now if there is one more interruption when it is 30 overs and 2 wickets lost, that means when they have 15 overs left losing 2 wickets, or have resources 42.6% and the match is further reduced to 40 overs, then after the interruption they have 10 overs left losing two wickets, or 30.8% resources available. So they lose more resources 42.6%-30.8% or 11.8%. So total resource lost is 5.7%+11.8% or 17.5%. So the total resources used by them is 100%-17.5% or 82.5%.

Now when the target score for the second team is to be set in an interrupted match, first it is to be calculated how much resources have been used by the first batting team and how much resources are going to be used by the second batting team. Then using this, the D\L Par Score of the first batting team is calculated. In the Standard Edition of D\L Method, there are two formulae for calculating the D\L Par Score of the first batting team.

- When first innings resources is greater than or equal to second innings resources, then D\L Par Score = (First Innings Total / First Innings Resources) * Second Innings Resource.
- When first innings resources is less than second innings resources, then D\L Par Score = (((Second Innings Resource - First Innings Resources) / 100) * G50) + First Innings Total.

If the D\L par score of the first team is a fraction, then the previous integer is considered as the D\L Par Score. For example, if it comes as 289.95 then the D\L par score will be 289.

And the second innings target = D\L par score + 1.

In cases when the match is stopped in the second innings and can’t be restarted again, then the match is declared finished and the percentages of resources used by both the teams is calculated. Then the D\L Par Score of the first team is found out. If second team’s score at the time of interruption is more than that, they win the match, otherwise they lose.

But according to Duckworth-Lewis method, it needs at least 20 overs in each innings to declare a match complete. That means, if a match is stopped before 20 overs of the second innings and can't be restarted, it is declared as a postponed match and no result is declared. And obviously if the match stops in the first innings and can't be restarted, it is declared as a postponed match.

G50: We have found a new term here called "G50". G50 is the average score expected from the team batting first in an uninterrupted 50 overs-per-innings match. Before 2002 the G50 declared by ICC for international one day matches was 225. From 2002-2009 it was 235 and after that it was 245 up to 2014. But it is not such that every time the G50 declared by ICC is properly maintained. It, in fact, varies from ground to ground and is, therefore, a matter of mutual understanding and settlement between the two teams before the match starts. But it is a matter to mention that G50 is required only when the total resources used by second team is more than that of the first team, otherwise not.

Implemenation in Matlab:

First of all we have given a 300 X 12 matrix in the program namely D\L table. It is the ICC approved Duckworth-Lewis Resource Table which shows the number of resources left at various combination of overs and wickets left.

While calculating Duckworth-Lewis Par Score using Matlab, at first we have to give the input as how many times the match has been interrupted in the first innings, n1. This is a number. It can be zero or more than zero, but can never be a fraction or a negative number. So at first we have to check whether it is valid or not. We have found out a way which turns every fraction input to its earlier integer and keeps integer input unchanged. Then if that integer is equal to the taken input, then it is understood that the input was an integer. Otherwise not. By this way we check whether the input is a fraction or not. That way is given below:

We use build in function "round" on the input. It turns the input to the earlier integer if the fractional part is less than .5 and to the next integer if the fractional part is more or equal to .5. Then we subtract the output from the input. If this output is less than zero, it indicates that the fractional part was more than or equal to .5 and now the output is the next integer. So the output is now to be changed by subtracting one. And if not, the output is to be kept unchanged. This way we filter out the earlier integer of a fraction. This way has been used many more times in the later to check out whether an input is an integer or not.

Now if the number of interruptions, n1 is a fraction or a negative number, the program will show the input invalid and will not proceed any more forward. And if it is not, it is zero, then it will directly ask for the first innings total as there was no interruption in the first innings.

Then considering no interruption in the first innings, the program will ask for the number of interruptions in the second innings, n2. Like the earlier way, it will be checked out first whether it is valid or not, it is invalid if it's a fraction or a negative number. If it's invalid then the program will show that and will not go forward. And if it is valid then we will move. If it's zero then there is no interruption in the second innings too and our program will simply show that the second innings target is first innings score plus one.

But if the number of interruptions in the second innings, n2 is more than zero then the program will ask for the information about each interruption. We have used for-loop here from 1 n2 with interval 1 to give the input as the information about each interruption. We have to give the over that was going on at the time of interruption with the number of balls bowled in that over, wickets fallen at that time and the reduced total number of overs after the interruption as input for each interruption. Our program receives each type of information in a column matrix of column number n2. It receives the number of overs bowled before the interruptions in the matrix Aa, number of balls bowled in those overs in the matrix Bb, Wickets lost during the interruptions in the matrix Ww1, Scores during the interruptions in matrix Ss, Reduced total overs in the matrix Cc and balls to be bowled in those overs in matrix Dd. We take two more row matrices Pp and Qq. Pp takes the total balls bowled before the interruption as its elements and Qq takes the total balls to be bowled in that innings as its elements.

After we finish giving the information, the program will check out whether there is any error in the given information or not. There are many errors a user can make, like any number of overs given as input cant be more than 50, less than 0 or a fraction. Similarly the number of balls in an over cant be more than 6, less than 0 or a fraction. Any score cant be a fraction or negative. The number of wickets cant be more than 10, less than 0 or a fraction. Any over during an interruption cant be less than that during the previous interruptions. It is same for runs scored and wickets lost. And any reduced total number of overs after an interruption cant be more than the previous one.

For filtering these errors, we have fixed the values of some variables. Each variable works for a particular possible error. If there is an error, that particular variable changes its value to 1 more. Then after finishing giving the information, we use if conditionals to check whether the values of all the variables are unchanged or not. If there is a change for the value of any variable, the program displays that that particular error defined by that variable has been made that number of times by which the value of the variable has increased. And the program will stop there, will not proceed further forward.

But if the values of the variables remain unchanged, then the program understands that there is no error and it moves forward. At first it forms two row matrices Ee and Ff whose of number of columns are equal to the total number of interruptions occurred, n2. The elements of Ee are the total number of balls played and lost before each interruption and the elements of Ff are the total number of balls played and lost after each interruption. For example, if the first interruption occurs when it is 12.5 overs, and the total number of overs is reduced to 45, then the first element of Ee is 77 and the first element of Ff is 107. Then if another interruption occurs at 25.5 over and the total over is reduced further to 40, then the second element Ee is 185 and the second element of Ff is 215.

When interruptions occurr, the resources available to a team before each interruption is the element in the (Ee+1)th row and (Ww1+2)th column of the DL_Table. And the resources available to a team after each interruption is the element in the (Ee+1)th row and (Ww1+2)th column of the DL_Table. So the resources lost in each interruption is the element in the (Ee+1)th row and (Ww1+2)th column of the DL_Table minus the element in the (Ee+1)th row and (Ww1+2)th column of the DL_Table.

At last the program forms one more row matrix RR2 of row number n2. It takes as its elements as the resources lost in each interruption. So sum(RR2) denotes the total amount of resources lost in the second innings. So the program takes 100-sum(RR2) as the total amount of resources used in the second innings.

In this situation there is no interruption in the first innings and interruptions in the second innings. So 100% resources have been used in the first innings and less than 100% resources have been used in the second innings. So according to the Duckworth-Lewis first formula the D\L Par Score of the first team is set and target of the second team is set as one plus it. After the last interruption in the second innings this new target is declared with the new total number of overs. Our program also displays the amount of runs which is still required by subtracting the last element of Ss from the reset target score, along with the left number of balls, by subtracting the last element of the matrix Qq by the last element of the matrix Pp. Our program ends there.

Sometimes it may happen that there is no play possible after an interruption in the second innings. In that case the reduced number of total balls in the innings after the last interruption is equal to the number of balls bowled before the last interruption. That means the last element of matrix Pp is equal to the last element of matrix Qq. In such cases target score of the second team is not declared. Rather the match is declared completed and the result of the match is declared if there was at least 20 overs possible in the second innings.The D\L par score of the first team is calculated by the D\L formulae. If second team's score at that moment is more than that, they win by the margin called 'difference' in our program. If their score is less, they lose and if the scores are equal the match is declared tied. Our program ends here.

If the match ends before 20 overs in the innings, the program shows that the match is postpones and there is no result. The program ends then.

Up to now, our program has worked on that there is no interruption in the second innings and interruptions in the first innings. But when there is interruptions in the first innings, n1 is greater than zero, then our program will ask for details about those interruptions. Similar to those discussed earlier, about the interruptions in the second innings but there is just one simple difference that scores during the interruptions are not required asked for, as in the calculation of D\L Par score, scores during the first innings interruptions are not required. Then the possible errors will be checked out in the same way discussed earlier. If there is no error, total resources used in the first innings will be calculated using the DL_Table.

Next the program will ask for the interruptions in the second innings when there is interruptions in the second innings. If there is no interruption in the second innings, total resources used in the second innings are 100%. But if there is interruptions, details will be asked for, errors will be checked out and second innings total resources going to be used are calculated in the same way like the earlier. And if the match cant be restarted after an interruption, the overs bowled before that interruption is considered as the reduced total number of overs and the total resources used in the second innings are found out in the same way.

From this point, we have a division of our paths. The program then checks which one is greater, total resources used in the first innings or that used or going to be used in the second innings. If the first one is greater, the program the calculates the D\L Par Score using the first formula stated in the theory section. And if the second one is greater, it asks for another term called G50, which has been discussed earlier. Obviously the G50 is checked out before taken into account, whether it is valid or not, if invalid showed there with no further proceed. And if it is valid, the D\L par score is calculated by using the second formula stated in the theory.

If the match ends there, the winner is declared along with the margin comparing D\L Par Score of the first team and the score at that point of the second team. But obviously for that at least 20 overs are required to be played in the second innings and otherwise the program shows no result.

And if there is still play to take place, then the reset target of the second team is displayed which is one plus the D\L Par Score of the first team. Our program also shows the amount of runs which is still required along with the left number of balls in the same way discussed earlier.

To tell in brief, the program that we have made works in this way.

BuilT-in Functions:

In our program we have used two Build-In Functions.

- **Round:** It turns any input as its nearest integer. Example: round (1.6)=2
- **Sum:** It adds all the elements of a row matrix or a column matrix.

<u>Example:</u>

On February 20, 2003, during the 2003 world cup, at Potchefstroom, South Africa, between the match between Australia and Netherlands, while batting first Australia's innings was interrupted three times. The first one was before starting the innings, and total over was reduced to 47. Then at the end of 25 overs, at the score 109/2. Total over was reduced to 44 then. And lastly at the end of 28 overs, at the score 123/2 reducing the total overs to 36. There was no interruption in the Netherlands innings. The G50 at that time was 235.

Number of interruption in the first innings: 3

<u>Interruption 1:</u>

Over during which the match is interrupted: 0.0

Wickets fallen at that time: 0

Reduced total overs after the match restarts: 47.0

<u>Interruption 2:</u>

Over during which the match is interrupted: 25.0

Wickets fallen at that time: 2

Reduced total overs after the match restarts: 44.0

<u>Interruption 3:</u>

Over during which the match is interrupted: 28.0

Wickets fallen at that time: 2

Reduced total overs after the match restarts: 36.0

First innings score: 170

Number of interruptions in the second innings: 0

G50 of the ground: 235

Target for second team is 198 runs in 36.0 overs(D/L).

Netherland’s target was 198 runs in 36 overs. They were able to score only 122 and at the end, lost by 75 runs.