

Automatic Calculation of Stability

and Control Derivatives

By

Bradley D. Beachler

AEROSPACE ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2002

TABLE OF CONTENTS

TABLE OF CONTENTS.....	ii
LIST OF FIGURES	iii
LIST OF TABLES.....	iv
ABSTRACT.....	1
BACKGROUND	2
PROCEDURE.....	3
LIMITATIONS.....	6
CESSNA EXAMPLE	7
FUTURE EXPANSION	9
USER MANUAL.....	10
APPENDIX.....	12
REFERENCES	20

LIST OF FIGURES

Figure 1: Cal Poly Flight Simulator.....	3
Figure 2: Dihedral Contribution to Rolling Moment due to Sideslip.....	4
Figure 3: Cessna 150 Three View.....	7
Figure 4: <i>Roscam Part VI</i>	10
Figure 5: Flap Pull Down Menu.....	11
Figure 6: Macro Execution Button.....	11
Figure 7: Output Format.....	11

LIST OF TABLES

Table 1: Dihedral Contribution to Rolling Moment due to Sideslip	5
Table 2: Worksheet Derivative Capabilities	6
Table 3: Longitudinal C-150 Derivative Comparisons.....	8
Table 4: Lateral C-150 Derivative Comparisons	8
Table 5: Raw Data Table	9

ABSTRACT

Calculating stability and control derivatives is time consuming and can waste valuable time in any aircraft design process. Because each iteration requires a new set of stability and control derivatives (a SAD file), a method to quickly obtain a SAD file would prove very beneficial. Therefore a Microsoft Excel workbook was created which will automatically calculate the SAD file for conventional aircraft designs.

BACKGROUND

Stability and control is an integral part of any aircraft or aircraft design process. Traditional methods of calculating stability and control derivatives involves following through a manual of Datcom or Roscam, following layered steps and relying on empirical data. Unfortunately this is very time consuming and challenging. When calculating these derivatives for any ongoing design process, it can mean many hours or days recalculating numbers to arrive at a new stability and derivative file, also known as a SAD file. This senior project addresses this challenging problem and simplifies the initial design process, allowing quick and easy access to approximate SAD files for any conventional aircraft.

PROCEDURE

This senior project will allow users to quickly get estimates on the flying qualities of their designed aircraft in the Cal Poly flight simulator located in the Advanced Technologies Laboratory seen in Figure 1. By inputting the SAD file that this workbook creates, the user will immediately be able to observe any improvement or degradation of design choices in an almost real time environment. By placing the aircraft in the flight simulator it will also be possible to have trained pilots fly the design and give valuable pilot ratings. If the user has knowledge of Simulink they will also be able to design and demonstrate simple control laws.



Figure 1: Cal Poly Flight Simulator

This goal was achieved using Microsoft Excel as the engine to calculate the SAD file. *Roscam part VI: Preliminary Calculations of Aerodynamic, Thrust and Power Characteristics* was the basis for the workbook. The workbook uses equations and methods directly out of the Roscam text. Many of these stability derivatives that are calculated using methods in *Part VI* rely heavily on charts in which a value is retrieved for a given set of parameters. Doing this by eye can be challenging because some needed values may lie between two charts and an educated guess must be made to retrieve the value. An example of this difficult to estimate chart is shown in Figure 2.

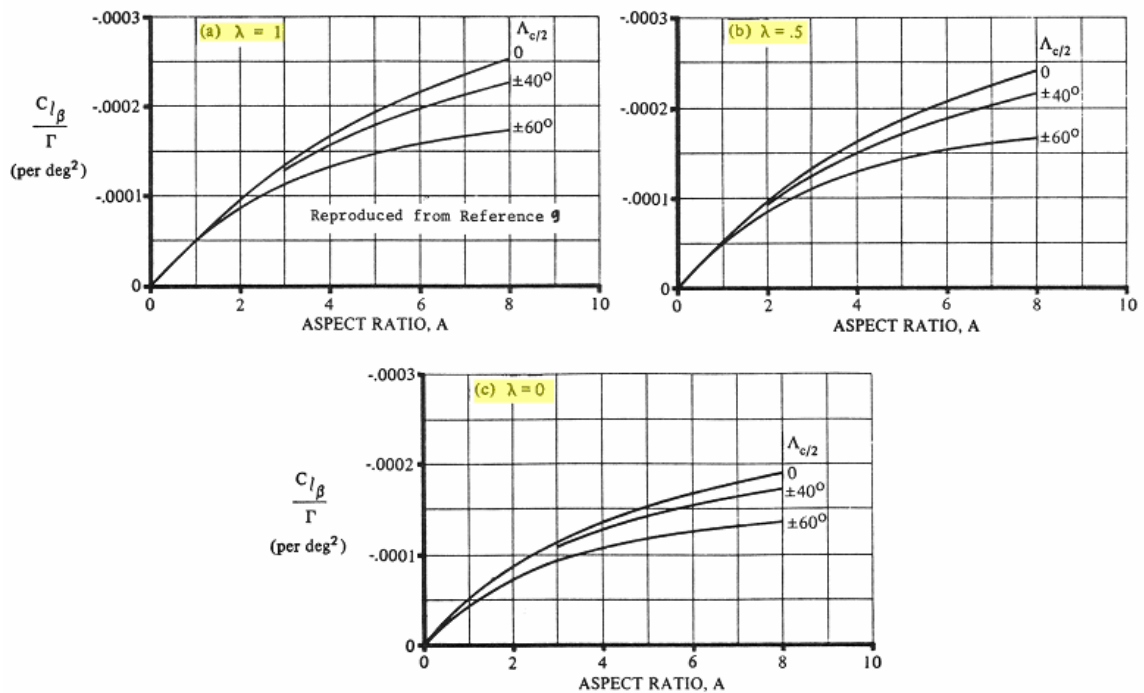


Figure 2: Dihedral Contribution to Rolling Moment due to Sideslip

This chart can be vary hard to get an accurate value of $\frac{C_{l\beta}}{\Gamma}$ (Dihedral contribution to rolling moment due to sideslip) by eye. If the taper ratio of the designed aircraft is any value other than 1, .5, or 0, an accurate value of $\frac{C_{l\beta}}{\Gamma}$ is all but impossible to obtain. If the values on the chart are input into a spreadsheet they can be numerically interpolated to arrive at a very accurate result. Table 1 displays the exact same data as Figure 2, for the three taper ratios all half cord sweep lines are plotted against aspect ratio.

Table 1: Dihedral Contribution to Rolling Moment due to Sideslip

λ	0.00			0.50			1.00		
$\Lambda_{c/2}$	0	40	60	0	40	60	0	40	60
AR	$(C_{l\beta})/\Gamma$								
0	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
1	-0.0000500	-0.0000500	-0.0000460	-0.0000500	-0.0000500	-0.0000500	-0.0000500	-0.0000500	-0.0000500
2	-0.0000850	-0.0000800	-0.0000750	-0.0001000	-0.0000970	-0.0000870	-0.0000980	-0.0000920	-0.0000850
3	-0.0001100	-0.0001050	-0.0000980	-0.0001300	-0.0001290	-0.0001100	-0.0001400	-0.0001300	-0.0001110
4	-0.0001350	-0.0001270	-0.0001100	-0.0001600	-0.0001550	-0.0001350	-0.0001700	-0.0001550	-0.0001300
5	-0.0001500	-0.0001400	-0.0001200	-0.0001900	-0.0001750	-0.0001450	-0.0001900	-0.0001750	-0.0001480
6	-0.0001630	-0.0001510	-0.0001250	-0.0002100	-0.0001900	-0.0001520	-0.0002100	-0.0001960	-0.0001540
7	-0.0001750	-0.0001670	-0.0001350	-0.0002250	-0.0002050	-0.0001600	-0.0002400	-0.0002150	-0.0001650
8	-0.0001900	-0.0001740	-0.0001400	-0.0002450	-0.0002200	-0.0001650	-0.0002550	-0.0002300	-0.0001750

Once all the necessary charts in *Part VI, Chapter 10* were inputted in the same format as Table 1, a method was needed to interpolate the actual value of $\frac{C_{l\beta}}{\Gamma}$. A code was written in Visual Basic which interpolates the value of $\frac{C_{l\beta}}{\Gamma}$ for any given configuration of taper ratio, half cord sweep, and aspect ratio. This VB code steps through all the tables created from *Part VI, Section 10*, outputting the required variables necessary to calculate stability and control derivatives.

LIMITATIONS

This worksheet was created for a conventional aircraft configuration. The user must limit the inputted aircraft to a standard single horizontal tail surface with no anhedral or dihedral. It was also assumed that the aircraft is subsonic with all wing and tail sweeps less than 60° . The worksheet was also built on the assumption that there is a single vertical tail with no cant angle. All other assumptions for this worksheet can be found in *Roscam Part VI: Preliminary Calculations of Aerodynamic, Thrust and Power Characteristics, Chapter 10*. The majority of them are limited by the range that the charts enclose. This Visual Basic code will display an error anytime that a needed input is outside the domain of the charts. The derivatives that the workbook is capable of calculating, along with their descriptions, are shown in Table 2.

Table 2: Worksheet Derivative Capabilities

<u>Longitudinal</u>		<u>Lateral</u>	
C_{Du}	Drag Due to Speed Derivative	$C_{M\dot{\delta}_e}$	Pitching Moment due to Elevator Derivative
C_{Lu}	Lift Due to Speed Derivative	$C_{l\dot{\beta}}$	Rolling Moment due to Sideslip Derivative
$C_{D\alpha}$	Drag due to Angle of Attack Derivative	$C_{n\dot{\beta}}$	Yawing Moment due Sideslip Derivative
$C_{L\alpha}$	Lift Due to Angle of Attack Derivative	$C_{y\dot{\beta}}$	Side Force due to Sideslip Derivative
$C_{M\alpha}$	Pitching Moment due to Angle of Attack Derivative (Static Longitudinal Stability)	$C_{l\dot{p}}$	Rolling Moment due to Roll Rate Derivative
$C_{L\alpha \dot{\alpha}}$	Lift due to Rate of Change of Angle of Attack Derivative	$C_{n\dot{p}}$	Yawing Moment due to Roll Rate Derivative
$C_{M\alpha \dot{\alpha}}$	Pitching Moment due to Rate of Change of Angle of Attack Derivative	$C_{y\dot{p}}$	Side Force due to Roll Rate Derivative
$C_{L\dot{q}}$	Lift due to Pitch Rate Derivative	$C_{l\dot{r}}$	Rolling Moment due to Yaw Rate Derivative
$C_{M\dot{q}}$	Pitching Moment due to Pitch Rate Derivative	$C_{n\dot{r}}$	Yawing Moment due to Yaw Rate Derivative
$C_{L\dot{\delta}_e}$	Lift due to Elevator Derivative	$C_{y\dot{r}}$	Side Force due to Yaw Rate Derivative
		$C_{l\dot{r}}$	Lift due to Rudder Derivative
		$C_{n\dot{r}}$	Yawing Moment due to Rudder Derivative
		$C_{y\dot{r}}$	Side Force due to Rudder Derivative

CESSNA EXAMPLE

To validate the worksheet a Cessna 150 SAD file was calculated using the worksheet and compared to Roscam's A1, Cessna 150 SAD file. For this validation a three view drawing was needed to measure and gather the geometric properties of the Cessna 150. This three view can be seen in Figure 3.

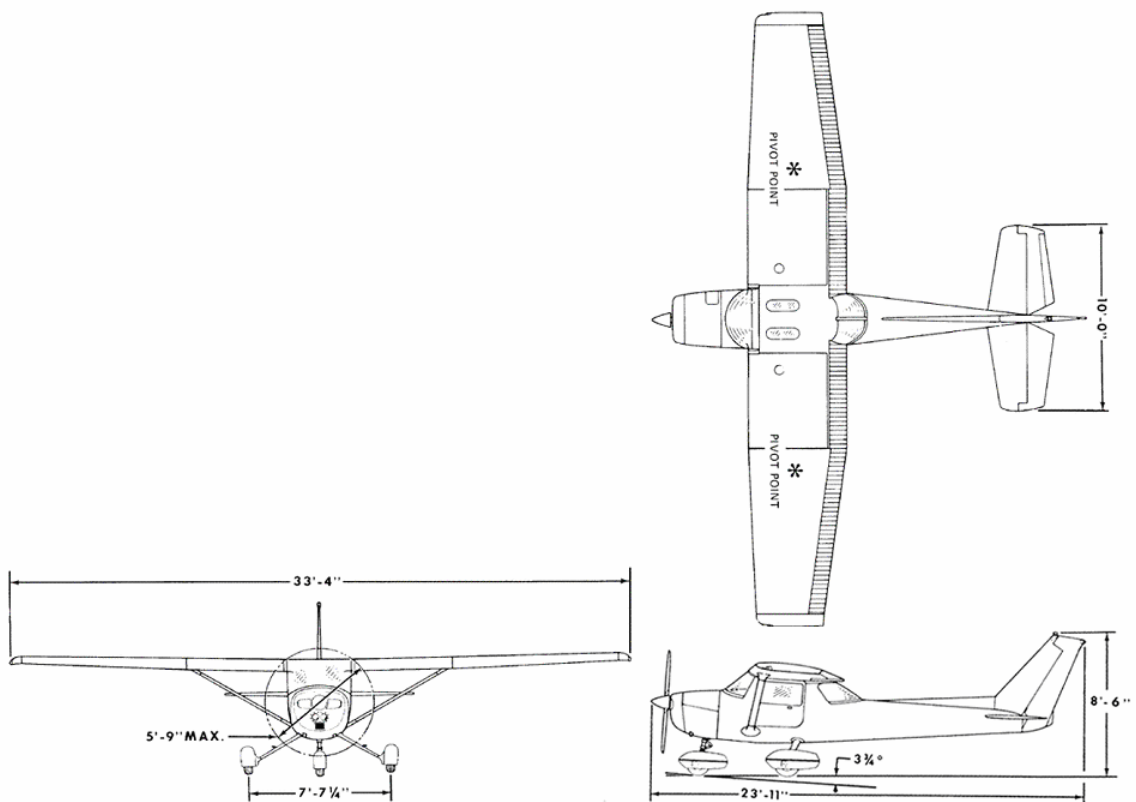


Figure 3: Cessna 150 Three View

After all geometric quantities needed were gathered from the three view they were inputted into the workbook and a SAD file was created for the C-150. In Table 3 and Table 4 the Roscam values are compared to the workbook calculated values.

Table 3: Longitudinal C-150 Derivative Comparisons

	C_{Du}	C_{Lu}	$C_{D\alpha}$	$C_{L\alpha}$	$C_{M\alpha}$	$C_{L\alpha \text{ dot}}$	$C_{M\alpha \text{ dot}}$	C_{Lq}	C_{Mq}	$C_{L\delta e}$	$C_{M\delta e}$
ROSCAM	0	0	0.130	4.600	-5.200	1.700	-0.890	3.900	-12.400	0.430	-1.280
Workbook	-0.186	0.008	0.107	3.764	-5.693	0.218	-1.856	3.602	-8.060	0.121	-0.509
Difference	NA	NA	0.023	0.836	0.493	1.482	0.966	0.298	4.340	0.309	0.771

Table 4: Lateral C-150 Derivative Comparisons

	$C_{l\beta}$	$C_{n\beta}$	$C_{y\beta}$	C_{lp}	C_{np}	C_{yp}	C_{lr}	C_{nr}	C_{yr}	$C_{l\delta r}$	$C_{n\delta r}$	$C_{y\delta r}$
ROSCAM	-0.089	0.065	-0.310	-0.470	-0.030	-0.037	0.096	-0.099	2.100E-01	1.470E-02	-6.570E-02	1.870E-01
Workbook	0.205	-12.660	-0.154	-6.167	0.000	0.000	0.085	-0.003	1.029E-04	1.550E-05	-1.836E-04	4.768E-04
Difference	0.294	12.725	0.156	5.697	0.030	0.037	0.011	0.096	0.210	-0.015	0.066	-0.187

All workbook values are acceptably accurate with the values in blue being at the outer limits of acceptability. These blue values are derivatives that are not extremely critical for a valuable flight simulation evaluation. The value in yellow, C_{Mq} , Pitching Moment due to Pitch Rate Derivative is off by a considerable amount. The most likely explanation for this anomaly is a difference in measured values off of the three view drawing. For this one value of C_{Mq} it is advisable to calculate the value by hand, or check it by another means.

USER MANUAL

This section is intended to provide a detailed description of how to utilize this workbook.

1. Obtain a copy of *Roscam Part VI: Preliminary Calculations of Aerodynamic, Thrust and Power Characteristics*, Figure 4



Figure 4: *Roscam Part VI*

2. Point an internet browser to http://aerosim.calpoly.edu/SAD_creator.htm
3. Download the file “SAD_creator.zip”
4. Extract the Zip file with WinZip or other extraction program
5. Open the Excel workbook titled “SAD_creator.xls”
6. Point to the sheet titled “Aircraft_Geometry”
7. Input the values required for Aircraft Geometry. Descriptions can be retrieved from *Roscam Part VI* for the appropriate figures.
8. Input the values in remaining fields under Other Quantities.

9. Point to the sheet titled “S&C”

10. Select flap type, plain or fowler from the pull down menu. Figure 5

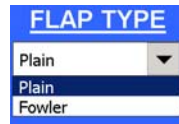


Figure 5: Flap Pull Down Menu

11. Point to the sheet titled “A”

12. Click the large button labeled “Get The SAD File” Figure 6

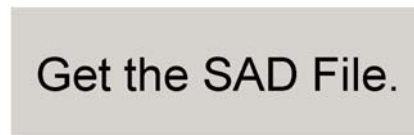


Figure 6: Macro Execution Button

13. Point to the sheet titled “S&C”

14. Retrieve the SAD File from the Boxes labeled Longitudinal and Lateral, Figure 7.

<u>Longitudinal</u>	<u>Lateral</u>
<u>Steady</u>	<u>Steady</u>
C_{Du}	$C_{l\beta}$
C_{Lu}	$C_{n\beta}$
$C_{D\alpha}$	$C_{y\beta}$
$C_{L\alpha}$	C_{lp}
$C_{M\alpha}$	C_{np}
$C_{L\alpha \text{ dot}}$	C_{yp}
$C_{M\alpha \text{ dot}}$	C_{lr}
C_{Dq}	C_{nr}
C_{Lq}	C_{yr}
C_{Mq}	<u>Controllable</u>
<u>Controllable</u>	$C_{l \delta\alpha}$
$C_{L \delta\alpha}$	$C_{n \delta\alpha}$
$C_{M \delta\alpha}$	$C_{y \delta\alpha}$
	$C_{l \delta r}$
	$C_{n \delta r}$
	$C_{y \delta r}$

Figure 7: Output Format

APPENDIX

Following is the code used to interpolate the coefficients from the tables in worksheet

“A”.

```
Sub DataReducer()  
  
Dim i, j, k, l, m, n, o, jmax As Integer  
Dim OuterCount As Integer  
Dim Countholder As Integer  
Dim SecondCount As Integer  
Dim ThirdCount As Integer  
Dim DownCount As Integer  
Dim Outer As Single  
Dim Second As Single  
Dim Third As Single  
Dim Down As Single  
Dim DownUp As Single  
Dim DownLow As Single  
Dim position(30, 30, 30) As Single  
Dim ValueDown(20) As Single  
Dim DownPositionA As Single  
Dim DownPositionB As Single  
Dim ValueThird(20) As Single  
Dim ThirdPositionA As Single  
Dim ThirdPositionB As Single  
Dim ValueSecond(20) As Single  
Dim SecondPositionA As Single  
Dim SecondPositionB As Single  
Dim ThirdAA As Single  
Dim ThirdAB As Single  
Dim ThirdBA As Single  
Dim ThirdBB As Single  
Dim ThirdA As Single  
Dim ThirdB As Single  
Dim FinalValue(50) As Single  
  
For i = 1 To 38  
  
    OuterCount = 0  
    SecondCount = 0  
    ThirdCount = 0  
    DownCount = 0  
  
    Outer = Cells((i * 30) - 14, 8)  
    Second = Cells((i * 30) - 13, 8)  
    Third = Cells((i * 30) - 12, 8)  
    Down = Cells((i * 30) - 11, 8)  
  
    For j = 1 To 72  
        If (Cells((i * 30) - 29, j + 1) <> "") Then  
            OuterCount = OuterCount + 1  
        End If  
    Next  
  
End Sub
```

```

    If (OuterCount > 0) Then
        jmax = 72 / OuterCount
    Else
        jmax = 24
    End If

For j = 1 To jmax
    If (Cells((i * 30) - 28, j) <> "") Then
        SecondCount = SecondCount + 1
    End If
Next

For j = 1 To jmax
    If (Cells((i * 30) - 27, j) <> "") Then
        ThirdCount = ThirdCount + 1
    End If
Next

ThirdCount = ThirdCount / SecondCount

For k = 1 To 10
    If (Cells(((k + 2) + ((i * 30) - 29)), 1) <> "") Then
        DownCount = DownCount + 1
    End If
Next

For l = 1 To SecondCount
    For m = 1 To DownCount
        For n = 1 To ThirdCount
            position(m, n, l) = Cells(((m + 2) + ((i * 30) - 29)), (((l * 8) - 7) + n))
        Next
    Next
Next

For m = 1 To DownCount
    ValueDown(m) = Cells(((i * 30) - 29) + (m + 2), 1)
    If (ValueDown(m) < Down) Then
        DownPositionB = m
        DownPositionA = m + 1
    End If
Next

For n = 1 To ThirdCount
    ValueThird(n) = Cells(((i * 30) - 27), n + 1)
    If (ValueThird(n) < Third) Then
        ThirdPositionB = n
        ThirdPositionA = n + 1
    End If
Next

For o = 1 To SecondCount
    ValueSecond(o) = Cells(((i * 30) - 27), ((o * 9) - 8))
    If (ValueSecond(o) < Second) Then
        SecondPositionA = o
        SecondPositionB = o - 1
    End If
Next

```

ThirdAA = (((Down - ValueDown(DownPositionA)) * (position(DownPositionB, ThirdPositionA, SecondPositionA) – position(DownPositionA, ThirdPositionA, SecondPositionA))) / (ValueDown(DownPositionB) - ValueDown(DownPositionA))) + position(DownPositionA, ThirdPositionA, SecondPositionA)

ThirdAB = (((Down - ValueDown(DownPositionA)) * (position(DownPositionB, ThirdPositionB, SecondPositionA) – position(DownPositionA, ThirdPositionB, SecondPositionA))) / (ValueDown(DownPositionB) - ValueDown(DownPositionA))) + position(DownPositionA, ThirdPositionB, SecondPositionA)

ThirdBA = (((Down - ValueDown(DownPositionA)) * (position(DownPositionB, ThirdPositionA, SecondPositionB) – position(DownPositionA, ThirdPositionA, SecondPositionB))) / (ValueDown(DownPositionB) - ValueDown(DownPositionA))) + position(DownPositionA, ThirdPositionA, SecondPositionB)

ThirdBB = (((Down - ValueDown(DownPositionA)) * (position(DownPositionB, ThirdPositionB, SecondPositionB) – position(DownPositionA, ThirdPositionB, SecondPositionB))) / (ValueDown(DownPositionB) - ValueDown(DownPositionA))) + position(DownPositionA, ThirdPositionB, SecondPositionB)

ThirdA = (((Third - ValueThird(ThirdPositionA)) * (ThirdAB - ThirdAA)) / (ValueThird(ThirdPositionB) – ValueThird(ThirdPositionA))) + ThirdAA

ThirdB = (((Third - ValueThird(ThirdPositionA)) * (ThirdBB - ThirdBA)) / (ValueThird(ThirdPositionB) – ValueThird(ThirdPositionA))) + ThirdBA

If ThirdA = ThirdB Then

 FinalValue(i) = ThirdA

Else

 FinalValue(i) = (((Second - ValueSecond(SecondPositionA)) * (ThirdB - ThirdA)) / (ValueSecond(SecondPositionB) – ValueSecond(SecondPositionA))) + ThirdA

End If

Cells((i * 30) - 9), 8) = FinalValue(i)

Next

Dim TaperRatio As Single
 Dim eta As Single
 Dim bark As Single
 Dim bcldel As Single
 Dim SweepBeta As Single
 Dim positionAB(72) As Single
 Dim etaup As Single
 Dim etalow As Single
 Dim BUpperHi As Single
 Dim BUpperLow As Single
 Dim AUpperLow As Single
 Dim ALowerLow As Single
 Dim BarkUpB As Single
 Dim BarkLowB As Single
 Dim BarkUpA As Single
 Dim BarkLowA As Single
 Dim Up As Single
 Dim Low As Single

For i = 1 To 72

positionAB(i) = Worksheets("S&C_Charts").Cells(401 + i, 20)
 TaperRatio = Worksheets("Aircraft_Geometry").Range("taperratio").Value
 eta = Worksheets("Aircraft_Geometry").Range("eta").Value
 SweepBeta = Worksheets("Aircraft_Geometry").Range("sweepbeta").Value
 bark = Worksheets("S&C").Range("bark").Value
 etaup = Worksheets("S&C_Charts").Cells(457, 24).Value
 etalow = Worksheets("S&C_Charts").Cells(458, 24).Value

Next

If (TaperRatio > 0.5) Then

If (bark > 4) Then

If (SweepBeta > 40) Then

$$BUpperHi = (((SweepBeta - 40) * (positionAB(36) - positionAB(35))) / (60 - 40)) + positionAB(35)$$

$$BLowerHi = (((SweepBeta - 40) * (positionAB(72) - positionAB(71))) / (60 - 40)) + positionAB(71)$$

$$BUpperLow = (((SweepBeta - 40) * (positionAB(32) - positionAB(31))) / (60 - 40)) + positionAB(31)$$

$$BLowerLow = (((SweepBeta - 40) * (positionAB(68) - positionAB(67))) / (60 - 40)) + positionAB(67)$$

$$AUpperHi = (((SweepBeta - 40) * (positionAB(24) - positionAB(23))) / (60 - 40)) + positionAB(23)$$

$$ALowerHi = (((SweepBeta - 40) * (positionAB(60) - positionAB(59))) / (60 - 40)) + positionAB(59)$$

$$AUpperLow = (((SweepBeta - 40) * (positionAB(20) - positionAB(19))) / (60 - 40)) + positionAB(19)$$

$$ALowerLow = (((SweepBeta - 40) * (positionAB(56) - positionAB(55))) / (60 - 40)) + positionAB(55)$$

ElseIf (SweepBeta > 0) Then

$$BUpperHi = (((SweepBeta - 0) * (positionAB(35) - positionAB(34))) / (40 - 0)) + positionAB(34)$$

$$BLowerHi = (((SweepBeta - 0) * (positionAB(71) - positionAB(70))) / (40 - 0)) + positionAB(70)$$

$$BUpperLow = (((SweepBeta - 0) * (positionAB(31) - positionAB(30))) / (40 - 0)) + positionAB(30)$$

$$BLowerLow = (((SweepBeta - 0) * (positionAB(67) - positionAB(66))) / (40 - 0)) + positionAB(66)$$

$$AUpperHi = (((SweepBeta - 0) * (positionAB(23) - positionAB(22))) / (40 - 0)) + positionAB(22)$$

ALowerHi = (((SweepBeta - 0) * (positionAB(59) - positionAB(58))) / (40 - 0)) + positionAB(58)
AUpperLow = (((SweepBeta - 0) * (positionAB(19) - positionAB(18))) / (40 - 0)) + positionAB(18)
ALowerLow = (((SweepBeta - 0) * (positionAB(55) - positionAB(54))) / (40 - 0)) + positionAB(54)

Else

BUpperHi = (((SweepBeta + 40) * (positionAB(34) - positionAB(33))) / (0 - 40)) + positionAB(33)
BLowerHi = (((SweepBeta + 40) * (positionAB(70) - positionAB(69))) / (0 - 40)) + positionAB(69)
BUpperLow = (((SweepBeta + 40) * (positionAB(30) - positionAB(29))) / (0 - 40)) + positionAB(29)
BLowerLow = (((SweepBeta + 40) * (positionAB(66) - positionAB(65))) / (0 - 40)) + positionAB(65)
AUpperHi = (((SweepBeta + 40) * (positionAB(22) - positionAB(21))) / (0 - 40)) + positionAB(21)
ALowerHi = (((SweepBeta + 40) * (positionAB(58) - positionAB(57))) / (0 - 40)) + positionAB(57)
AUpperLow = (((SweepBeta + 40) * (positionAB(18) - positionAB(17))) / (0 - 40)) + positionAB(17)
ALowerLow = (((SweepBeta + 40) * (positionAB(54) - positionAB(53))) / (0 - 40)) + positionAB(53)

End If

BarkUpB = (((bark - 4) * (BUpperHi - BUpperLow)) / (8 - 4)) + BUpperLow

BarkLowB = (((bark - 4) * (BLowerHi - BLowerLow)) / (8 - 4)) + BLowerLow

BarkUpA = (((bark - 4) * (AUpperHi - AUpperLow)) / (8 - 4)) + AUpperLow

BarkLowA = (((bark - 4) * (ALowerHi - ALowerLow)) / (8 - 4)) + ALowerLow

Else

If (SweepBeta > 40) Then

BUpperHi = (((SweepBeta - 40) * (positionAB(32) - positionAB(31))) / (60 - 40)) + positionAB(31)
BLowerHi = (((SweepBeta - 40) * (positionAB(68) - positionAB(67))) / (60 - 40)) + positionAB(67)
BUpperLow = (((SweepBeta - 40) * (positionAB(28) - positionAB(27))) / (60 - 40)) + positionAB(27)
BLowerLow = (((SweepBeta - 40) * (positionAB(64) - positionAB(63))) / (60 - 40)) + positionAB(63)
AUpperHi = (((SweepBeta - 40) * (positionAB(20) - positionAB(19))) / (60 - 40)) + positionAB(19)
ALowerHi = (((SweepBeta - 40) * (positionAB(56) - positionAB(55))) / (60 - 40)) + positionAB(55)
AUpperLow = (((SweepBeta - 40) * (positionAB(16) - positionAB(15))) / (60 - 40)) + positionAB(15)
ALowerLow = (((SweepBeta - 40) * (positionAB(52) - positionAB(51))) / (60 - 40)) + positionAB(51)

ElseIf (SweepBeta > 0) Then

BUpperHi = (((SweepBeta - 0) * (positionAB(31) - positionAB(30))) / (40 - 0)) + positionAB(30)
BLowerHi = (((SweepBeta - 0) * (positionAB(67) - positionAB(66))) / (40 - 0)) + positionAB(66)
BUpperLow = (((SweepBeta - 0) * (positionAB(27) - positionAB(26))) / (40 - 0)) + positionAB(26)
BLowerLow = (((SweepBeta - 0) * (positionAB(63) - positionAB(62))) / (40 - 0)) + positionAB(62)
AUpperHi = (((SweepBeta - 0) * (positionAB(19) - positionAB(18))) / (40 - 0)) + positionAB(18)

ALowerHi = (((SweepBeta - 0) * (positionAB(55) - positionAB(54))) / (40 - 0)) + positionAB(54)
AUpperLow = (((SweepBeta - 0) * (positionAB(15) - positionAB(14))) / (40 - 0)) + positionAB(14)
ALowerLow = (((SweepBeta - 0) * (positionAB(51) - positionAB(50))) / (40 - 0)) + positionAB(50)

Else

BUpperHi = (((SweepBeta + 40) * (positionAB(30) - positionAB(29))) / (0 - 40)) + positionAB(29)
BLowerHi = (((SweepBeta + 40) * (positionAB(66) - positionAB(65))) / (0 - 40)) + positionAB(65)
BUpperLow = (((SweepBeta + 40) * (positionAB(26) - positionAB(25))) / (0 - 40)) + positionAB(25)
BLowerLow = (((SweepBeta + 40) * (positionAB(62) - positionAB(61))) / (0 - 40)) + positionAB(61)
AUpperHi = (((SweepBeta + 40) * (positionAB(18) - positionAB(17))) / (0 - 40)) + positionAB(17)
ALowerHi = (((SweepBeta + 40) * (positionAB(54) - positionAB(53))) / (0 - 40)) + positionAB(53)
AUpperLow = (((SweepBeta + 40) * (positionAB(14) - positionAB(13))) / (0 - 40)) + positionAB(13)
ALowerLow = (((SweepBeta + 40) * (positionAB(50) - positionAB(49))) / (0 - 40)) + positionAB(49)

End If

BarkUpB = (((bark - 2) * (BUpperHi - BUpperLow)) / (4 - 2)) + BUpperLow
BarkLowB = (((bark - 2) * (BLowerHi - BLowerLow)) / (4 - 2)) + BLowerLow
BarkUpA = (((bark - 2) * (AUpperHi - AUpperLow)) / (4 - 2)) + AUpperLow
BarkLowA = (((bark - 2) * (ALowerHi - ALowerLow)) / (4 - 2)) + ALowerLow
End If
Up = (((TaperRatio - 0.5) * (BarkUpB - BarkUpA)) / (1 - 0.5)) + BarkUpA
Low = (((TaperRatio - 0.5) * (BarkLowB - BarkLowA)) / (1 - 0.5)) + BarkLowA

Else

If (bark > 4) Then

If (SweepBeta > 40) Then

AUpperHi = (((SweepBeta - 40) * (positionAB(12) - positionAB(11))) / (60 - 40)) + positionAB(11)
ALowerHi = (((SweepBeta - 40) * (positionAB(48) - positionAB(47))) / (60 - 40)) + positionAB(47)
AUpperLow = (((SweepBeta - 40) * (positionAB(8) - positionAB(7))) / (60 - 40)) + positionAB(7)
ALowerLow = (((SweepBeta - 40) * (positionAB(44) - positionAB(43))) / (60 - 40)) + positionAB(43)
BUpperHi = (((SweepBeta - 40) * (positionAB(24) - positionAB(23))) / (60 - 40)) + positionAB(23)
BLowerHi = (((SweepBeta - 40) * (positionAB(60) - positionAB(59))) / (60 - 40)) + positionAB(59)
BUpperLow = (((SweepBeta - 40) * (positionAB(20) - positionAB(19))) / (60 - 40)) + positionAB(19)
BLowerLow = (((SweepBeta - 40) * (positionAB(56) - positionAB(55))) / (60 - 40)) + positionAB(55)

ElseIf (SweepBeta > 0) Then

AUpperHi = (((SweepBeta - 0) * (positionAB(11) - positionAB(10))) / (40 - 0)) + positionAB(10)

ALowerHi = (((SweepBeta - 0) * (positionAB(47) - positionAB(46))) / (40 - 0)) + positionAB(46)

AUpperLow = (((SweepBeta - 0) * (positionAB(7) - positionAB(6))) / (40 - 0)) + positionAB(6)

ALowerLow = (((SweepBeta - 0) * (positionAB(43) - positionAB(42))) / (40 - 0)) + positionAB(42)

BUpperHi = (((SweepBeta - 0) * (positionAB(23) - positionAB(22))) / (40 - 0)) + positionAB(22)

BLowerHi = (((SweepBeta - 0) * (positionAB(59) - positionAB(58))) / (40 - 0)) + positionAB(58)

BUpperLow = (((SweepBeta - 0) * (positionAB(19) - positionAB(18))) / (40 - 0)) + positionAB(18)

BLowerLow = (((SweepBeta - 0) * (positionAB(55) - positionAB(54))) / (40 - 0)) + positionAB(54)

Else

AUpperHi = (((SweepBeta + 40) * (positionAB(10) - positionAB(9))) / (0 - 40)) + positionAB(9)

ALowerHi = (((SweepBeta + 40) * (positionAB(46) - positionAB(45))) / (0 - 40)) + positionAB(45)

AUpperLow = (((SweepBeta + 40) * (positionAB(6) - positionAB(5))) / (0 - 40)) + positionAB(5)

ALowerLow = (((SweepBeta + 40) * (positionAB(42) - positionAB(41))) / (0 - 40)) + positionAB(41)

BUpperHi = (((SweepBeta + 40) * (positionAB(22) - positionAB(21))) / (0 - 40)) + positionAB(21)

BLowerHi = (((SweepBeta + 40) * (positionAB(58) - positionAB(57))) / (0 - 40)) + positionAB(57)

BUpperLow = (((SweepBeta + 40) * (positionAB(18) - positionAB(17))) / (0 - 40)) + positionAB(17)

BLowerLow = (((SweepBeta + 40) * (positionAB(54) - positionAB(53))) / (0 - 40)) + positionAB(53)

End If

BarkUpB = (((bark - 4) * (BUpperHi - BUpperLow)) / (8 - 4)) + BUpperLow

BarkLowB = (((bark - 4) * (BLowerHi - BLowerLow)) / (8 - 4)) + BLowerLow

BarkUpA = (((bark - 4) * (AUpperHi - AUpperLow)) / (8 - 4)) + AUpperLow

BarkLowA = (((bark - 4) * (ALowerHi - ALowerLow)) / (8 - 4)) + ALowerLow

Else

If (SweepBeta > 40) Then

AUpperHi = (((SweepBeta - 40) * (positionAB(8) - positionAB(7))) / (60 - 40)) + positionAB(7)

ALowerHi = (((SweepBeta - 40) * (positionAB(44) - positionAB(43))) / (60 - 40)) + positionAB(43)

AUpperLow = (((SweepBeta - 40) * (positionAB(4) - positionAB(3))) / (60 - 40)) + positionAB(3)

ALowerLow = (((SweepBeta - 40) * (positionAB(40) - positionAB(39))) / (60 - 40)) + positionAB(39)

BUpperHi = (((SweepBeta - 40) * (positionAB(20) - positionAB(19))) / (60 - 40)) + positionAB(19)

BLowerHi = (((SweepBeta - 40) * (positionAB(56) - positionAB(55))) / (60 - 40)) + positionAB(55)

BUpperLow = (((SweepBeta - 40) * (positionAB(16) - positionAB(15))) / (60 - 40)) + positionAB(15)

BLowerLow = (((SweepBeta - 40) * (positionAB(52) - positionAB(51))) / (60 - 40)) + positionAB(51)

ElseIf (SweepBeta > 0) Then

AUpperHi = (((SweepBeta - 0) * (positionAB(7) - positionAB(6))) / (40 - 0)) + positionAB(6)

ALowerHi = (((SweepBeta - 0) * (positionAB(43) - positionAB(42))) / (40 - 0)) + positionAB(42)

AUpperLow = (((SweepBeta - 0) * (positionAB(3) - positionAB(2))) / (40 - 0)) + positionAB(2)

ALowerLow = (((SweepBeta - 0) * (positionAB(39) - positionAB(38))) / (40 - 0)) + positionAB(38)

BUpperHi = (((SweepBeta - 0) * (positionAB(19) - positionAB(18))) / (40 - 0)) + positionAB(18)

BLowerHi = (((SweepBeta - 0) * (positionAB(55) - positionAB(54))) / (40 - 0)) + positionAB(54)

BUpperLow = (((SweepBeta - 0) * (positionAB(15) - positionAB(14))) / (40 - 0)) + positionAB(14)

BLowerLow = (((SweepBeta - 0) * (positionAB(51) - positionAB(50))) / (40 - 0)) + positionAB(50)

Else

AUpperHi = (((SweepBeta + 40) * (positionAB(6) - positionAB(5))) / (0 - 40)) + positionAB(5)

ALowerHi = (((SweepBeta + 40) * (positionAB(42) - positionAB(41))) / (0 - 40)) + positionAB(41)

AUpperLow = (((SweepBeta + 40) * (positionAB(2) - positionAB(1))) / (0 - 40)) + positionAB(1)

ALowerLow = (((SweepBeta + 40) * (positionAB(38) - positionAB(37))) / (0 - 40)) + positionAB(37)

BUpperHi = (((SweepBeta + 40) * (positionAB(18) - positionAB(17))) / (0 - 40)) + positionAB(17)

BLowerHi = (((SweepBeta + 40) * (positionAB(54) - positionAB(53))) / (0 - 40)) + positionAB(53)

BUpperLow = (((SweepBeta + 40) * (positionAB(14) - positionAB(13))) / (0 - 40)) + positionAB(13)

BLowerLow = (((SweepBeta + 40) * (positionAB(50) - positionAB(49))) / (0 - 40)) + positionAB(49)

End If

BarkUpB = (((bark - 2) * (BUpperHi - BUpperLow)) / (4 - 2)) + BUpperLow

BarkLowB = (((bark - 2) * (BLowerHi - BLowerLow)) / (4 - 2)) + BLowerLow

BarkUpA = (((bark - 2) * (AUpperHi - AUpperLow)) / (4 - 2)) + AUpperLow

BarkLowA = (((bark - 2) * (ALowerHi - ALowerLow)) / (4 - 2)) + ALowerLow

End If

Up = (((TaperRatio - 0) * (BarkUpB - BarkUpA)) / (0.5 - 0)) + BarkUpA

Low = (((TaperRatio - 0) * (BarkLowB - BarkLowA)) / (0.5 - 0)) + BarkLowA

End If

FinalValueA = (((eta - etalow) * (Up - Low)) / (etaup - etalow)) + Low

Worksheets("S&C_Charts").Cells(461, 23) = FinalValue

End Sub

REFERENCES

1. Roscam, Dr. Jan, Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics, Ottawa, KA, Roscam Aviation and Engineering Corporation., 250-489, 1990.

DISCLAIMER

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the flight simulator, inaccurate outputs, or infringement of patent and copyright laws. California Polytechnic State University San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.