Blended Wing Body form factor code and Model Center integration

A Senior Project

presented to

the Faculty of the Aerospace Engineering Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Simo Alberti

June, 2010

© 2010 Simo Alberti

Blended Wing Body form factor code and Model Center integration

Simo Alberti

California Polytechnic State University, San Luis Obispo Sr. Project – Dr. R. McDonald

This senior project investigated a method of calculating the form factor of an arbitrarily shaped wing, such as a blended wing body, from its pressure coefficient distribution. A Matlab script was coded and integrated into an existing Model Center tool for blended wing bodies. Model Center allows a VSP model to be analyzed with PMARC, a panel code, to determine the pressure distribution over a wing. The Matlab code can then calculate the form factor and critical Mach number. It was found that the code can accurately model simple planar wings, validated against empirical equations, and is robust for blended wing bodies.

Objective

The objective of my senior project is to incorporate new functionality in the Aero Tools for Rapid BWB Analysis initially created by Dr. McDonald. I created several Matlab codes and functions which interface, in Model Center, with previous modules. Model Center is an environment which allows interfacing between different solid modeling and computational programs, such as VSP, PMARC, and Matlab, in one single project space.

Aero Tools project space was designed to perform aerodynamic calculations using PMARC on a wing geometry created with VSP. The raw data provided by PMARC is then processed using Matlab codes which calculate lift curves, drag polars, aerodynamic centers and pressure coefficient distributions over the wing and its wake. Working from the code which calculated pressure coefficient distributions, I created codes which can calculate the form factor and critical Mach number from aerodynamic conditions on the surface of the wing. The idea came from a paper by Takahashi, *Zero Lift Drag and Drag Divergence Prediction for Finite Wings in Aircraft Design*, which suggested that a mathematical solution to the form factor can be calculated from a know pressure distribution. This method would thus be more accurate than empirical equations for form factor, which loose fidelity when applied to unconventional wings, such as blended wing bodies. The equations in Takahashi's paper were traced back to Thwaites' *Incompressible Aerodynamics* text book, which allowed me to formulate a general equation for form factor independent of Reynolds's number and friction coefficient. This allowed for a more flexible Matlab code with less assumption.

Form Factor Equation

Common form factor equations found today in aircraft design books are all based on empirical data. These equations are calibrated and accurate in small ranges of wing geometry and often include less than two variables, thickness-to-chord, $\frac{t}{c}$, and sweep, Λ ,

$$FF = f(\frac{t}{c}, \Lambda, M)$$
(1)

These empirical equations are considerably different from one another but, if applied to the correct wing, can yield good results. The problem arises when trying to compute a form factor for a blended wing body which is far from the applicable ranges of the empirical form factor equations. Moreover, blended wing bodies have the added advantage of 3 dimensional flow relieving, a phenomenon that is not considered by any empirical form factor

equations. For these reasons it was necessary to find a method which can calculate the form factor of a blended wing body using flow data rather than empirical equations.

The paper by Takahashi formulated a combination of equations, mostly derived from Thwaites, which could calculate form factor from pressure coefficients, calculated using a 3D potential flow code, VORLAX. The equation of form factor used is,(Takahashi, German, Shajanian, Daskilewicz, & Donovan, 2010)

$$FF = \frac{C_{D_f}}{C_f} \tag{2}$$

Assuming fully turbulent flow of the wing, it is possible to use Thwaites first approximation equation for drag which uses on-body inviscid velocity field. The relative velocities, $\frac{U}{U_{\infty}}$, can be calculated from the pressure coefficient using the following relation,

$$\frac{U}{U_{\infty}} = \sqrt{1 - C_p} \tag{3}$$

Thwaites' equation for drag coefficient is,

$$c_{D_f} = \sum \left[\frac{0.02429}{Re^{\frac{1}{5}}} \int_0^1 \left(\frac{U}{U_{\infty}} \right)^4 d\left(\frac{x}{c} \right) \right]^{\frac{3}{6}}$$
(4)

E

Where the summation refers to the upper and lower surfaces of the airfoil.(Thwaites, 1960)

Skin friction coefficient equations are empirical, curves that fit experimental data points. Many skin friction equations exist, all with considerably different trends applicable to different flow conditions and airfoil geometries, Thwaites outlined seven of the most common in his book, shown in Figure 1(Thwaites, 1960).



Figure 1: The local skin friction coefficient for a turbulent boundary layer on a flat plate

The Prandtl-Schlichting equation, Equation 5, was arbitrarily chosen in Takahashi's paper as the one that best fit their conditions and was applied at a reference Reynolds Number of 10⁷.(Takahashi, German, Shajanian, Daskilewicz, & Donovan, 2010)

$$C_f = \frac{0.455}{\log_{10} Re^{2.58}} \tag{5}$$

Combing Equations 4 and 5 into Equation 2 gives the final form factor equation used in, Zero Lift Drag and Drag Divergence Prediction for Finite Wings in Aircraft Design (Takahashi, German, Shajanian, Daskilewicz, & Donovan, 2010),

$$FF = 0.104 \frac{\log_{10} Re^{2.58}}{Re^{\frac{1}{6}}} \sum \left[\int_0^1 \left(\frac{U}{U_{\infty}} \right)^4 d\left(\frac{x}{c} \right) \right]^{\frac{5}{6}}$$
(6)

It is important to notice that there are two sets of Reynolds Numbers in the final form factor equation. This requires a velocity condition to be set in order to calculate the form factor. This means that the form factor calculated from Equation 6 is not a general solution because it is associated with a Reynolds Number and thus only valid for one flow condition.

Thwaites' Derivation of the Form Factor Equation

The final form of the form factor equation derived in Zero Lift Drag and Drag Divergence Prediction for Finite Wings in Aircraft Design was not satisfactory. A proper equation should be able to calculate the form factor independent of Reynolds Number, making its result applicable to any flow condition without any assumptions. It can be seen that both the c_{d_f} and C_f equation include Reynolds Number, but the Prandtl-Schlichting equation does not allow for the Re to cancel. Choosing a different equation for the skin friction confection would allow for Re to cancel from the two equations.

To find the correct skin friction coefficient equation it was necessary to investigate the origins of Equation 4. This equation for drag coefficient must have started from an assumption for skin friction; matching the C_f equation with this assumption would make the Re cancel and make the equations consistent.

Equation 4 is the full drag coefficient equation in Incompressible Aerodynamics shown below,

$$C_{D} = 2\sum_{t} \left[\left(\frac{\delta_{2t}}{c} \right)^{\frac{6}{5}} \left(\frac{U_{t}}{U_{\infty}} \right)^{\frac{21}{5}} Re^{-\frac{3}{5}} + 0.0106 Re^{-\frac{1}{5}} \int_{\frac{x_{t}}{c}}^{1} \left(\frac{U}{U_{\infty}} \right)^{4} d\left(\frac{x}{c} \right) \right]^{\frac{3}{6}}$$
(7)

Where the first term is for laminar flow and the second term is for turbulent flow, the second term is used in Equation 4. The term inside the square brackets follows from the equation for predicting momentum thickness, Equation 8, derived several chapters earlier in *Incompressible Aerodynamics*.

$$\left(\frac{\delta_2}{c}\right)^{1+n} \left(\frac{U}{U_0}\right)^{\beta+n} = \alpha R e^{-n} \int_{\frac{x_0}{c}}^{\frac{x}{c}} \left(\frac{U}{U_0}\right)^{\beta} d\left(\frac{x}{c}\right) + K$$
(8)

The constant n, α and β are given by extrapolation from experimental results, seven different sets are given in Thwaits' book. The values proposed by Spence were used in Equation 7 and are shown in Table 1 (Thwaites, 1960).

Table 1: Constants in the momentum equation, Equation 8

Author	n	α	β
Spence (1956)	0.2	0.0106	4.0

The first assumption made to arrive to Equation 8 was proposed by Ludwieg and Tillmann and is,

$$c_f = 2\left(Re_{\delta_2}\right)^{-n} G(H) \tag{9}$$

With this assumption the momentum equation can be rewritten as Equation 10, where $\Theta = \delta (Re_{\delta_2})^n$ and $\Gamma = -(\Theta/U)(dU/dx)$.

$$\frac{d\Theta}{dx} = (1+n)G(H) + \Gamma\{(1+n)H + 2 + n\}$$
(10)

Equation 10 was found to be a good linear approximation as a function of only Γ , by Thwaites. With this second assumption, Equation 10 was simplified to Equation 11, which can then be integrated to obtain Equation 8.

$$\frac{d}{dx} = \alpha + \beta \Gamma \tag{11}$$

Using the values from Table 1 and Equations 10 and 11, it is possible to find the value of G(H) found in Equation 9 for this particular application (Thwaites, 1960).

$$G(H) = 0.00883$$
 (12)

It was found that Equation 4 originated from an assumption of the skin friction coefficient, Equation 9, as previously theorized. It is now possible to find a skin friction coefficient equation which matches the one used to derive Equation 4 to implement in Equation 2 in order to find a more accurate and consistent equation for form factor.

By looking at the different skin friction coefficient equations proposed by Thwaits, it is possible to notice that Equation 9 is remarkably similar to the power law equation,

$$c_f = 2C \left(Re_{\delta_2} \right)^{-n} \tag{13}$$

Where C is the equivalent of G(H), in Equation 9, whose value is shown in Equation 12. Three different authors give different values for *n* and *C*, based on different experimental data. Young offers a set of values with *n* matching the value proposed by Spence, shown in Table 1. Table 2 shows the permanents in Young's power law equation for the skin friction coefficient. Comparing Tables 1 and 2 shows that Equations 9 and 13 are identical. It can thus be concluded that Equation 4 is derived from Young's power law skin friction coefficient equation. Using the Young power law equation for C_f , instead of the Prandtl-Schlichting equation (Equation 5) to find the form factor will result in consistent equation for form factor with canceling terms, unlike Equation 6 (Thwaites, 1960).

Fable 2: Constants	; in skin	friction	coefficient	, Equation 14
---------------------------	-----------	----------	-------------	---------------

Author	п	С	$2[(1+n)C]^{1/(1+n)}$
Young (1953)	0.2	0.00883	0.0450

By integrating the momentum equation, using Equation 13, it is possible to find the 3 dimensional skin friction coefficient in terms of Re_x , shown in Equation 14. Inputting Young's parameters from Table 2 will yield the equation which will substitute Equation 5,

$$C_f = 2[(1+n)C]^{\frac{1}{1+n}} Re_x^{\frac{n}{1+n}} = 0.0450 Re_x^{\frac{1}{6}}$$
(14)

The form factor equation can now be reformulated from Equation 2 by combining Equations 4 and 14. As can be seen in Equation 15, all the terms outside the summation cancel out.

$$FF = \frac{0.0450 Re^{-\frac{1}{6}} \sum \left[\int_{0}^{1} \left(\frac{U}{U_{\infty}} \right)^{4} d\left(\frac{x}{c} \right) \right]^{\frac{3}{6}}}{0.0450 Re^{-\frac{1}{6}}}$$
(15)
$$FF_{2D} = \sum \left[\int_{0}^{1} \left(\frac{U}{U_{\infty}} \right)^{4} d\left(\frac{x}{c} \right) \right]^{\frac{5}{6}}$$
(16)

This results in an elegant equation for form factor in 2 dimensions independent of the Reynolds Number, shown in Equation 16. Most importantly, the cancelling of the coefficients in front of the integral means that Equation 16 is now free of the initial assumptions of skin friction. Thus, the values obtained from Equation 16 for form factor are independent of the power law skin friction equation, from which the equation originated, and applicable with any skin friction coefficient equations.

Equation 16 defined the form factor in two dimensions. To find the form factor equation in three dimensions, it is necessary to integrate over the span of the wing, as shown in Equation 17.

$$FF = \frac{1}{S_{ref}} \int_0^b c(y) \sum \left[\int_0^1 \left(\frac{U}{U_{\infty}} \right)^4 d\left(\frac{x}{c} \right) \right]^{\frac{5}{6}} dy$$
(17)

Where c(y) is the chord at the respective y location. The complete equation must include a S_{ref}/S_{wet} term, which comes from the original drag coefficient equation in three dimensions,

$$C_D = FF \frac{S_{wet}}{S_{ref}}$$
(18)

Combining Equation 17 and 18 gives the final form factor equation,

$$FF = \frac{1}{S_{wet}} \int_0^b c(y) \sum \left[\int_0^1 \left(\frac{U}{U_{\infty}} \right)^4 d\left(\frac{x}{c} \right) \right]^{\frac{5}{6}} dy$$
(19)

Critical Mach Number Equation

The critical Mach number, M_{cr} , of a wing can be calculated by knowing the critical pressure coefficient on the wing surface. Equation 20, gives the critical pressure coefficient in terms of the critical Mach number and can be used to calculate M_{cr} .

$$C_{p,cr} = \frac{2}{\gamma M_{cr}^2} \left[\left(\frac{1 + \frac{\gamma - 1}{2} M_{cr}^2}{1 + \frac{\gamma - 1}{2}} \right)^{\frac{\gamma}{\gamma - 1}} \right]$$

(20)

Equation 20 must be combined with a compressibility correction to give the most accurate result. The Karman-Tsien rule, shown in Equation 21, was chosen to for this application.

$$C_{p} = \frac{C_{p,0}}{\sqrt{1 - M_{\infty}^{2}} + \left[\frac{M_{\infty}^{2} \left(1 + \frac{\gamma - 1}{2} M_{\infty}^{2}\right)}{2\sqrt{1 - M_{\infty}^{2}}}\right] C_{p,0}}$$
(21)

Where $C_{p,0}$ is the incompressible critical pressure coefficient. Equations 20 and 21 must be combined to find the critical Mach number, Figure 2 shows how to find the critical Mach number with Equations 20 and 21 graphically (Anderson, 2007).



Figure 2: Graphical estimation of critical Mach number

The critical Mach number can be found by setting Equations 20 and 21 equal to each other, resulting in Equation 22. M_{cr} and M_{∞} are equal, and can be solved for with Matlab once the incompressible critical pressure coefficient, $C_{p,0}$, is known.

$$\frac{2}{\gamma M_{cr}^2} \left[\left(\frac{1 + \frac{\gamma - 1}{2} M_{cr}^2}{1 + \frac{\gamma - 1}{2}} \right)^{\frac{\gamma}{\gamma - 1}} \right] = C_{p,cr} = C_p = \frac{C_{p,0}}{\sqrt{1 - M_{\infty}^2} + \left[\frac{M_{\infty}^2 \left(1 + \frac{\gamma - 1}{2} M_{\infty}^2 \right)}{2\sqrt{1 - M_{\infty}^2}} \right]} C_{p,0}$$
(22)

Code Explanation

The form factor Matlab code, $FF_03_x.m$, was written to integrate into Model Center taking inputs from *pmFltCond*, a component which runs the PMARC panel code. The inputs/outputs in the form factor code were,

- % variable: FF double[] output % variable: pnames string[] output
- % variable: fileDATA22 string input

Where *fileDATA22* is the PMARC output, *FF* is the form factor matrix, and *pnames* is a matrix with the names of each patch. The input, *fileDATA22*, must be reorganized by the *parse22.m* function, extrapolated from one of Dr. McDonald's codes. This function return the coordinates on the wing surfaces as cell arrays, *x*, *y*, *z*, where each cell corresponds to a different patch, or wing component (i.e. left wing, right wing, etc.), the pressure coefficients distribution as a cell array, *c*, and a matrix with the names of each patch. The cell arrays, *x*, *y*, *z*, and *c*, were then rearranged such that the top and bottom surfaces of each patch were in separate matrices with the leading edge in the first row, trailing edge in the last row, root chord in the first column, and the wing tip in the last column. After the original cell array name there was added a *t* for top surface and a *b* for bottom surface, each cell still represents the different patches. Figure 3 shows how the matrices were arranged.



Figure 3: Matrix structure

The next portion of the code is dedicated to calculating the wetted area of each patch. This was done using Heron's formula shown in Equation 17.

$$Aera = \sqrt{s(s-a)(s-b)(s-c)}$$
(17)

Where a, b, and c are the length of each side of the triangle and s is the semi perimeter defined in Equation 18.

$$s = \frac{a+b+c}{2} \tag{18}$$

To implement this equation all the points on the wing surface, defined in x, y, and z, were connected into triangles. Figure 4 shows how the surface of the wing was divided into triangles, with the corresponding indices. The colored triangles represent the two triangles calculated from the point (i, j) = (2, 1), shown by the red dot. The area of each triangle was then summed to find the total surface area of the patch. The wetted area of the wing is necessary to properly calculate the form factor.

The last section in the code calculates the form factor for each patch. Equation 3 is used to calculate the velocity distribution, after which, Equation 16 is applied to calculate the 2 dimensional form factor. The trapz() function in Matlab was used to perform the integration over the upper and lower surface of the airfoils at each of the span-wise locations defined by the y matrix. Up to this point, the upper and lower surfaces of the patch were considered independently and all values kept in separate matrices. To calculate the 2 dimensional form factor, the upper and lower surface values were summed, as intended by the summation in Equation 16. Once the 2

dimensional form factors we known along the wing, they were integrated, using the trapz() function, and divided by S_{wet} to obtain the 3 dimensional form factor as given by Equation 19.



Figure 4: Triangles used to calculate wetted area of wing

The critical Mach number was calculated in a separate code Matlab code, *Mcrit_x.m*, using the same input as the form factor code explained above. The input/output header for this file is as follows,

% variable: Mcr[] double output % variable: fileDATA22 string input

The two codes integrate into model center identically from *pmFltCond* which supplies the fileDATA22 data from PMARC.

Matlab can easily find the minimum pressure coefficients from the C_p matrix, c, returned from PMARC for each patch. Equation 22 is used in the *fzero()* Matlab function to find the critical Mach number. M_{cr} is calculated independently for each patch to get better insight.

The final result, with all the code integrated into Model Center is shown in Figure 5.



Figure 5: Model Center project

Results

It was important to compare the form factor values obtained, from the method described above, in Model Center with empirical methods on simple wings. Three empirical equations were found for conventional wings which did not require setting a Reynolds number, all of which had thickness-to-chord as the only variable. These methods were described by Hoerner, Torrenbeek, and Roskam and applicable to simple, planar wings. To create the most comparable results, two wings were tested with Model Center. The first was a high aspect ratio (AR = 100) non swept wing, used to simulate an infinite wing. The second was a 30 degree swept wing with an aspect ratio of 10, used to simulate realistic wing. The thickness was varied between 12% and 26%, the results are shown in Figure 6.





It is possible to see that both the infinite wing and finite swept wing calculations, using Model Center, fit within the empirical equation results when the thickness-to-chord is between 12% to 20%. At lower and higher thicknesses, the trends start to diverge. This is expected because the empirical equations are reliable only within a small range. Because the Model Center code calculates the form factor from the pressure coefficient distribution, it is much more accurate and can take into consideration all parameters of the wing, such as sweep and span.

Having proven that the Model Center code can calculate accurate form factors for simple wings, it can now be applied to the more complex wing geometries of blended wing bodies. The form factor was calculated for the two BWBs showing in Figure 7and Figure 8. The first model has a maximum thickness-to-chord of 26% and the wings are swept by 30 degrees. The form factor was calculated to be 1.45. This value is considerably lower than that calculated for a simple wing of 26% thickness, seen in Figure 6. The empirical equations could never have been able to calculate the form factor for such a complex body accurately.



Figure 7: BWB, 26% thick body

The second model has a very thick center body, with a thickness-to-chord of 30%. Because the chord changes greatly from the center of the body to the outer section, the thickness drops drastically from over 22 feet to just 6 feet in the center section of the body. This shape allows for significant 3 dimensional affects to occur, the form factor for this type of blended wing body should be very low. The form factor was, in fact, calculated to be 1.38, comparable to that of a 12% thick planar wing calculated using Torrenbeek's method.



Figure 8: BWB, 30% thick body

The Matlab code for form factor and critical Mach number give good results applicable to blended wing bodies. Because they are derived from the aerodynamic properties calculated from PMARC, the results are very accurate and free of many assumptions. The code is based on form factor and critical Mach number equations derived from fundamental equations, hence making the code very robust and reliable.

Bibliography

Abbott, I. H., & von Doenhoff, A. E. (1959). Theory of Wing Sections. New York: Dover Publication, Inc

Cebeci, T., & Smith, A. M. (1968). Calculation of Profile Drag of Airfoils at Low Mach Numbers. Long Beach, CA: AIAA.

Karman, T. v. (1941). Compressibility Effects in Aerodynamics. Journal of the Aeronautical Sciences, 337-356.

McDonald, R. (2009). Aero Tools for rapid BWB Analysis.

Takahashi, T., German, B., Shajanian, A., Daskilewicz, M., & Donovan, S. (2010). Zero Lift Drag and Drag Divergence Prediction for Finite Wings in Aircraft Design. *48th AIAA Aerospace Sciences Meeting*. Orlando, FL: AIAA.

Thwaites, B. (1960). Incompressible Aerodunamics. Oxford: Oxford University Press.

Zucrow, M. J., & Hoffman, J. D. (1976). Gas Dybamics, Volume 1. John Wiley & Sons.

Appendix

```
FF_03_x.m, form factor code:
00
% FormFactor calculates the form factor of a wing/body from the Cp
% distribution.
0
% variable: FF double[] output
% variable: pnames string[] output
% variable: fileDATA22 string input
% Simo Alberti
% Senior Project - R. McDonald
% Cal Poly, San Luis Obispo
% v. 3.0 - 04/05/2010
% - Re was eliminated using a different Cf equation from Young.
% - Swet calculated from triangle theorem and x,y,z geometry
% - indecies changed to calculate patches seperatly with unlimited patches
% - Removied Mcr calculations, moved to separate M-file
% Patch Assumptions
8
   Each patch is a wing. First row is the trailing edge, middle row is the
2
    leading eadge, and last row is trailing edge. The first half of rows is
2
    of the bottom half of the wing, the second half of the rows is of the
   top half of the wing. The first column is at the root of the wing, the
8
8
    last column is at the wing tip. This mus the true for x, y, z, and c
8
   and all must be of equal size.
%% Load, set, measure
[x, y, z, c, pnames] = parse22(fileDATA22);
gamma = 1.4;
npatch = length(x);
[row, col] = size(x{1});
row2 = row/2 + .5;
%% Matrix Reorganization
% t = top of wing; b = bottom of wing; k = patch number
for k = 1:npatch
    ° x
    xb{k} = flipdim(x{k}(1:row2,:),1);
    xt\{k\} = x\{k\} (row2:row,:);
    ° v 8
    yb{k} = flipdim(y{k}(1:row2,:),1);
    yt{k} = y{k} (row2:row,:);
```

8 7

zb{k} = flipdim(z{k}(1:row2,:),1);

```
zt{k} = z{k} (row2:row,:);
    % Cp
    cpb{k} = flipdim(c{k}(1:row2,:),1);
    cpt{k} = c{k} (row2:row,:);
end
%% Swet
% To find wetted area of eatch patch Heron's formula is used, which
% calculates the area from the length of each side, a, b, and c.
% Top and Bottop flip from patch 1 and 2
for k = 1:npatch
for j = 1:length( xb{1}(1,:))-1
    for i = 1:length( xb{1}(:,1) )
        % Triangle 1: (i, j) (i, j+1) (i-1, j)
        % Triangle 2: (i, j) (i, j+1) (i+1, j+1)
        %Triangle 1
        % must be skipped in first row
        if i ~= 1
        ab 1 = norm( [xb{k}(i,j), yb{k}(i,j), zb{k}(i,j)] - [xb{k}(i,j+1),
yb{k}(i,j+1), zb{k}(i,j+1)]);
        at_1 = norm( [xt{k}(i,j), yt{k}(i,j), zt{k}(i,j)] - [xt{k}(i,j+1),
yt{k}(i,j+1), zt{k}(i,j+1)]);
        bb 1 = norm( [xb{k}(i,j), yb{k}(i,j), zb{k}(i,j)] - [xb{k}(i-1,j),
yb{k}(i-1,j), zb{k}(i-1,j)]);
        bt 1 = norm( [xt\{k\}(i,j), yt\{k\}(i,j), zt\{k\}(i,j)] - [xt\{k\}(i-1,j),
yt{k}(i-1, \overline{j}), zt{k}(i-1, \overline{j})]);
        cb 1 = norm( [xb{k}(i-1,j), yb{k}(i-1,j), zb{k}(i-1,j)] -
[xb{k}(i,j+1), yb{k}(i,j+1), zb{k}(i,j+1)]);
        ct 1 = norm( [xt{k}(i-1,j), yt{k}(i-1,j), zt{k}(i-1,j)] -
[xt{k}(i,j+1), yt{k}(i,j+1), zt{k}(i,j+1)]);
        else
            ab 1 = 0;
            at_1 = 0;
            bb 1 = 0;
            bt 1 = 0;
            cb \ 1 = 0;
            ct 1 = 0;
        end
        %Triangle 2
        % must be skippeded in last row
        if i ~= length( xb{1}(:,1) )
        ab 2 = norm( [xb{k}(i,j), yb{k}(i,j), zb{k}(i,j)] - [xb{k}(i,j+1),
yb\{k\}(i,j+1), zb\{k\}(i,j+1)]);
```

```
at 2 = norm( [xt{k}(i,j), yt{k}(i,j), zt{k}(i,j)] - [xt{k}(i,j+1),
yt{k}(i,j+1), zt{k}(i,j+1)]);
        bb 2 = norm( [xb{k}(i,j+1), yb{k}(i,j+1), zb{k}(i,j+1)] -
[xb{k}(i+1,j+1), yb{k}(i+1,j+1), zb{k}(i+1,j+1)]);
        bt 2 = norm( [xt{k}(i,j+1), yt{k}(i,j+1), zt{k}(i,j+1)] -
[xt{k}(i+1,j+1), yt{k}(i+1,j+1), zt{k}(i+1,j+1)]);
        cb 2 = norm( [xb{k}(i,j), yb{k}(i,j), zb{k}(i,j)] - [xb{k}(i+1,j+1),
yb{k}(i+1,j+1), zb{k}(i+1,j+1)]);
        ct 2 = norm( [xt{k}(i,j), yt{k}(i,j), zt{k}(i,j)] - [xt{k}(i+1,j+1),
yt{k}(i+1,j+1), zt{k}(i+1,j+1)]);
        else
            ab 2 = 0;
            at^{2} = 0;
            bb^{-}2 = 0;
            bt_2 = 0;
            cb^{2} = 0;
            ct^{2} = 0;
        end
        %Triangle 1
        sb 1 = (ab 1+bb 1+cb 1)/2;
        st 1 = (at 1+bt 1+ct 1)/2;
        %Triangle 2
        sb 2 = (ab 2+bb 2+cb 2)/2;
        st 2 = (at 2+bt 2+ct 2)/2;
        Ab{k}(i,j) = sqrt( sb_1*(sb_1-ab_1)*(sb_1-bb_1)*(sb_1-cb_1)) + sqrt(
sb 2*(sb 2-ab 2)*(sb 2-bb 2)*(sb 2-cb 2));
        At{k}(i,j) = sqrt( st 1*(st 1-at 1)*(st 1-bt 1)*(st 1-ct 1)) + sqrt(
st 2*(st 2-at 2)*(st 2-bt 2)*(st 2-ct 2));
    end
end
Swb\{k\} = sum(sum(Ab\{k\}));
Swt\{k\} = sum(sum(At\{k\}));
Swet{k} = Swb{k} + Swt{k};
end
%% Form Factor
for k = 1:npatch
    % eqn 16 from AIAA 2010-846
    UUb\{k\} = sqrt(1-cpb\{k\});
    UUt\{k\} = sqrt(1 - cpt\{k\});
% setting x in terms of chrod, x/c
    for j = 1:col
```

```
chord(j) = xb{k}(row2,j) - xb{k}(1,j);
xcb{k}(:,j) = (xb{k}(:,j) - xb{k}(1,j)) ./ (chord(j));
xct{k}(:,j) = (xt{k}(:,j) - xt{k}(1,j)) ./ (chord(j));
end
% eqn 17 from AIAA 2010-846
for j = 1:col
FF2d_b{k}(j) = ( (trapz(xcb{k}(:,j),UUb{k}(:,j).^4)) ) ^(5/6);
FF2d_t{k}(j) = ( (trapz(xct{k}(:,j),UUt{k}(:,j).^4)) ) ^(5/6);
FF2d{k}(j) = FF2d_b{k}(j) + FF2d_t{k}(j);
end
%abs because top and bottom flip from patch 1 and 2
FF(k) = abs(trapz(yt{k}(1,:),FF2d{k}.*chord) / Swet{k});
```

```
end
```

% eqn 18 from AIAA 2010-846
% FF = Cdf/Cf;

Mcrit_x.m, critical Mach number code:

```
8
% Mcrit calculates the critical Mach Number of a wing/body from the Cp
% distribution.
2
% variable: Mcr[] double output
% variable: fileDATA22 string input
% Simo Alberti
% Senior Project - R. McDonald
% Cal Poly, San Luis Obispo
% v. 1.0 - 02/06/2010
%% Load
[x, y, z, c] = parse22(fileDATA22);
npatch = size(c);
gamma = 1.4;
%% Calc
for k = 1:npatch
    Cp min(k,:) = min(c{k});
% Cp star is the incompressible pressure coefficient for each patch
    Cp star(k) = min(Cp min(k));
% This is a combination of the Karmen-Tsien compressability correction
```

% (11.54 - Aerodynamics, Anderson) and the Cp,crit equation (11.60 - % Aerodynamics, Anderson)

```
\begin{aligned} & \mbox{Mcr}(k) = fzero \ (@(Mcr) \ (Cp_star(k) \ / \ (sqrt(1-Mcr^2) \ + \ Mcr^2/(1 \ + \ sqrt(1+Mcr^2)) \ (Cp_star(k) \ / 2) \ ) \ ) \ - \ (2/(gamma \ + \ Mcr^2) \ \ ( \ ( \ (2 \ + \ (gamma \ - \ 1) \ + \ Mcr^2) \ / \ (gamma \ + \ 1) \ ) \ ) \ (gamma \ - \ 1)) \ , \ 0.5); \end{aligned}
```

end