Experiment #1 STALL SPEED and $C_{L_{max}}$ Determination

Stall Flight

Data Card

Objectives

1. Observe the characteristics at the stall at different c.g. positions, configurations, and power settings.
2. Determine the warning speeds and stall speeds as above.
3. ~~Observe flow separation patterns.~~

Instrumentation

1. Airspeed and altimeter calibrated for instrument and position errors.
2. Angle of attack (α) meter. ← From GIST Station
3. ~~Wing tufts.~~

Flight Test Techniques

The aircraft is stabilized at approximately 1.5 times the stalling speed in the configuration to be tested. All forces are trimmed out. Then, using pitch alone, the aircraft speed is reduced at 1 kt/sec until the aircraft reaches the stall. The planned configurations* are:

1. Clean, power for level flight.
2. Clean, idle power.
3. Approach flaps, idle power. — 1/2 Flaps
4. Landing flaps, idle power. — FULL FLAPS

Set altimeter to 29.92 in Hg.

Flow Visualization

~~The first stall in each configuration will be done for data. Then each stall will be repeated to allow students to see the propagation of the stall over the wing.~~

* Do two configurations per team.

Stall Flight

Data Card

Aircraft:			Date:	
Zero fuel weight: 3772 lbf			c.g:	
Pitot-static system:			Console:	
	Configuration			
Trim Conditions	1	2	3	4
V_i				
H_i	5000	5000	5000	5000
w_f				
RPM				
M.P. or torque				
flaps				
gear				
Data				
V_i (warning)				
V_i (stall)				
H_i (stall)				
α (stall)				
$\Delta\phi$ (bank)				
$\Delta\psi$ (heading)				
H_i (recovery)				

Stall Flight					
Data Reduction					
1	H_i (ft)				
2	V_i (kts)				
3	ΔV_{ic} (kts)				
4	ΔV_{pc} (kts)				
5	ΔV_c (kts)				
6	V_e (kts)				
7	W_t (lbs)				
8	$C_{L\,max}$				
9	V_{S1} (kts)				
10	V_i (kts)				
11	ΔV_{ic} (kts)				
12	ΔV_{pc} (kts)				
13	ΔV_c (kts)				
14	V_e (kts)				
15	V_{warn} (kts)				
16	ΔV (kts)				
17	α (μA)				
18	α (degree)				

See Experiment #1 handout for a MATLAB script that implements this data reduction.

Stall Flight

Data Reduction

1. H_i (ft) Indicated pressure altitude.
2. V_i (kts) Indicated **stall** airspeed.
3. ΔV_{ic} (kts) Airspeed instrument error correction.
4. ΔV_{pc} (kts) Airspeed position error correction.
5. ΔV_c (kts) Scale altitude error correction (approximately 0 below 250 kts, below 10,000 ft)
6. V_e (kts) Equivalent airspeed: **[2]** + **[3]** + **[4]** + **[5]**.
7. W_t (lbs) Test weight = zero fuel weight + fuel weight
8. C_L Stall lift coefficient = $\frac{2W}{\rho_0 V_e^2 S} = \frac{2[7]}{.002377([6]\times 1.689)^2 S}$
   where S is the wing area.
9. V_{S1} (kts) Stall speed corrected for weight = $V_e\sqrt{\frac{W_S}{W_t}} = [6]\sqrt{\frac{W_S}{[7]}}$
   where W_S is the standard weight
10. V_i (kts) Indicated **stall warning** airspeed (buffet, horn, light)
11. ΔV_{ic} (kts) Airspeed instrument error correction.
12. ΔV_{pc} (kts) Airspeed position error correction.
13. ΔV_c (kts) Scale altitude error correction (approximately 0 below 250 kts, below 10,000 ft)
14. V_e (kts) Equivalent airspeed: **[10]** + **[11]** + **[12]** + **[13]**.
15. V_{warn} (kts) Warning speed corrected for weight = $V_e\sqrt{\frac{W_S}{W_t}} = [14]\sqrt{\frac{W_S}{[7]}}$
    where W_S is the standard weight
16. ΔV (kts) Stall warning = $V_{warn} - V_{S1} = [15]-[9]$
17. α (μA) Indicated α at stall
18. α (deg) Stall α converted to degrees (use aircraft calibration book)

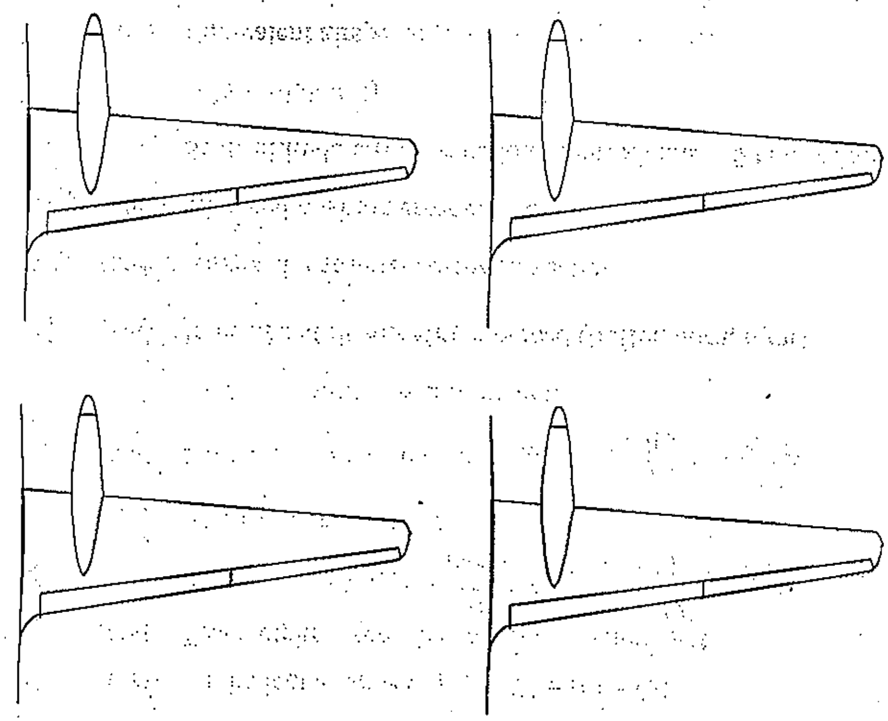
Stall Flight
Data Analysis

1. Does aircraft have adequate stall warning for each configuration?
2. Are stall speed warning results repeatable or do you require more data to obtain statistical confidence?
3. How does stall α change with configuration?
4. Using $C_{L_{max}}$ in the clean/idle power configuration as the baseline value, how does $C_{L_{max}}$ change with configuration?
5. Does aircraft meet FARs for heading change in a stall?
6. Does aircraft meet FARs for bank change in a stall?
7. Does aircraft lose more than 100 feet in a stall?
8. Are above results dependent of pilot technique?

2001
Spring

Airflow Separation Analysis

1. Sketch the separation pattern for each configuration tested.
2. How does power effect the pattern?
3. How does the flap setting effect the pattern?
4. Which appears to be most benign? Most adverse?
5. Is separation sudden? How does it correlate to buffet?



Experiment #1 Continued
Error Analysis

You are to perform the error analysis for the computation of CLmax. To do this you need to use the provided error analysis software (see course web site) and to write a MATLAB function that returns a scalar CLmax.

Error analysis software has been provided to you at the course web site.

http://roger.ecn.purdue.edu/~andrisan/Courses/AAE490A_S2003/Index.html

Note that CLmax is computed in this experiment from only 5 input quantities, Hi, Vi, Wt, flap and wing area. So we can build a vector of inputs x as follows:

```
x(1)=Hi(j)        Indicated altitude, feet
x(2)=Vi(j         Indicated airspeed at stall, knots
x(3)=Wt(j)        Weight, pounds
x(4)=flap(j)      Flap setting flap=0 for no flap %flap=0.5
                  for half flap %flap=1 for full flap
x(5)=S            Wing area, ft^2.
```

You will need to compute the error analysis for j=1 to the length of Hi. If your flight experiment involved flying three cases, then Hi, Vi, Ht and flap will be vectors of length three (j=1,2,3).

Unfortunately, the measured values of the first four of there these quantities are all slightly in error. These errors mean that the computed values for maximum lift coefficient will also be in error.

You need to write a MATLAB script to study the relationship between the errors in the five inputs (Hi, Vi, Wt, flap, and S) and the error in the output (CLmax).

First you need to define the array of errors

```
error=[100, 1, 100, .0001, .00001]
```

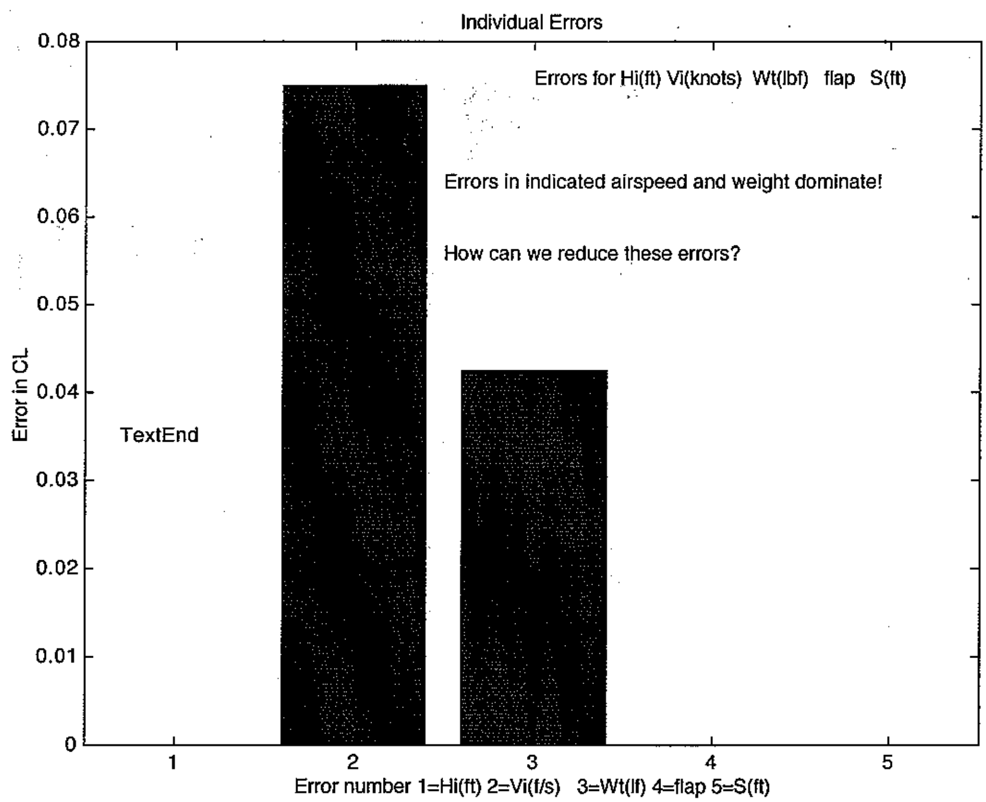
These are the assumed errors in Hi, Vi, Wt, flap, and S. Units are feet, knots, pounds, non-dimensional, square feet. Note that the errors in flap and S are deliberately quite small since we actually know them quite well.

If your flight experiment had three different cases, then you will need to generate a table of error analysis results of the following form.

	Error Analysis Data Reduction			
1	Error in Hi(ft)	100.0	100.0	100.0
2	Error in Vi(kts)	1.0	1.0	1.0
3	Error in Wt(lbf)	100.0	100.0	100.0
4	Abs Err in CLmax	0.11	0.11	0.12
5	Rss Err in CLmax	0.08	0.08	0.09

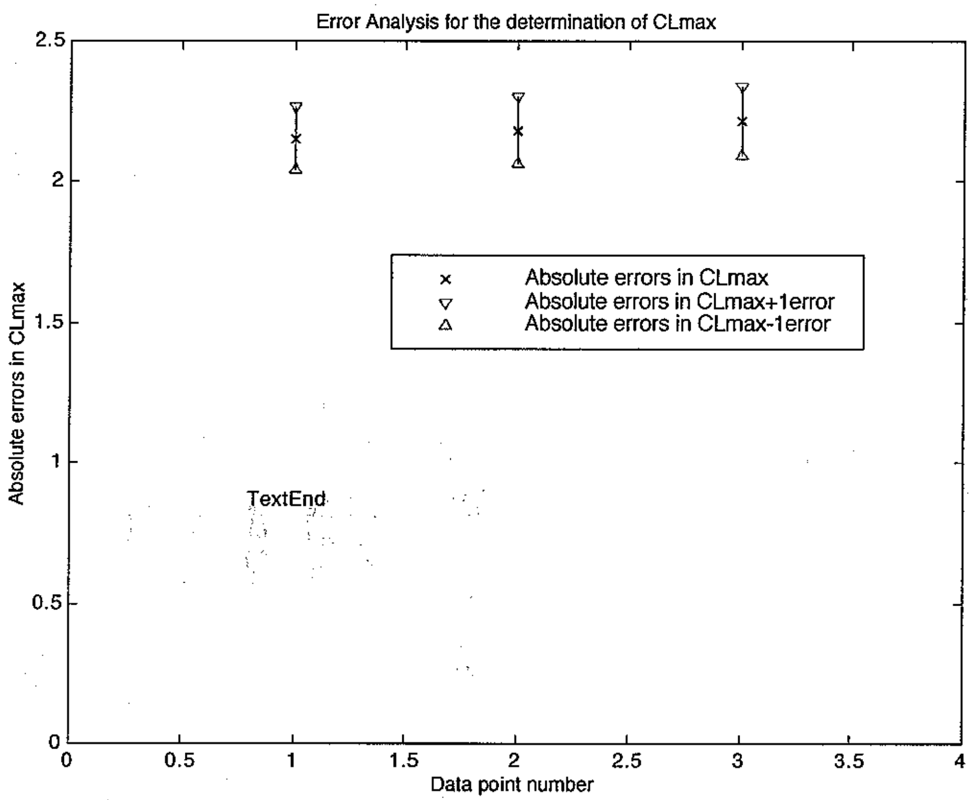
This table shows the results of using the provided software to compute the absolute and rss errors in the computation of CLmax.

Using the provided software, you need to generate a bar chart showing the individual errors from the five input quantities (Hi, Vi, Wt, flap, and S). An example is shown below.

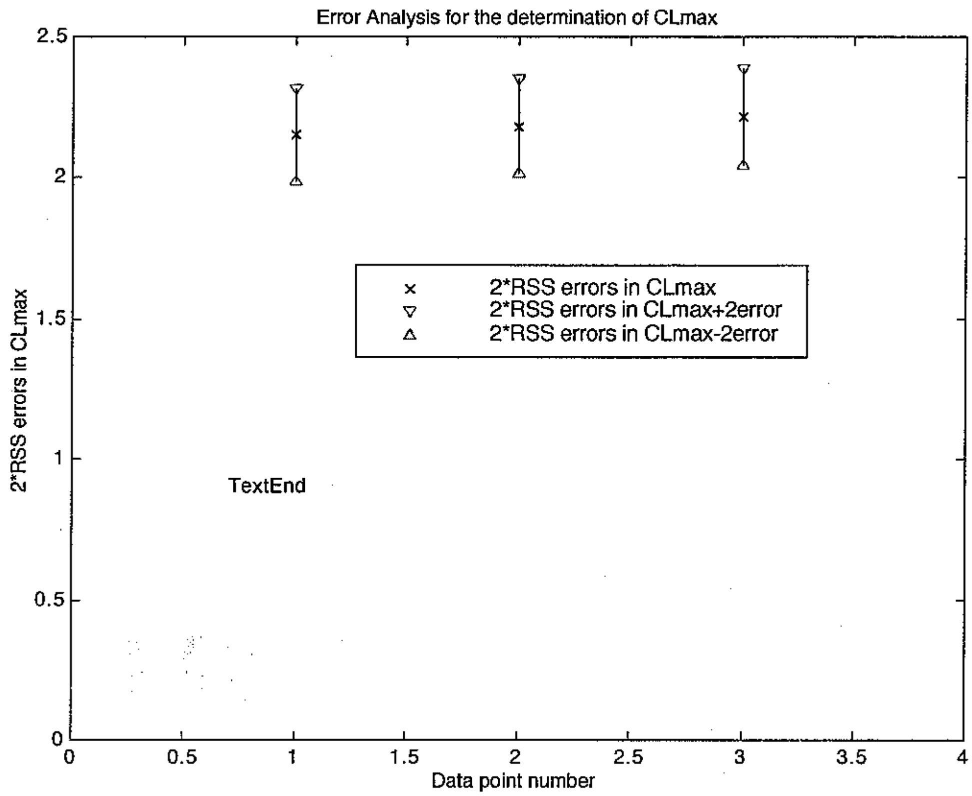


Please interpret these results. Do errors in altitude affect the accuracy of our determination of CLmax? What factors most affect the accuracy of our determination of CLmax?

You need also to plot CLmax that you determined for each of the cases you flew and plot them as shown below. Put the error bounds on the figure to indicate graphically how accurately you have computed CLmax. Do this under the assumption that the errors specified above are to be interpreted as absolute errors and again if the errors are to be interpreted as rss errors.



Shown below are the computed values of CLmax under the assumption that the errors are rss errors. The error bars reflect +- 2 times the rss errors.



In your opinion which error model is the most appropriate to use, the absolute error model or the rss error model? Why?

MATLAB SCRIPT TO PERRFORM THE FIRST PART OF THE EXPERIMENT DATA ANALYSIS (not including the error analysis)

The following script and the script MakeOutputTable can be found at the course web site.

http://roger.ecn.purdue.edu/%7Eandrisan/Courses/AAE490A_S2006/Buffer/Exp1/

```
% Experiment #1 Stall Speed and CLmax Determination
% Ref: NTPS, pages D24-D26.
% This MATLAB script computes stall airspeed and CLmax
% given Vindicated, flap setting, indicated geopotential pressure
altitude.
% It assumes standard outside air temperature for the calibrated
pressure altitude.
disp(' ')
disp('Start here')
disp('Experiment 1') %Example 1
disp(' ')
clear all
echo on
% List all inputs: Data Inputs
Hi=[4500,4100,4400]        % Item 1 of the Table of Outputs: indicated
geopotential feet at stall. Put your data in this vector.
Vi=  [61,60,59]            % Item 2: stall speed knots. Put your data in
this vector and the one below.
flap=[ 0, 0, 0]            % Flap setting %flap=0 for no flap %flap=0.5
for half flap %flap=1 for full flap
FuelWeight=[900,800,700]  % Pounds (lbf).
Viwarn=[64,66,63]          % Item 10, indicated stall warning speed
knots.
ZeroFuelWeight=4500        % Pounds <<< Check this number. Use the
number CÈline gives you.
Ws=5500                    % Pounds (lbf), standard weight is maximum
takeoff weight (Baron 58 manual p. 1-12)
S=199.2;                   % Wing area, ft squared
% When you do your experiment you will replace this data with your
data.
% All three vectors above must be of the same length.

% Compute stall speed and Clmax and build a Table of Outputs
% as called for in the Data Reduction section of the experiment.

% Item 1 in the Table of Outputs is indicated pressure altitude
% for the flight experiment and has been given above.

% Item 2 in the Table of Outputs is indicated stall airspeed
% for the flight experiment and has been given above.

% The following step is not listed in the NTPS Data Reduction
procedure,
% but we will do this in this and other experiments.
% We will use the function ALTBaron58 which performs two useful
% data corrections at once. It computes calibrated pressure altitude
% from indicated pressure altitude, indicated airspeed and flap
setting.
% In the process it determines the instrument corrections and
```

```
% static position error corrections. Unfortunately it does not
% return these two errors individually as is required for the Table of
Outputs,
% so we will have to do a few awkward steps to get data in the required
format
% for the Table of Outputs

help ALTBaron58

% Correct indicated altitude for instrument and position errors.
Hcalibrated=ALTBaron58(Hi,Vi,flap) % Altitude units are feet.
% Notice that we are using array operations to compute Hcalibrated,
i.e.,
% Hi and Vi are arrays and returned variable Hcalibrated is an array.
% For most computations done below we will use array operations.


% Items 3 and 4  are the individual corrections applied to
% airspeed measurements,
%  dVic is the speed increment to add to indicated airspeed to
% correct for instrument error.
%         Vic=Vi+dVic
% dVpc is the increment to add to airspeed to correct for
% static position error.
%         Vcalibrated=Vic+dVpc
%                    =Vi+dVic+dVpc

% Item 3 in the Table of Outputs: airspeed instrument correction (dVic)
% The Baron manual assumes that Vinstrumentcorrected (Vic) = Vindicated
(Vi)
% so dVic = 0
dVic=zeros(size(Vi)); % for Item 3
Vic=Vi+dVic; % sort of trivial since we assumed dVic=0
% Airspeed units are KNOTS.

% Item 4 in the Table of Outputs: airspeed position correction (dVpc)
% To do this we will use a function called AS1Baron58. This
% function corrects indicated airspeed for instrument error
% and static position error. Unfortunately it does not
% return these two errors individually, as is required for the Table of
Outputs,
% so we will have to do a few awkward steps to get data in the required
format.

help AS1Baron58

Vcalibrated=AS1Baron58(Vi,flap) % Airspeed units are KNOTS.
dVpc=Vcalibrated-Vic; % for Item 4. This is the awkward step.
% Airspeed units are KNOTS.

% Items 5 and 6 Compute dVc and equivalent airspeed
% dVc is the Scale Altitude Correction a.k.a. "Compressibility
Correction".

% Item 6a Get standard pressure (p, lbf/ft^2) for calibrated pressure
altitude.
GeometricFlag=0 %Hvector is Geopotential altitude
[temp,p,rho,Hgeopvector]=atmosphere4(Hcalibrated,GeometricFlag);
```

```
p
% Item 6b Get equivalent airspeed (Vequivalent)
% To do this we will use the function AS2 to compute
% equivalent velocity from calibrated velocity and static pressure at
the test
% condition. Unfortunately, this function does not return dVc, so we
% will have to perform an awkward step to compute it.

help AS2

[Vequivalent,qcOverp0,qcOverp]=AS2(Vcalibrated,p); % for line 6, Knots
dVc=Vequivalent-Vcalibrated; % for Item 5. This is the awkward step.
Vequivalent                  % Airspeed units are KNOTS.

% Item 7 Compute test weights.
Wt=ZeroFuelWeight+FuelWeight % for Item 7 of the Table of Outputs


% Compute Item 8: CLmax
% To do this we will use two different methods.

% Item 8 done using equivalent airspeed (Vequivalent, knots)
rho0=0.00237691267925741;  % Sea level density, slugs per cubic foot
Vefps=1.687808*Vequivalent % convert knots to ft/sec
Clmax=2*Wt./(rho0*Vefps.*Vefps*S) % for Item 9 of the Table of Outputs

% Item 8 of the Table of Outputs done using true airspeed (Vtrue,
knots)
% In this calculation we assume standard temperature for the calibrated
pressure altitude.
% It would be better if we had measured outside air temperature, but we
did not in this experiment.
temp     %degree Rankine
% To do this we use the function AS3. This function computes true
airspeed
% from equivalent airspeed, outside air temperature and calibrated
altitude.

help AS3

[Vtrue,M,aknots,rho,sigma]=AS3(Vequivalent,temp,Hcalibrated); % knots
Vtfps=1.687808*Vtrue % convert knots to ft/sec
Clmax2=2*Wt./(rho.*Vtfps.*Vtfps*S) % Note that this is dene with array
operations.

% Item 9 of the Table of Outputs: Stall speed corrected to the standard
weight
Vsl=Vequivalent.*sqrt(Ws./Wt) % for Item 9 of the Table of Outputs,
knots


% *********
% Compute warning speed. To do this we reepeat all the airspeed
calibration steps.

% Item 10 Indicated airspeed at stall warning.
% This was input above in the Data Inputs section of this script.
% The following step is not listed in the NTPS Data Reduction
```

```
procedure,
% but we will do this in this and other experiments.
% Correct indicated altitude for instrument and position errors.
Hcalibratedwarn=ALTBaron58(Hi,Viwarn,flap)
% The Baron manual assumes that Vinstrumentcorrected (Vic) = Vindicated
(Vi)
% so dVic = 0
dVicwarn=zeros(size(Viwarn)); % forItem 11
Vicwarn=Viwarn+dVicwarn;

% Items 11 and 12 Correct for indicated airspeed for instrument,
position errors.
Vcalibratedwarn=AS1Baron58(Viwarn,flap) % Airspeed units are KNOTS.
dVpcwarn=Vcalibratedwarn-Vicwarn; % for Item 12

% Items 13 and 14 Compute equivalent airspeed
% Item 14a Get standard pressure (p, lbf/ft^2) for calibrated pressure
altitude.
GeometricFlag=0 %Hcalibratedwarn is Geopotential altitude
[temp,pwarn,rho,Hgeopvector]=atmosphere4(Hcalibratedwarn,GeometricFlag)
;
pwarn
% Item 14b Get equivalent airspeed (Vequivalent)
[Vequivalentwarn,qcOverp0,qcOverp]=AS2(Vcalibratedwarn,pwarn); %Knots
Vequivalentwarn % for Item 14
dVcwarn=Vequivalentwarn-Vcalibratedwarn; % for Item 13

% Item 15 Warning speed corrected for weight
Vwarn=Vequivalentwarn.*sqrt(Ws./Wt) %for Item 15,  knots

% Item 16 Stall warning
StallWarning=Vwarn-Vs1 % for Item 16, knots

% Now let's call a script that writes a formatted table
%to the command window in a pretty format. We use a script here simply
to
% cut down on the size of this main script (which is already too long).
echo off
MakeOutputTable
```

Results of running the script Exp1.m

```
Start here
Experiment 1

% List all inputs: Data Inputs
Hi=[4500,4100,4400]        % Item 1 of the Table of Outputs: indicated
geopotential feet at stall. Put your data in this vector.

Hi =

        4500        4100        4400

Vi=  [61,60,59]            % Item 2: stall speed knots. Put your data in
this vector and the one below.

Vi =

    61    60    59

flap=[ 0, 0, 0]            % Flap setting %flap=0 for no flap %flap=0.5
for half flap %flap=1 for full flap

flap =

     0     0     0

FuelWeight=[900,800,700]  % Pounds (lbf).

FuelWeight =

   900   800   700

Viwarn=[64,66,63]          % Item 10, indicated stall warning speed
knots.

Viwarn =

    64    66    63

ZeroFuelWeight=4500        % Pounds <<< Check this number. Use the
number C�line gives you.

ZeroFuelWeight =

        4500

Ws=5500                    % Pounds (lbf), standard weight is maximum
takeoff weight (Baron 58 manual p. 1-12)

Ws =

        5500

S=199.2;                   % Wing area, ft squared
% When you do your experiment you will replace this data with your
data.
% All three vectors above must be of the same length.
```

```
% Compute stall speed and Clmax and build a Table of Outputs
% as called for in the Data Reduction section of the experiment.
% Item 1 in the Table of Outputs is indicated pressure altitude
% for the flight experiment and has been given above.
% Item 2 in the Table of Outputs is indicated stall airspeed
% for the flight experiment and has been given above.
% The following step is not listed in the NTPS Data Reduction
procedure,
% but we will do this in this and other experiments.
% We will use the function ALTBaron58 which performs two useful
% data corrections at once. It computes calibrated pressure altitude
% from indicated pressure altitude, indicated airspeed and flap
setting.
% In the process it determines the instrument corrections and
% static position error corrections. Unfortunately it does not
% return these two errors individually as is required for the Table of
Outputs,
% so we will have to do a few awkward steps to get data in the required
format
% for the Table of Outputs
help ALTBaron58
 function Hcalibrated=ALTBaron58(Hindicated,Vindicated,flap)
  Correct indicated altitude for instrument error and static position
error.
  Data is extracted from Pilots Operating Handbook Baron 58, page 5-17
(Normal System).
  It is assumed that the Beechcraft Baron 58 has zero instrument
correction.
  Therefore
              Hinstrumentcorrected (Hic) = Hindicated (Hi)
  Given below are the position error corrections to get Hcalibrated.
  They are a function of Hindicated, Vindicated, and flap position.
              Hcalibrated=f(Hindicated,Vindicated,flap)
  Hindicated is indicated pressure altitude in geopotential feet.
  Hcalibrated is calibrated pressure altitude in geopotential feet.
  Vindicated is indicated airspeed in knots.
                    flap=0   for no flap
                    flap=0.5 for half flap
                    flap=1   for full flap
  For accuracy, Vindicated must be in the range 60-200 knots for the no
flap case.
  For accuracy, Vindicated must be in the range 60-130 knots for the
full flap case.
  For accuracy, Hindicated must be in the range 0-20000 ft for the no
flap case.
  For accuracy, Hindicated must be in the range 0-10000 ft for the full
flap case.
  Use of this function for the half flap case is somewhat questionable.
  This function uses the 2D table lookup function interp2.


% Correct indicated altitude for instrument and position errors.
Hcalibrated=ALTBaron58(Hi,Vi,flap) % Altitude units are feet.

Hcalibrated =
```

```
          4497.5334176209                4097.607365
4397.57666

% Notice that we are using array operations to compute Hcalibrated,
i.e.,
% Hi and Vi are arrays and returned variable Hcalibrated is an array.
% For most computations done below we will use array operations.
% Items 3 and 4  are the individual corrections applied to
% airspeed measurements,
%  dVic is the speed increment to add to indicated airspeed to
% correct for instrument error.
%         Vic=Vi+dVic
% dVpc is the increment to add to airspeed to correct for
% static position error.
%         Vcalibrated=Vic+dVpc
%                    =Vi+dVic+dVpc
% Item 3 in the Table of Outputs: airspeed instrument correction (dVic)
% The Baron manual assumes that Vinstrumentcorrected (Vic) = Vindicated
(Vi)
% so dVic = 0
dVic=zeros(size(Vi)); % for Item 3
Vic=Vi+dVic; % sort of trivial since we assumed dVic=0
% Airspeed units are KNOTS.
% Item 4 in the Table of Outputs: airspeed position correction (dVpc)
% To do this we will use a function called AS1Baron58. This
% function corrects indicated airspeed for instrument error
% and static position error. Unfortunately it does not
% return these two errors individually, as is required for the Table of
Outputs,
% so we will have to do a few awkward steps to get data in the required
format.
help AS1Baron58
  function Vcalibrated=AS1Baron58(Vindicated,flap)
  Correct indicated airspeed for instrument error and static position
error.
  Data is extracted from Pilots Operating Handbook Baron 58, page 5-16
(Normal System).
  It is assumed that the Beechcraft Baron 58 has zero instrument
correction.
  Therefore
              Vinstrumentcorrected (Vic) = Vindicated (Vi)
  Given below are the position error corrections to get Vcalibrated.
  They are a function of Vindicated and flap position.
              Vcalibrated=f(Vindicated,flap)
                    flap=0   for no flap
                    flap=0.5 for half flap
                    flap=1   for full flap
  Vindicated and Vcal are a row vectors.
  Half flap data is linearly interpolated between no flap and full flap
data.
  Units are KNOTS.


Vcalibrated=AS1Baron58(Vi,flap) % Airspeed units are KNOTS.

Vcalibrated =
```

```
    61      60      59

dVpc=Vcalibrated-Vic; % for Item 4. This is the awkward step.
% Airspeed units are KNOTS.
% Items 5 and 6 Compute dVc and equivalent airspeed
% dVc is the Scale Altitude Correction a.k.a. "Compressibility
Correction".
% Item 6a Get standard pressure (p, lbf/ft^2) for calibrated pressure
altitude.
GeometricFlag=0 %Hvector is Geopotential altitude

GeometricFlag =

     0

[temp,p,rho,Hgeopvector]=atmosphere4(Hcalibrated,GeometricFlag);
p

p =

          1794.15802524556            1821.07726301237
1800.85576404759

% Item 6b Get equivalent airspeed (Vequivalent)
% To do this we will use the function AS2 to compute
% equivalent velocity from calibrated velocity and static pressure at
the test
% condition. Unfortunately, this function does not return dVc, so we
% will have to perform an awkward step to compute it.
help AS2
  function [Vequivalent,qcOverp0,qcOverp]=AS2(Vcalibrated,p)
  Computes the Scale Altitude Correction a.k.a. "Compressibility
Correction"
  Vcalibrated is calibrated velocity.
  p is static pressure at which Vcalibrated is known.
  Vcalibrated and p are row vectors.
  Assumes Vcalibrated <=ao.
  Units are knots and lbf/ft^2.
  Theses equations are a slightly different version
  of quantities in equation 2.15.2 in the book
  "Flight Test Engineering" by Ward and Srtganac (page 13).
  This function works for all different types of aircraft
  as long as Vcalibrated <=ao.


[Vequivalent,qcOverp0,qcOverp]=AS2(Vcalibrated,p); % for line 6, Knots
dVc=Vequivalent-Vcalibrated; % for Item 5. This is the awkward step.
Vequivalent                  % Airspeed units are KNOTS.

Vequivalent =

          60.9883956985937            59.9900282280775
58.9897545503723
```

```
% Item 7 Compute test weights.
Wt=ZeroFuelWeight+FuelWeight % for Item 7 of the Table of Outputs

Wt =

        5400         5300         5200


% Compute Item 8: CLmax
% To do this we will use two different methods.
% Item 8 done using equivalent airspeed (Vequivalent, knots)
rho0=0.00237691267925741;  % Sea level density, slugs per cubic foot
Vefps=1.687808*Vequivalent % convert knots to ft/sec

Vefps =

          102.936702167252          101.251649563575
99.5633796481548

Clmax=2*Wt./(rho0*Vefps.*Vefps*S) % for Item 9 of the Table of Outputs

Clmax =

          2.15268603376695          2.18373074765527
2.2158048954023


% Item 8 of the Table of Outputs done using true airspeed (Vtrue,
knots)
% In this calculation we assume standard temperature for the calibrated
pressure altitude.
% It would be better if we had measured outside air temperature, but we
did not in this experiment.
temp     %degree Rankine

temp =

          502.631076227417          504.057276519232
502.987538018174

% To do this we use the function AS3. This function computes true
airspeed
% from equivalent airspeed, outside air temperature and calibrated
altitude.
help AS3
  function
[VtrueKnots,M,aknots,rho,sigma]=AS3(Vequivalent,TdegR,Hpcalibrated)
  This function computes true arispeed from equivalent airspeed
  and works for all different types of aircraft.
  Hpcalibrated is calibrated pressure altitude (geopotential feet)
  Hpcalibrated can be a vector.
  TdegR is actual temperature in degR
  Vequivalent is equivalent airspeed in knots.
  Vtrue is true airspeed in knots.
  M is mach number (nondimensional)
  aknots is the speed of sound in knots
  rho is actual air density (slugs/ft^3)
  sigma=rho/rho0 nondimensional
```

```
[Vtrue,M,aknots,rho,sigma]=AS3(Vequivalent,temp,Hcalibrated); % knots
Vtfps=1.687808*Vtrue % convert knots to ft/sec

Vtfps =

          110.052373973644           107.60010827523
106.285402211383

Clmax2=2*Wt./(rho.*Vtfps.*Vtfps*S) % Note that this is dene with array
operations.

Clmax2 =

          2.15268603376695          2.18373074765527
2.2158048954023


% Item 9 of the Table of Outputs: Stall speed corrected to the standard
weight
Vs1=Vequivalent.*sqrt(Ws./Wt) % for Item 9 of the Table of Outputs,
knots

Vs1 =

           61.550512610895          61.1114342162578
60.6675228040291


% *********
% Compute warning speed. To do this we reepeat all the airspeed
calibration steps.
% Item 10 Indicated airspeed at stall warning.
% This was input above in the Data Inputs section of this script.
% The following step is not listed in the NTPS Data Reduction
procedure,
% but we will do this in this and other experiments.
% Correct indicated altitude for instrument and position errors.
Hcalibratedwarn=ALTBaron58(Hi,Viwarn,flap)

Hcalibratedwarn =

          4497.43439548361          4097.41522249815
4397.4783755843

% The Baron manual assumes that Vinstrumentcorrected (Vic) = Vindicated
(Vi)
% so dVic = 0
dVicwarn=zeros(size(Viwarn)); % forItem 11
Vicwarn=Viwarn+dVicwarn;

% Items 11 and 12 Correct for indicated airspeed for instrument,
position errors.
Vcalibratedwarn=AS1Baron58(Viwarn,flap) % Airspeed units are KNOTS.
```

```
Vcalibratedwarn =

    64    66    63

dVpcwarn=Vcalibratedwarn-Vicwarn; % for Item 12

% Items 13 and 14 Compute equivalent airspeed
% Item 14a Get standard pressure (p, lbf/ft^2) for calibrated pressure
altitude.
GeometricFlag=0 %Hcalibratedwarn is Geopotential altitude

GeometricFlag =

     0

[temp,pwarn,rho,Hgeopvector]=atmosphere4(Hcalibratedwarn,GeometricFlag)
;
pwarn

pwarn =

          1794.16465036291          1821.09027436524
1800.86235967757

% Item 14b Get equivalent airspeed (Vequivalent)
[Vequivalentwarn,qcOverp0,qcOverp]=AS2(Vcalibratedwarn,pwarn); %Knots
Vequivalentwarn % for Item 14

Vequivalentwarn =

          63.9866024589078          65.986736356799
62.9875315476322

dVcwarn=Vequivalentwarn-Vcalibratedwarn; % for Item 13

% Item 15 Warning speed corrected for weight
Vwarn=Vequivalentwarn.*sqrt(Ws./Wt) %for Item 15,  knots

Vwarn =

          64.5763531974026          67.2202400486068
64.7790033313736


% Item 16 Stall warning
StallWarning=Vwarn-Vs1 % for Item 16, knots

StallWarning =

          3.02584058650753          6.10880583234901
4.11148052734453


% Now let's call a script that writes a formatted table
%to the command window in a pretty format. We use a script here simply
to
% cut down on the size of this main script (which is already too long).
echo off
```

Stall Flight Data
Data Reduction

1	Hi(ft)	4500.0	4100.0	4400.0
2	Vi(kts)	61.0	60.0	59.0
3	dVic(kts)	0.0	0.0	0.0
4	dVpc(kts)	0.0	0.0	0.0
5	dVc(kts)	-0.0	-0.0	-0.0
6	Ve(kts)	61.0	60.0	59.0
7	Wt(lbf)	5400.0	5300.0	5200.0
8	CLmax	2.2	2.2	2.2
9	Vs1(kts)	61.6	61.1	60.7
10	Vi(kts)	64.0	66.0	63.0
11	dVic(kts)	0.0	0.0	0.0
12	dVpc(kts)	0.0	0.0	0.0
13	dVc(kts)	-0.0	-0.0	-0.0
14	Ve(kts)	64.0	66.0	63.0
15	Vwrn(kts)	64.6	67.2	64.8
16	dV(kts)	3.0	6.1	4.1
17	ignore			
18	ignore			