MathWorks | Training Services

Appendix B: Exercises

MATLAB® Fundamentals for Aerospace Applications

Exercises

All exercise and solution files are found in the `exercises` subfolder of the `C:\class\coursefiles\mlbe_o` folder created by the course installer.

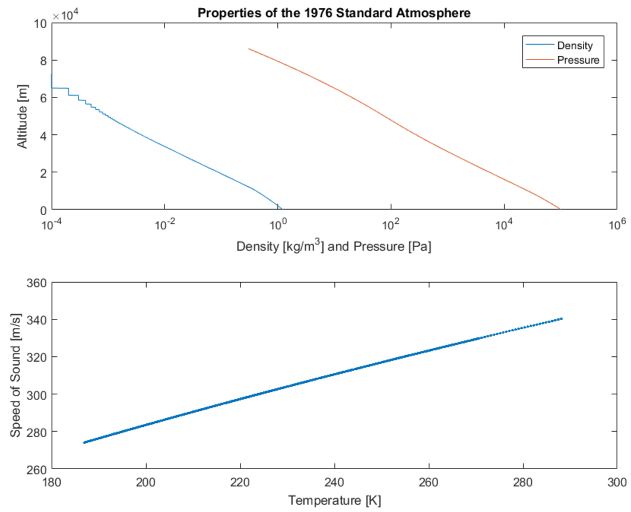
Standard Atmosphere I

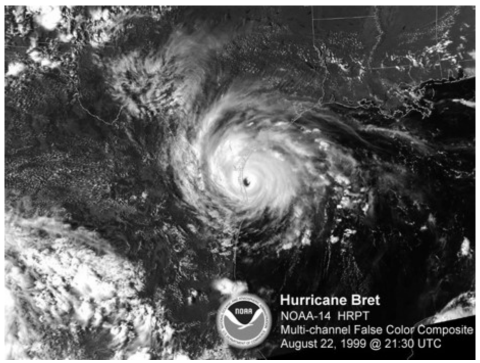
Reference: Chapter 2
Difficulty: I (Fundamental)

Exercise

1. Open the file `Standard_Atmosphere.txt` as text.
2. Use the Import Tool to load the data in the file into a 861-by-5 array `data` in the MATLAB® workspace.
3. Create new variables `alt`, `temp`, `press`, `den`, and `sos` from the columns of `data`.
4. In the Workspace browser, select `den` and `alt` and use the plotting options on the **Plot** tab of the toolstrip to plot the variation of density with altitude. Although altitude is the independent variable and density is the dependent variable, it is traditional to plot altitude on the vertical axis. Use the Plot Tools to format and label the plot, and to change the density axis scale to logarithmic.
5. Add the pressure variation with altitude to the same plot.
6. Divide the figure window into two subplots. In the second subplot, plot `sos` against `temp`. Add labels, and change the line properties to have no line and points as markers.
7. Hide Plot Tools and save the figure as a `.fig` file and as a JPEG-file.
8. Interactively save your five variables (`alt`, `temp`, `press`, `den`, `sos`) in a MAT-file from the Workspace browser.

Solution
`stdAtmo1.fig`





COESA:
The Standard Atmosphere was adopted October 15, 1976 by the United States Committee on Extension to the Standard Atmosphere (COESA), representing 29 U.S. scientific and engineering organizations.

You can find further information in *U.S. Standard Atmosphere, 1976* published by the U.S. Government Printing Office, Washington, D.C. 20401.

Standard Atmosphere II

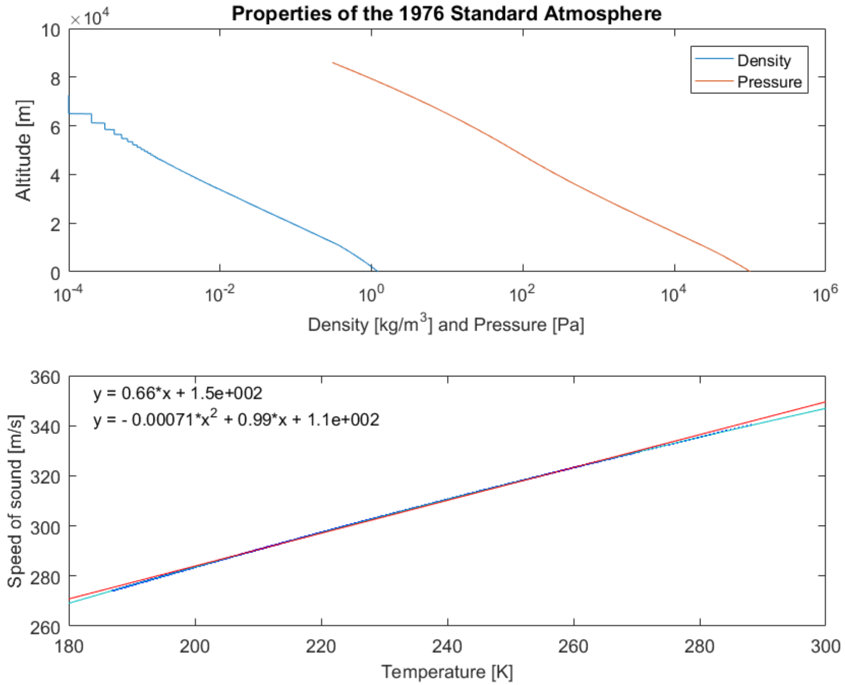
Solution
`stdAtmo2.fig`

Reference: Chapter 2
Difficulty: II (Intermediate)

Exercise

This exercise continues from the exercise "Standard Atmoshphere I".

1. Open your figure from the previous exercise, or the provided solution `stdAtmo1.fig`.
2. Open the Basic Fitting Tool from the figure window.
3. Determine the equation for a linear fit of `sos` against `temp`.
4. Use the **Zoom** and **Pan** buttons to examine the linear fit. Do you see any problems with the fit? Try a quadratic fit as well.



The Cassini-Huygens Spacecraft I

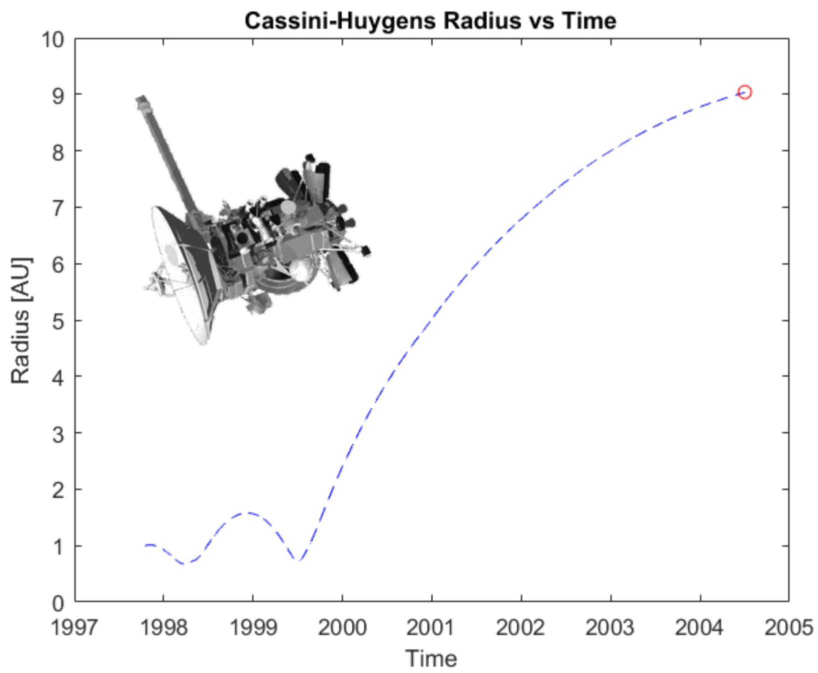
Solution
`cassiniPath1.mlx`

Reference: Chapter 3
Difficulty: I (Fundamental)

Exercise

The Cassini-Huygens mission was to study the planet Saturn and its moons. The payload consisted of the Cassini orbiter and the Huygens probe. The craft was launched in October 1997 and reached Saturn in 2004. Cassini-Huygens' path consisted of four different gravity assist maneuvers.

1. Create a new MATLAB live script. Load the file `cassiniData1.mat`. This file contains column vectors for Time, Year, Month, Day, Radius, Latitude, and Longitude. (Time consists of year and fractional year and is included for ease of plotting against time.).
2. Plot `Radius` versus `Time` with a blue, dashed line.
3. Add annotations to the plot including a title and *x*- and *y*-axis labels.
4. The maximum radius occurs at the final time. What is the maximum radius? Plot the maximum as a red circle on the plot.



Following a Flight I

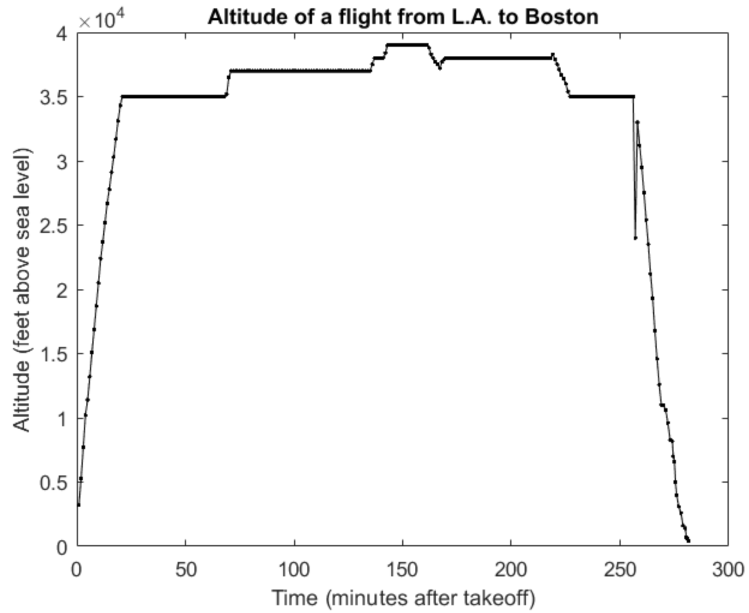
Solution
`plotFlight1.mlx`

Reference: Chapter 3
Difficulty: I (Fundamental)

Exercise

In this exercise, you will plot the altitude of a commercial aircraft that flew from Los Angeles International Airport (airport code LAX) to Boston Logan International Airport (BOS) on July 27, 2009. The plane took off from LAX at 11:11 p.m. Pacific Daylight Time.

1. Create a new MATLAB live script. Load the data from the file `flightData.mat`. The variables represent the time (`t`) in minutes after takeoff from LAX and the altitude (`altitude`) in feet above sea level.
2. Plot the altitude of the aircraft versus the time in black with points as markers and a solid line connecting the points.
3. Add a title to your graph as well as labels to the *x*- and *y*-axes.



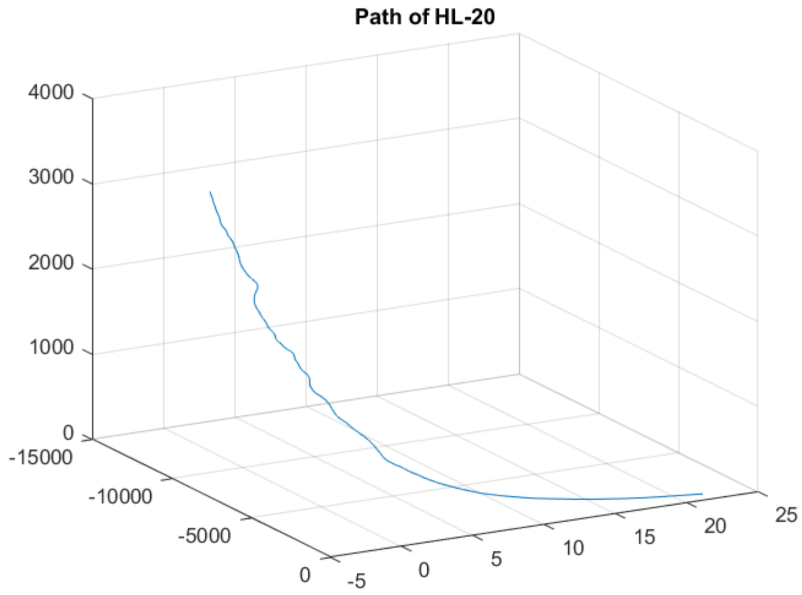
HL-20 Flight Data

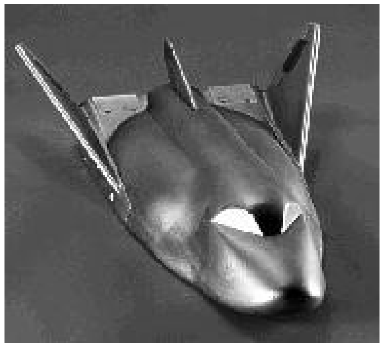
Solution
HL20flight.mlx

Reference: Chapter 3
Difficulty: I (Fundamental)

Exercise

1. Load the data in `HL20data.mat`. The data represents the location (in Cartesian coordinates) of an aircraft, as a function of time.
2. Plot the path of the aircraft, using `plot3`.
3. Annotate your plot with a title and add a grid. Use the 3D Rotate tool or the `view` command to change the angle from which the plot is viewed.

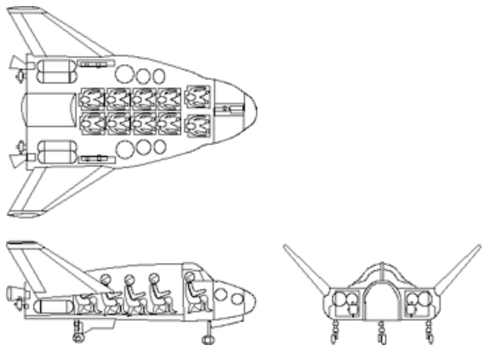




NASA designed the HL-20 lifting body as a manned spaceplane backup to the space shuttle (in case the shuttle was grounded).

It would also serve as a Crew Emergency Return Vehicle (CERV) from the space station Freedom.

The lifting body reentry vehicle is loosely based on the Russian BOR-4 design. It is intended for two flight crew and eight passengers. The spacecraft is capable of piloted landings at a designated airfield on rolling landing gear.



The Cassini-Huygens Spacecraft II

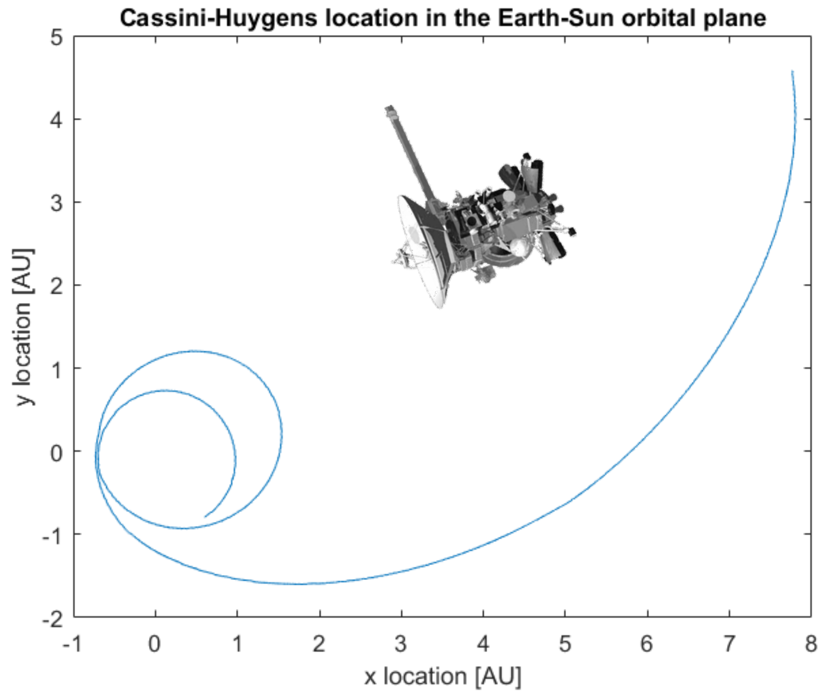
Solution
`cassiniPath2.mlx`

Reference: Chapter 4
Difficulty: I (Fundamental)

Exercise

1. Open your solution to the exercise "The Cassini-Huygens Spacecraft I" or the provided solution `cassiniPath1.mlx`. Save the file with a new name. This script loads the data and plots the distance of the Cassini-Huygens probe from the Sun as a function of time.
2. Using the `min` function, find the spacecraft's smallest distance to the sun and the year and month that it occurred. Does this agree with the plot?
3. Plot the probe's path in Cartesian coordinates. To do this:
   a. Convert the latitude and longitude from degrees to radians. (**Hint** 1 degree = $\pi/180$ radians.)
   b. Use the `sph2cart` function to convert the probe location to x, y, and z coordinates, using the latitude and longitude in radians. (**Note** In spherical coordinates, latitude is also known as *elevation*, and longitude as *azimuth*.)
   c. Plot the probe's x-y location and add appropriate annotations.

Bonus: Plot the x-y-z location using the `plot3` function.



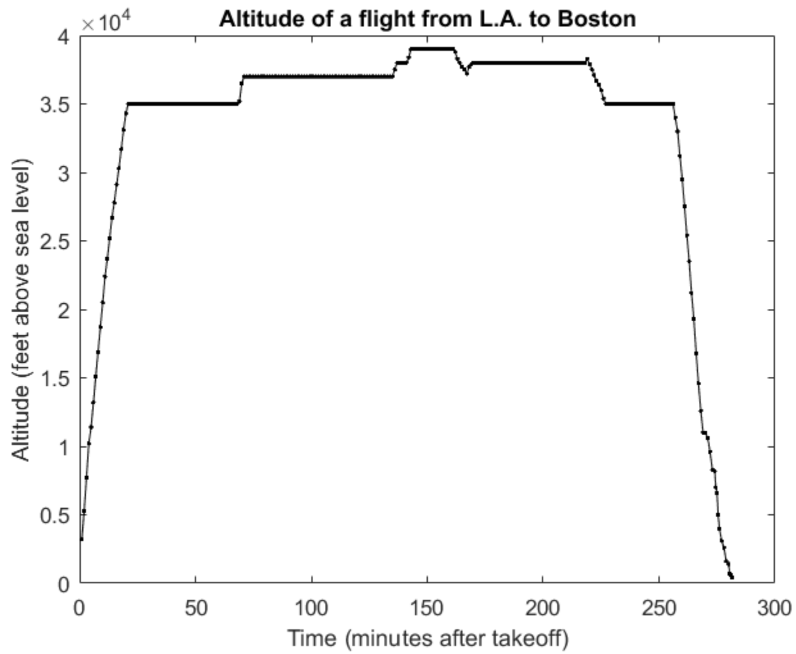
Following a Flight II

Solution
`plotFlight2.mlx`

Reference: Chapter 4
Difficulty: II (Intermediate)

Exercise

1. Open your solution to the exercise "Following a Flight I" or the provided solution `plotFlight1.mlx`. Save the file with a new name.
2. From the plot created in Part I, it appears there is an altitude sometime after `t = 250` that was recorded incorrectly. Zoom in to the plot to find `tBad`, the time at which that point occurred.
3. Using MATLAB commands, replace the value of the altitude at time `tBad` with the average of the altitude at times `tBad-1` and `tBad+1`.
4. Replot using the same plot style as before (black dots connecting a straight line.) Make sure to include a title and axes labels.



Following a Flight III

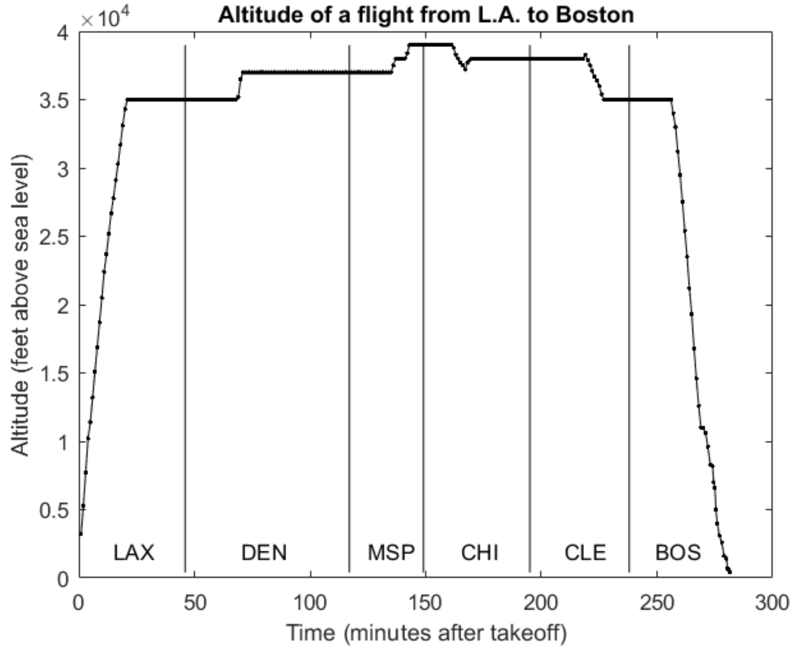
Solution
`plotFlight3.mlx`

Reference: Chapter 4
Difficulty: II (Intermediate)

Exercise

The `flightData` file includes information on which air traffic control tower the flight was in contact with at what times. In this exercise, you will add this information to the plot.

1. Open your solution to the exercise "Following a Flight II" or the provided solution `plotFlight2.mlx`. Save the file with a new name.
2. Starting with the plot created in Part 2, plot vertical lines at the times between centers to separate the graph into regions. This information is stored in the `centers` variable. For example, the transition from the Los Angeles to Denver centers occurred at $t = 46$; draw a vertical line at $t = 46$ to indicate this transition.
3. Add text on your plot to indicate which center each region of your graph corresponds to. Your graph should look similar to the one to the right. (**Hint** For more help, enter `doc text`. If you need to move or delete text after you display it, use Plot Tools. Use the three-letter codes that are given in the `cities` variable as the text.)



Spectral Measurement

Reference: Chapter 4
Difficulty: II (Intermediate)

Exercise

You can determine the speed of a star relative to the Earth by calculating the amount of *redshift* – the change in the wavelengths of light emitted by the star. One way to determine the amount of redshift is to compare the wavelength of a notable feature with its known experimental value. This exercise uses hydrogen-alpha radiation, which has a canonical wavelength of 656.28 nm.

The spectra of most stars have a distinctive "spike" or "dip" near 650 nm, corresponding to H_α radiation. This could be detected by looking for either the maximum or minimum flux in a given range around 650 nm. To cover both possibilities, a more robust approach is to look for the largest deviation from the average.

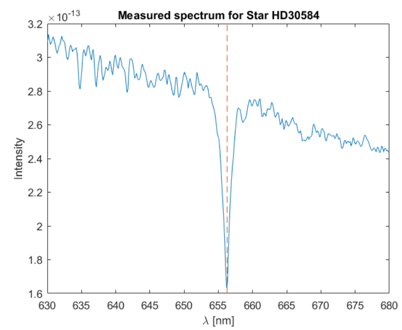
The spreadsheet `spectra.xlsx` contains 10 columns, each representing the measured spectrum of a star. The value in each row is the observed intensity of a particular wavelength of light. The header information provides the wavelength of the first row (λ_0) and the difference between subsequent values of the wavelength ($\Delta\lambda$). Wavelengths are given in nm.

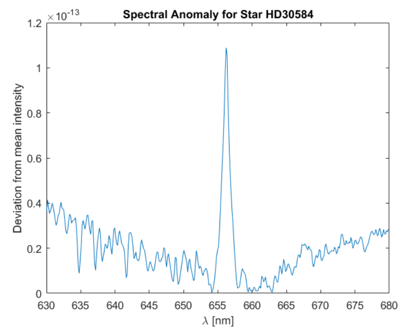
1. Use the Import Tool to import the first column (`HD30584`) from the file `spectra.xlsx`.
2. Create four scalar variables `lambdastart`, `dlambda`, `nObs`, and `lambdaend` containing the values of λ_0 (630.02), $\Delta\lambda$ (0.14), the number of observations (357), and the last value of the wavelength (that is, the wavelength corresponding to the last observation), according to the formula

$$\lambda_{end} = \lambda_0 + (n_{Obs} - 1)\Delta\lambda$$

Solution
`spectrum.mlx`

3. Create a column vector `lambda` that starts at `lambdastart`, and goes in steps of `dlambda` to `lambdaend`.
4. Plot `HD30584` as a function of `lambda`. Label your plot.
5. To draw a line on your plot from (x_1,y_1) to (x_2,y_2), use the command `plot(x,y)` where `x` is a vector of the points x_1 and x_2, and `y` is a vector of y_1 and y_2. Create a vector `x` containing the two values 656.28 and 656.28 (i.e., the same value twice). Create a vector `y` containing the two values 1.6×10^{-13} and 3.2×10^{-13} (**Note** You can enter a value such as 1.6×10^{-13} using the notation `1.6e-13`). Use `x` and `y` to add a dashed line to your plot.
6. Create a variable `meanFlux` that contains the average (arithmetic mean) of `HD30584`.
7. Use the `abs` function to calculate the absolute difference between the flux (`HD30584`) and its mean (`meanFlux`).
8. Plot the anomaly calculated in the previous step as a function of wavelength (`lambda`).
9. Find the wavelength of the maximum anomaly. (**Hint** Remember that the `max` function can optionally return the index of the maximum value.)
10. Calculate the redshift $z = \lambda_{max}/656.28 - 1$, where λ_{max} is the wavelength calculated in step 9. From this, calculate the speed (in km/s) of the star relative to Earth, by multiplying z by 300000 (the approximate speed of light in km/s).





Delta-v for a Booster Rocket I

Reference: Chapter 5
Difficulty: I (Fundamental)

Exercise

The *delta-v* (Δv) of a rocket is the total change of velocity that can be achieved by converting the chemical energy of the fuel to kinetic energy (in an idealized setting). Requirements for orbital maneuvers are typically given in terms of Δv; for example, a low earth orbit requires a Δv of 9.5 – 10 km/s. Delta-v is related to the change in mass of the rocket, which can be broken into the mass of the fuel and the mass of the rocket structure. However, increasing the mass of the fuel also increases the size of the rocket and, thus, the mass of the structure. Hence, for a fuel mass of m, the rocket equation for Δv becomes

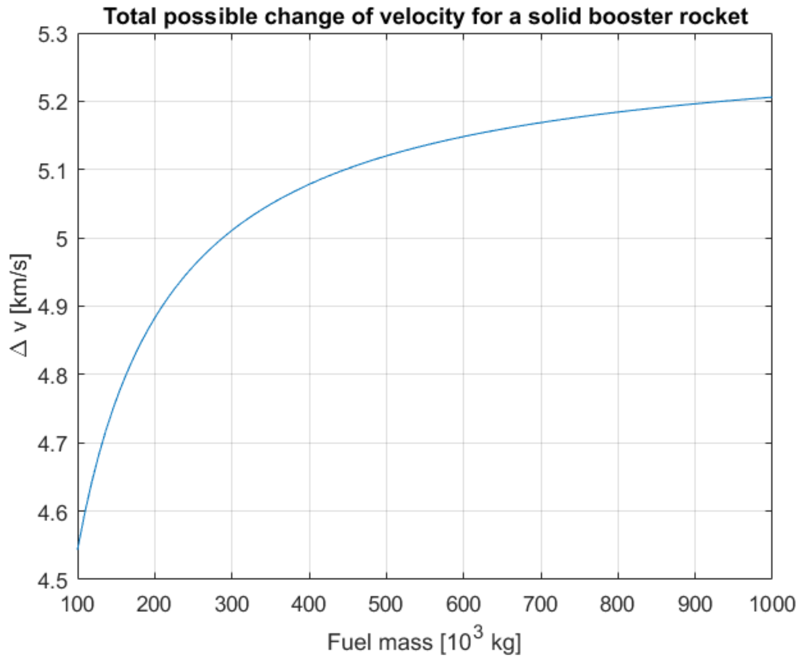
$$\Delta v = v_e \ln\left(1 + \frac{m}{\alpha m + m_{fix}}\right)$$

where v_e is the effective exhaust velocity, m_{fix} is the fixed mass of the rocket structure (such as the engine, which doesn't depend on the fuel mass), and α is a dimensionless parameter that depends on the geometry of the rocket and the densities of the fuel and structure. Create a script to calculate Δv for a range of values of m. To keep the numbers manageable, you may want to work in units of km and tonnes (1 t = 1000 kg), rather than m and kg.

1. Create scalar variables for v_e (2.6 km/s), alpha (0.15), and m_{fix} (6.1 t).
2. Create a vector of masses from 100 t to 1000 t.
3. Calculate and plot Δv as a function of m. (**Hint** In the calculation, you must prefix the division operator with a dot (`./`) to perform a vector operation instead of a matrix operation.)
4. Add labels, a title, and a grid to your plot. Format your script to provide help and documentation.

Solution
`deltav1.mlx`

Bonus: Experiment with increasing and decreasing the values of α and m_{fix} by a fixed percentage. Observe the effect on the plot. If you had to decide to spend Research and Development funds on reducing the weight of either the rocket engine (m_{fix}) or the rocket body (α), which would you choose (assuming that similar improvements are likely in either case for the same amount of effort)?



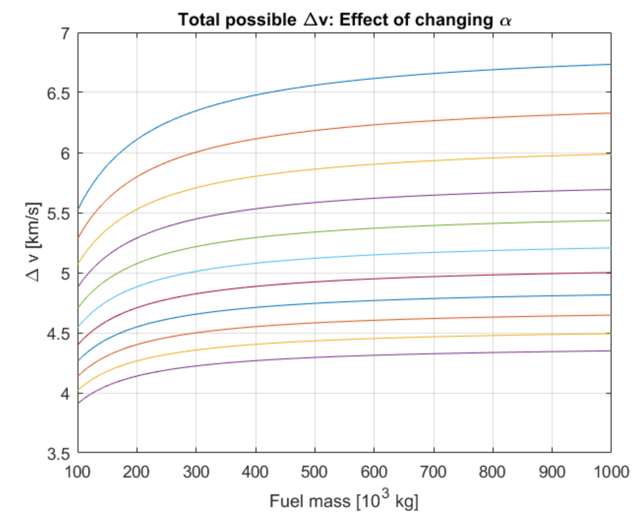
Delta-v for a Booster Rocket II

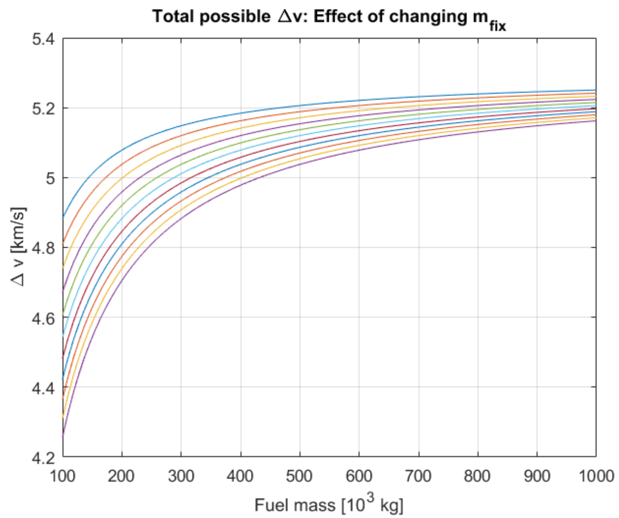
Solution
deltav2.mlx

Reference: Chapter 5
Difficulty: II (Intermediate)

Exercise

1. Open your solution to the exercise "Delta-v for a Booster Rocket I" or the provided solution `deltav1.mlx`. Save the file with a new name. This script determines the Δv for a rocket for a given range of values of the fuel mass. Two design parameters, α and m_{fix}, are fixed.
2. Create a column vector, `perc`, of values from 0.5 to 1.5 in steps of 0.1.
3. Multiply `alpha` by the vector `perc` in the calculation of Δv. Modify the calculation to use array multiplication where necessary. Note that the calculation results in a matrix. Plot the matrix as a function of fuel mass.
4. Repeat the previous step, this time multiplying the value of m_{fix} by the vector `perc` instead.
5. If you had to decide to spend Research and Development funds on reducing the weight of either the rocket engine (m_{fix}) or the rocket body (α), which would you choose (assuming that similar improvements are likely in either case for the same amount of effort)?





Wright Flyer Wind Tunnel Test I

Solution
`wrightFlyer1.mlx`

Reference: Chapter 5
Difficulty: I (Fundamental)

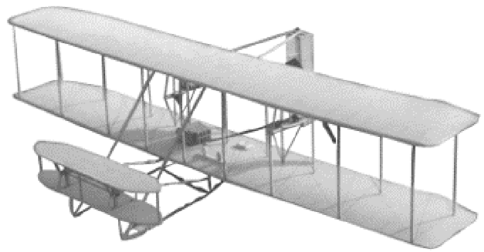
Exercise

1. Load the file `wrightWindTunnel.mat` and read the description.
2. Calculate the mean and standard deviation of the lift and drag forces for each of the five tests. Verify that they are reasonably consistent across the different tests.
3. Use array operations to calculate the coefficient of drag for each measurement, using the formula

$$C_D = \frac{F_D}{A \cdot P_d}$$

where F_D is the drag force, P_d is the dynamic pressure, and A is the wing area (510 ft^2 for the Wright flyer).

4. Plot the five curves of C_D versus angle of attack simultaneously. Add labels and a legend to your plot.

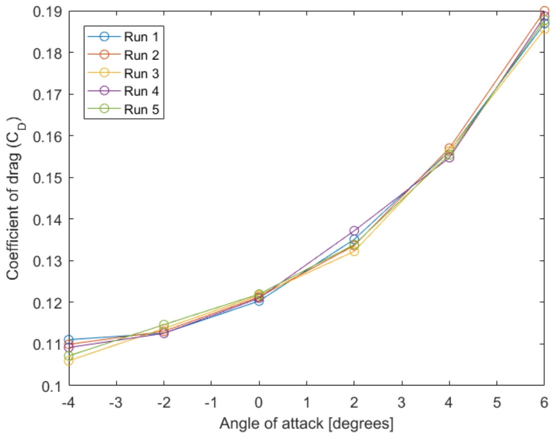


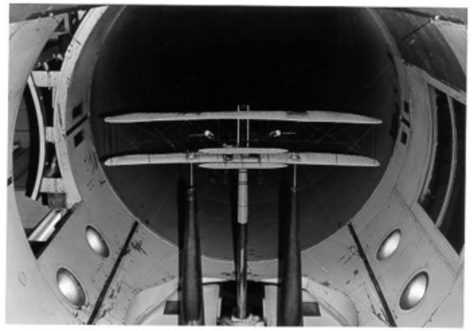
In March of 1999, aeronautical engineers used the NASA Ames Research Center in Mountain View, CA to test a full scale replica of the original Wright Flyer. This model was tested in the 40'x80' Low Speed Wind Tunnel.

The project was conducted to commemorate the Wright Brothers' accomplishment nearly a century earlier.

The goal was to generate a complete body of experimental data and analyses that describe the characteristics of the first successfully piloted airplane.

`http://www.wrightflyer.org/WindTunnel`





Wright Flyer Wind Tunnel Test II

Solution
`wrightFlyer2.mlx`

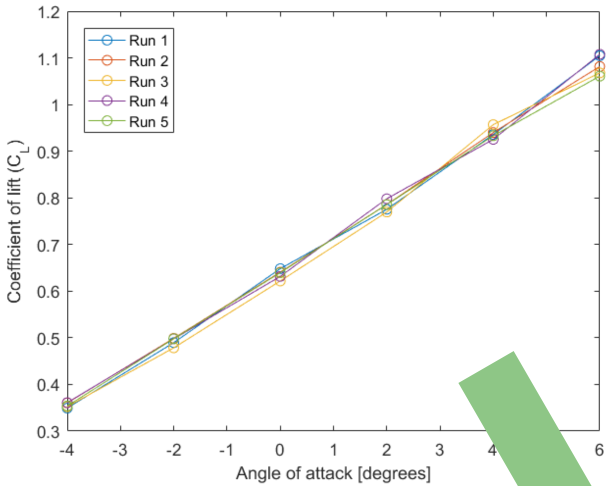
Reference: Chapter 5
Difficulty: II (Intermediate)

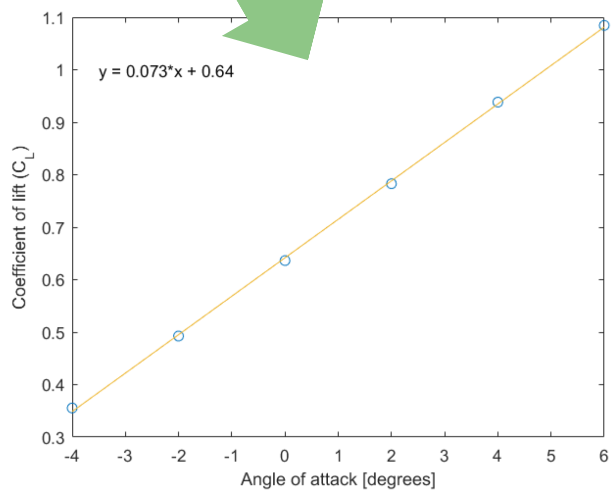
Exercise

1. Load the file `wrightWindTunnel.mat` and read the description.
2. Use array operations to calculate the coefficient of lift for each measurement, using the formula

$$C_L = \frac{F_L}{A \cdot P_d}$$

   where F_L is the lift force, P_d is the dynamic pressure, and A is the wing area (510 ft^2 for the Wright flyer).
3. Plot the five curves of C_L versus angle of attack simultaneously.
4. Calculate the mean coefficient of lift for each angle of attack (i.e., average across the five tests).
5. Make a scatter plot of the mean coefficient of lift as a function of angle of attack. Add labels to your plot. Use the Basic Fitting Tool to add a linear least-squares line of best fit to the data.





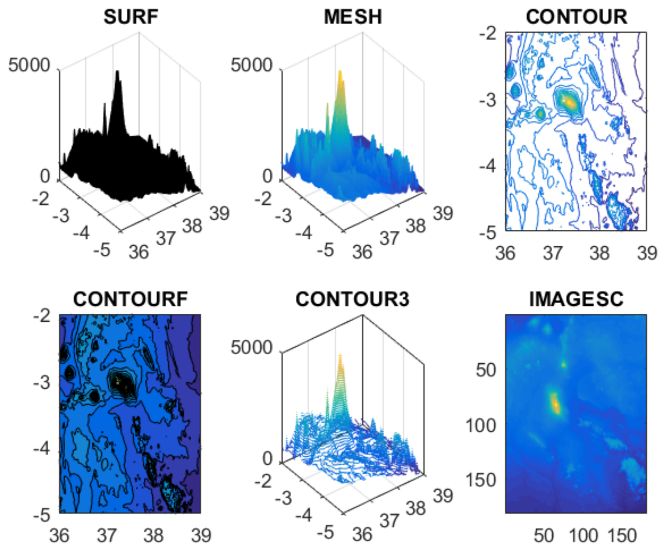
Satellite Terrain Image

Solution
`satelliteImage.mlx`

Reference: Chapter 5
Difficulty: II (Intermediate)

Exercise

1. Load the file `surfaceData.mat`.
2. Display the matrix `Altitude` (versus `Longitude` and `Latitude`, when possible) in separate figures using the following functions:
   - `surf`
   - `mesh`
   - `contour` (with 20 levels)
   - `contourf` (with 20 levels)
   - `contour3` (with 50 levels)
   - `imagesc`



The first satellite photographs of Earth were made April 1, 1960 by the weather satellite TIROS-1.

The Landsat program is the longest running enterprise for acquisition of imagery of Earth from space.

The first Landsat satellite was launched in 1972; the most recent, Landsat 7, was launched on April 15, 1999.

The instruments on the Landsat satellites have acquired millions of images throughout the program.

UAV Capabilities I

Solution
`UAV1.mlx`

Reference: Chapter 6
Difficulty: I (Fundamental)

Exercise

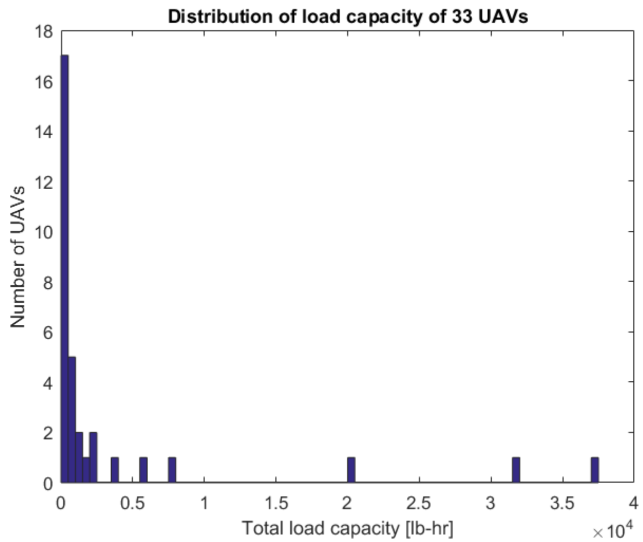
The file `UAVData.xlsx` contains four columns of data on 33 unmanned aerial vehicles (UAVs). The first column contains the names of the UAVs; the next three columns contain numeric data: endurance (hours), payload (pounds), and operational altitude ceiling (feet), respectively.

1. Use the `readtable` command to import the data as a table with four variables.
2. Use array multiplication to calculate the total load capacity – that is, the product of the endurance and the payload – of each UAV (in lb-hr). Add this as a fifth variable in the table.
3. Sort the table by total load capacity (calculated in the previous step). Display the names of the UAVs in descending order of total load capacity.
4. Create a histogram of total load capacity with a bin width of 500 lb-hr. (**Hint** Use an optional input to `histogram` to set the `BinWidth` property to 500.)

Cruising at extremely high altitudes, UAVs such as Northrop-Grumman's Global Hawk, can survey large geographic areas with pinpoint accuracy in near real time.

Once mission parameters are programmed into Global Hawk, it can autonomously taxi, take off, fly, remain on station capturing imagery, return, and land.

Note the resemblance of the Global Hawk aircraft to the `U2` reconnaissance aircraft of a previous generation.



UAV Capabilities II

Solution
`UAV2.mlx`

Reference: Chapter 7
Difficulty: I (Fundamental)

Exercise

The file `UAVData.xlsx` contains four columns of data on 33 unmanned aerial vehicles (UAVs). The first column contains the names of the UAVs; the next three columns contain numeric data: endurance (hours), payload (pounds), and operational altitude ceiling (feet), respectively.

1. Use the `readtable` command to import the data as a table with four variables.
2. How many UAVs are able to carry a load of 500 lb or more?
3. What is the average payload capability of the UAVs that can fly at 30,000 ft or higher?
4. Display the names of the UAVs that can carry a load of at least 200 lb at 30,000 ft or higher for at least 6 hours.

Cypher	2.5	45	5000
Darkstar	8	1000	45000
Eagle Eye	8	300	20000
Firebee	1.25	470	60000
Freewing	3.5	50	15000
Hawk-i 7B	1	3	3000
Hawk-i 7F	2	12	3000
Hawk-i 7H	1	5	3000
Huntair	7.5	80	17000
Hunter	12	200	15000
Javelin	1.5	6	3000
Model 324	2.5	200	43000
Model 350	1	400	40000
Model 410	12	300	30000
Outrider	4	160	15000
Pathfinder	16	88	70000
Perseus B	72	441	65620
Porter	4	75	5000
Predator	29	700	40000
Prowler	6	50	21000
Raptor	8	75	65000
SASS Lite	5	100	9850
Seabat	3	50	10000
Shadow 600	14	100	17000
Skyeye	10	175	18000

The Cassini-Huygens Spacecraft III

Solution
`cassiniPath3.mlx`

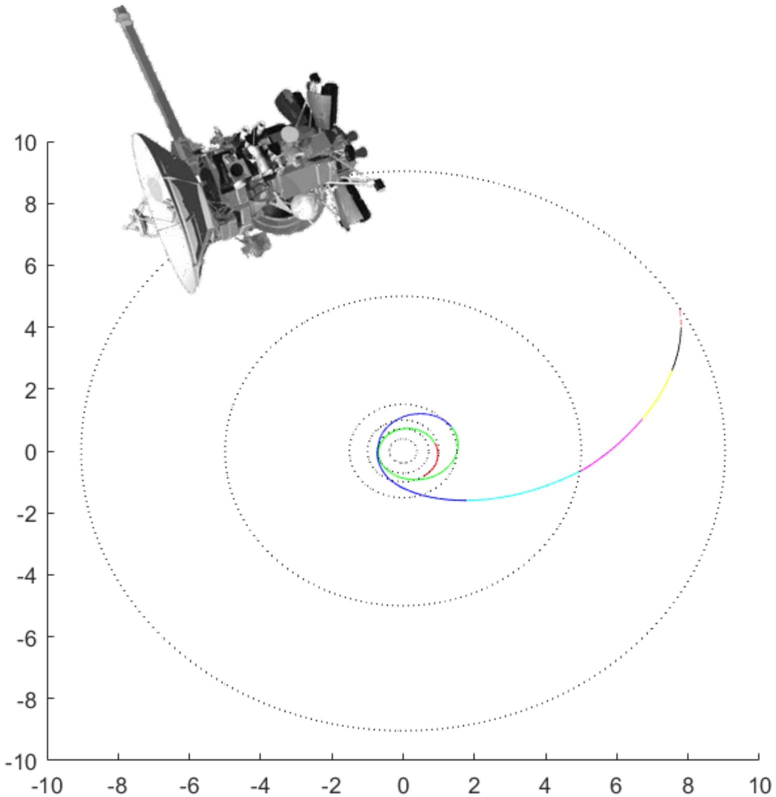
Reference: Chapter 7
Difficulty: III (Challenging)

Exercise

1. Load the file `cassiniData2.mat` and extract the columns Year, Month, Day, X, Y, and Z (columns 1-6, respectively) into separate variables.
2. Using X, Y, and Z, compute the radius.
3. The orbit radii of the 8* planets are given in the following table. Determine the dates on which the craft crossed each planet's orbit.

Planet	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Orbit Radius (au)	0.39	0.72	1.0	1.51	5.0	9.04	19.18	30.06

4. Plot the X-Y position of the spacecraft for each year in a different color or style on the same plot.
5. Add the orbits of Mercury, Venus, Earth, Mars, Jupiter, and Saturn.



* According to the 2006 IAU definition of a planet

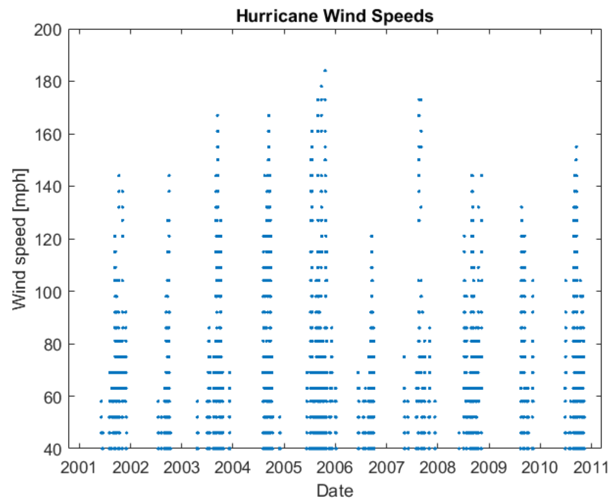
Atlantic Hurricanes

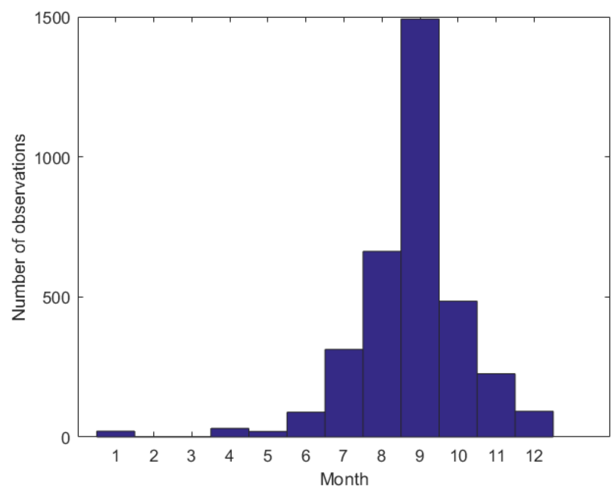
Solution
`hurricanes.mlx`

Reference: Chapter 8
Difficulty: I (Fundamental)

Exercise

1. Load the data in `hurricaneData.mat`. This file contains two variables: `Dates` (as a `datetime` variable) and `Windspeed` (in miles per hour). The data represents observations of Atlantic storms. Observations are made every 6 hours of storms where the wind speed is at least 39 mph (the official definition of a "tropical storm").
2. Create a plot of the wind speeds through time, marked with points.
3. Sort the data by wind speed (descending) using the `sort` function. Find when the top 10 wind speeds were recorded. (**Hint** The `sort` function can return the original index as an optional output.)
4. Use an appropriate function to extract the month (as a number 1 – 12) for each observation.
5. Use this vector to make a histogram of the number of observations in each month using the `histogram` function. Use the months (1 – 12) as the bins for the histogram by using an optional input to `histogram` to set the `BinMethod` property to `integers`.





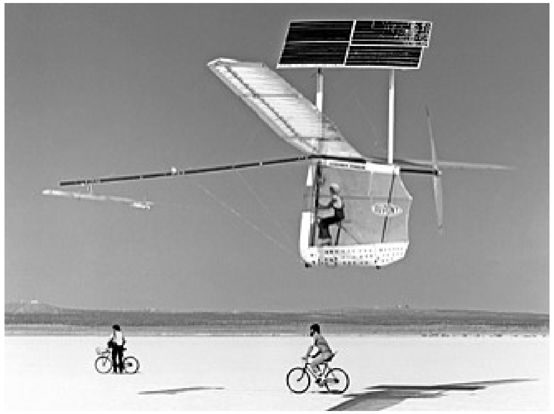
Wing Loading

Solution
`wingLoad.mlx`

Reference: Chapter 9
Difficulty: I (Fundamental)

Exercise

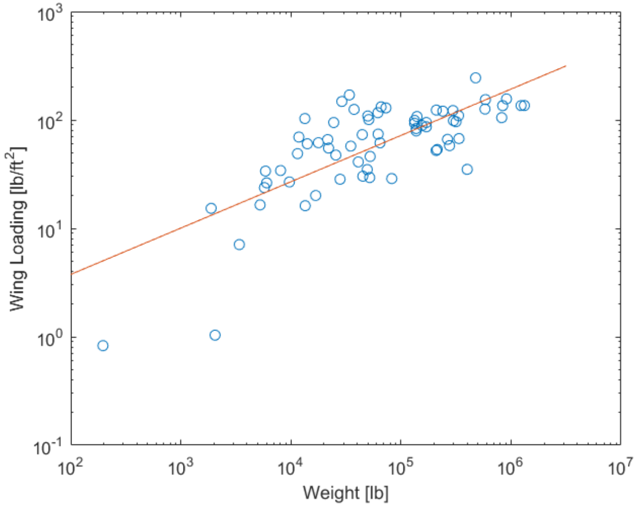
1. Load the data in `WingLoading.mat` into the MATLAB workspace. This file contains the wing loading, `wload`, in lb/ft², and the maximum takeoff weight, `mtow`, in lb, for 68 aircraft.
2. Make a scatter plot of `wload` against `mtow`. Make a second scatter plot using logarithmic scales on both axes. Note that a linear relationship, which wasn't apparent with linear axes, appears using the logarithmic axes.
3. What happens if you try to use the Basic Fitting Tool to determine the linear relationship on the logarithmic plot? Why?
4. Create new variables, `wload_log` and `mtow_log`, containing the logarithms (base 10) of `wload` and `mtow`, respectively. Use `polyfit` to determine the regression coefficients for a linear fit.
5. Create a vector of theoretical (logarithmic) weights. Use `polyval` to calculate the corresponding (logarithmic) wing loadings.
6. Use exponentials to transform the theoretical logarithmic values back to the original scale.
7. Plot the raw data and the fitted model on log-log axes.



The Gossamer Albatross II represents an aircraft with one of the lowest values of wing loading ever flown.

The original Gossamer Albatross was a human-powered aircraft built by Dr. Paul MacCready. On June 12, 1979 it successfully completed a crossing of the English Channel.

Author Henk Tennekes, in his book *The Simple Science of Flight: From Insects to Jumbo Jets* explores what he refers to as "The Great Flight Diagram" and demonstrates that all flying objects obey this fundamental relationship between wing loading and weight.



Solar Radiation I

Solution
`solar1.mlx`

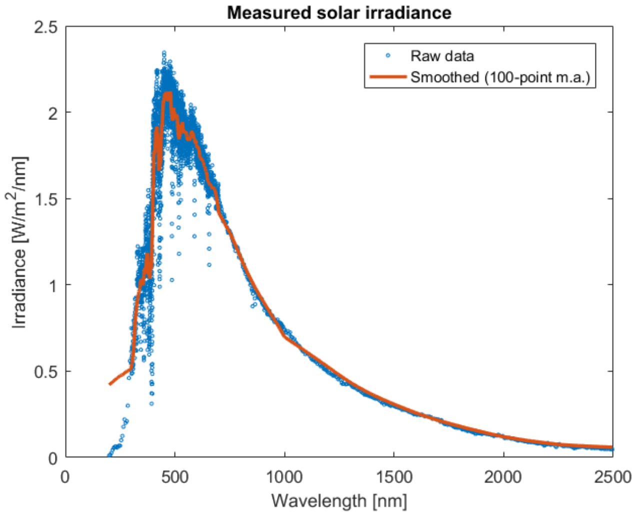
Reference: Chapter 9
Difficulty: I (Fundamental)

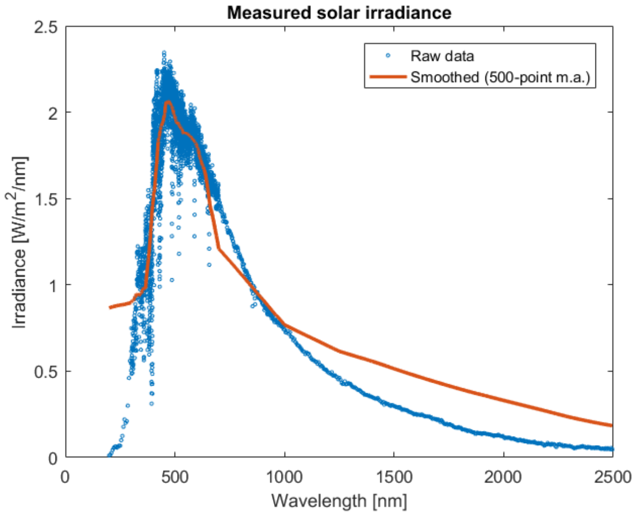
Exercise

1. Load the data in the MAT-file `solar.mat`. This file contains measurements of solar radiation from an airborne observatory. The data consists of vectors of wavelength (`lambda`) in nm and measured irradiance (`irrad`) in W/m^2/nm.
2. Smooth the data by finding the 100-point moving average of the measured irradiance.
3. Plot the raw radiation data with small markers (such as points or circles) and no line. Overlay the moving average with a thick solid line. Add appropriate annotations to your plot.
4. Smooth the data further by finding the 500-point moving average of the measured irradiance. Create the same plot as in step 3, but with the 500-point moving average. Add appropriate annotations to your plot.

Airborne observatories such as NASA's Kuiper Airborne Observatory (KAO) allow researchers to observe celestial bodies without atmospheric interference that can distort images.

NASA's newest airborne observatory Stratospheric Observatory For Infrared Astronomy (SOFIA) will allow scientists to reach altitudes above 40,000 feet to study the universe in the infrared spectrum.





Solar Radiation II

Solution
`solar2.mlx`

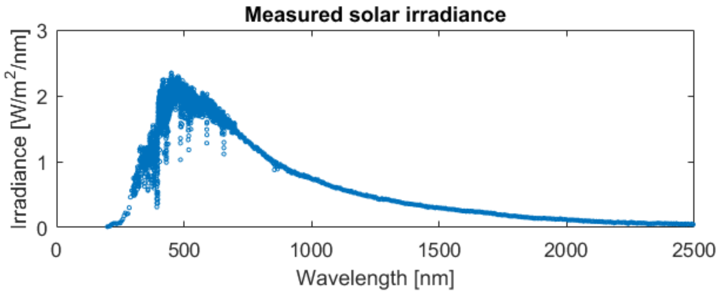
Reference: Chapter 9
Difficulty: II (Intermediate)

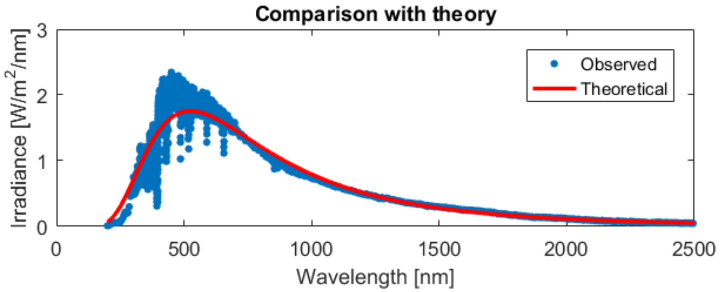
Exercise

1. Load the data in the MAT-file `solar.mat`. This file contains measurements of solar radiation from an airborne observatory. The data consists of vectors of wavelength (`lambda`) in nm and measured irradiance (`irrad`) in W/m^2/nm.
2. Plot the raw radiation data with small markers (such as points or circles) and no line. Add appropriate annotations to your plot.
3. Create a vector of wavelengths from 200 to 2500 nm, in (equally spaced) 1 nm increments.
4. Calculate the theoretical irradiance spectrum using Planck's Law

$$u(\lambda) = \frac{\alpha}{\lambda^5\left(e^{\beta/\lambda} - 1\right)}$$

   where $\alpha = 9.9214 \times 10^{15}$ W·nm^4/m^2, $\beta = 2604.1$ nm, and λ is wavelength (in nm).
5. Create a second plot of the raw radiation data with small markers and no line. Overlay the theoretical spectrum calculated in the previous step as a thick red line. Add appropriate annotations to your plot.
6. The solar constant is the total energy flux (in W/m^2) incident on the earth. This can be calculated by integrating the irradiance spectrum. Because the spectrum in step 4 was calculated for wavelengths in 1 nm increments, the integral can be approximated by simply adding all the irradiance values. Compute the solar constant and compare with the standard measured value of 1.37 kW/m^2. (`http://en.wikipedia.org/wiki/Sunlight#Solar_constant`)





SOFIA will be the largest airborne observatory in the world. It will allow researchers to make observations that are impossible for even the largest and highest of ground-based telescopes.

NASA has modified a Boeing® 747-SP aircraft and installed a 2.5 meter reflecting telescope within the fuselage.

UAV Capabilities III

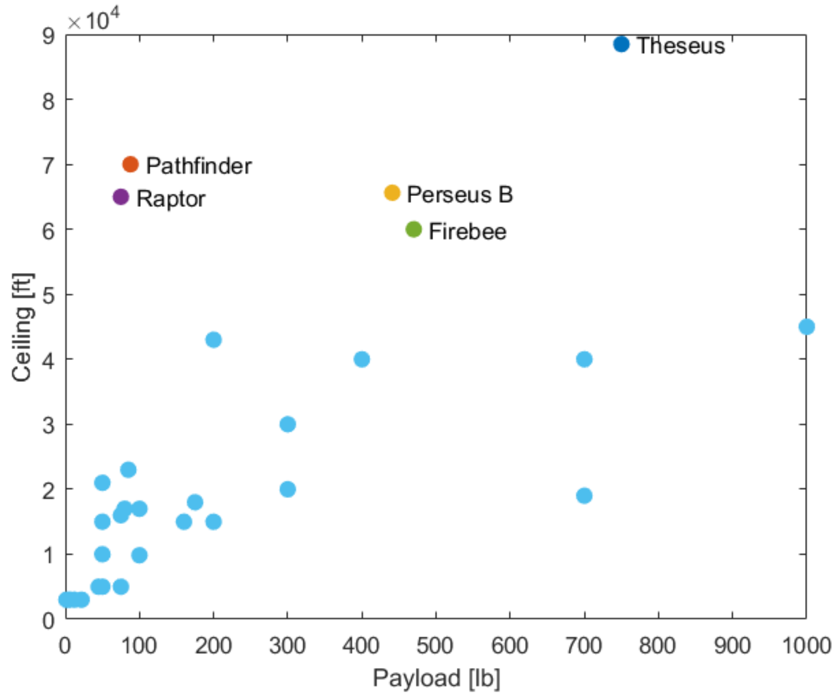
Solution
`UAV3.mlx`

Reference: Chapter 10
Difficulty: I (Fundamental)

Exercise

The file `UAVData.xlsx` contains four columns of data on 33 unmanned aerial vehicles (UAVs). The first column contains the names of the UAVs; the next three columns contain numeric data: endurance (hours), payload (pounds), and operational altitude ceiling (feet), respectively.

1. Use the `readtable` command to import the data as a table with four variables.
2. Sort the UAVs according to ceiling.
3. Plot the ceiling as a function of payload with large dots as markers. Create the plot such that the top five UAVs by ceiling are each a different color and labeled by name. Markers for all other UAVs are the same color.



Boeing® 747® Wing Area

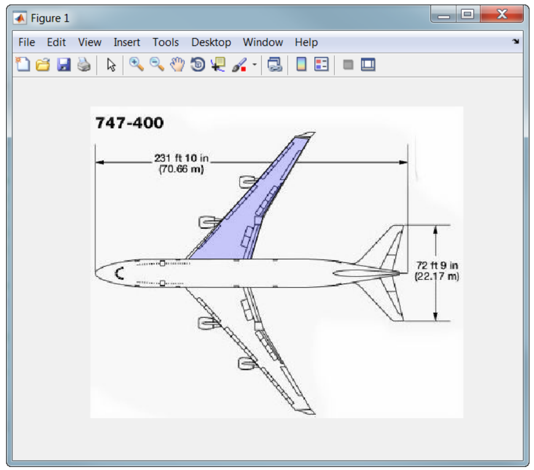
Reference: Chapter 10
Difficulty: II (Intermediate)

Template
`wingArea_template.mlx`

Solution
`wingArea.mlx`

Exercise

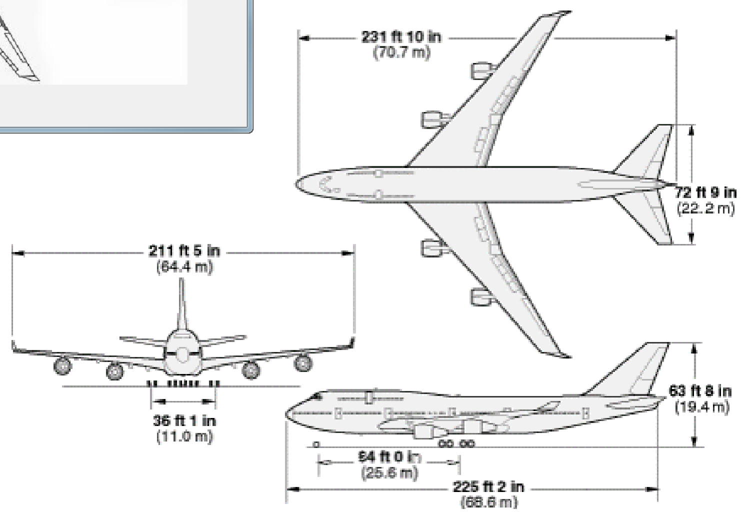
1. Open the file `wingArea_template.mlx` for editing. This script contains the commands to load and display the image file `747.jpg`.
2. Using the `msgbox` and `waitfor` commands, prompt the user to select six points around the edge of the wing. The sixth point does not need to coincide with the first point.
3. Use `ginput` to obtain the bounding coordinates of the wing from the image and save them to the variables `x` and `y`. (The `patch` command will add a colored polygon to the image to show the area selected.)
4. Use `polyarea` to determine the area of the region enclosed by the coordinates. This will yield an area in *image units* ("pixels²").
5. Convert the wing area from pixels² to ft² using the conversion factor of 1.45 pixels/ft. Multiply the final result by 2 to get the total wing area for the aircraft.
6. Report the final estimated wing area back to the user.



The first 747®, the 747-100, rolled out of the Seattle works on September 30, 1968, and flew on February 9, 1969. The first commercial flight was with Pan Am between London and New York on January 22, 1970.

The National Aeronautics and Space Administration later modified two 747-100s into Shuttle Carrier Aircraft.

The 747-400 rolled out in 1988. Its wingspan is 212 feet, and it has 6-foot-high "winglets" on the wing tips.



Escape Velocity I

Solution
escapeVelocity1.mlx

Reference: Chapter 11
Difficulty: I (Fundamental)

Exercise

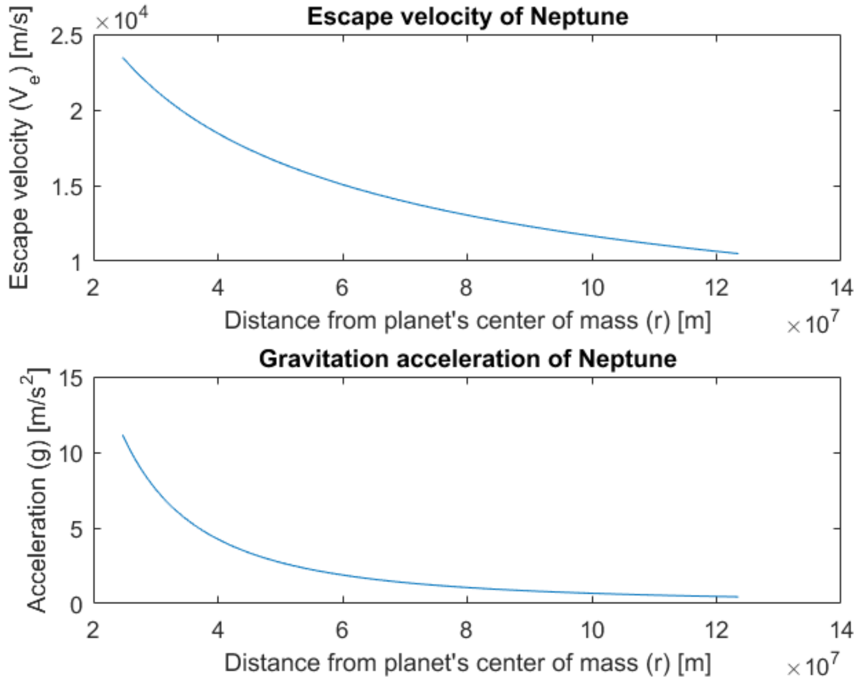
The escape velocity for a body a distance r from the center of mass of a planet of mass M is

$$V_e = \sqrt{\frac{2GM}{r}}$$

where G is the gravitational constant (G = 6.67300 × 10^{-11} $m^3/kg.s^2$). The acceleration due to gravity experienced by the body is

$$g = \frac{GM}{r^2}$$

1. Write a live script that sets the values of M and R for Neptune: $M = 1.02 \times 10^{26}$ kg, $R = 2.47 \times 10^7$ m.
2. Create a local function to compute V_e and g. The function should accept two inputs: the mass M of the planet, and the radius R of the planet (both numeric scalars). The function should return three numeric vectors: a vector of distances (r), and the corresponding values of V_e and g.
3. In the function, create a vector r of radius values from the given planetary radius R to $5R$. Then calculate V_e and g according to the above functions.
4. Call your function using the values of M and R for Neptune.
5. Plot the escape velocity and acceleration due to gravity for Neptune as a function of the distance from Neptune's center of mass. Annotate your plots.



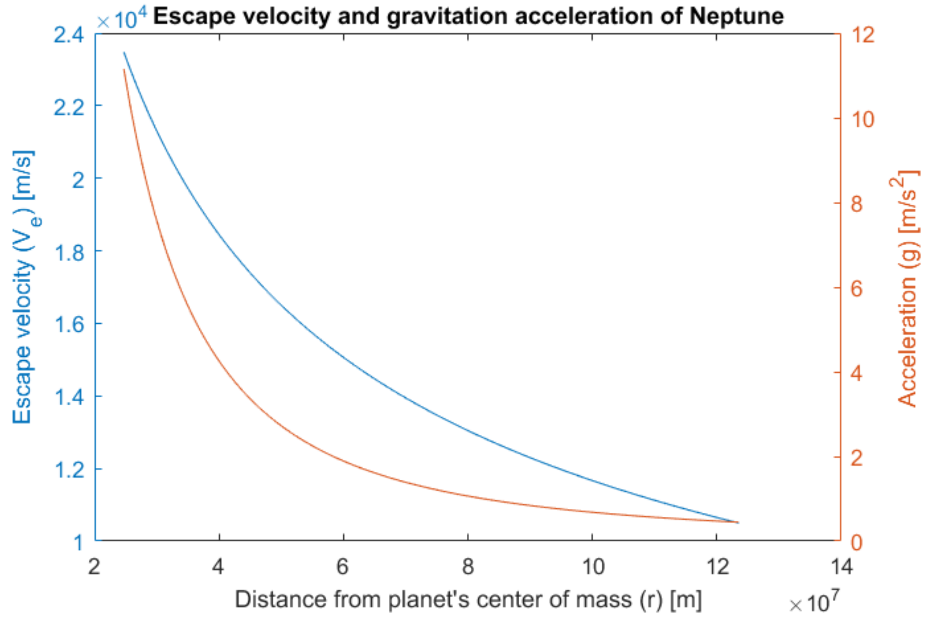
Escape Velocity II

Solution
`escapeVelocity2.mlx`

Reference: Chapter 11
Difficulty: II (Intermediate)

Exercise

1. Open your solution to the "Escape Velocity I" exercise, or the provided solution `escapeVelocity1.mlx`.
2. Modify your function to accept a character vector containing the name of the planet, in addition to the mass and radius.
3. Include the visualization commands in your function (instead of the calling script). Plot V_e and g together on a plot with two y-axes using the command `yyaxis`. Use `yyaxis left` to plot escape velocity on the left y-axis, and `yyaxis right` to plot gravitational acceleration on the right y-axis. Annotate your plot. Add the name of the planet to the title of the plot.



In total, your local function should:

- Create the vector of distances.
- Calculate V_e and g.
- Plot the results using `yyaxis`.
- Label the plot with the planet's name.
- Return the distance, escape velocity, and gravitational acceleration vectors.

Escape Velocity III

Reference: Chapter 11
Difficulty: II (Intermediate)

> Solution
> **`veandg.m`**
> **`escapeVelocity3.mlx`**

Exercise

1. Open your solution to the "Escape Velocity II" exercise, or the provided solution `escapeVelocity2.mlx`. Copy your local function into a plain text code file in the MATLAB Editor and save the file. (Don't forget to change the name of the function in the declaration line to match the file name.)
2. Modify the function to accept a single input: a structure containing the name of the planet, the mass, and the radius.
3. Write a script that creates a structure array containing the data for the planets in the table below. Use a `for`-loop to plot the escape velocity and gravitational acceleration curves for each planet.

Planet	Mass [kg]	Radius [m]
Mercury	3.30×10^{23}	2.44×10^{6}
Mars	6.42×10^{23}	3.40×10^{6}
Jupiter	1.90×10^{27}	7.15×10^{7}
Uranus	8.68×10^{25}	2.56×10^{7}