

ATSC 5010 – Physical Meteorology I Lab

Lab 2 – numerical integration; height/pressure relationships

→ Begin with IDEAL GAS Law:

$$PV = mR_X T \quad \text{or} \quad P\alpha = R_X T$$

note: $\frac{1}{\alpha} = \rho$ thus:

$$P = R_X \rho T$$

This is true for a mixture of gases, for dry air the mixing ratios of the constituent gases is relatively constant:

$$P_d = R_d \rho_d T \quad \text{or} \quad \rho_d = \frac{P_d}{R_d T}$$

→ For a gas in hydrostatic balance, the force exerted downward due to gravity is exactly equal and opposite to the force exerted due to the pressure gradient (for a given volume of gas the pressure below the volume is greater than the pressure above the volume, this exerts a force on the volume in the upward direction). This can be expressed:

$$-dp = g\rho dz \quad \text{or} \quad \frac{dp}{\rho} = -gdz$$

assuming an atmosphere devoid of water vapor:

$$\frac{dp_d}{\rho_d} = -gdz$$

From the ideal gas law, substitute for dry - air density:

$$-gdz = R_d T \frac{dp_d}{p_d} \quad \text{integrate both sides:}$$

$$\int -gdz = R_d \int T \frac{dp_d}{p_d} \quad \text{if we integrate from some initial height } z_0 \text{ to some height } z \text{ we have:}$$

$$\int_{z_0}^z -g dz = R_d \int_{P(z_0)}^{P(z)} T \frac{dp_d}{P_d}$$

over the altitudes of the troposphere, the variation in gravity is quite small.

Thus, we replace g with the constant g_0

thus:

$$z - z_0 = \frac{-R_d}{g_0} \int_{P(z_0)}^{P(z)} T \frac{dp_d}{P_d}$$

At this point, Let's consider two models-

(A) the isothermal model (temperature is constant) with height over the depth of the atmosphere, in this case T can be replaced by T_0 , moved outside of the integral and the solution is simple:

$$z = z_0 + \frac{R_d T_0}{g_0} \ln \left(\frac{P(z_0)}{P(z)} \right)$$

(B) the explicit model (temperature varies with height)

here, we need a measure of temperature (from an atmospheric sounding) and we can then integrate over each level in the atmosphere (numerically) to calculate the height of that level. We will use the mean temperature between two discrete levels as the temperature of that level. If we let z_1 denote the height of our first level, then the height of our next level up, z_2 is given by:

$$Z_2 = Z_1 + \left(\frac{R_d}{g_0} \cdot \overline{T}_{12} \cdot \frac{1}{P_{d12}} \cdot (p_{d1} - p_{d2}) \right)$$

→ We will use pressure and temperature data from a sounding. Knowing the original height, we will compute the height of each measurement using both models. We will compare our computed heights with each other and the actual (measured) height from the sounding.

Lab 3 Exercise:

all computations should be double precision! LABEL YOUR PLOT AXES

Sounding data:

pressure in millibars

temperature in Celsius

height in meters

Computations require appropriate units: temperature in Kelvin, pressure in Pascal....

You will also be using two files that are available for download from the class website:

the idl function file 'read_sounding.pro'

the text file 'sounding_lab2.txt'

Your program should also be generic in sizing, that is, it should not depend on the number of measurements that are contained in the sounding.

1. Create a procedure 'atsc5010_yourname_lab2.pro'.
2. In the first line of your procedure, run your procedure *phycon* (this sets values of constants as environmental variables). Through the remainder of your procedure use these environmental variables rather than 'hardwiring' values of constants.
3. Call 'read_sounding' from within your procedure and extract pressure, temperature, and height data. 'read_sounding' is a function, the usage is on the next page.
4. Assuming an isothermal atmosphere, in which the temperature at all levels is equal to the temperature at the lowest level ($T = T_0$), calculate the height at each pressure level.
5. Assuming the explicit model, using temperature and pressure at all levels, calculate the height at each pressure level. Note that there are 100 levels. The height of the first level is given, so you should compute 99 thicknesses (ie 99 height levels).
6. **Make a first plot**, Plot the measured pressure (in millibars) on the X-axis from 1000 to 0 and calculated height from the isothermal model (*in kilometers*) on the Y-axis, use a solid red line
7. On your plot in (6), overplot the measured pressure and calculated height from the explicit model, using a dashed blue line
8. **Make a second plot** that shows computed height (isothermal model) as a function of actual (measured) height, solid red line. Plot the heights in kilometers. Overplot the computed height (explicit model) as a function of actual height, dashed blue line, thickness=2.
9. **Make a third plot** that shows the difference between the computed and actual height (in meters, *computed minus actual*) as a function of actual height (*in kilometers*). Use a solid red line for the isothermal model and a dashed blue line for the explicit model.

Questions

1. Comparing the two models, at the top of the sounding (lowest pressure), which has the greatest calculated height?
2. Consider the assumptions used in the two models and the actual temperature as a function of altitude, explain the difference in calculated height between the two models. Why does temperature matter?
3. Does the difference between the two models increase, decrease, or stay about the same as altitude increases? Why?

Functions & Procedures to use:

Supplied by the lab instructor

'read_sounding':

USAGE: *result = read_sounding([temperature=temperature], [height=height])*

Upon calling, read_sounding prompts the user to select a sounding.txt file to read

Result: named variable that returns the one-dimensional array of pressure measurements in millibars

Keywords:

Temperature is named variable that upon exit contains the one dimensional array of temperature measurements (in celcius)

Height is named variable that upon exit contains the one dimensional array of height measurements (in meters)

Built in IDL procedures & functions

DBLARR – for creating an array of Doubles

N_ELEMENTS – for determining how many elements are in an array

SHIFT – to shift all elements of a one-dimensional array one direction or another

TOTAL (with keyword CUMULATIVE) – to determine the total values of all elements in an array, in a cumulative sense

PLOT (as we've used in earlier labs...)

Some hints for programming:

npts = N_ELEMENTS(arr1)

arr2 = DBLARR(npts)

arr2 is then an array of doubles and has the same number of elements as arr1

IF

Array1 = [4, 6, 9, 3, 2]

and

Array2 = SHIFT(Array1, 1) = [2, 4, 6, 9, 3]

Array1 and Array2 look the same, but the elements are shifted by 1

then

Array3 = Array1 – Array2 = [4-2, 6-4, 9-6, 3-9, 2-3]

Note if we dispose of the zeroth element then

Array3[1] = array1[1]-array1[0]

Array3[2] = array1[2]-array1[1]

Array3[3] = array1[3]-array1[2] etc.....

This is similar to the statement FOR i=0,3 DO array3[i] = array1[i+1]-array[i] (but the above does not use a FOR loop and is much faster in IDL (and easier to read and debug)

IF

$X = [1, 2, 3, 2, 3, 5]$

then

$TOTAL(x) = 16$ (the total)

and

$TOTAL(x, /cumulative) = [1, 3, 6, 8, 11, 16]$ (the cumulative summation)