Assignment 1 - Andrew Loeppky

Create contour plots of concentration (µg/m^3) using Stull eq.(19.20) with eqs. (19.13) within the domain for:

- a) horizontal (x,y) slice at earth's surface (z = 0)
- b) horizontal (x,y) slice at height of plume centerline (z = zcl).
- c) vertical along-wind (x,z) slice of thru plume centerline (y = 0)
- d) vertical crosswind (y,z) slice at a downwind distance x from the stack

Hints:

- Use any computer language; e.g., R, MatLab, python, fortran, excel. (I used R.)
- Use a contour interval of 20 µg/m^3.
- Check that your answer to part (a) is similar to the sample application (Solved Example) in Stull p734 before you do the other parts. (It won't be exactly equal, because of different inputs in this HW assignment.)
- Turn in your contour plots AND your code.
- Please have your name on everything you turn in.

In [118…

```
# Gaussian Plume - simple
# R. Stull, 8 Feb 2016, Modified 4 Jul 2018.

# Givens ==============
# meteorology
m = 20.           # wind speed (m/s)
zi = 2000.        # mixed layer depth (m)
sigma_v = 1.3     # lateral velocity variance (m/s)
sigma_w = 1.02    # vertical velocity variance (m/s)
tl = 60.          # Lagrangian time scale (s)

# plume
zcl = 150.        # plume centerline height (m)
q = 300.          # emission rate (g/s)
q_ug = q * 1e6    # convert emission to micrograms

# domain, assuming origin (x, y, z) = (0,0,0) is a base of smoke stack
xmax = 10000    # x downwind domain size for x = 0 to xmax (m)
ymax = 500      # y crosswind domain size for y = -ymax to +ymax (m)
zmax = 500      # z vertical domain size for z = 0 to zmax (m)

# spatial resolution for calculations of concentration
delx = 200      # x increment (m)
dely = 20       # y increment (m)
delz = 20       # z increment(m)

xslice = 3000.   # x-location of crosswind (y,z) slice (m)
```

In [2]:

```
import xarray as xr
import numpy as np
import matplotlib.pyplot as plt
```

Gaussian Plume Equations from PrMet

$$\sigma_y^2 = 2\sigma_v^2 t_L^2 \cdot \left[\frac{x}{Mt_L} - 1 + exp\left(-\frac{x}{Mt_L} \right) \right] \qquad (19.13a)$$

$$\sigma_z^2 = 2\sigma_w^2 t_L^2 \left[\frac{x}{Mt_L} - 1 + exp\left(-\frac{x}{Mt_L} \right) \right] \qquad (19.13b)$$

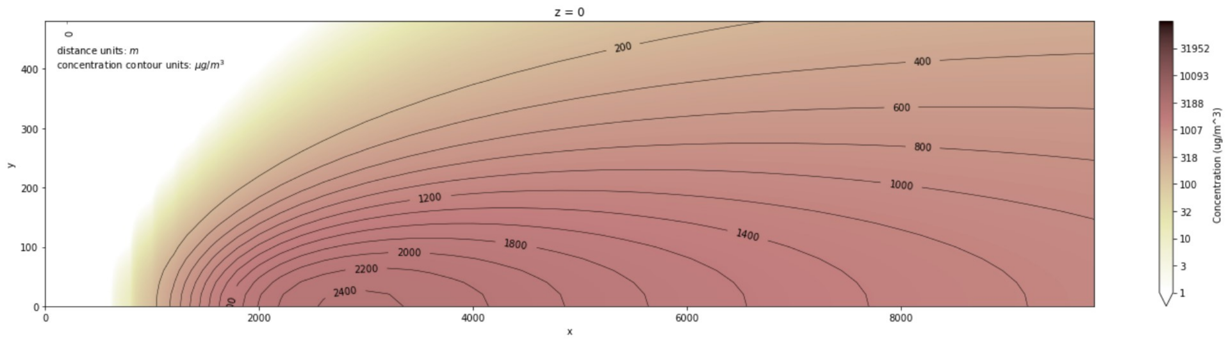
$$c = \frac{Q}{2\pi\sigma_y\sigma_z M} \cdot exp\left[-0.5 \cdot \left(\frac{y}{\sigma_y} \right)^2 \right] \left\{ exp\left[-0.5 \cdot \left(\frac{z - z_{CL}}{\sigma_z} \right)^2 \right] + exp\left[-0.5 \cdot \left(\right. \right. \right.$$

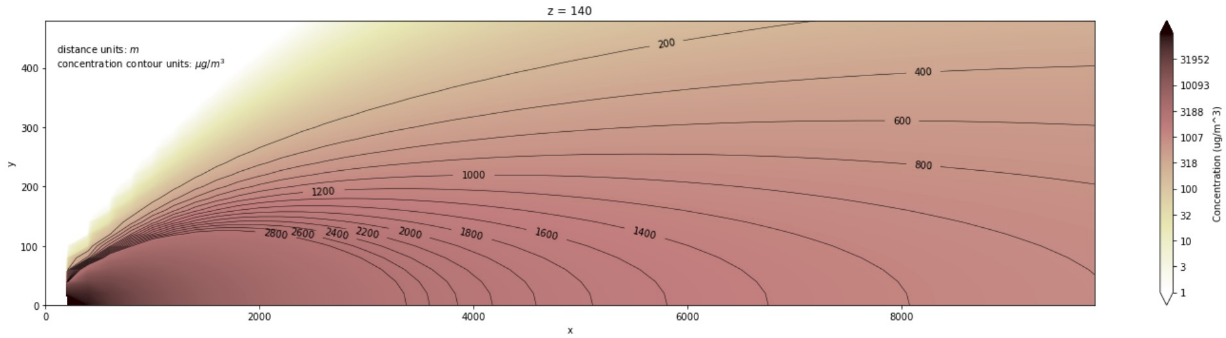
In [67]:

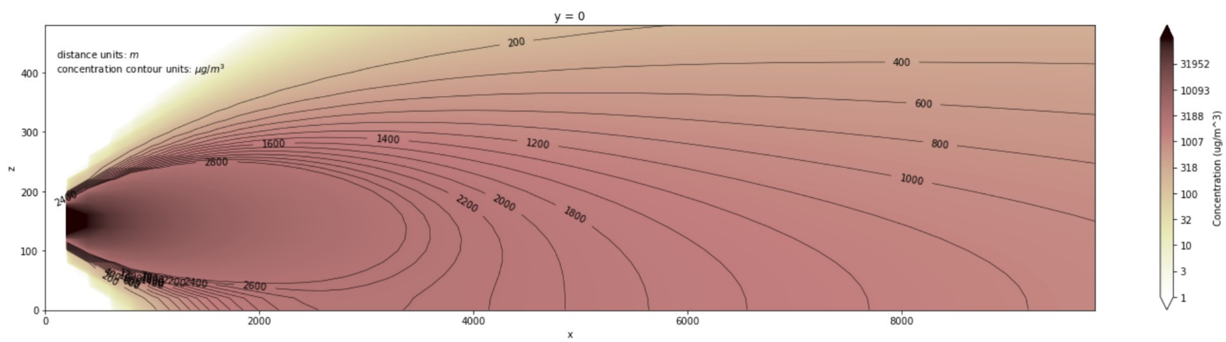
```
##### CALCULATION #####

# create an empty concentration array with specified resolution
c = xr.DataArray(np.zeros([xmax // delx, ymax // dely, zmax // delz]),
                 coords={"x":np.arange(0,xmax, delx),
                         "y":np.arange(0,ymax, dely),
                         "z":np.arange(0,zmax, delz)},
                 dims=["x","y","z"],
                 name="Concentration (ug/m^3)")

# calculate sigma in y and z dimensions
sigma_y = (2 * sigma_v ** 2 * tl ** 2 \
           * ( (c["x"]) / (m * tl) - 1 + np.exp(-(c["x"]) / (m * tl)) )) ** 0
sigma_z = (2 * sigma_w ** 2 * tl ** 2 \
           * ( (c["x"]) / (m * tl) - 1 + np.exp(-(c["x"]) / (m * tl)) )) ** 0


# fill the concentration DataArray with values from 19.20
c += q_ug / (2 * np.pi * sigma_y * sigma_z) \
      * np.exp(-0.5 * (c["y"] / sigma_y) ** 2) \
      * (np.exp(-0.5 * ((c["z"] - zcl) / sigma_z) ** 2) \
         + np.exp(-0.5 * ((c["z"] + zcl) / sigma_z) ** 2))
```

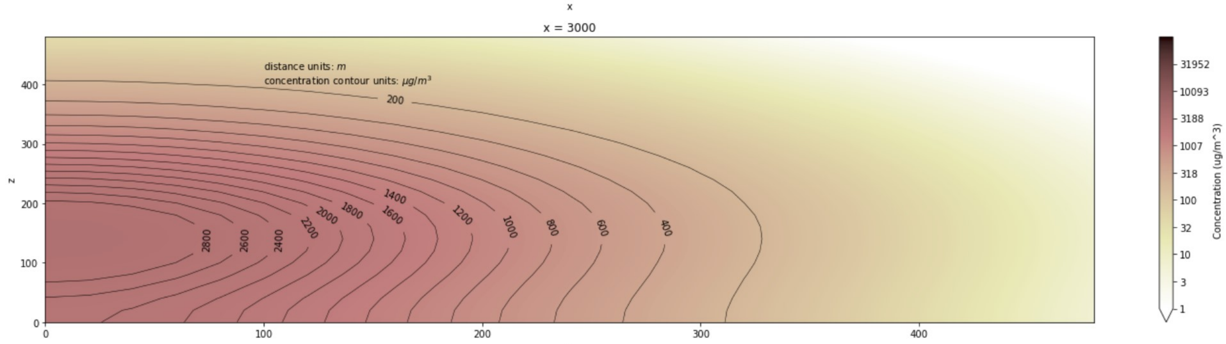
In [122…

```
##### PLOTTING #####

# apply the same keyword arguments to each plot, including a common scaled co

# kwargs for contour plot
con_kwargs = {
    "levels": np.arange(0, 3000, 200),
    "linewidths":0.5,
    "cmap": "black",
}

# kwargs for background fill
fill_kwargs = {
    "add_colorbar": True,
    "levels":np.logspace(0, 5, 1000),
    #"levels":np.arange(0, 20000, 10),
    "linewidths":0.5,
    #"antialiased": True,
    "cmap": "pink_r",
}

# do the spefified plots on a single figure
fig, ax = plt.subplots(4, figsize=(20, 20))

# a) horizontal (x,y) slice at earth's surface (z = 0)
slice1 = c.isel(z=0).T.plot.contour(ax=ax[0], **con_kwargs)
c.isel(z=0).T.plot.contourf(ax=ax[0], **fill_kwargs)
ax[0].clabel(slice1, slice1.levels, inline=True, fontsize=10)

# b) horizontal (x,y) slice at height of plume centerline (z = zcl).
slice2 = c.isel(z=int(zcl / delz)).T.plot.contour(ax=ax[1], **con_kwargs)
c.isel(z=int(zcl / delz)).T.plot.contourf(ax=ax[1], **fill_kwargs)
ax[1].clabel(slice2, slice2.levels, inline=True, fontsize=10)

# c) vertical along-wind (x,z) slice of thru plume centerline (y = 0)
slice3 = c.isel(y=0).T.plot.contour(ax=ax[2], **con_kwargs)
c.isel(y=0).T.plot.contourf(ax=ax[2], **fill_kwargs)
ax[2].clabel(slice3, slice3.levels, inline=True, fontsize=10)

# d) vertical crosswind (y,z) slice at a downwind distance x - xslice = 3000.
slice4 = c.isel(x=int(xslice / delx)).T.plot.contour(ax=ax[3], **con_kwargs)
c.isel(x=int(xslice / delx)).T.plot.contourf(ax=ax[3], **fill_kwargs)
ax[3].clabel(slice4, slice4.levels, inline=True, fontsize=10)

fig.tight_layout()
for i in [0,1,2,3]:
    ax[i].annotate("distance units: $m$\nconcentration contour units: $\mu g/r
```


z = 0
distance units: m
concentration contour units: $\mu g/m^3$
200
400
600
800
1000
1200
1400
1800
2000
2200
2400
x
y
Concentration (ug/m^3)
31952
10093
3188
1007
318
100
32
10
3
1


z = 140
distance units: m
concentration contour units: $\mu g/m^3$
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
x
y
Concentration (ug/m^3)


y = 0
distance units: m
concentration contour units: $\mu g/m^3$
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
x
z
Concentration (ug/m^3)


x = 3000
distance units: m
concentration contour units: $\mu g/m^3$
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
y
z
Concentration (ug/m^3)