**Model output coding:**

* 15 files from Dec1990-Feb1992
* m01s02i162 = Met Office Unified Model ID – 162 refers to extinction at 550nm.
* E.G. Dec1990 – Each file has 3 lots of cubes with “time: 80” and 1360. 80 refers to outputs per day (3 hourly measurements) for 10 days of the month.
* Concatenated – these cubes have “**time: 240**” – as outputs are then for 8 x 30 days (30 days in model months).
* **1360** refers to **16 sites** (lidar sites based on SPARC (?) report) **x 85 model levels**.
* When loading one file – i.e. Dec1990, once concatenated if you slice e.g. “Dec\_1990\_Extinction [239]” that gives the last date of that file.
* **3600** = **15\*240** – applies for when ALL files loaded.

**Output all dates for individual sites:**

* Set level\_number\_coord = icoords.DimCoord(range(1,86), standard\_name=’model\_level\_number’) = making a model level number coord.
* Load data – in this case all data from all files (i.e. all data from Dec1990-Feb1992).
* Loads them all in as 3 cases each – i.e. each case with 10 days.
* Concatenate\_cube() – makes each case for each month.
* Dir\_out = directory for all site cases to be saved.
* Sets n=0

For I in range (0,16):

Site = extinction[:,n:n+85] = sets site 1 to 1-85, 2 to 86-170 etc.

Site.remove\_coord(‘model\_level\_number’)

Site.remove\_coord(‘level\_height’)

Site.remove\_coord(‘sigma’)

Site.add\_dim\_coord(level\_number\_coord, 1)

Site.rename(‘Extinction\_550nm\_site\_’+str(i)) = sets up string for filename

Iris.save(site,dir\_out+site.name()+’.nc’) = saves the file

N+=85 = +85 for the next site

SO – with this code, we can read in every month of model output, concatenate to make a whole month of data and supply model levels. We can then adapt the files before outputting them to a folder for each lidar site.

NOW – aim to use one month’s data.

* Read in the data
* Concatenate to make a cube for the month
* Output X,Y plot for time vs model level height (for now)

Aim is to have “data” as the actual extinction levels.

* When reading straight from model output, if the cube is set up (i.e. concatenated) then ‘name\_of\_cube’.data is how you can output the actual extinction values. E.g. setting Dec\_1990\_Extinction to the concatenated cube, using “Dec\_1990\_Extinction.data” outputs the extinction values for (240, 1360) = (time, height at each site).
* Setting data to “Dec\_1990\_Extinction.data” makes plotting easier.
* Setup = plt.contourf(x,y,data)
* X and y need to have x = range(0,’time’) and y = list(level\_number\_coord.points).
* When looking at the RAW data – need to use slicing for “data” e.g. plt.contourf(x,y,np.transpose(data[:,0:85])) – specify first “:” for ALL timesteps, and 0:85 for first site.

**IF USING NETCDF FILES:**

* E.g. new\_data = iris.load\_cube(‘path\_to\_.nc\_file’)
* No need to specify x and y with qplt
* BUT:
* Plt.figure()
* plt.contourf(range(0,3600),y,np.transpose(new\_data.data)) – plots all data for that site. SO for EACH SITE can plot OVER TIME PERIOD. SLICING will help refine time step. E.g.
* plt.contourf(range(0,3600),y,np.transpose(new\_data.data[0:240]))

**Model heights:**

hybrid\_ht = 20, 53.33334, 100, 160, 233.3333, 320, 420, 533.3334, 659.9999,

799.9999, 953.3337, 1120, 1300, 1493.333, 1700, 1920, 2153.333, 2400,

2659.999, 2933.333, 3220, 3520, 3833.333, 4160, 4500, 4853.333, 5220,

5600, 5993.333, 6400, 6820, 7253.333, 7700, 8160.001, 8633.34, 9120.007,

9620.02, 10133.37, 10660.08, 11200.16, 11753.64, 12320.55, 12900.93,

13494.88, 14102.48, 14723.88, 15359.24, 16008.82, 16672.9, 17351.9,

18046.29, 18756.7, 19483.89, 20228.78, 20992.53, 21776.51, 22582.39,

23412.16, 24268.18, 25153.22, 26070.59, 27024.11, 28018.26, 29058.23,

30150.02, 31300.54, 32517.71, 33810.59, 35189.52, 36666.24, 38254.03,

39967.93, 41824.85, 43843.83, 46046.21, 48455.83, 51099.35, 54006.43,

57210.02, 60746.7, 64656.96, 68985.52, 73781.77, 79100.02, 85000

**PLOTTING:**

Plt.contourf(x,y,np.transpose(MLO\_data.data))

EASIER = data = np.transpose(MLO\_data.data)

**SLICING:**

e.g. Dec1990 = data[0:72,0:240]

NOW have model data, with correct heights and ways to slice for different heights and times.

TO DO: convert from 0,3600 time to real time and work up obs data.

**Converting observational data to 2D array to use within Python:**

* e.g. np.array(zeros,4,4)
* At the moment model data is read in by iris as a cube
* Neely help with Pandas – create dataframe of time vs height (saved in iPython notebook).

**TO DO THIS WEEK (17-21ST JULY):**

* Sort timescale bar for model plots.
* Use code for model plots to create same plot as lidar plot.
* Sort coding for Jan 1991-June 1991 – different altitude dataset for different times?

3600