notebook

December 16, 2020

1 Plotting maps of LMDZ variables using Cartopy : a quick example

1.1 Loading libraries

[1]:
```
import netCDF4
from pylab import *
import numpy as np
import cartopy.crs as ccrs
from cartopy.util import add_cyclic_point
```

1.2 Reading the netCDF file

[2]:
```
ncfile="input.nc"
ncgcm = netCDF4.Dataset(ncfile)
# Grid
longi = ncgcm.variables['lon'][:]
lati = ncgcm.variables['lat'][:]
# Variables
SWdnTOA = np.mean(ncgcm.variables['SWdnTOA'][:],axis=0)
SWdnTOAclr = np.mean(ncgcm.variables['SWdnTOAclr'][:],axis=0)
SWupTOA = np.mean(ncgcm.variables['SWupTOA'][:],axis=0)
SWupTOAclr = np.mean(ncgcm.variables['SWupTOAclr'][:],axis=0)
topl = np.mean(ncgcm.variables['topl'][:],axis=0)
topl0 = np.mean(ncgcm.variables['topl0'][:],axis=0)
CREsw6A = (SWdnTOA - SWupTOA) - (SWdnTOAclr - SWupTOAclr)
CRElw6A = topl0 - topl
```

1.3 Plot user's parameters

[3]:
```
dataplt = CREsw6A+CRElw6A
titleplt = r"Net Cloud Radiative Forcing (W m$^{-2}$) - LMDZ6A"
pngfile = 'CREnet6A.png'
#clevs = np.linspace(-20,80,21) # LW
```

```
#clevs = np.linspace(-120,0,25) # SW
clevs = np.linspace(-80,20,21) # NET
```

1.4 Filling the gap at longitude 0

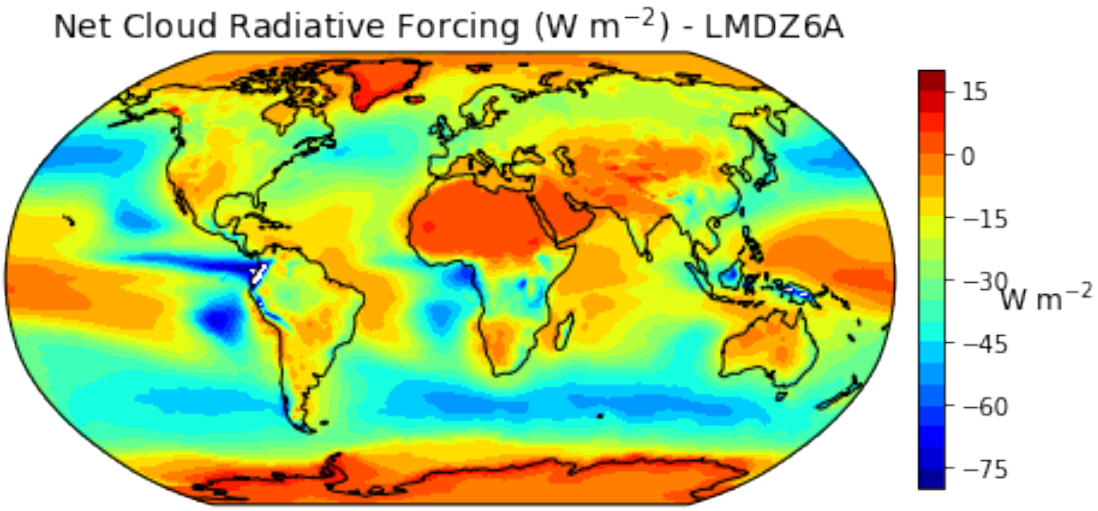
[4]:
```
dataplt, longiplt = add_cyclic_point(dataplt, coord=longi)
```

1.5 2D maps using the Robinson projection

[5]:
```
fig = plt.figure(figsize=(8,4))
ax = plt.axes(projection=ccrs.Robinson())
ax.set_global()
ax.coastlines(resolution="110m",linewidth=1)
plt.contourf(longiplt, lati, dataplt, clevs, transform=ccrs.
 ↪PlateCarree(),cmap=plt.cm.jet)
plt.title(titleplt, size=14)
cb = plt.colorbar(ax=ax, orientation="vertical", pad=0.02, aspect=16, shrink=0.
 ↪8)
cb.set_label(r'W m$^{-2}$',size=12,rotation=0,labelpad=15)
cb.ax.tick_params(labelsize=10)

fig.savefig(pngfile, dpi=200)
plt.show()
```

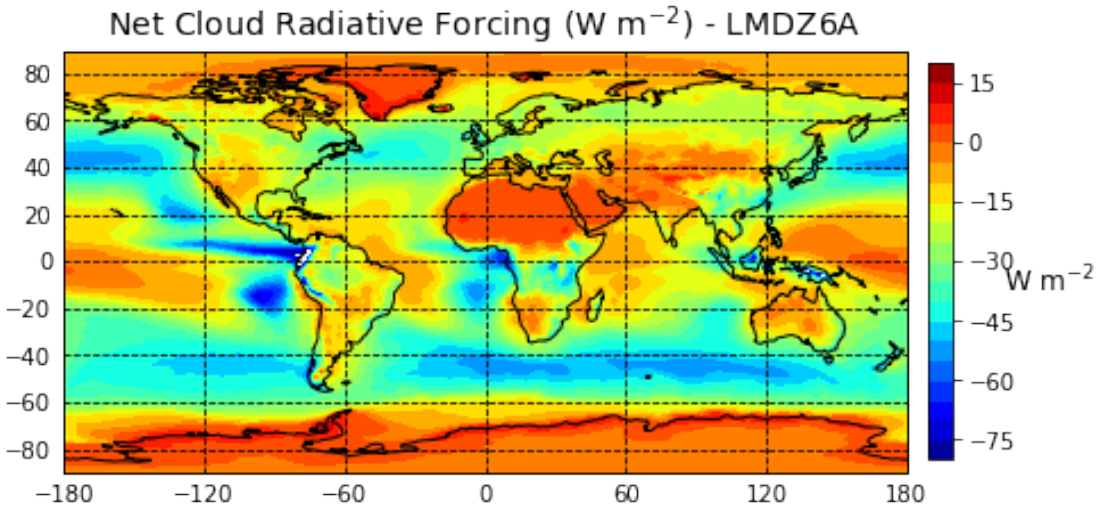
```
/home/jmadeleine/PROGRAMS/miniconda3/envs/python3/lib/python3.7/site-
packages/cartopy/mpl/geoaxes.py:388: MatplotlibDeprecationWarning:
The 'inframe' parameter of draw() was deprecated in Matplotlib 3.3 and will be
removed two minor releases later. Use Axes.redraw_in_frame() instead. If any
parameter follows 'inframe', they should be passed as keyword, not positionally.
  inframe=inframe)
```



1.6 2D maps using a classical equirectangular projection

[6]:
```
fig = plt.figure(figsize=(8,4))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.set_global()
ax.coastlines(resolution="110m",linewidth=1)
gl = ax.gridlines(linestyle='--',color='black',
             draw_labels=True)
gl.xlabels_top = False
gl.ylabels_right = False
plt.contourf(longiplt, lati, dataplt, clevs, transform=ccrs.
 ↪PlateCarree(),cmap=plt.cm.jet)
plt.title(titleplt, size=14)
cb = plt.colorbar(ax=ax, orientation="vertical", pad=0.02, aspect=16, shrink=0.
 ↪8)
cb.set_label(r'W m$^{-2}$',size=12,rotation=0,labelpad=15)
cb.ax.tick_params(labelsize=10)

fig.savefig(pngfile, dpi=200)
plt.show()
```



1.7 Global mean CRF

[7]:
```
area = ncgcm.variables['aire'][:]
totarea = np.sum(np.sum(area[:,:],axis=0),axis=0)
print("Annual mean LW CRF (W/m2) : ",
      np.mean(np.mean(CRElw6A[:,:]*area,axis=0),
              axis=0)/totarea*area.size)
print("Annual mean SW CRF (W/m2) : ",
      np.mean(np.mean(CREsw6A[:,:]*area,axis=0),
              axis=0)/totarea*area.size)
print("Annual mean Net CRF (W/m2) : ",np.mean(np.mean(
    (CRElw6A[:,:]+CREsw6A[:,:])*area,axis=0),axis=0)/
      totarea*area.size)
```

```
Annual mean LW CRF (W/m2) :  24.97136061452329
Annual mean SW CRF (W/m2) :  -48.32627287879586
Annual mean Net CRF (W/m2) :  -23.354914661496878
```