Comment ajouter un fond de carte et faire une animtion

Ajout de fond de carte

En météorologie, on peut vouloir ajouter des informations géographiques sur nos cartes.
Pour cela, une solution est de faire appel à la librairie `cartopy`.

On montrera ici deux exemples :

- Un cas où l'information géographique est présent dans la bibliothèque
- Un cas où l'information provient d'un fichier tiers (fichier de type `geojson`).

Ajouter les frontières et la délimitation Terre/Mer

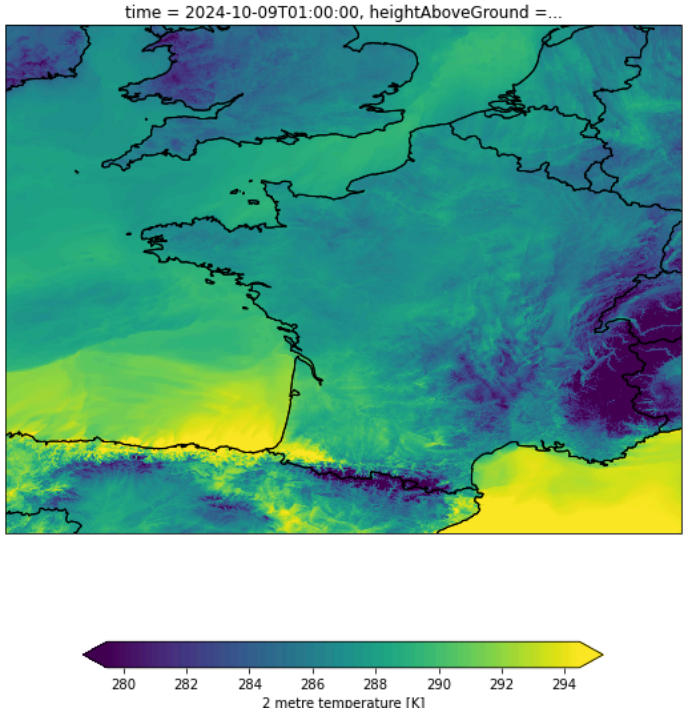
```
import xarray as xr
import cartopy.feature as cfeature
import cartopy.crs as ccrs
import matplotlib.pyplot as plt

fig, axis = plt.subplots(1, 1,
      subplot_kw=dict(projection=ccrs.PlateCarree()),figsize=(10,10))
# Fonction de plot
ds["t2m"].isel(time=0).plot(
    ax=axis,
    transform=ccrs.PlateCarree(),  # this is important!
    #On specifie la geometrie de la carte
    cbar_kwargs={"orientation": "horizontal", "shrink": 0.7},
    robust=True,
)
axis.coastlines() # Ajout du trait de côtes

# Ajout des frontières des pays
border_10m = cfeature.NaturalEarthFeature('cultural',
     'admin_0_countries', '10m',edgecolor='k',facecolor='none')
axis.add_feature(border_10m,linestyle='-', alpha=1)

plt.show()
```

Exemple



Fonction permettant l'ajout de polygones

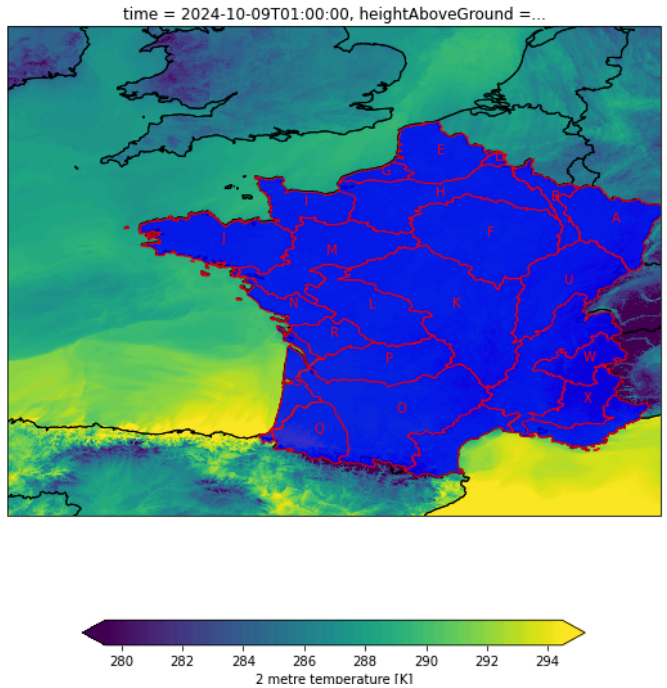
```python
def add_polygons(axis, file:str, subdomain:str=None):
    # Ouverture du fichier json
    import json
    with open(file) as fp:
        data = json.load(fp)
    # Insérer ici une fonctionnalité de tri por selectionner les features de la zone qui nous intéresse
    if subdomain:
        pass
    #
    for elt in data["features"]:
        # convertion du json en une shape (shapely)
        geom = shape(elt["geometry"])
        # Ajout de la shape à la carte.
        # On peut jouer sur la couleur
        axis.add_geometries([geom], crs=ccrs.PlateCarree(),facecolor="b", edgecolor='red', alpha=0.5)
        # On peut ajouter un message (par exemple ici le nom de la zone)
        # Si le centroid est bien dans la zone
        ID = elt["properties"]["ID"]
        centroid = geom.centroid
        if geom.contains(centroid):
            axis.annotate(ID, xy=(centroid.x, centroid.y),color="red")
        else:
            print(f"Centroid not in area for id :{ID}")
```

Exemple d'ajout de polygon

```
FILE = "/home/newton/ienm2021/chabotv/COURS_CS/masks/BASSIN_VERSANT_LVL1.geojson"

fig, axis = plt.subplots(1, 1,
subplot_kw=dict(projection=ccrs.PlateCarree()),figsize=(10,10))
ds["t2m"].isel(time=0).plot(
    ax=axis,
    # On specifie la geometrie de la carte
    transform=ccrs.PlateCarree(),
    cbar_kwargs={"orientation": "horizontal", "shrink": 0.7},
    robust=True,
)
axis.coastlines() # Ajout du trait de côtes
# Ajout des frontières des pays
border_10m = cfeature.NaturalEarthFeature('cultural',
'admin_0_countries', '10m',edgecolor='k',facecolor='none')
axis.add_feature(border_10m,linestyle='-', alpha=1)
# Ajout des polygones contenus dans le fichier
add_polygons(axis,FILE)
plt.show()
```

Exemple

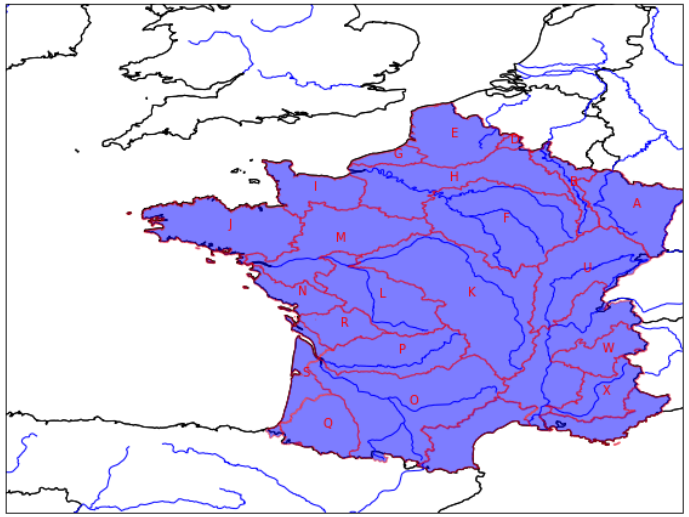


Carte sans donnée météorologique

Visualisation pouvant être intéressante pour présenter une carte de "type risque" (en mettant une couleur selon l'intensité "du risque" par sous-zone).

```
fig, axis = plt.subplots(1, 1,
      subplot_kw=dict(projection=ccrs.PlateCarree()),figsize=(10,10))
# On définit la zone d'interet
#(min_lon, max_lon, min_lat, max_lat)
axis.set_extent([-8,8,41.4,53.4,])
axis.coastlines() # Ajout du trait de côtes
# Ajout des frontières des pays
border_10m = cfeature.NaturalEarthFeature('cultural',
          'admin_0_countries', '10m', edgecolor='k',facecolor='none')
axis.add_feature(border_10m,linestyle='-', alpha=1)
rivers = cfeature.NaturalEarthFeature('physical',
         'rivers_lake_centerlines', '10m',edgecolor='b',facecolor='none')
axis.add_feature(rivers)
add_polygons(axis,FILE)
plt.show()
```

Exemple



Faire une animation

Pour faire une animation on va décomposer le problème en deux étapes :

- Une fonction qui créé des images à un instant donnée (et les sauve dans la RAM)
- une fonction qui crée un gif à partir d'une liste d'image

1. Comment ça s'articule

```
from PIL import Image
from io import BytesIO
OUTPUT_GIF = "my_super_video.gif"

image_list = []
# On va mettre 4 images dans notre gif
for i in range(0,4):
    # On va créé une image a partir d'un dataset
    image_file = make_image(ds,i)
    # On se place au début "du fichier" dans la RAM
    image_file.seek(0)
    # On ajoute l'image à la liste d'image
    image_list.append(Image.open(image_file))

# On créé l'animation à partir de la liste d'image
animation(image_list, OUTPUT_GIF)
```

2. Fonction make_image

```
def make_image(ds,time_sel):
    """
    Crée et sauve une image.
    Cette image est sauvée en RAM
    """
    ds["t2m"].isel(time=time_sel).plot()
    # On crée un espace dans la RAM pour stocker l'image
    image_file=BytesIO()
    plt.savefig(image_file,bbox_inches="tight")
    plt.close()
    return image_file
```

3. Fonction créant l'animation

```
def animation(images:list, gif_filepath:str):
    """
    A partir d'une liste d'Image (PIL) créé une animation
    Cette animation est sauvegardée dans gif_filepath
    """
    nbimages=len(images)
    # create a tuple of display durations, one for each frame
    first_last = 1000 #show the first and last frames for 100 ms
    standard_duration = 1000 #show all other frames for 5 ms
    durations = tuple([first_last] + [standard_duration] * (nbimages - 2) + [first_last])
    # load all the static images into a list
    # save as an animated gif
    gif = images[0]
    gif.info['duration'] = durations #ms per frame
    gif.info['loop'] = 0 #how many times to loop (0=infinite)
    gif.save(fp=gif_filepath, format='gif', save_all=True, append_images=images[1:])
```