ecco_vs_era5_heatflux

June 10, 2022

0.1 Compare ECCO and ERA5 net air-sea heat fluxes

[1]:
```
import numpy as np
import pandas as pd
import xarray as xr
from pathlib import Path
import cmocean
import matplotlib.pyplot as plt
import cartopy
import cartopy.crs as ccrs
import cartopy.feature as cfeature
```

Open ECCO heat flux data

[2]:
```
# define root directory for location of all downloaded NetCDF files
root_dir = Path('../data/ECCO_V4r4_PODAAC')

# define the directory where the files specific to desired dataset are stored
nc_heat_dir = root_dir / "ECCO_L4_HEAT_FLUX_05DEG_MONTHLY_V4R4"
```

[3]:
```
# get all files in each folder for import
heat_nc_files = list(nc_heat_dir.glob('*nc'))
```

[4]:
```
# import 26 years of ecco heat flux data
ecco_heat_ds = xr.open_mfdataset(heat_nc_files, parallel=True,␣
↪data_vars='minimal',\
                             coords='minimal', compat='override')

ecco_heat_ds
```

[4]:
```
<xarray.Dataset>
Dimensions:          (time: 312, latitude: 360, longitude: 720, nv: 2)
Coordinates:
  * time             (time) datetime64[ns] 1992-01-16T18:00:00 … 2017-12-16T…
  * latitude         (latitude) float32 -89.75 -89.25 -88.75 … 89.25 89.75
  * longitude        (longitude) float32 -179.8 -179.2 -178.8 … 179.2 179.8
    time_bnds        (time, nv) datetime64[ns] dask.array<chunksize=(1, 2),
meta=np.ndarray>
```

```
    latitude_bnds   (latitude, nv) float32 dask.array<chunksize=(360, 2),
meta=np.ndarray>
    longitude_bnds  (longitude, nv) float32 dask.array<chunksize=(720, 2),
meta=np.ndarray>
Dimensions without coordinates: nv
Data variables:
    EXFhl           (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFhs           (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFlwdn         (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFswdn         (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFqnet         (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    oceQnet         (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    SIatmQnt        (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    TFLUX           (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFswnet        (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    EXFlwnet        (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    oceQsw          (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
    SIaaflux        (time, latitude, longitude) float32 dask.array<chunksize=(1,
360, 720), meta=np.ndarray>
Attributes: (12/57)
    acknowledgement:             This research was carried out by the Jet Pr…
    author:                      Ian Fenty and Ou Wang
    cdm_data_type:               Grid
    comment:                     Fields provided on a regular lat-lon grid. …
    Conventions:                 CF-1.8, ACDD-1.3
    coordinates_comment:         Note: the global 'coordinates' attribute de…
    …                          …
    time_coverage_duration:      P1M
    time_coverage_end:           1992-02-01T00:00:00
    time_coverage_resolution:    P1M
    time_coverage_start:         1992-01-01T12:00:00
    title:                       ECCO Ocean and Sea-Ice Surface Heat Fluxes …
    uuid:                        73ea7d5c-4158-11eb-8d61-0cc47a3f812d
```

[5]:
```
# import the geometry data file that provides area and volume information for␣
↪grid cells
```

```python
geometry_ds = xr.open_dataset('../data/ECCO_V4r4_PODAAC/
↪ECCO_L4_GEOMETRY_05DEG_V4R4/GRID_GEOMETRY_ECCO_V4r4_latlon_0p50deg.nc')
geometry_ds
```

[5]:
```
<xarray.Dataset>
Dimensions:          (Z: 50, latitude: 360, longitude: 720, nv: 2)
Coordinates:
  * Z                (Z) float32 -5.0 -15.0 -25.0 … -5.461e+03 -5.906e+03
  * latitude         (latitude) float32 -89.75 -89.25 -88.75 … 89.25 89.75
  * longitude        (longitude) float32 -179.8 -179.2 -178.8 … 179.2 179.8
    latitude_bnds    (latitude, nv) float32 …
    longitude_bnds   (longitude, nv) float32 …
    Z_bnds           (Z, nv) float32 …
Dimensions without coordinates: nv
Data variables:
    hFacC            (Z, latitude, longitude) float64 …
    Depth            (latitude, longitude) float64 …
    area             (latitude, longitude) float64 …
    drF              (Z) float32 …
    maskC            (Z, latitude, longitude) bool …
Attributes: (12/57)
    acknowledgement:                 This research was carried out by the Jet…
    author:                          Ian Fenty and Ou Wang
    cdm_data_type:                   Grid
    comment:                         Fields provided on a regular lat-lon gri…
    Conventions:                     CF-1.8, ACDD-1.3
    coordinates_comment:             Note: the global 'coordinates' attribute…
    …                               …
    references:                      ECCO Consortium, Fukumori, I., Wang, O.,…
    source:                          The ECCO V4r4 state estimate was produce…
    standard_name_vocabulary:        NetCDF Climate and Forecast (CF) Metadat…
    summary:                         This dataset provides geometric paramete…
    title:                           ECCO Geometry Parameters for the 0.5 deg…
    uuid:                            b4795c62-86e5-11eb-9c5f-f8f21e2ee3e0
```

Open ERA5 heat flux data

[35]:
```python
# inlcudes 1979-2021 variables: Surface net solar radiation, Surface net␣
↪thermal radiation
solar_thermal = xr.load_dataset("../data_climatology/
↪era5_1m_1979to2021_50to90N_-120Wto40E_025025_solar_thermal_heat.grib",␣
↪engine='cfgrib')
```

```
Ignoring index file '../data_climatology/era5_1m_1979to2021_50to90N_-120Wto40E_0
25025_solar_thermal_heat.grib.923a8.idx' incompatible with GRIB file
```

[36]:
```python
# inlcudes 1979-2021 variables: Surface latent heat flux, Surface sensible heat flux
latent_sensible = xr.load_dataset("../data_climatology/era5_1m_1979to2021_50to90N_-120Wto40E_025025_latent_sensible_heat.grib", engine='cfgrib')
```

```
Ignoring index file '../data_climatology/era5_1m_1979to2021_50to90N_-120Wto40E_0
25025_latent_sensible_heat.grib.923a8.idx' incompatible with GRIB file
```

[165]:
```python
latent_sensible
```

[165]:
```
<xarray.Dataset>
Dimensions:     (time: 516, latitude: 161, longitude: 641)
Coordinates:
    number      int32 0
  * time        (time) datetime64[ns] 1978-12-31T18:00:00 … 2021-11-30T18:0…
    step        timedelta64[ns] 12:00:00
    surface     float64 0.0
  * latitude    (latitude) float64 90.0 89.75 89.5 89.25 … 50.5 50.25 50.0
  * longitude   (longitude) float64 -120.0 -119.8 -119.5 … 39.5 39.75 40.0
    valid_time  (time) datetime64[ns] 1979-01-01T06:00:00 … 2021-12-01T06:0…
Data variables:
    slhf        (time, latitude, longitude) float32 -1.383e+05 … -5.358e+05
    sshf        (time, latitude, longitude) float32 1.947e+05 … 2.053e+05
Attributes:
    GRIB_edition:             1
    GRIB_centre:              ecmf
    GRIB_centreDescription:   European Centre for Medium-Range Weather Forecasts
    GRIB_subCentre:           0
    Conventions:              CF-1.7
    institution:              European Centre for Medium-Range Weather Forecasts
    history:                  2022-06-10T09:33 GRIB to CDM+CF via cfgrib-0.9.1…
```

[ ]:
```python
# Note: the `valid_step` coordinate is equal to latent_sensible.time + latent_sensible.step
```

[128]:
```python
# The units are joules per square metre (J m^-2 ).
# To convert to watts per square metre (W m^-2 ), the accumulated values should be divided by the accumulation period expressed in seconds.
# must divide values by 86,400 (sec/day)
era5_qnet_da = (solar_thermal.ssr/86400 + solar_thermal.str/86400 + latent_sensible.slhf/86400 + latent_sensible.sshf/86400)*(-1)
```

Isolate subpolar gyre region

ECCO:

[6]:
```
# pull out Depth from geometry file to use for creating the mask
ecco_depth = geometry_ds.Depth
```

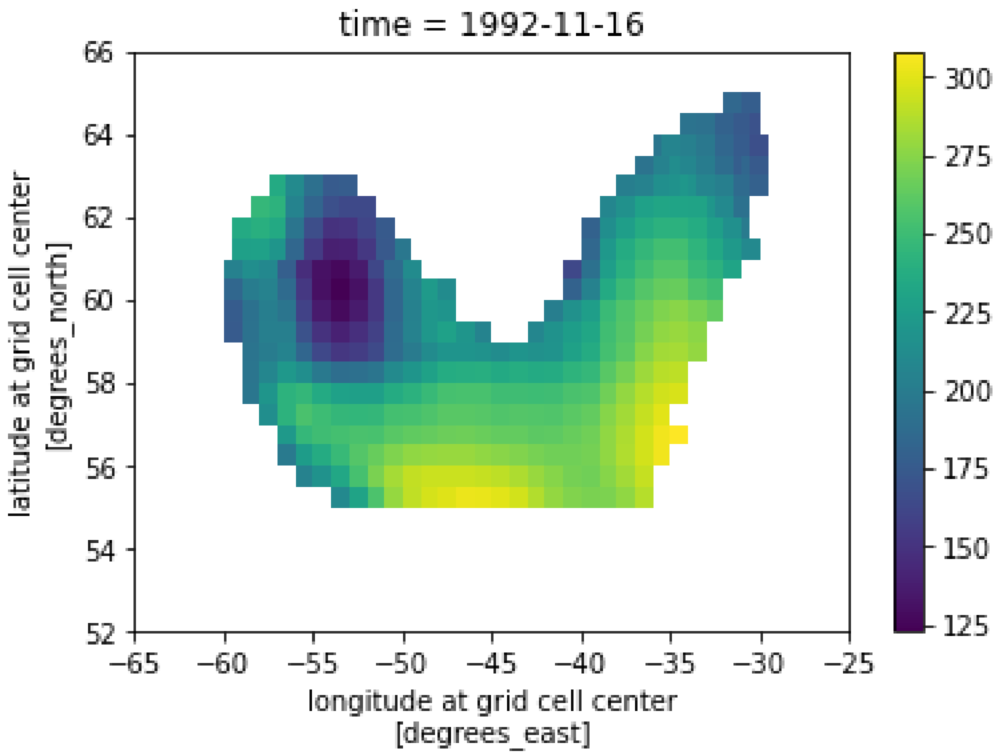
[7]:
```
# meshgrid for longitude and latitude 1D arrays
data_x_mg, data_y_mg = np.meshgrid(ecco_depth.longitude,ecco_depth.latitude)
```

[8]:
```
# create mask of subpolar gyre using 2000 m contour north of 55 deg.
# the last two logical statements remove small boxes from the desired region
gyre_mask = np.where(ecco_depth > 2000, 1, np.nan)*np.where(np.
 ↪logical_and(data_y_mg > 55, data_y_mg < 70), 1, np.nan)*\
            np.where(data_x_mg > -60, 1, np.nan)*np.where(data_x_mg < -29, 1,␣
 ↪np.nan)*\
            np.where(np.logical_and(data_y_mg < 58.5, data_x_mg > -34), np.nan,␣
 ↪1)*\
            np.where(np.logical_and(data_y_mg < 55.5, data_x_mg > -35), np.nan,␣
 ↪1)
```

[10]:
```
gyre_mask_da = xr.DataArray(gyre_mask, dims=("latitude","longitude"),\
                            coords=dict(latitude=ecco_heat_ds.EXFqnet.
 ↪latitude,longitude=ecco_heat_ds.EXFqnet.longitude),\
                            name='mask')
```

[11]:
```
# multiply ecco heat flux by mask to isolate data for the subpolar gyre
ecco_EXFqnet_gyre_da = ecco_heat_ds.EXFqnet*gyre_mask_da
```

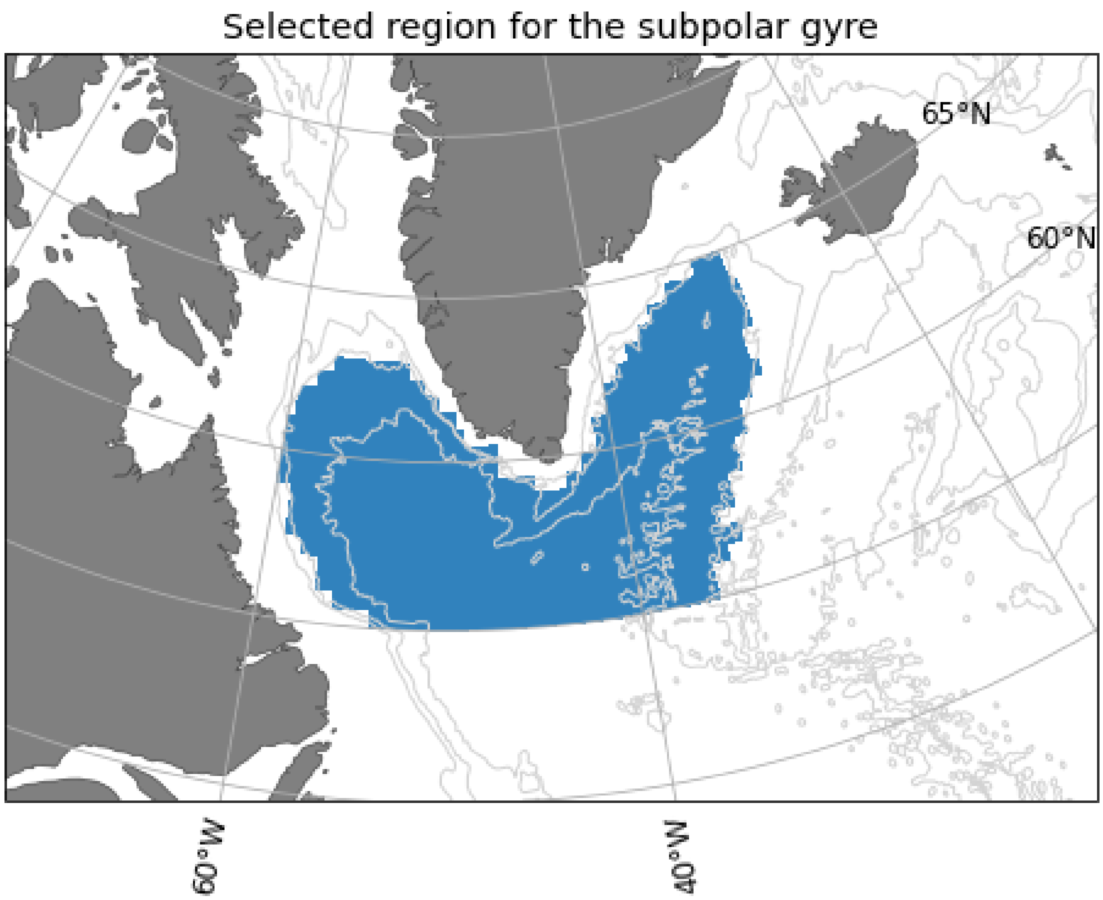
[13]:
```
# sanity check plot to make sure the mask worked
ecco_EXFqnet_gyre_da.isel(time=10).plot()
plt.xlim(-65,-25)
plt.ylim(52,66);
```



[15]:

```
# Plot region selected for analysis
plt.rcParams['font.size'] = 12
fig = plt.figure(figsize=[8,10])

ax1=plt.subplot(1,1,1, projection=ccrs.NorthPolarStereo(central_longitude=-50))
ax1.set_extent([-70, -20, 50, 70], ccrs.PlateCarree()) # Limit the map extent
ax1.add_feature(cfeature.COASTLINE, edgecolor='k',linewidth=0.2)
ax1.add_feature(cfeature.LAND, color='gray')
bathym = cfeature.NaturalEarthFeature(name='bathymetry_J_1000', scale='10m',␣
 ↪category='physical')
ax1.add_feature(bathym, facecolor='none', edgecolor='lightgray')
bathym = cfeature.NaturalEarthFeature(name='bathymetry_I_2000', scale='10m',␣
 ↪category='physical')
ax1.add_feature(bathym, facecolor='none', edgecolor='lightgray')
bathym = cfeature.NaturalEarthFeature(name='bathymetry_H_3000', scale='10m',␣
 ↪category='physical')
ax1.add_feature(bathym, facecolor='none', edgecolor='lightgray', label=True)

gl = ax1.gridlines(draw_labels=True)

gl.top_labels=False
```

```
data_cartopy_crs = ccrs.PlateCarree()
ecco_EXFqnet_gyre_da.isel(time=100).plot(ax=ax1,transform=ccrs.
 ↪PlateCarree(),cmap='tab20c',vmin= -1, vmax=1,add_colorbar=False)

ax1.set_title("Selected region for the subpolar gyre");
```



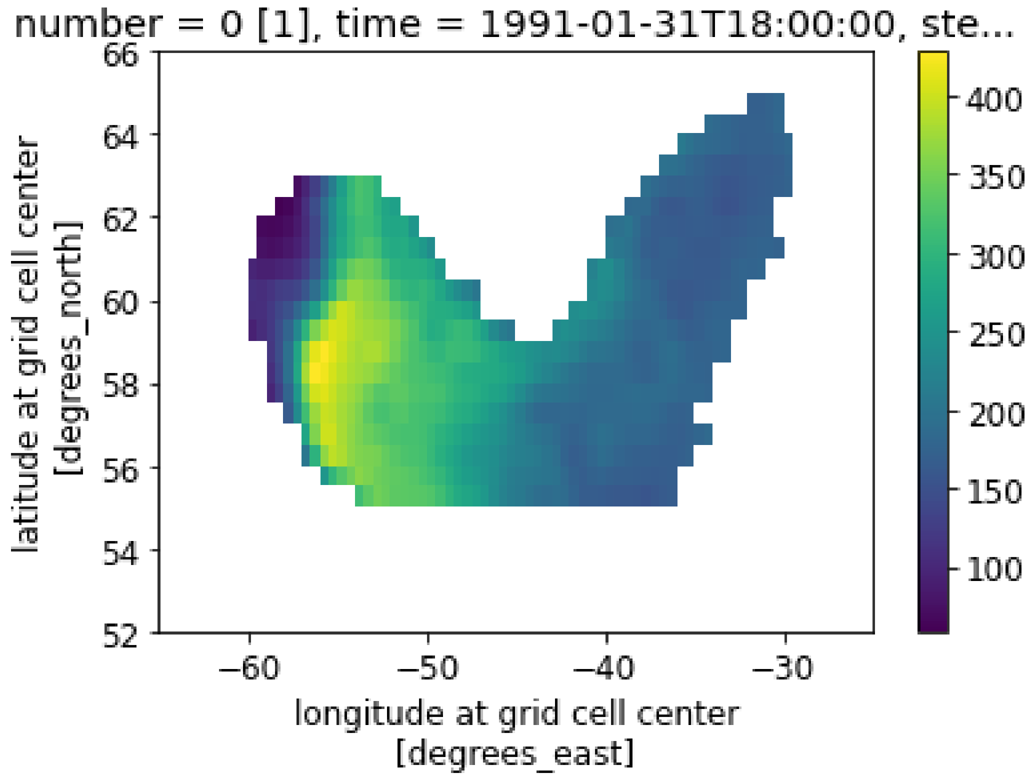
ERA5: Need to put the ERA5 dataset on the same grid as the ECCO data in order to match the geometry mask for the gyre

[130]:
```
era5_interp_da = era5_qnet_da.interp(latitude=ecco_heat_ds.latitude,␣
 ↪longitude=ecco_heat_ds.longitude, method="nearest")
era5_qnet_da = era5_interp_da.sel(time=slice("1991","2017"))
```

[131]:
```
era5_qnet_gyre_da = era5_qnet_da*gyre_mask_da
```

[132]:
```
# sanity check plot
era5_qnet_gyre_da.isel(time=0).plot()
plt.xlim(-65,-25)
```

```
plt.ylim(52,66);
```



Look at time coordinate of both datasets ECCO's time coordinate indicates the center time of the averaging period (the middle of each month). ERA5's time coordinate indicates a specific point in time at which a forecast starts (initialization time; also called "forecast_reference_time") - 'time' can be 06:00 or 18:00 - 'step' indicates hours after the initialization time.

ERA5 reference links: https://confluence.ecmwf.int/pages/viewpage.action?pageId=85402030 (terminology) https://confluence.ecmwf.int/display/CKB/ERA5%3A+data+documentation#ERA5:datadocumentat Monthlymeans (how monthly means are calculated)

[133]:
```
# slice out two years to investigate phase shift
ecco_slice = ecco_EXFqnet_gyre_da.sel(time=slice('2007','2008')).
 ↪mean(dim=['latitude','longitude'])
era5_slice = era5_qnet_gyre_da.sel(time=slice('2007','2008')).
 ↪mean(dim=['latitude','longitude'])
```

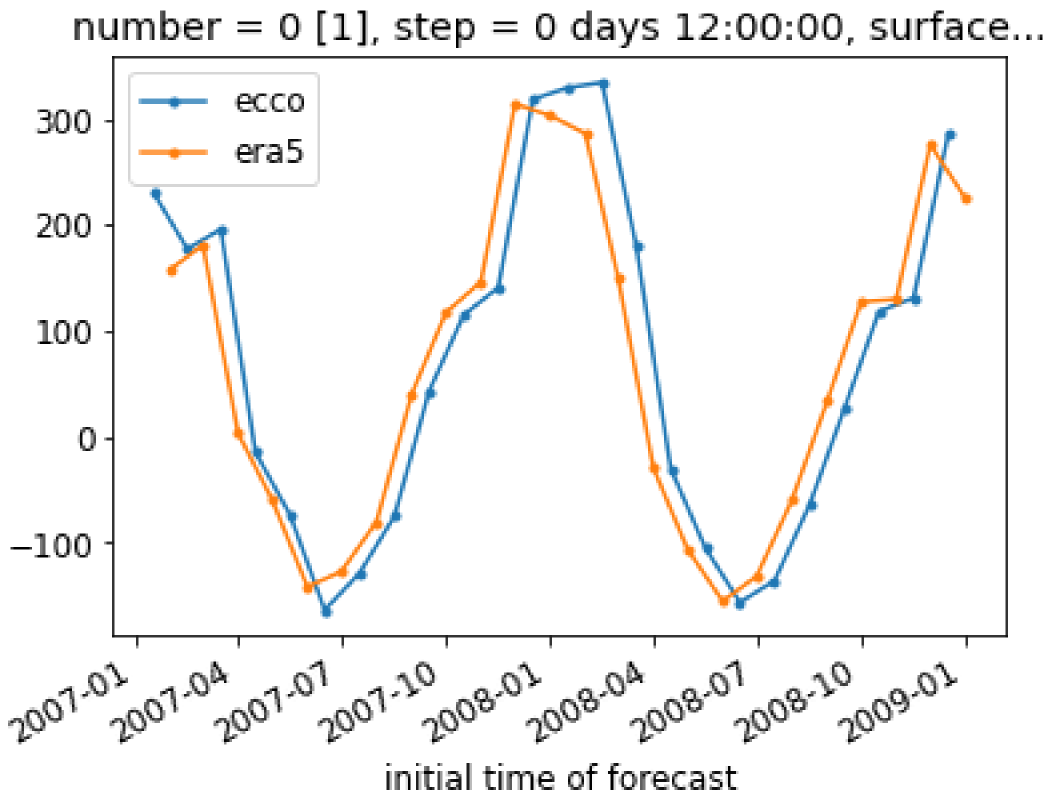
[134]:
```
ecco_slice.time[0:3].values
```

[134]:
```
array(['2007-01-16T12:00:00.000000000', '2007-02-15T00:00:00.000000000',
       '2007-03-16T12:00:00.000000000'], dtype='datetime64[ns]')
```

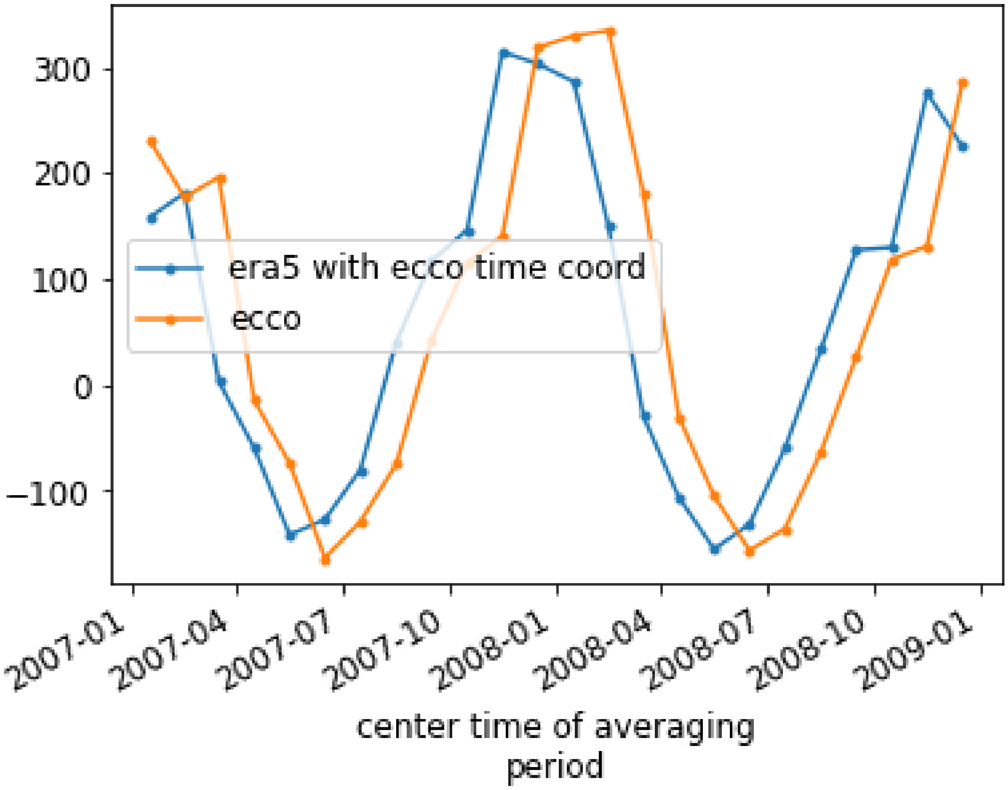
[135]:
```
era5_slice.time[0:3].values
```

[135]:
```
array(['2007-01-31T18:00:00.000000000', '2007-02-28T18:00:00.000000000',
       '2007-03-31T18:00:00.000000000'], dtype='datetime64[ns]')
```

[136]:
```
# plot the two datasets
ecco_slice.plot(marker=".",label='ecco')
era5_slice.plot(marker=".",label='era5')
plt.legend();
```



[137]:
```
# I can use the x coords of ecco but plot the y values (qnet) from ERA5
plt.plot(ecco_slice.time,era5_slice.values,marker=".",label='era5 with ecco␣
 ↪time coord')
# then plot the ecco data for comparison
ecco_slice.plot(marker=".",label='ecco')
plt.legend();
```



^ ah hah…so we see where the phase shift came from

What we need is the `valid_time` coordinate in the dataset instead of `time`. However, `valid_time` is not a dimensional coordinate so we cannot use it for indexing (e.g., using '.sel'). We can use 'swap_dims' to make it a dimensional coordinate.

[142]:
```
# swap dims
era5_qnet_gyre_swap = era5_qnet_gyre_da.swap_dims({'time':'valid_time'})
# remove `time` coord and rename `valid_time` to `time` dim so the rest of my␣
↪code works as is
era5_qnet_gyre_valid_da = era5_qnet_gyre_swap.drop('time').rename({'valid_time':
↪'time'}).sel(time=slice("1992","2017"))
```

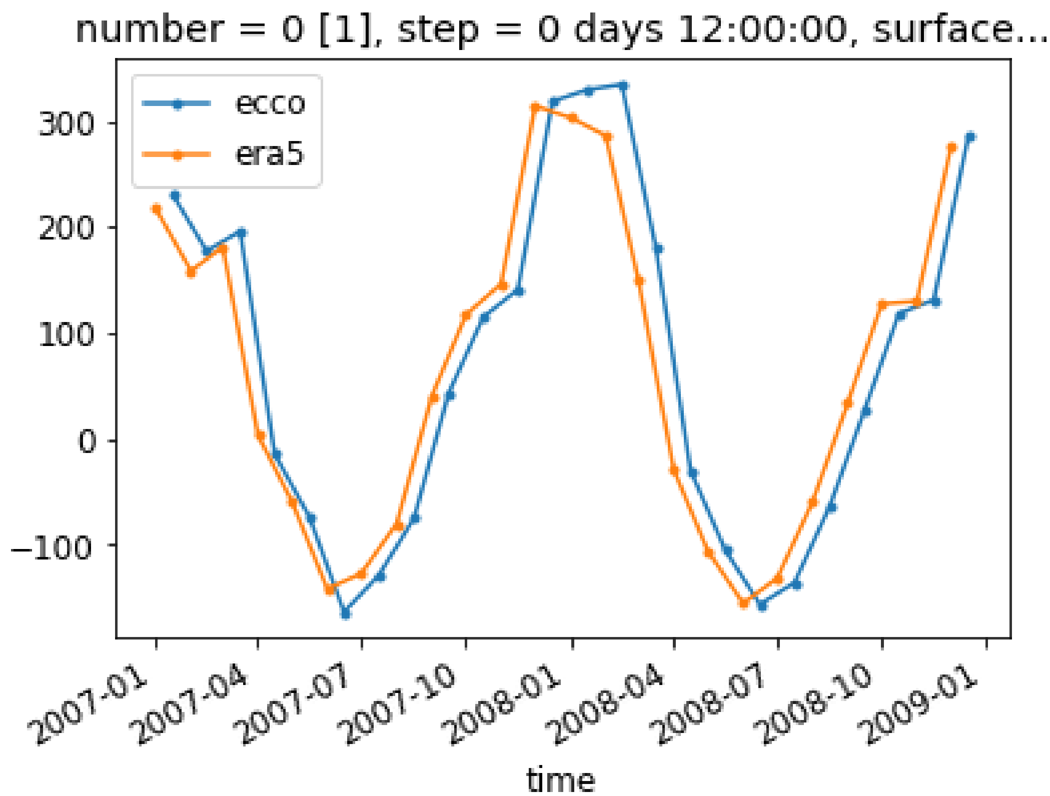
[164]:
```
era5_qnet_gyre_valid_da
```

[164]:
```
<xarray.DataArray (time: 312, latitude: 360, longitude: 720)>
array([[[nan, nan, nan, …, nan, nan, nan],
        [nan, nan, nan, …, nan, nan, nan],
        [nan, nan, nan, …, nan, nan, nan],
        …,
        [nan, nan, nan, …, nan, nan, nan],
        [nan, nan, nan, …, nan, nan, nan],
```

```
          [nan, nan, nan, …, nan, nan, nan]],

         [[nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          …,
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan]],

         [[nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          …,
…
          …,
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan]],

         [[nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          …,
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan]],

         [[nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          …,
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan],
          [nan, nan, nan, …, nan, nan, nan]]])
Coordinates:
    number     int32 0
    step       timedelta64[ns] 12:00:00
    surface    float64 0.0
  * time       (time) datetime64[ns] 1992-01-01T06:00:00 … 2017-12-01T06:00:00
  * latitude   (latitude) float32 -89.75 -89.25 -88.75 … 88.75 89.25 89.75
  * longitude  (longitude) float32 -179.8 -179.2 -178.8 … 178.8 179.2 179.8
```

[144]:
```python
# plot the two datasets again
era5_slice_v2 = era5_qnet_gyre_valid_da.sel(time=slice('2007','2008')).
 ↪mean(dim=['latitude','longitude'])
```

```
ecco_slice.plot(marker=".",label='ecco')
era5_slice_v2.plot(marker=".",label='era5')
plt.legend();
```



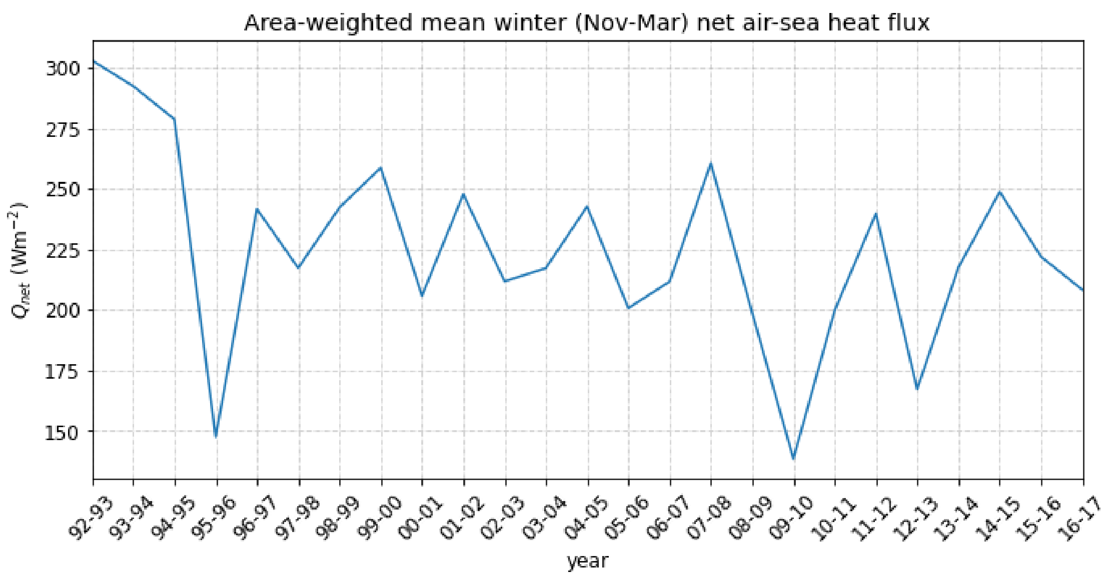
Function to calculate and plot mean winter air-sea heat fluxes

[145]:
```
def calc_winter_Qnet(qnet_da, years, mask_da):
    """
    Function to calculate area-weighted net air-sea heat fluxes. Can be easily␣
↪modified to other datasets for wintertime means.

    `qnet_da` is a DataArray containing all heat flux data that includes lat␣
↪and lon coordinates
    `years` is a list of years (e.g., list(range(1992, 2017, 1))) over which␣
↪you want to calculate winter heat fluxes
    `mask_da` is a DataArray that defines the area within the dataset to be␣
↪used for calculations and weight-averages

    """
    # loop through each year and calculate only winter season (Nov-Mar) heat␣
↪fluxes
```

```
    heat_winter = []

    for year in years:
        # subset data for winter season using consecutive years and take mean
        data_tmp = qnet_da.sel(time=slice('11-'+str(year),'03-'+str(year+1)))

        # cos(lat) is proportional to grid cell area
        cos_lat = np.cos(np.deg2rad(data_tmp.latitude))
        weights_lon, weights_lat = np.meshgrid(data_tmp.longitude, cos_lat)
        weights = xr.DataArray(weights_lat*mask_da,␣
↪dims=("latitude","longitude"),\
                                coords=dict(latitude=data_tmp.
↪latitude,longitude=data_tmp.longitude),name='weights')
        qnet_weighted = (data_tmp*weights).sum(dim=["latitude","longitude"])/
↪weights.sum()
        winter_heat = qnet_weighted.mean(dim=["time"])
        # assign to output
        heat_winter.append(winter_heat.values)

    # Create output DataArray
    winter_heat_da = xr.DataArray(heat_winter, dims='start_year',␣
↪coords={'start_year': years}, name='Qnet_winter')

    heat_avg_winter = winter_heat_da.mean()

    print(f'mean winter (Nov-Mar) air-sea heat flux 1992-2017: {heat_avg_winter.
↪round(2).values} W m^-2\n')
    print(winter_heat_da)

    # plot winter mean heat flux
    plt.figure(figsize=[11, 5])
    plt.rcParams['font.size'] = '12'
    winter_heat_da.plot()
    # ticks
    year_span = []
    for year in winter_heat_da.start_year:
        year_span.append(str(year.values)[-2:]+'-'+str(year.values+1)[-2:])
    plt.margins(x=0)
    plt.grid(linestyle='-.', linewidth=0.5)
    plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
    plt.xticks(ticks=winter_heat_da.start_year, labels=year_span, rotation=45)
    plt.xlabel("year")
    plt.title("Area-weighted mean winter (Nov-Mar) net air-sea heat flux");

    return winter_heat_da, heat_avg_winter
```

[146]:
```python
# run function for ECCO data
years = list(range(1992, 2017, 1))
ecco_winter_heat_da, ecco_heat_avg_winter =␣
↪calc_winter_Qnet(ecco_EXFqnet_gyre_da, years, gyre_mask_da)
```

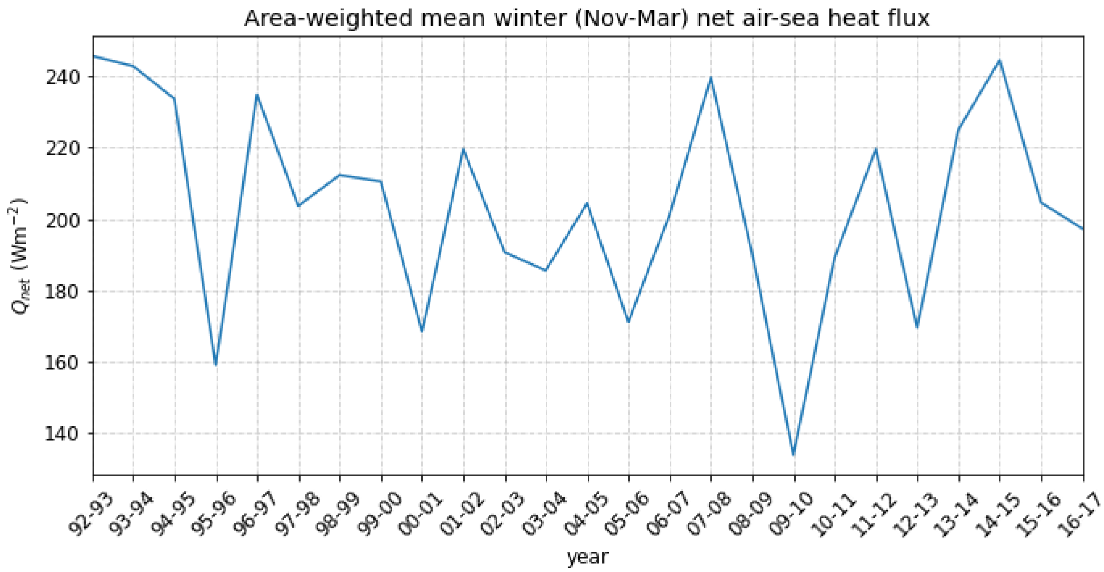
```
mean winter (Nov-Mar) air-sea heat flux 1992-2017: 224.83 W m^-2

<xarray.DataArray 'Qnet_winter' (start_year: 25)>
array([302.96020752, 292.33229329, 278.63559666, 147.67159446,
       241.60664724, 217.29177999, 242.24969947, 258.64126046,
       205.85495617, 247.73113624, 211.79820885, 217.197308  ,
       242.65953721, 200.72937585, 211.69056367, 260.57720135,
       198.90764985, 138.63073922, 199.72819136, 239.79489015,
       167.37363356, 217.52088526, 248.74931359, 221.97220353,
       208.37612733])
Coordinates:
  * start_year  (start_year) int32 1992 1993 1994 1995 … 2013 2014 2015 2016
```



[147]:
```python
# run function for ERA5 data
era5_winter_heat_da, era5_heat_avg_winter =␣
↪calc_winter_Qnet(era5_qnet_gyre_valid_da, years, gyre_mask_da)
```

```
mean winter (Nov-Mar) air-sea heat flux 1992-2017: 203.88 W m^-2

<xarray.DataArray 'Qnet_winter' (start_year: 25)>
array([245.81325067, 242.95441471, 233.71745117, 158.95688274,
       234.94325079, 203.65044606, 212.28656871, 210.53983993,
       168.37197542, 219.58698379, 190.62996162, 185.4760554 ,
```

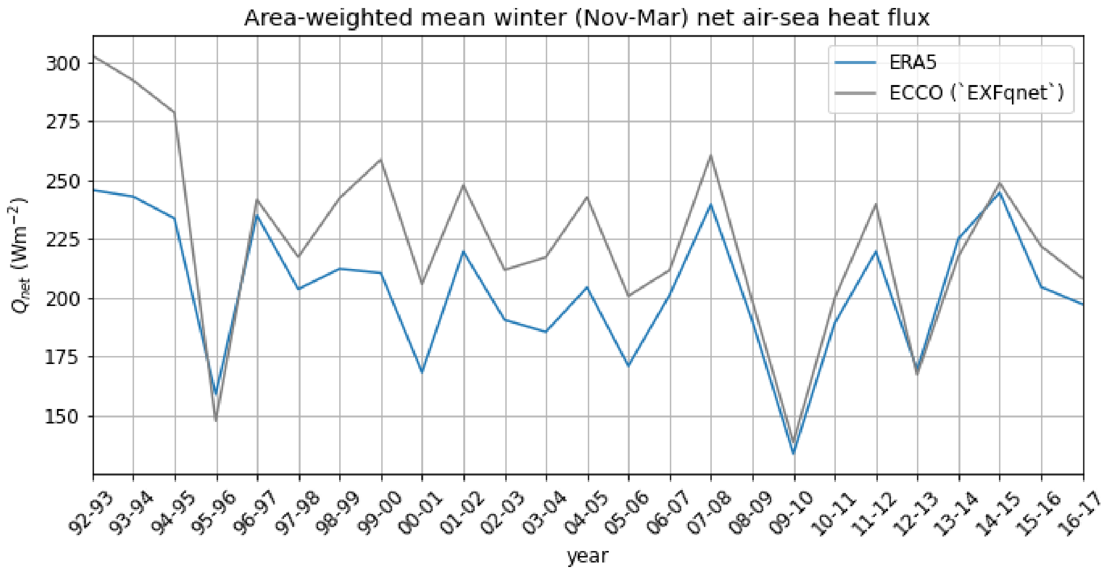
```
       204.39778542, 171.03550848, 201.11110979, 239.67056607,
       190.38447673, 133.73774753, 189.10496946, 219.65285219,
       169.47042915, 225.04539102, 244.6584558 , 204.60845526,
       197.28651998])
Coordinates:
  * start_year  (start_year) int32 1992 1993 1994 1995 … 2013 2014 2015 2016
```



0.1.1 Compare ERA5 and ECCO winter heat fluxes

[172]:
```python
plt.figure(figsize=[11, 5])
plt.rcParams['font.size'] = '12'

era5_winter_heat_da.plot(color='tab:blue',label='ERA5')
ecco_winter_heat_da.plot(color='tab:gray',label='ECCO (`EXFqnet`)')

# ticks
year_span = []
for year in ecco_winter_heat_da.start_year:
    year_span.append(str(year.values)[-2:]+'-'+str(year.values+1)[-2:])

plt.margins(x=0)
plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
plt.xticks(ticks=era5_winter_heat_da.start_year, labels=year_span, rotation=45)
plt.title("Area-weighted mean winter (Nov-Mar) net air-sea heat flux")
plt.xlabel("year")
```

```
plt.legend()
plt.grid();
```



Plot monthly mean net air-sea heat fluxes

ECCO:

[149]:
```
# heat flux weighted by grid-cell area
area_gyre = geometry_ds.area*gyre_mask_da
area_gyre_total = area_gyre.sum(['latitude','longitude'])

EXFqnet_gyre_mean_weighted = (ecco_EXFqnet_gyre_da*area_gyre).
 ↪sum(dim=["latitude","longitude"])/area_gyre_total

# now get the mean for each month from 1992-2017
EXFqnet_gyre_mean_month_weighted = EXFqnet_gyre_mean_weighted.groupby("time.
 ↪month").mean(dim=["time"]) # average
EXFqnet_gyre_std_month_weighted = EXFqnet_gyre_mean_weighted.groupby("time.
 ↪month").std(dim=["time"]) # standard dev
```

[150]:
```
ecco_heat_months_ordered = xr.concat([EXFqnet_gyre_mean_month_weighted.
 ↪sel(month=slice(5,12)),EXFqnet_gyre_mean_month_weighted.
 ↪sel(month=slice(1,4))],dim='month')
ecco_heat_months_sd_ordered = xr.concat([EXFqnet_gyre_std_month_weighted.
 ↪sel(month=slice(5,12)),EXFqnet_gyre_std_month_weighted.
 ↪sel(month=slice(1,4))],dim='month')
```

```
[151]: ecco_heat_months_ordered['month'] =␣
        ↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
       ecco_heat_months_sd_ordered['month'] =␣
        ↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
```

ERA5:

```
[158]: # heat flux weighted by grid-cell area
       area_gyre = geometry_ds.area*gyre_mask_da
       area_gyre_total = area_gyre.sum(['latitude','longitude'])

       era5_gyre_mean_weighted = (era5_qnet_gyre_valid_da*area_gyre).
        ↪sum(dim=["latitude","longitude"])/area_gyre_total

       # now get the mean for each month from 1992-2017
       era5_gyre_mean_month_weighted = era5_gyre_mean_weighted.groupby("time.month").
        ↪mean(dim=["time"]) # average
       era5_gyre_std_month_weighted = era5_gyre_mean_weighted.groupby("time.month").
        ↪std(dim=["time"]) # standard dev
```

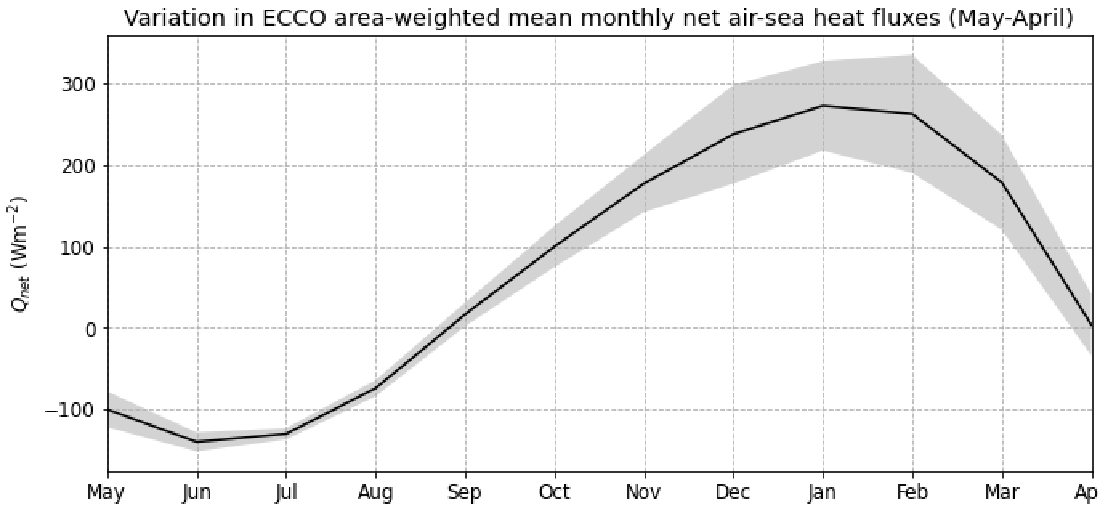
```
[159]: era5_heat_months_ordered = xr.concat([era5_gyre_mean_month_weighted.
        ↪sel(month=slice(5,12)),era5_gyre_mean_month_weighted.
        ↪sel(month=slice(1,4))],dim='month')
       era5_heat_months_sd_ordered = xr.concat([era5_gyre_std_month_weighted.
        ↪sel(month=slice(5,12)),era5_gyre_std_month_weighted.
        ↪sel(month=slice(1,4))],dim='month')
```

```
[160]: era5_heat_months_ordered['month'] =␣
        ↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
       era5_heat_months_sd_ordered['month'] =␣
        ↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
```

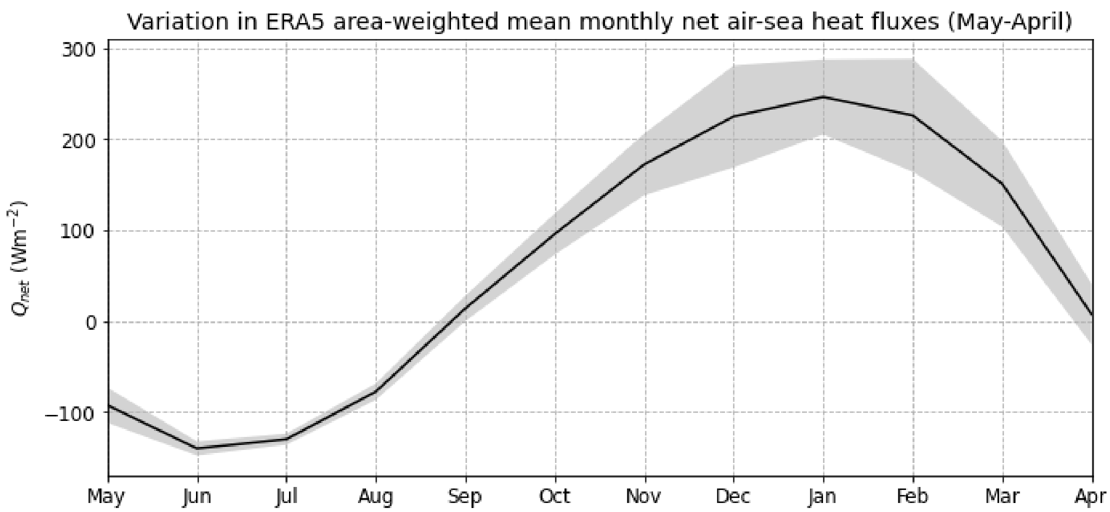
Plot:

```
[152]: plt.figure(figsize=[11, 5])
       plt.rcParams['font.size'] = '12'
       plt.margins(x=0)

       plt.plot(ecco_heat_months_ordered.month,ecco_heat_months_ordered.values,␣
        ↪color='k')
       plt.fill_between(ecco_heat_months_sd_ordered.month, ecco_heat_months_ordered.
        ↪values-ecco_heat_months_sd_ordered.values,\
                        ecco_heat_months_ordered.values+ecco_heat_months_sd_ordered.
        ↪values, facecolor='lightgray', interpolate=True)

       plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
```

```
# plt.xticks(ticks=heat_months_ordered.month,␣
↪labels=['M','J','J','A','S','O','N','D','J','F','M','A'])
plt.grid(linestyle='--')
plt.title("Variation in ECCO area-weighted mean monthly net air-sea heat fluxes␣
↪(May-April)");
```

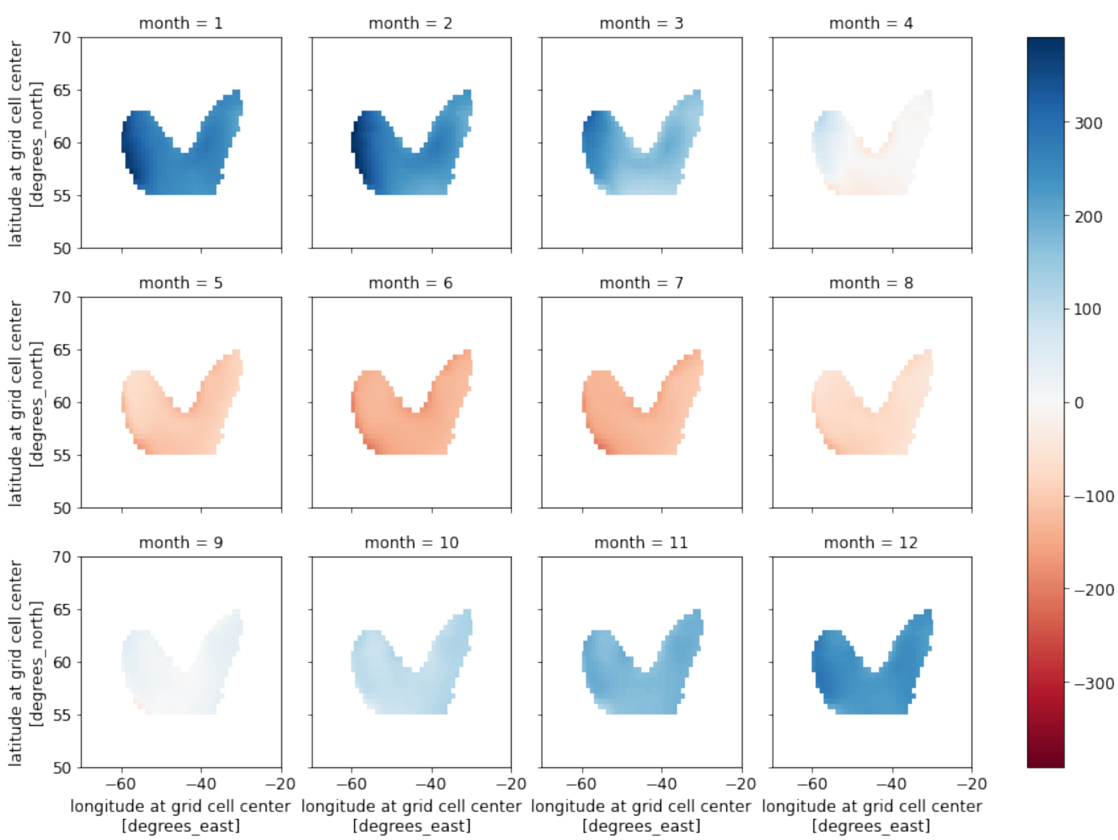


[161]:
```
plt.figure(figsize=[11, 5])
plt.rcParams['font.size'] = '12'
plt.margins(x=0)

plt.plot(era5_heat_months_ordered.month,era5_heat_months_ordered.values,␣
↪color='k')
plt.fill_between(era5_heat_months_ordered.month, era5_heat_months_ordered.
↪values-era5_heat_months_sd_ordered.values,\
                 era5_heat_months_ordered.values+era5_heat_months_sd_ordered.
↪values, facecolor='lightgray', interpolate=True)

plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
# plt.xticks(ticks=heat_months_ordered.month,␣
↪labels=['M','J','J','A','S','O','N','D','J','F','M','A'])
plt.grid(linestyle='--')
plt.title("Variation in ERA5 area-weighted mean monthly net air-sea heat fluxes␣
↪(May-April)");
```

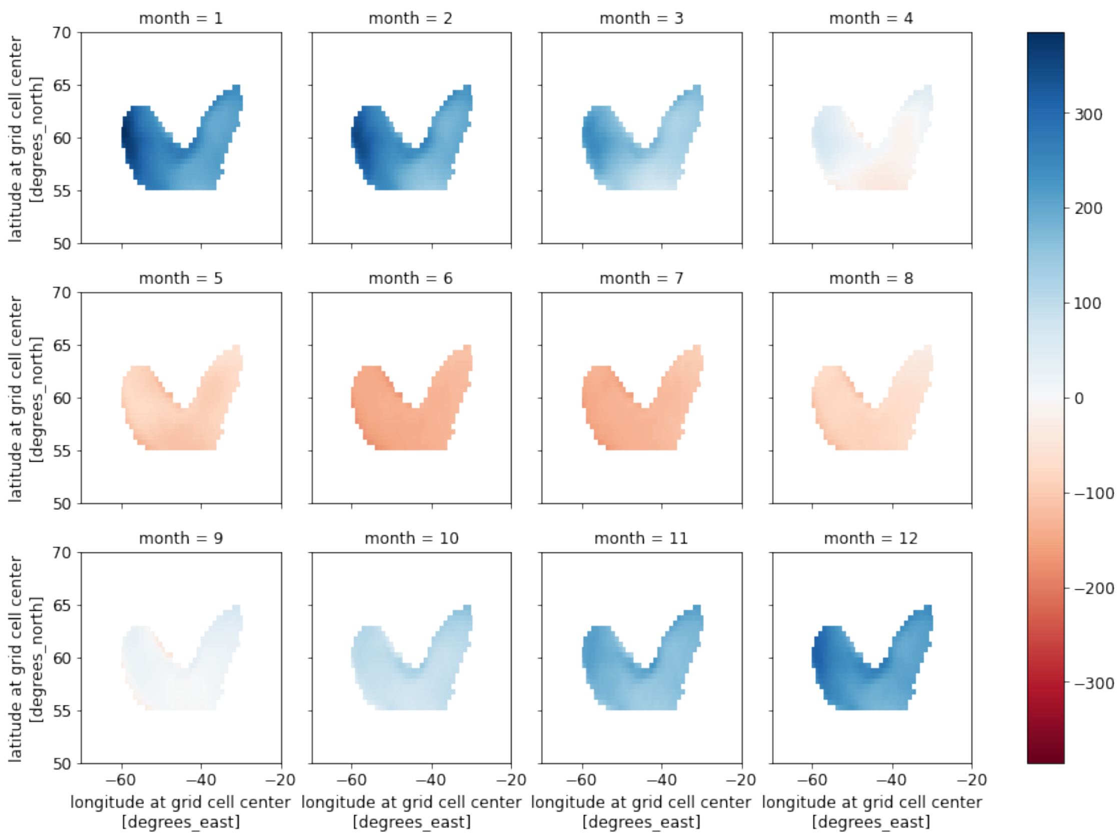


Finally, take a quick look at spatial variation in monthly means:

[162]:
```
ecco_EXFqnet_gyre_da.sel(latitude=slice(50,70), longitude=slice(-70,-20)).
    ↪groupby("time.month").mean("time").
    ↪plot(x="longitude",y="latitude",col="month",col_wrap=4,cmap='RdBu');
```



```
[163]: era5_qnet_gyre_valid_da.sel(latitude=slice(50,70), longitude=slice(-70,-20)).
         ↪groupby("time.month").mean("time").
         ↪plot(x="longitude",y="latitude",col="month",col_wrap=4,cmap='RdBu');
```



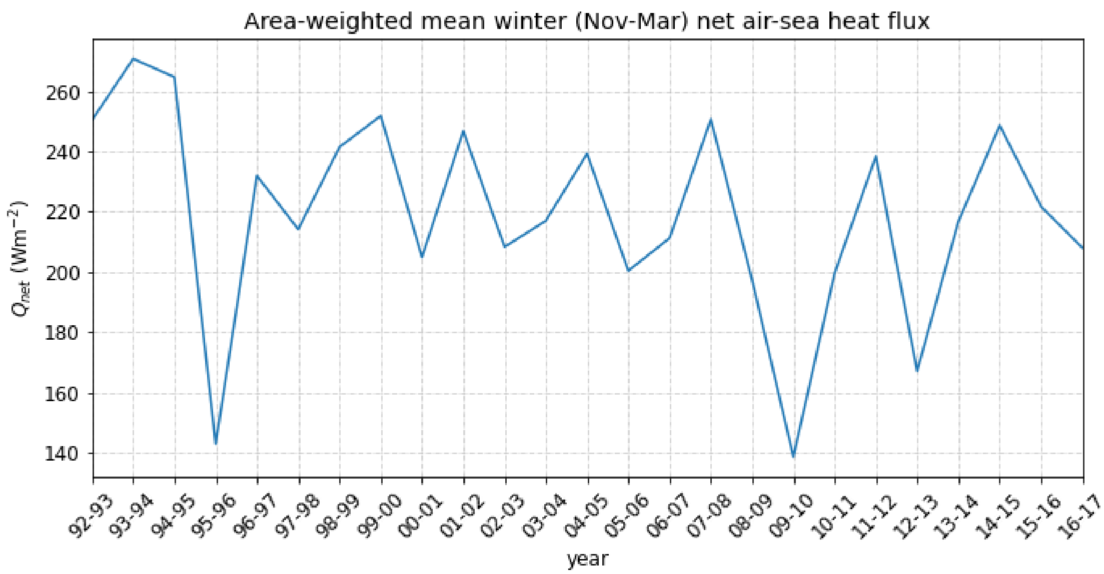
Run again but use SIatmQnet

```
[168]: # multiply ecco heat flux by mask to isolate data for the subpolar gyre
       ecco_SIatmQnt_gyre_da = ecco_heat_ds.SIatmQnt*gyre_mask_da
```

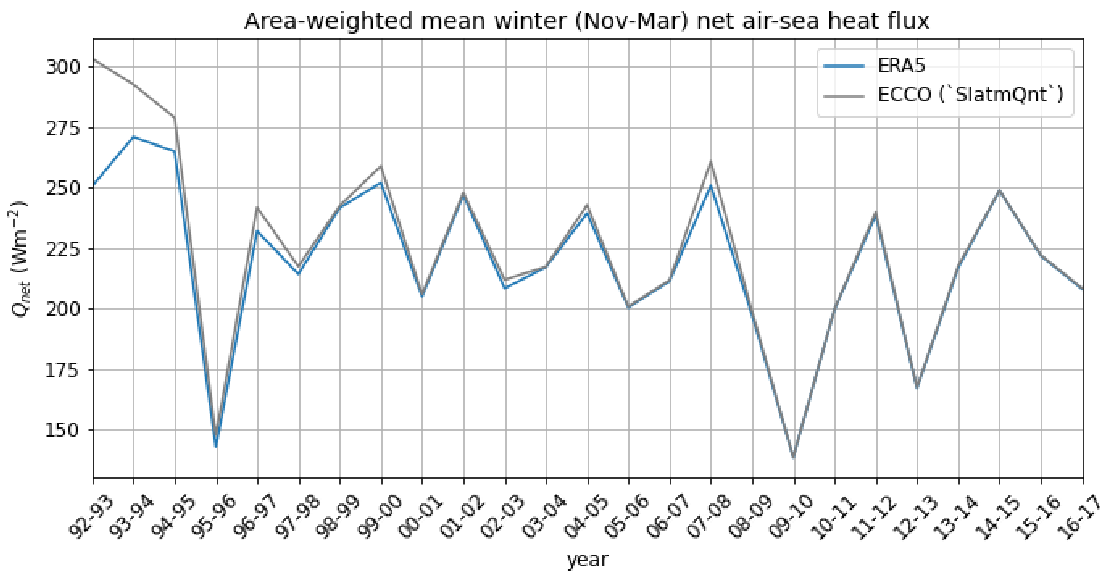
```
[169]: # run function for ECCO data
       years = list(range(1992, 2017, 1))
       ecco_winter_SIatmQnt_da, ecco_SIatmQnt_avg_winter =␣
         ↪calc_winter_Qnet(ecco_SIatmQnt_gyre_da, years, gyre_mask_da)
```

```
mean winter (Nov-Mar) air-sea heat flux 1992-2017: 219.29 W m^-2

<xarray.DataArray 'Qnet_winter' (start_year: 25)>
array([250.32226951, 270.73933902, 264.67021222, 142.86749335,
       231.87605045, 214.1247012 , 241.43634089, 251.79870221,
```

```
       204.8777801 , 246.74298721, 208.27436204, 216.94680997,
       239.22742062, 200.35553493, 211.19378451, 250.67074377,
       197.29400508, 138.57795807, 199.54721528, 238.44031145,
       167.09379526, 216.72759625, 248.61362399, 221.73462141,
       208.03828923])
Coordinates:
  * start_year  (start_year) int32 1992 1993 1994 1995 … 2013 2014 2015 2016
```



[171]:
```
plt.figure(figsize=[11, 5])
plt.rcParams['font.size'] = '12'

ecco_winter_SIatmQnt_da.plot(color='tab:blue',label='ERA5')
ecco_winter_heat_da.plot(color='tab:gray',label='ECCO (`SIatmQnt`)')

# ticks
year_span = []
for year in ecco_winter_heat_da.start_year:
    year_span.append(str(year.values)[-2:]+'-'+str(year.values+1)[-2:])

plt.margins(x=0)
plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
plt.xticks(ticks=era5_winter_heat_da.start_year, labels=year_span, rotation=45)
plt.title("Area-weighted mean winter (Nov-Mar) net air-sea heat flux")
plt.xlabel("year")

plt.legend()
plt.grid();
```



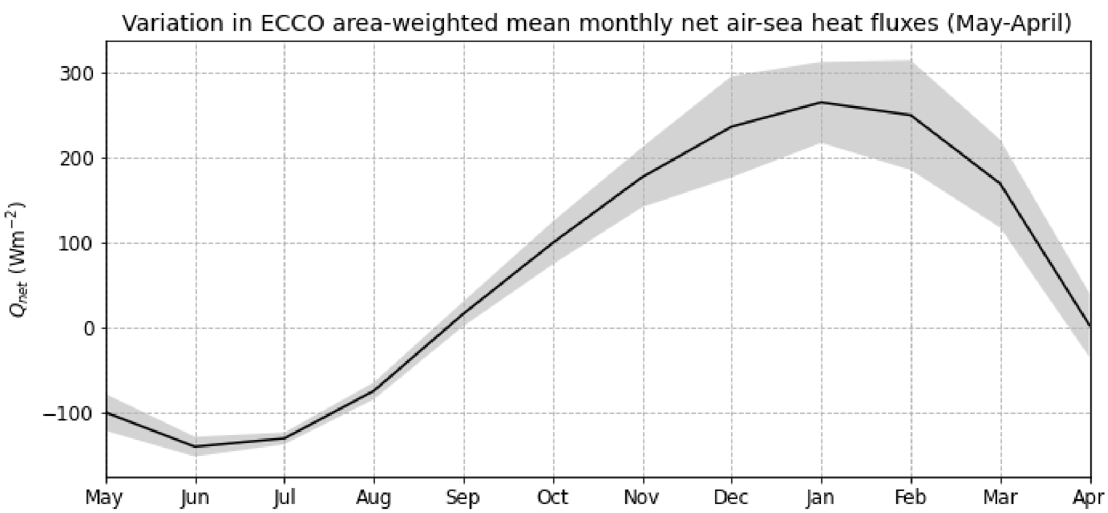
[173]:
```
# heat flux weighted by grid-cell area
SIatmQnt_gyre_mean_weighted = (ecco_SIatmQnt_gyre_da*area_gyre).
↪sum(dim=["latitude","longitude"])/area_gyre_total

# now get the mean for each month from 1992-2017
SIatmQnt_gyre_mean_month_weighted = SIatmQnt_gyre_mean_weighted.groupby("time.
↪month").mean(dim=["time"]) # average
SIatmQnt_gyre_std_month_weighted = SIatmQnt_gyre_mean_weighted.groupby("time.
↪month").std(dim=["time"]) # standard dev
```

[174]:
```
ecco_heat_months_ordered = xr.concat([SIatmQnt_gyre_mean_month_weighted.
↪sel(month=slice(5,12)),SIatmQnt_gyre_mean_month_weighted.
↪sel(month=slice(1,4))],dim='month')
ecco_heat_months_sd_ordered = xr.concat([SIatmQnt_gyre_std_month_weighted.
↪sel(month=slice(5,12)),SIatmQnt_gyre_std_month_weighted.
↪sel(month=slice(1,4))],dim='month')
```

[175]:
```
ecco_heat_months_ordered['month'] =␣
↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
ecco_heat_months_sd_ordered['month'] =␣
↪['May','Jun','Jul','Aug','Sep','Oct','Nov','Dec','Jan','Feb','Mar','Apr']
```

[176]:
```
plt.figure(figsize=[11, 5])
plt.rcParams['font.size'] = '12'
plt.margins(x=0)
```

```
plt.plot(ecco_heat_months_ordered.month,ecco_heat_months_ordered.values,␣
↪color='k')
plt.fill_between(ecco_heat_months_sd_ordered.month, ecco_heat_months_ordered.
↪values-ecco_heat_months_sd_ordered.values,\
                 ecco_heat_months_ordered.values+ecco_heat_months_sd_ordered.
↪values, facecolor='lightgray', interpolate=True)

plt.ylabel("$Q_{net}$ (Wm$^{-2}$)")
# plt.xticks(ticks=heat_months_ordered.month,␣
↪labels=['M','J','J','A','S','O','N','D','J','F','M','A'])
plt.grid(linestyle='--')
plt.title("Variation in ECCO area-weighted mean monthly net air-sea heat fluxes␣
↪(May-April)");
```



[ ]: