# Mapping Meteorological Fields from WRF Output

*Lecture 5*

Ph.D., Senior lecturer Madina Tursumbayeva

Department of Meteorology and Hydrology

# Lecture Objective and Learning Outcomes

## Lecture objective

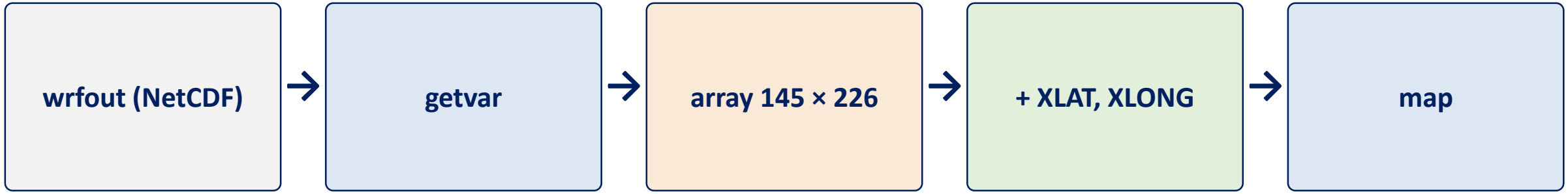
To teach how to turn WRF output into clear and correct meteorological maps in Python: coordinates and projections, filled fields, contour lines, wind vectors and the fair comparison of maps.

## Learning outcomes

*By the end of this lecture, students will be able to:*

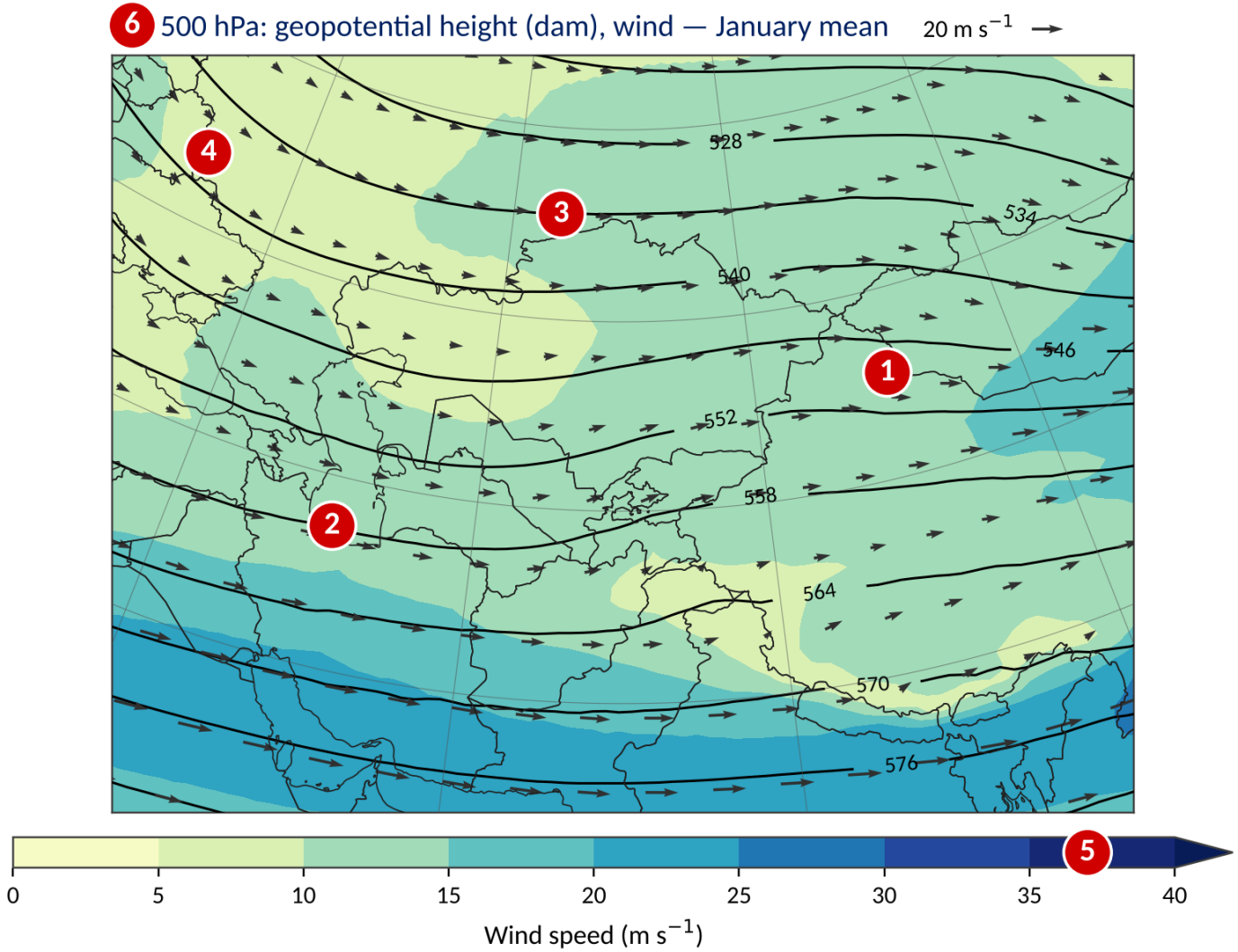
1. list the elements of a meteorological map;
2. plot a WRF field with matplotlib and cartopy, using the 2-D coordinates XLAT and XLONG and the correct transform;
3. choose between pcolormesh, contourf and contour lines, and between sequential and diverging colour scales;
4. plot wind vectors with thinning and a reference arrow;
5. compare maps correctly, and build and interpret a difference map.

# From a File to a Map




- Lecture 4: we learned to read a variable from a wrfout file
- An array of numbers tells us very little
- A map places every value at its position on the Earth
- Today: coordinates, projections, colours, contours, wind vectors, and how to compare maps

# Ingredients of a Meteorological Map




1 Filled field: wind speed

2 Contour lines: geopotential height

3 Vectors: wind

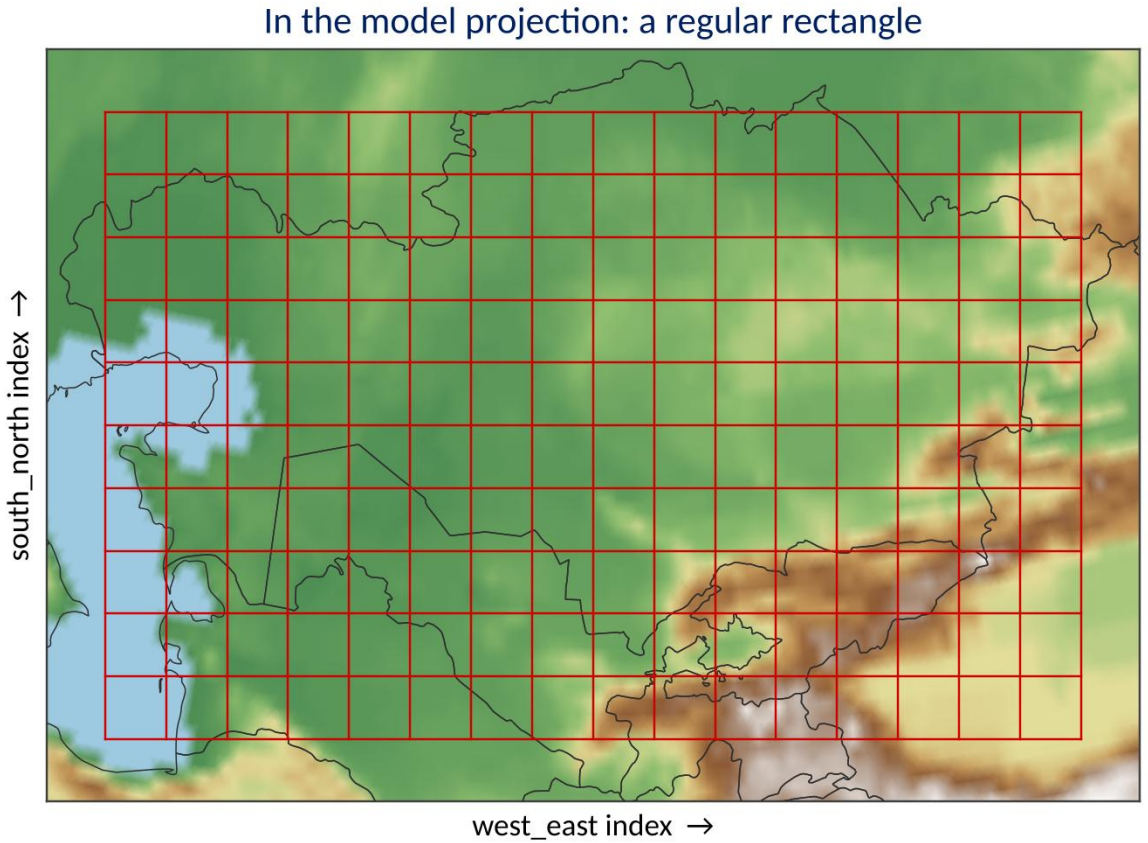
4 Geography: borders, coastlines

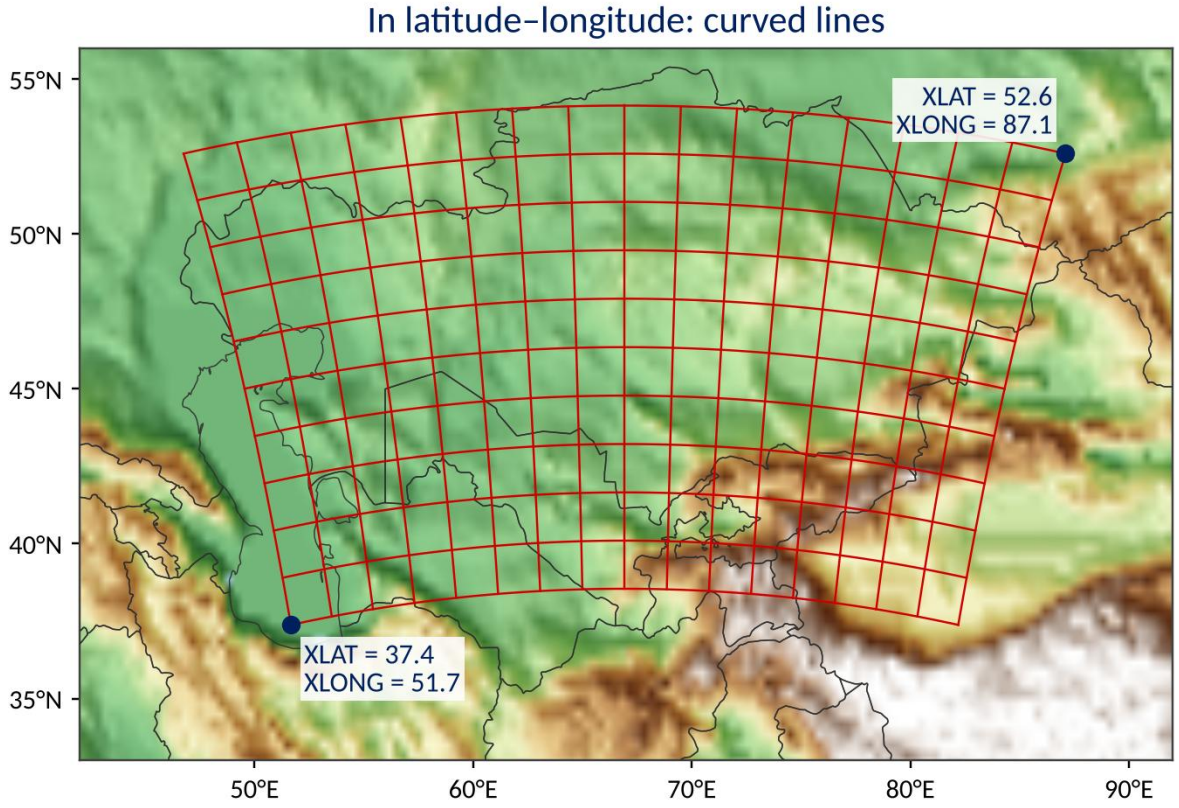
5 Colour bar with units

6 Title: variable, level, time

*Data: ERA-Interim reanalysis, monthly means (xarray tutorial dataset)*

# Coordinates: XLAT and XLONG Are Two-Dimensional




**Every grid point has its own latitude and longitude: XLAT(south_north, west_east), XLONG(south_north, west_east)**

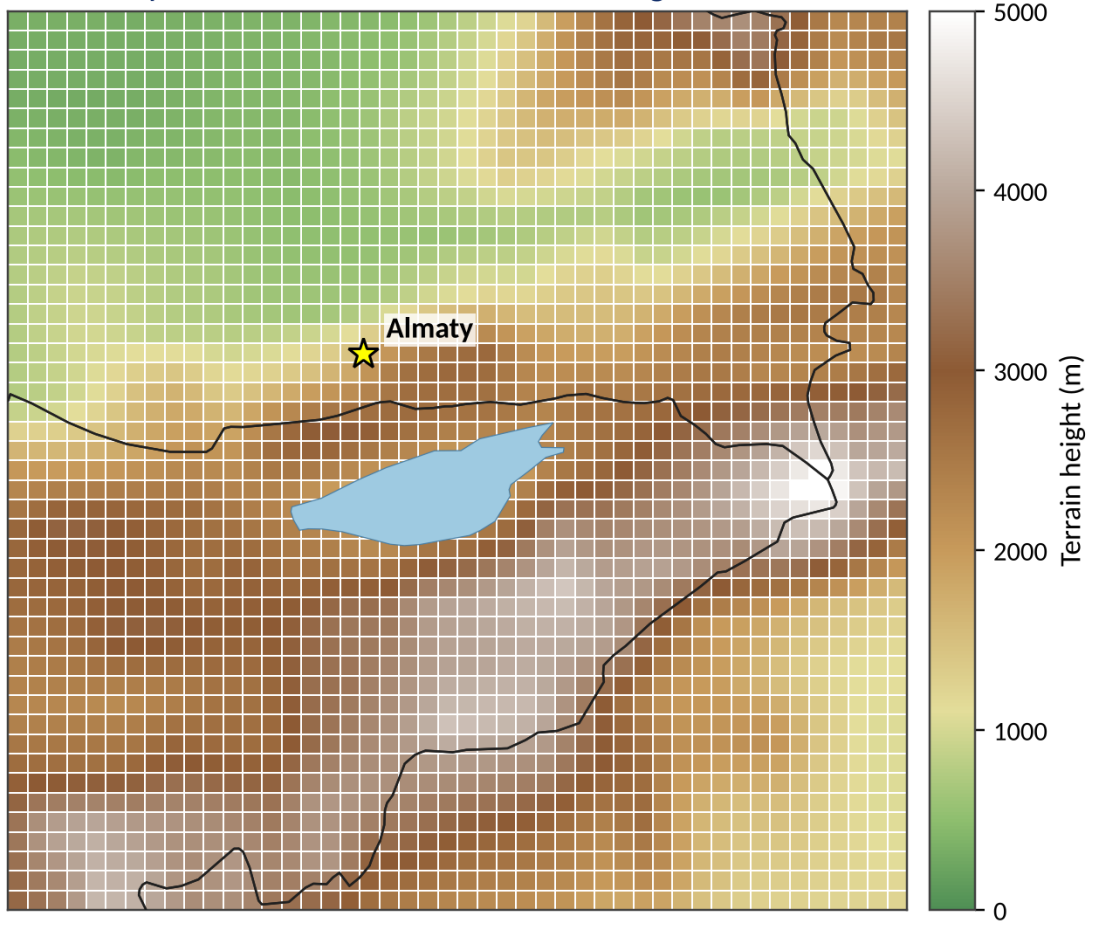
# Drawing a Map in Python: cartopy

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs, cartopy.feature as cfeature
from wrf import getvar, to_np, latlon_coords, get_cartopy

ter = getvar(nc, "ter")             # terrain height, m
lats, lons = latlon_coords(ter)   # 2-D coordinates

ax = plt.axes(projection=get_cartopy(ter))
cf = ax.contourf(to_np(lons), to_np(lats), to_np(ter),
                 levels=20, cmap="terrain",
                 transform=ccrs.PlateCarree())

ax.add_feature(cfeature.BORDERS); ax.coastlines()
plt.colorbar(cf, label="Terrain height (m)")
```

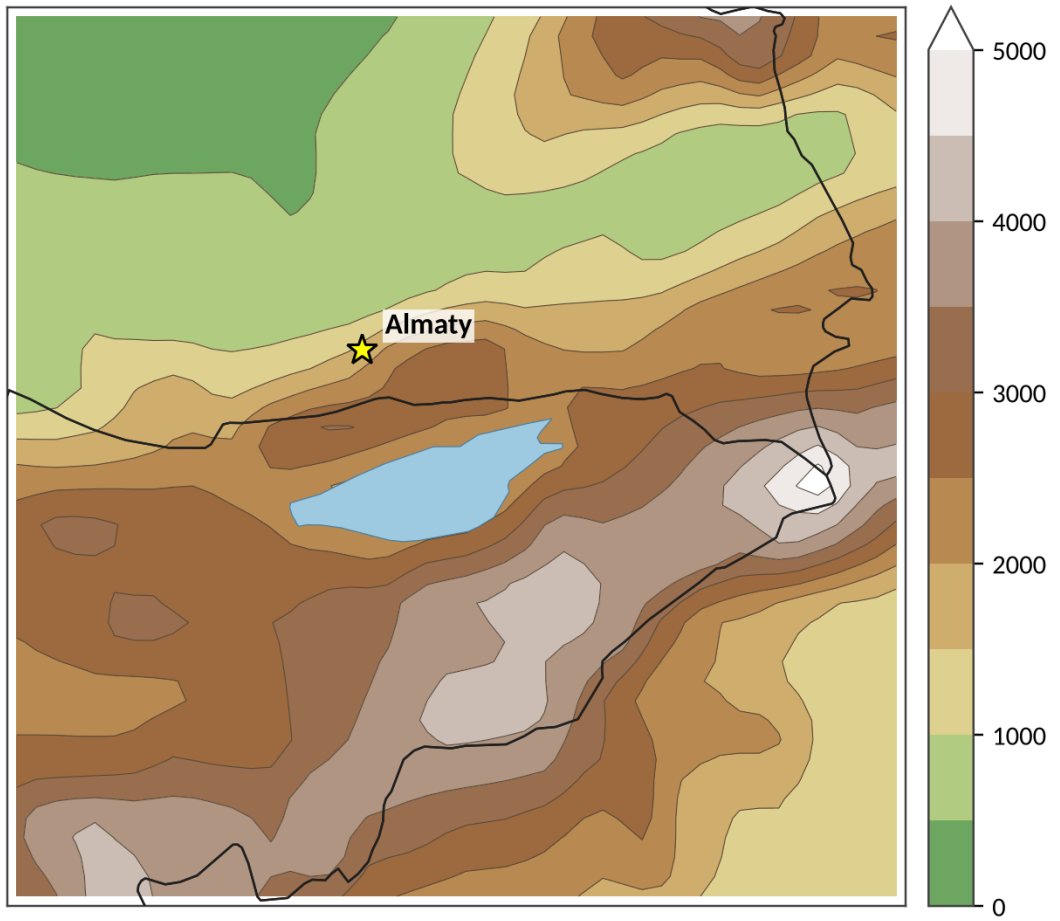
- latlon_coords: 2-D latitude and longitude
- get_cartopy: the projection of the model
- transform=PlateCarree(): “my coordinates are latitude and longitude”
- Forgetting transform is the most common mistake

# Filled Maps: pcolormesh and contourf

pcolormesh: one colour for each grid cell

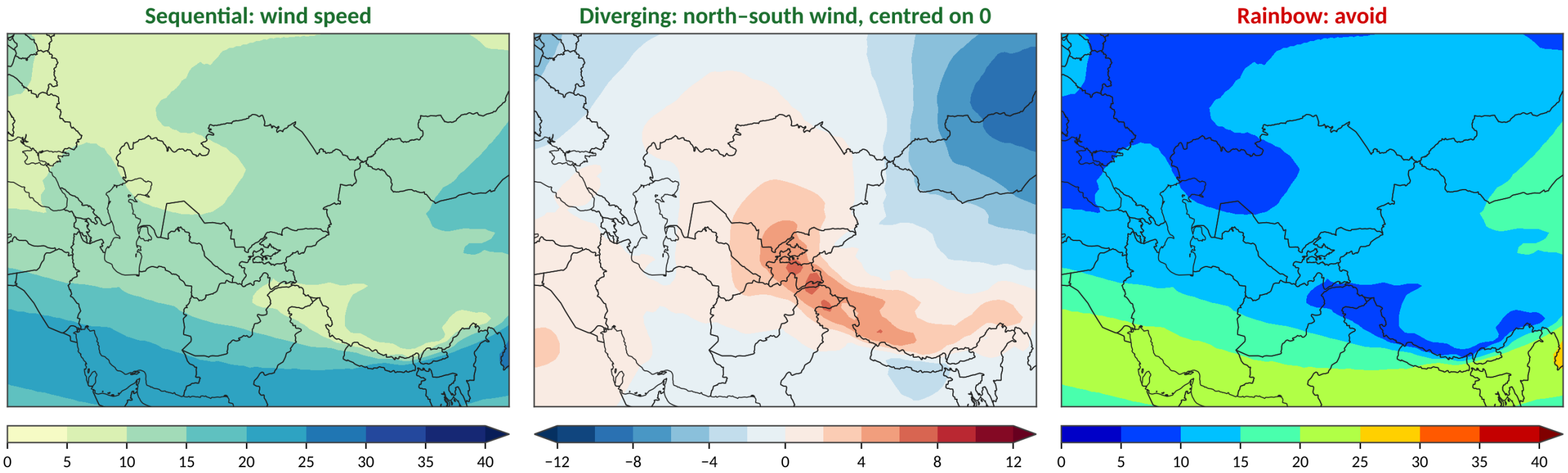



contourf: smooth bands between chosen levels




**pcolormesh: what the model really sees.** **contourf: smoother and easier to read**

# Choosing a Colour Scale

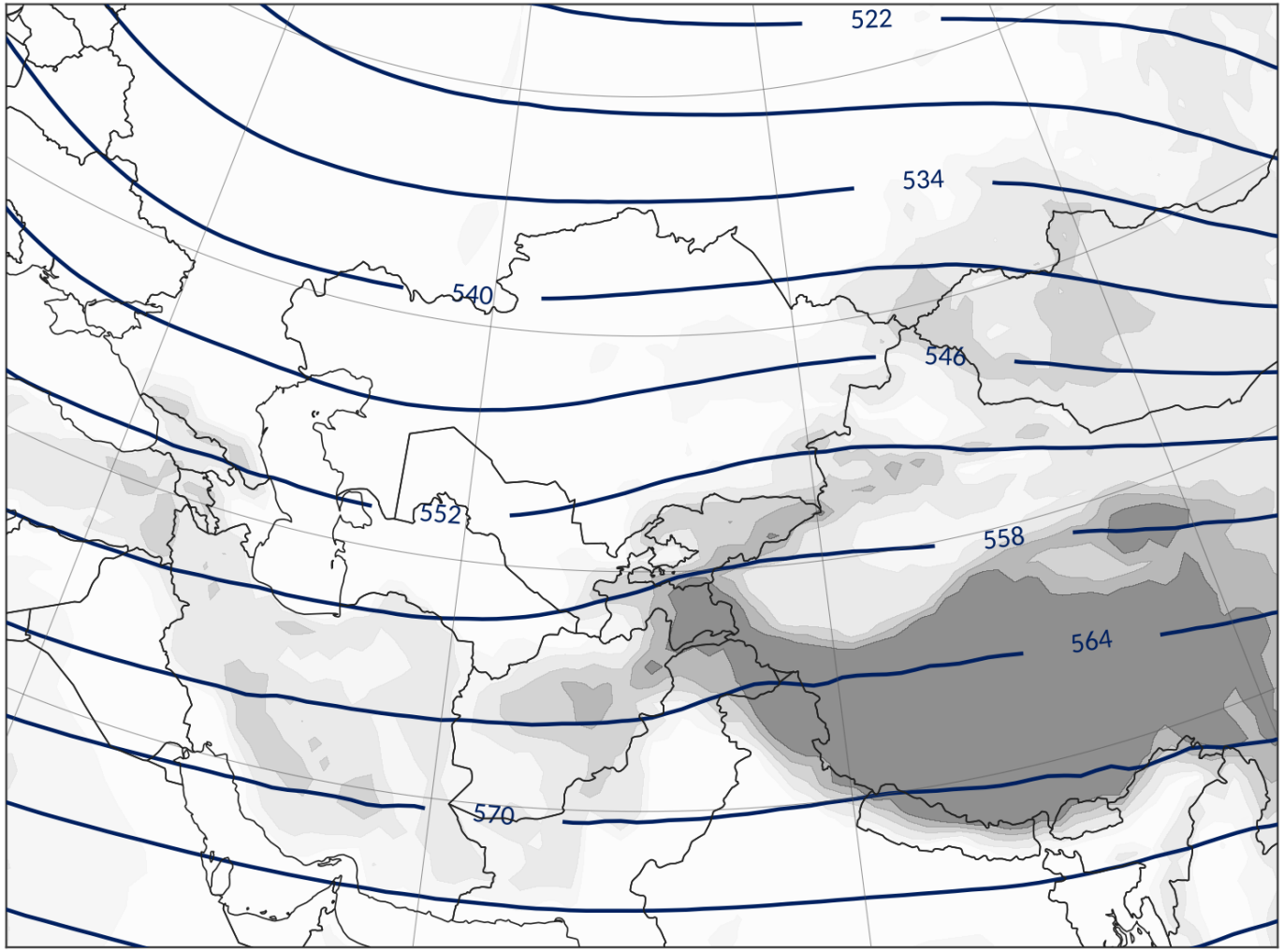



- Sequential: amounts (wind speed, precipitation, terrain). Diverging: differences, anomalies, values around zero
- Rainbow scales create false boundaries and hide real detail

*Data: ERA-Interim reanalysis, monthly means (xarray tutorial dataset), 500 hPa, January*

# Contour Lines

500 hPa geopotential height (dam), January mean; terrain in grey

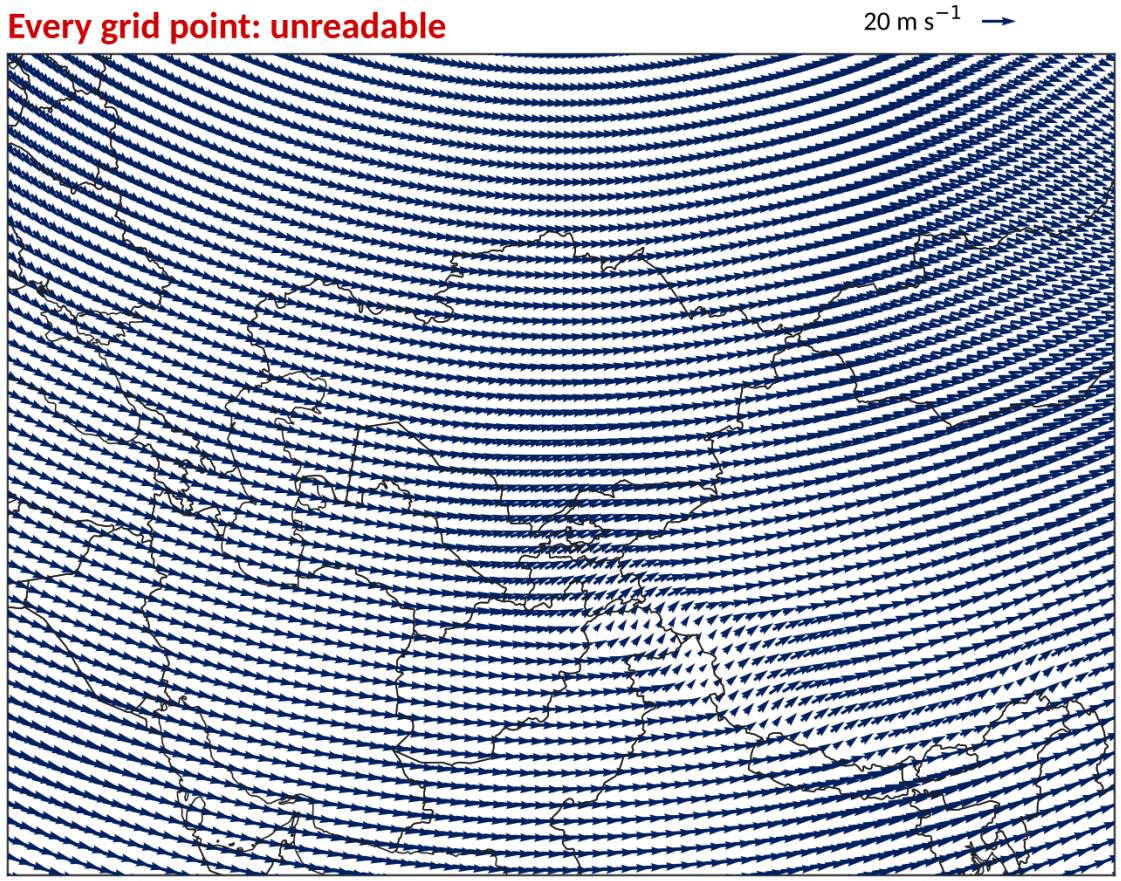



- Traditional for pressure and geopotential height
- Labels on the lines: no colour bar needed
- Lines close together: strong gradient, strong wind
- Can be drawn on top of a filled field

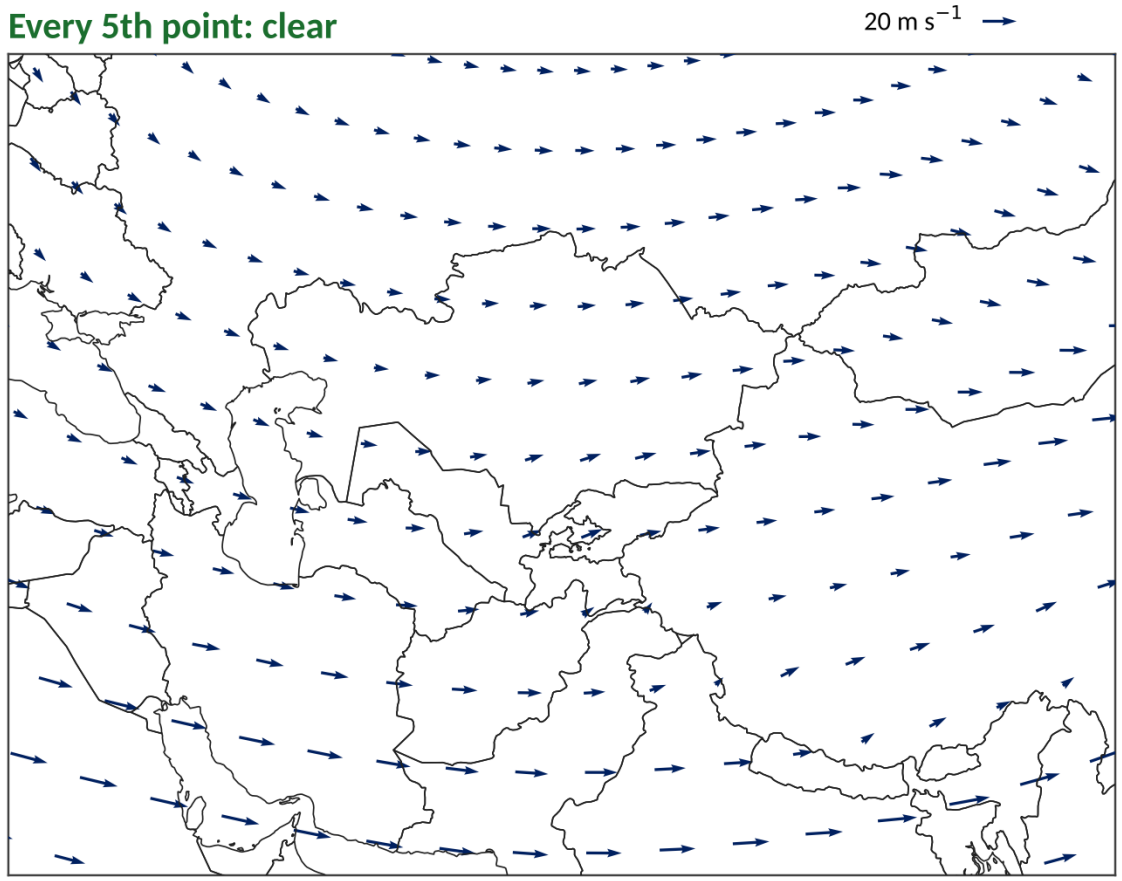
*Data: ERA-Interim reanalysis, monthly means (xarray tutorial dataset)*

# Wind: Vectors and Thinning

**Every grid point: unreadable**




**Every 5th point: clear**




**Draw every n-th arrow, and always add a reference arrow**

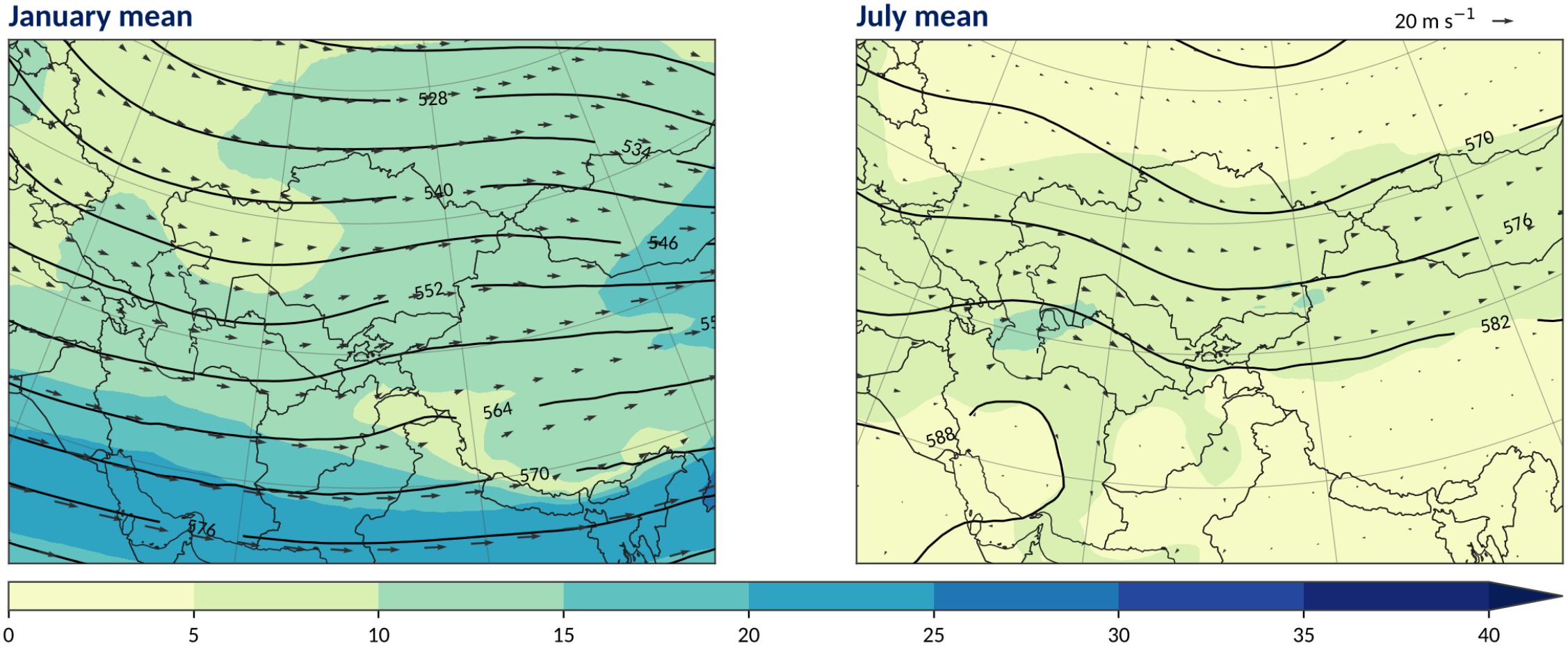
*Data: ERA-Interim reanalysis, monthly means (xarray tutorial dataset), 500 hPa, January*

# Code: Pressure Contours and Wind Vectors

```
from wrf import getvar, smooth2d, to_np

slp = smooth2d(getvar(nc, "slp", timeidx=12), 3)
u10, v10 = getvar(nc, "uvmet10", timeidx=12)

cs = ax.contour(to_np(lons), to_np(lats), to_np(slp),
                levels=range(980, 1050, 4), colors="k",
                transform=ccrs.PlateCarree())
ax.clabel(cs, fmt="%d")

k = 10        # thinning: every 10th point
ax.quiver(to_np(lons[::k, ::k]), to_np(lats[::k, ::k]),
          to_np(u10[::k, ::k]), to_np(v10[::k, ::k]),
          transform=ccrs.PlateCarree())
```

- slp: sea-level pressure, a diagnostic of getvar
- smooth2d: removes noise over mountains
- uvmet10: wind rotated to east and north
- [::k, ::k]: every k-th point

# Comparing Maps: January and July




**Same colour scale, same contour interval, same arrow scale in both panels**

*Data: ERA-Interim reanalysis, monthly means (xarray tutorial dataset)*

# Time: Which Moment Do We Plot?

**A single moment**

getvar(nc, "T2", timeidx=12)

Times in the file are in UTC

**An average over a period**

For example, the monthly mean for 13–17 local time and for 03–07 local time

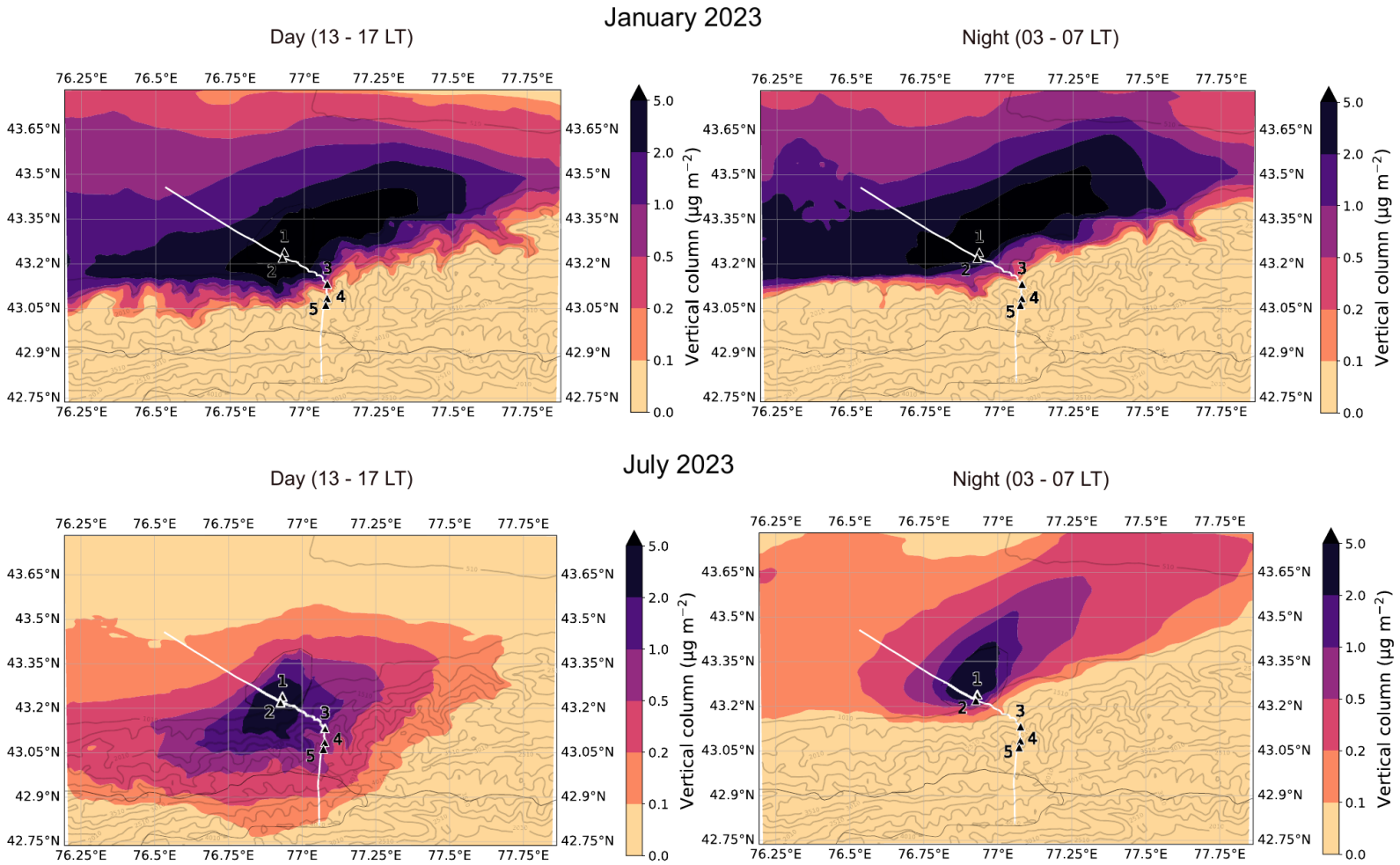
**An accumulated value**

Precipitation: total at the end minus total at the beginning

*Averaging periods as in Tursumbayeva et al. (2026))*

**Always state the time on the figure: a map without a time is a map without a meaning**

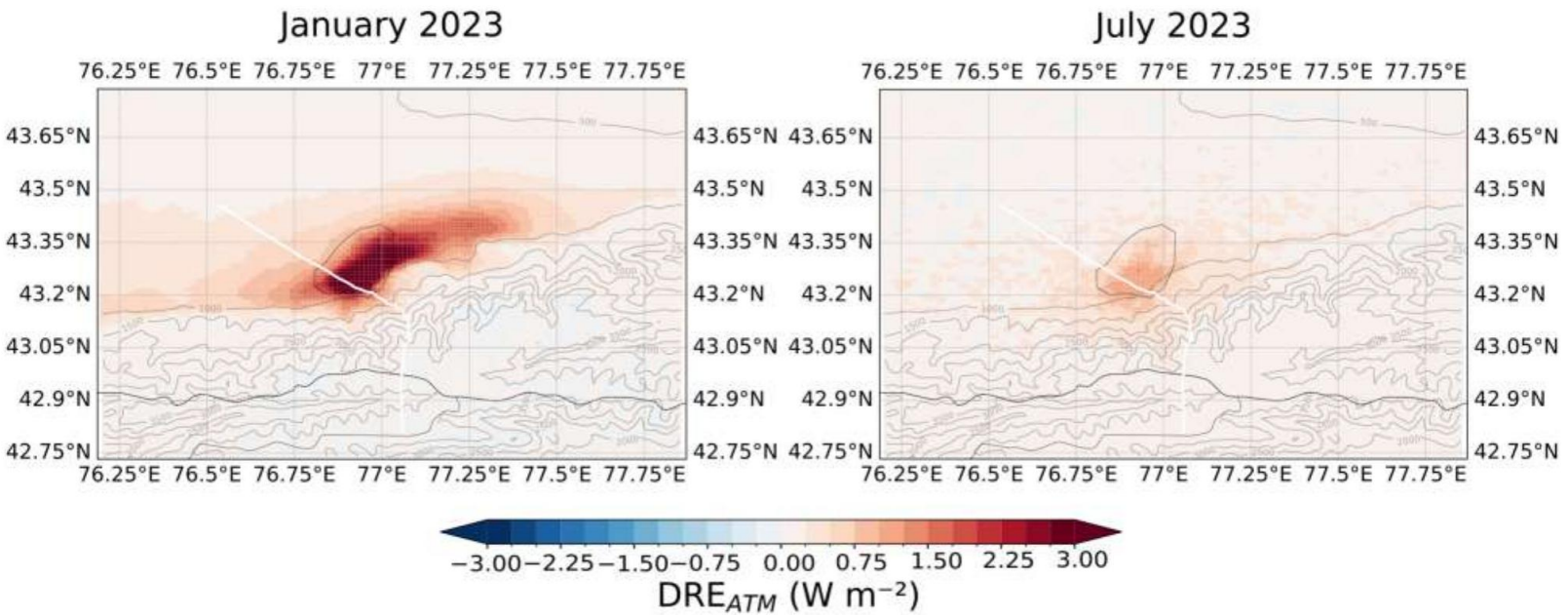
# Research Example: Maps on the 1-km Domain




- Filled contours, sequential colour scale
- Levels are denser at low values
- Thin grey lines: terrain contours
- One colour scale for all four panels
- January: BC fills the basin. July: weaker, but it reaches the mountains

*Column black carbon, WRF-CHIMERE, 1 km. Source: Tursumbayeva et al. (2026), Meteorological Applications, 33, e70183*

# Research Example: A Difference Map




*Direct radiative effect of BC in the atmosphere. Source: Tursumbayeva et al. (2026), Meteorological Applications, 33, e70183*

- Difference between two simulations: with and without the effect of BC on radiation
- Diverging colour scale, centred on zero
- Only the effect of interest remains
- Stronger in January than in July

**Difference map = simulation A − simulation B: only the effect of what differs between them remains**

# Checklist for a Good Map

**Title**

Variable, level and time

**Colour bar**

Always with units

**Colour scale**

Sequential or diverging; no rainbow

**Wind**

Thinned arrows and a reference arrow

**Comparison**

The same scale in all panels

**Not overloaded**

At most three layers on one map

# Summary

**Key points**

- A map needs the data and the 2-D coordinates XLAT and XLONG
- matplotlib + cartopy, in the projection of the model; transform=PlateCarree()
- Scalars: filled colours or contour lines.  Wind: thinned arrows
- Choose the colour scale, the time and the title carefully
- Compare maps only with the same scale

**Next lecture**

Vertical structure of the atmosphere

Model levels, pressure levels and vertical profiles

**Thank you for your attention!**

# Self-Check Questions

1. Which elements must a meteorological map contain?
2. Why are XLAT and XLONG two-dimensional in WRF output?
3. What does transform=ccrs.PlateCarree() tell cartopy, and what happens if it is forgotten?
4. When is a sequential colour scale appropriate and when a diverging one? Why should rainbow scales be avoided?
5. Why are wind arrows thinned, and why is a reference arrow needed?
6. Which rules must be followed when comparing two maps, and what does a difference map show?

# References


1. Skamarock, W. C., et al. (2019). A Description of the Advanced Research WRF Model Version 4. NCAR Technical Note NCAR/TN-556+STR. https://doi.org/10.5065/1dfh-6p97
2. Tursumbayeva, M., Ciarelli, G., Di Antonio, L., Bettineschi, M., Kissebayev, D., & Baimatova, N. (2026). Modelling the Black Carbon Variability and Dynamics in Almaty, Kazakhstan, During Winter and Summer Seasons. Meteorological Applications, 33, e70183. https://doi.org/10.1002/met.70183
3. NSF NCAR Mesoscale and Microscale Meteorology Laboratory. WRF Tutorial: recorded lectures (YouTube, wrfhelp NCAR-MMM, 2021) and lecture slides, including “Post-Processing Tools: Python”. https://www2.mmm.ucar.edu/wrf/wrf_tutorial/lectures.html
4. Ladwig, W. (2017). wrf-python (Version 1.3) [Software]. UCAR/NCAR. https://doi.org/10.5065/D6W094P1
5. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55
6. Met Office. Cartopy: a cartographic Python library with a Matplotlib interface. https://scitools.org.uk/cartopy
7. Crameri, F., Shephard, G. E., & Heron, P. J. (2020). The misuse of colour in science communication. Nature Communications, 11, 5444. https://doi.org/10.1038/s41467-020-19160-7
8. Dee, D. P., et al. (2011). The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Quarterly Journal of the Royal Meteorological Society, 137(656), 553–597. https://doi.org/10.1002/qj.828
9. Figure data: ERA-Interim monthly means from the xarray tutorial datasets (github.com/pydata/xarray-data); NOAA NGDC ETOPO global relief model; Natural Earth.