Spectral Transform

Andreas Mueller

ECMWF, Bonn, Germany

andreas.mueller@ecmwf.int

Overview

10 minutes	• Fourier transform • Spectral transform
30 minutes	• hands-on exercises
20 minutes	• aliasing • parallelization • performance
rest	• time for questions

IFS (Integrated Forecast System)

technology applied at ECMWF for
the last 40 years

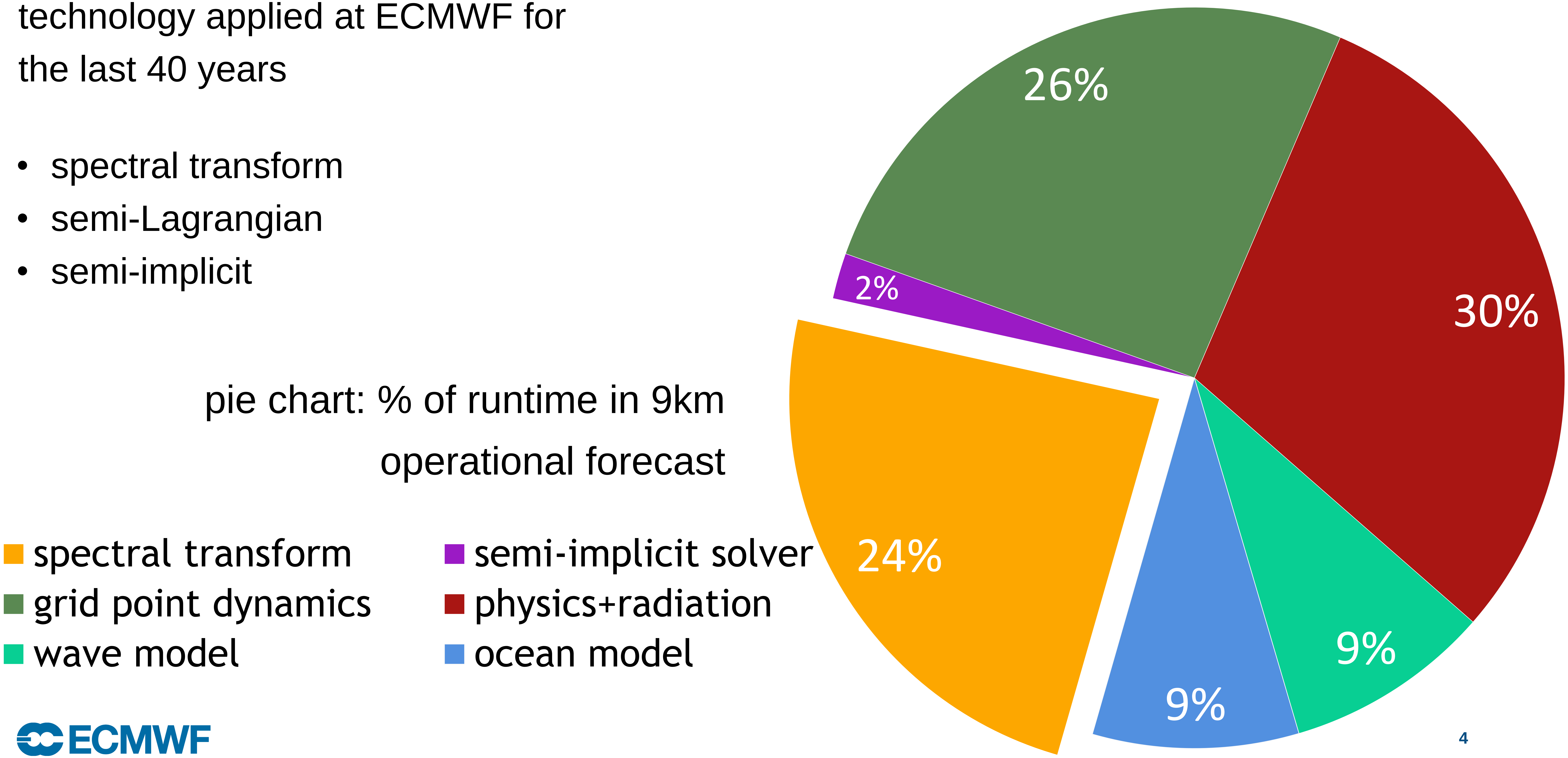
- spectral transform
- semi-Lagrangian
- semi-implicit

IFS (Integrated Forecast System)

technology applied at ECMWF for the last 40 years

- spectral transform
- semi-Lagrangian
- semi-implicit

pie chart: % of runtime in 9km operational forecast



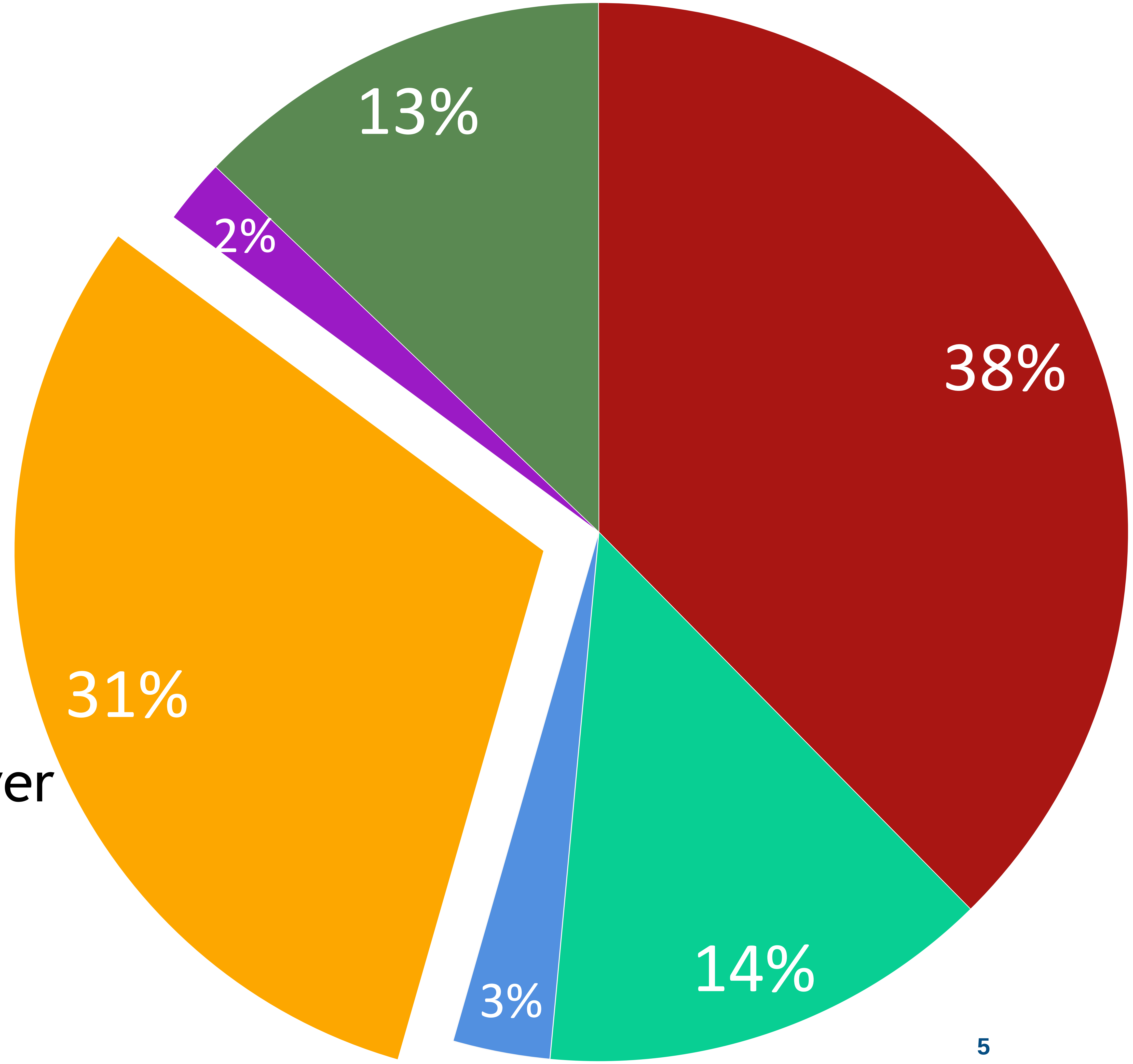
ECMWF

IFS (Integrated Forecast System)

technology applied at ECMWF for the last 40 years

- spectral transform
- semi-Lagrangian
- semi-implicit

pie chart: % of runtime in 5km forecast (future operational)



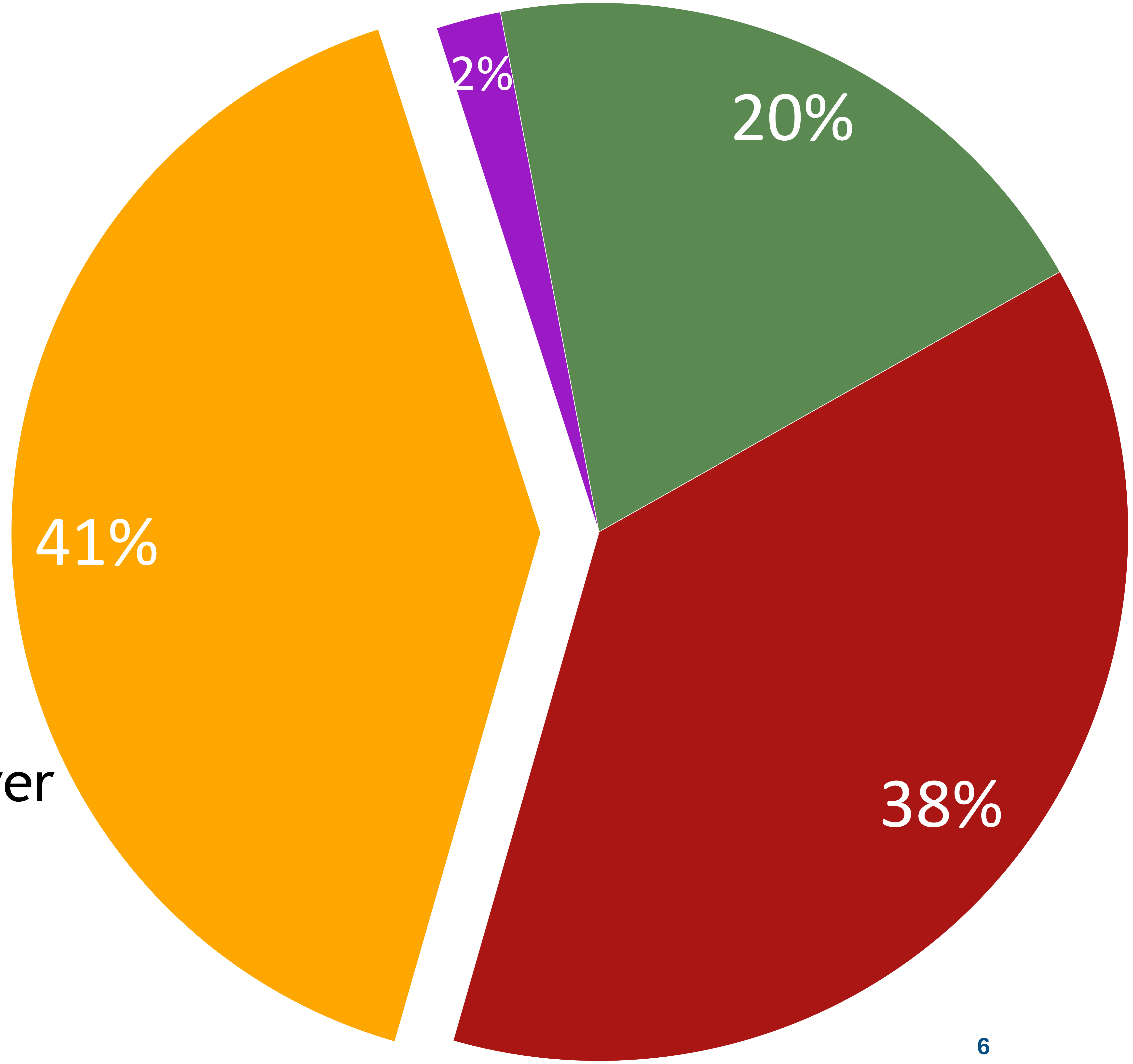
ECMWF

IFS (Integrated Forecast System)

technology applied at ECMWF for the last 40 years

- spectral transform
- semi-Lagrangian
- semi-implicit

pie chart: % of runtime in 1.25km forecast (experiment, no ocean)

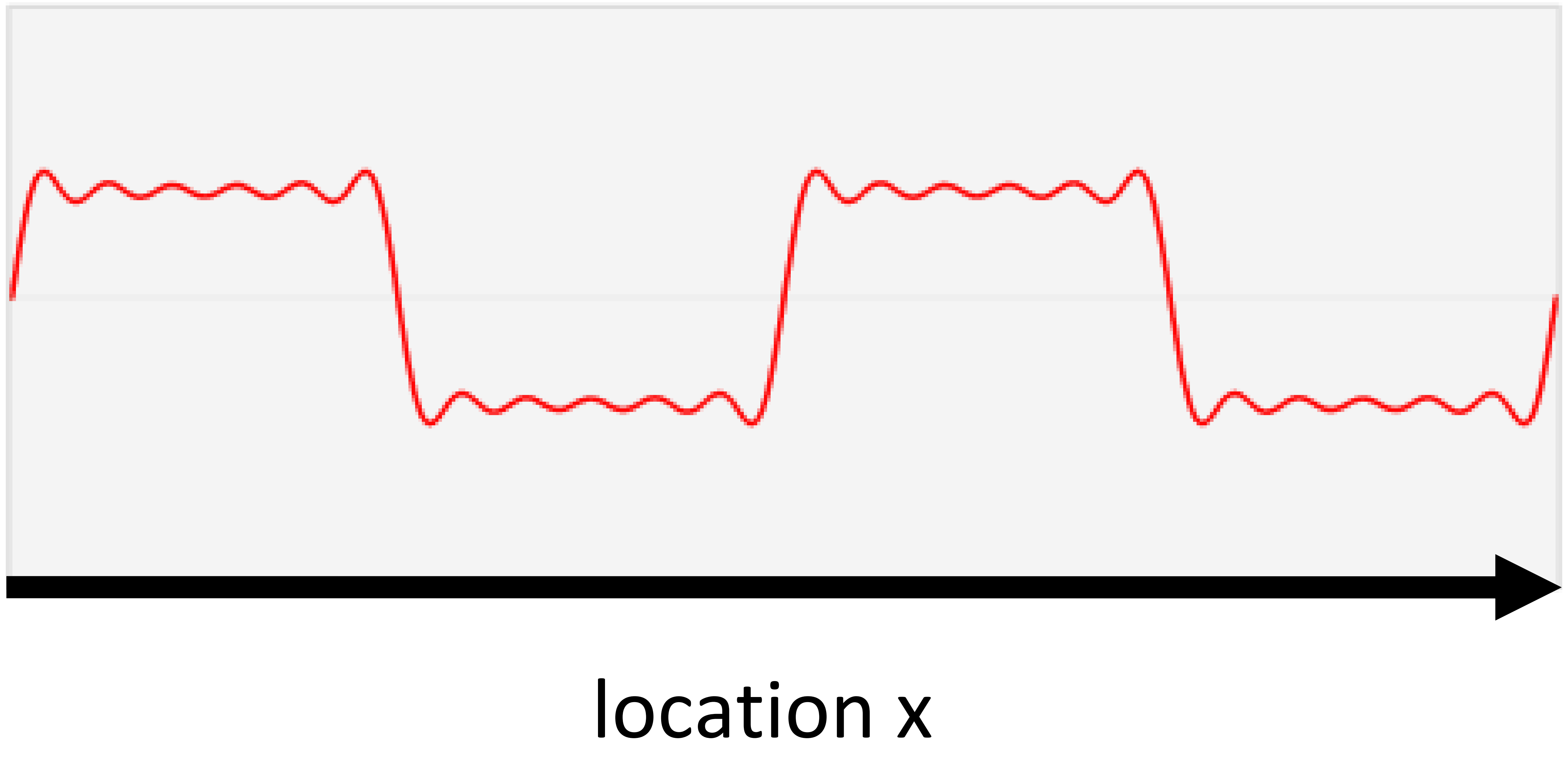


- spectral transform
- semi-implicit solver
- grid point dynamics
- physics+radiation
- wave model
- ocean model

ECMWF

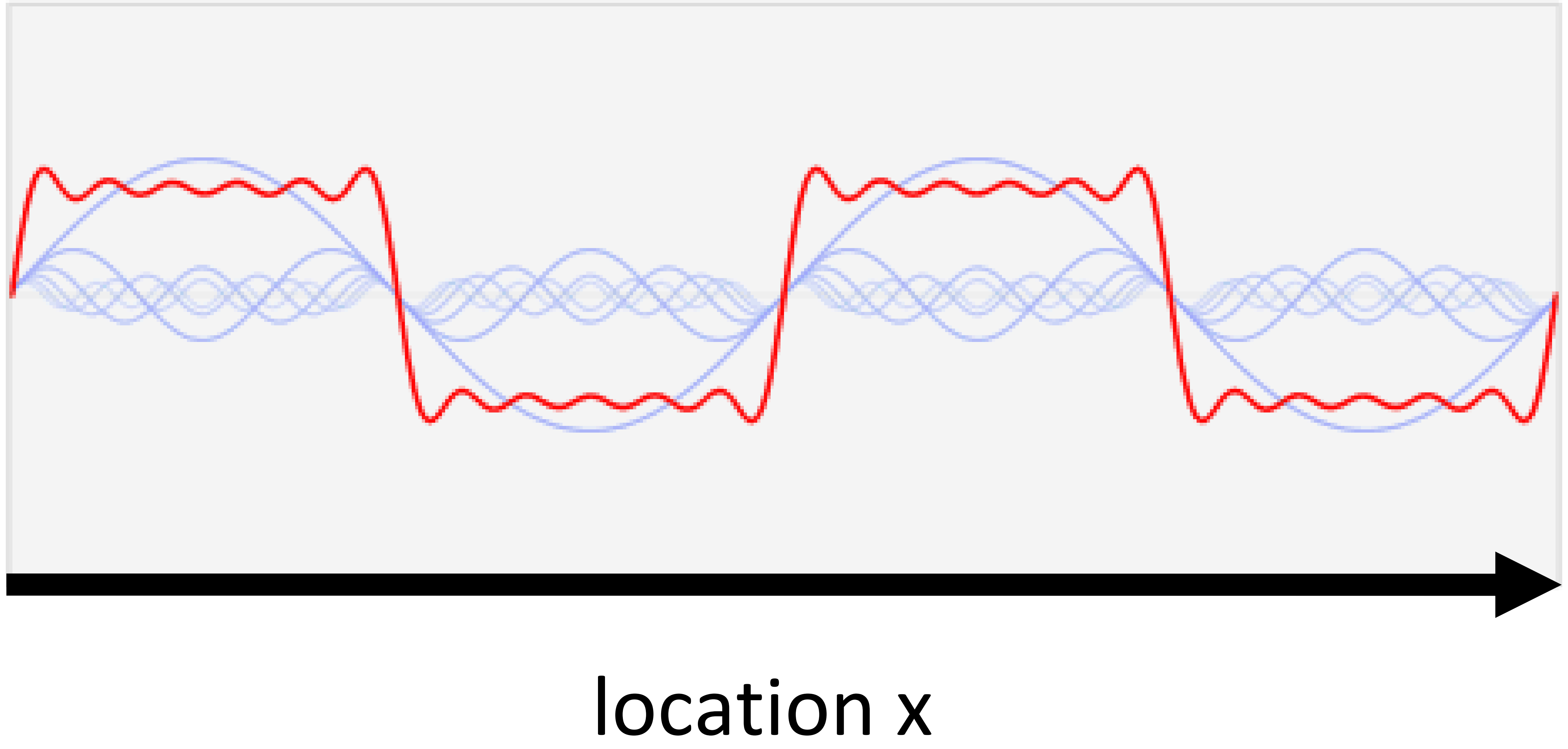
Fourier transform

Fourier transform = Spectral transform in 1D



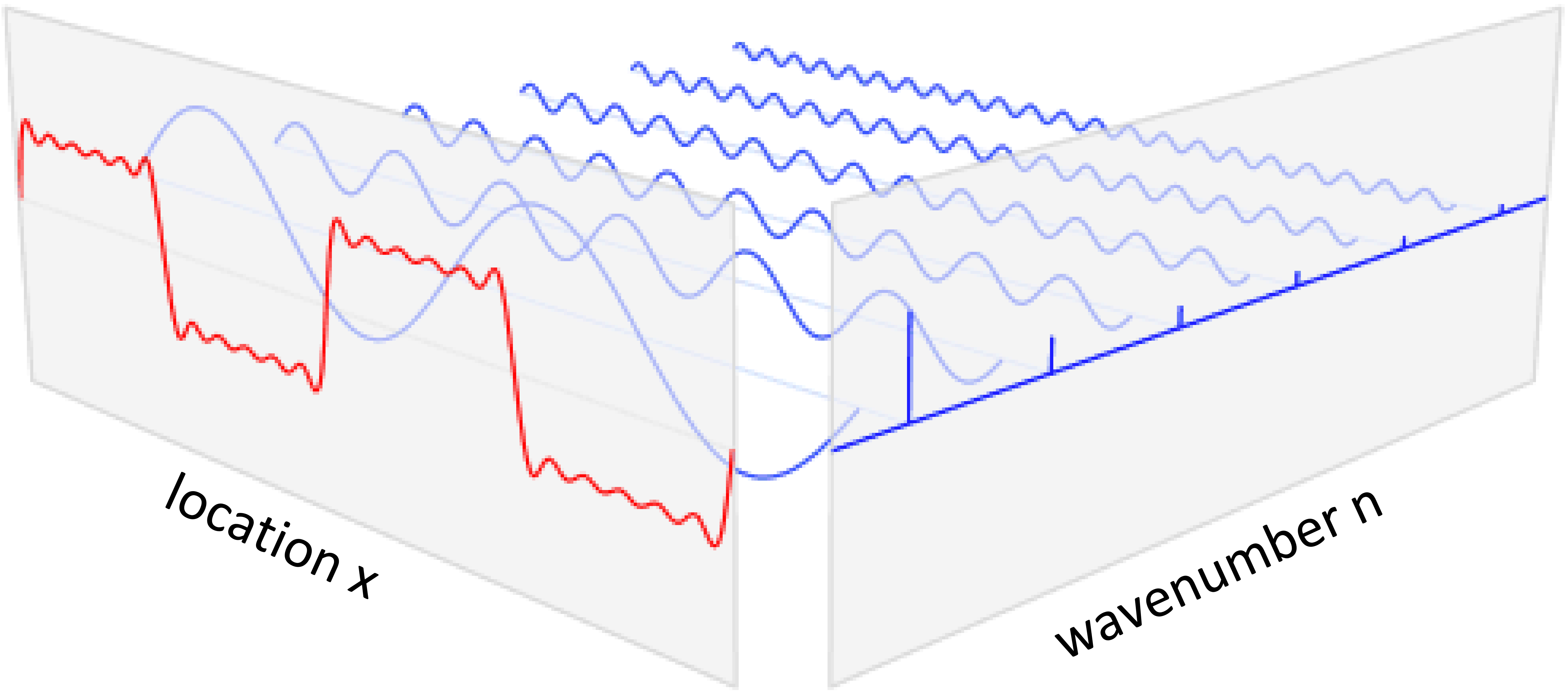
Fourier transform

Fourier transform = Spectral transform in 1D



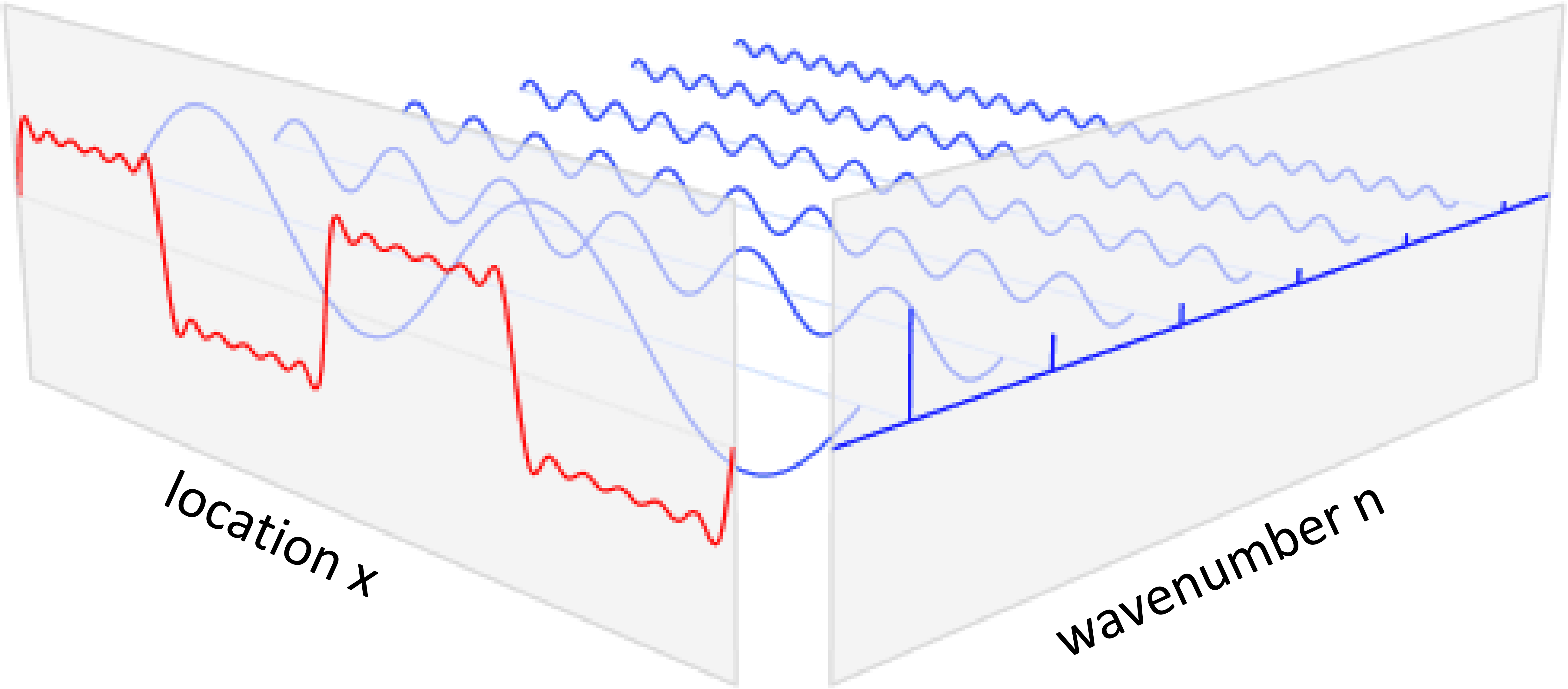
Fourier transform

Fourier transform = Spectral transform in 1D



grid point space **Fourier space**

Fourier transform



function in grid point space

Fourier coefficients

$$f(x) = \sum_n f_n \cdot \mathrm{e}^{-2\pi \mathrm{i}\, n\, x}$$

Fourier transform

function in grid point space

Fourier coefficients

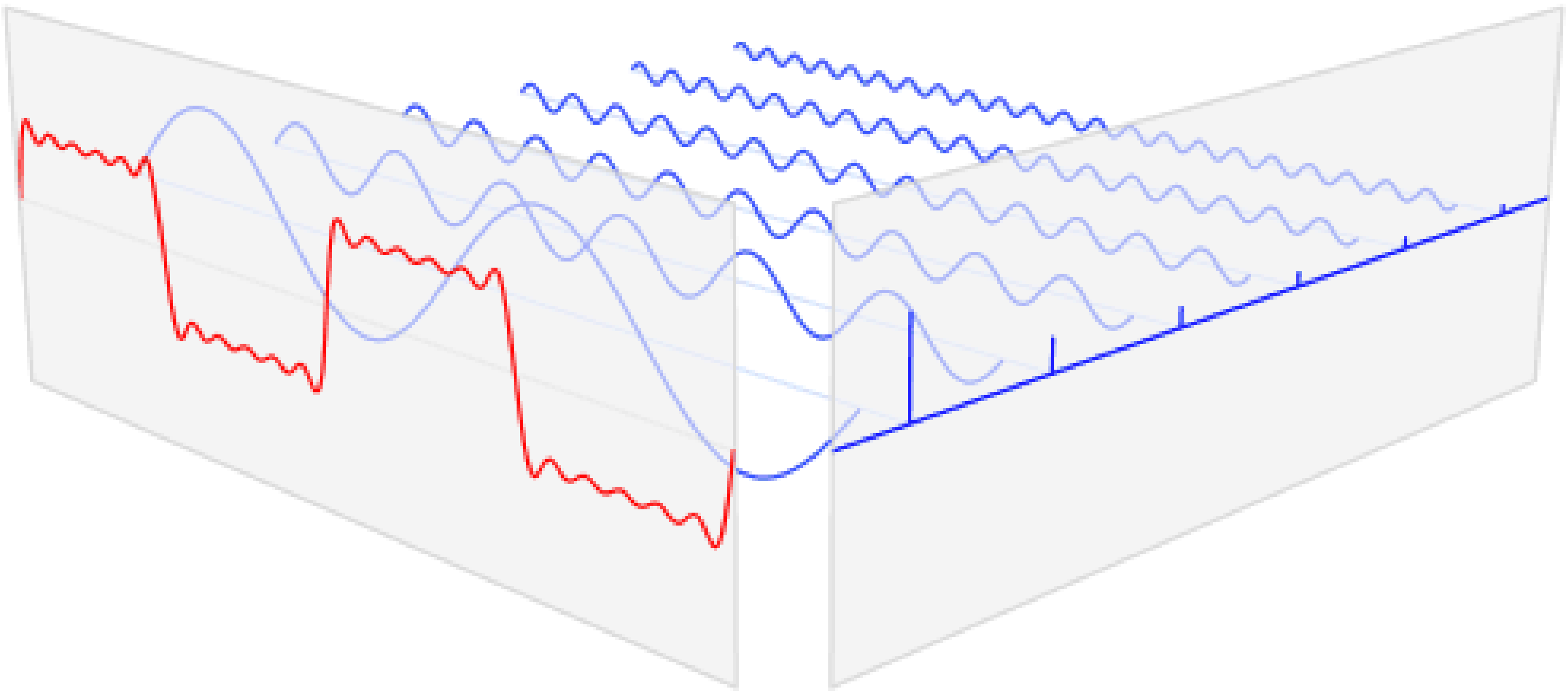
$$f(x) = \sum_n f_n \cdot \mathrm{e}^{-2\pi \mathrm{i}\, n\, x}$$

differentiation

simple multiplication

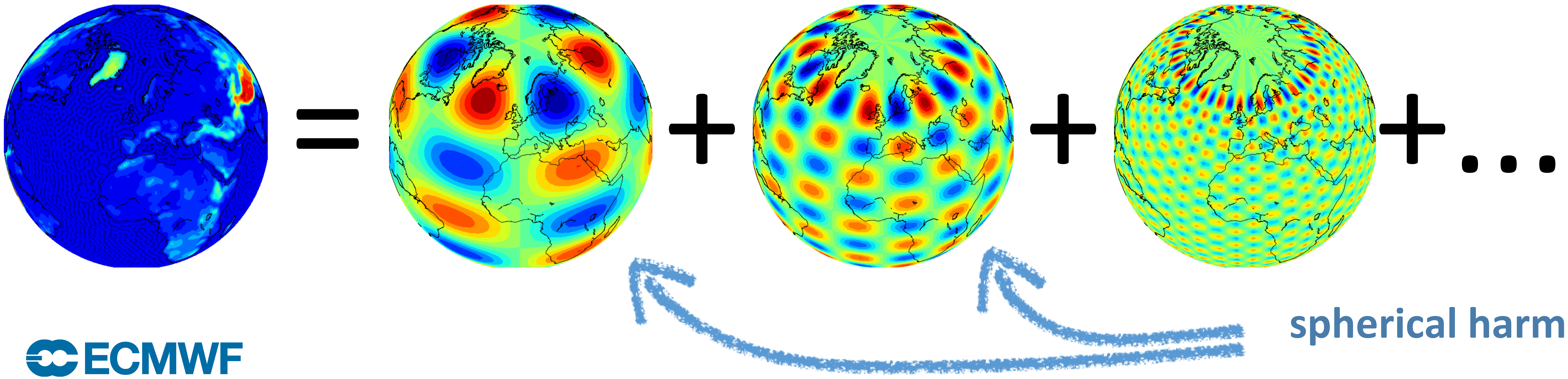
$$\frac{\mathrm{d}f(x)}{\mathrm{dx}} = \sum_n (-2\pi \mathrm{i}\, n\, f_n) \cdot \mathrm{e}^{-2\pi \mathrm{i}\, n\, x}$$

on the sphere: spectral transform



grid point space

spectral space



spherical harmonics

on the sphere: spectral transform

Spectral coefficients

Spherical harmonics

Latitude

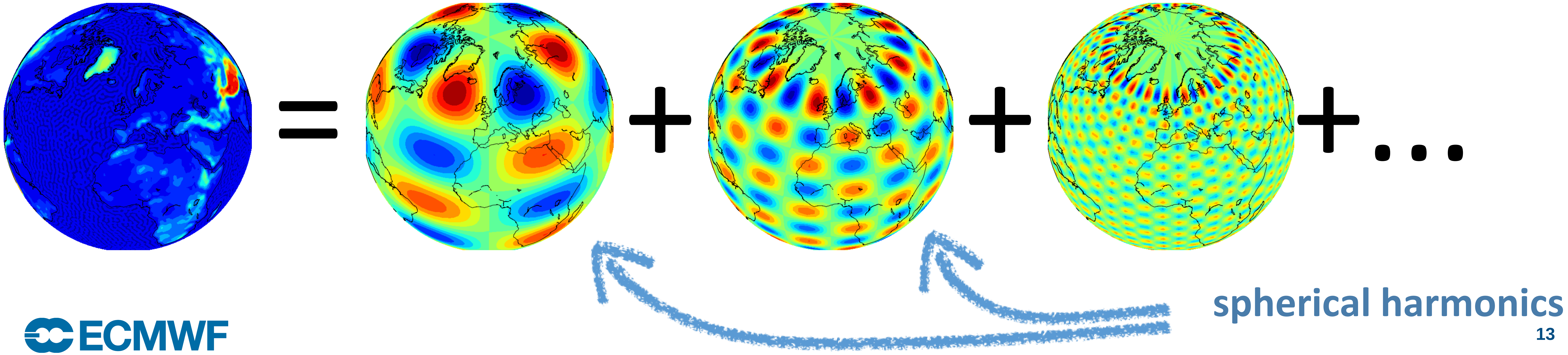
Longitude

Grid point variable

$$f(\phi,\lambda) = \Re\left(\sum_{m=0}^{M}\sum_{n=m}^{M} f_{m,n}\, Y_n^m(\phi,\lambda)\right)$$

m: zonal wavenumber
n: total wavenumber
M: truncation

grid point space **spectral space**



ECMWF

on the sphere: spectral transform

Spectral coefficients

Spherical harmonics

Latitude

Longitude

Grid point variable

$$f(\phi, \lambda) = \Re\left(\sum_{m=0}^{M}\sum_{n=m}^{M} f_{m,n}\, Y_n^m(\phi, \lambda)\right)$$

m: zonal wavenumber
n: total wavenumber
M: truncation

Legendre polynomials

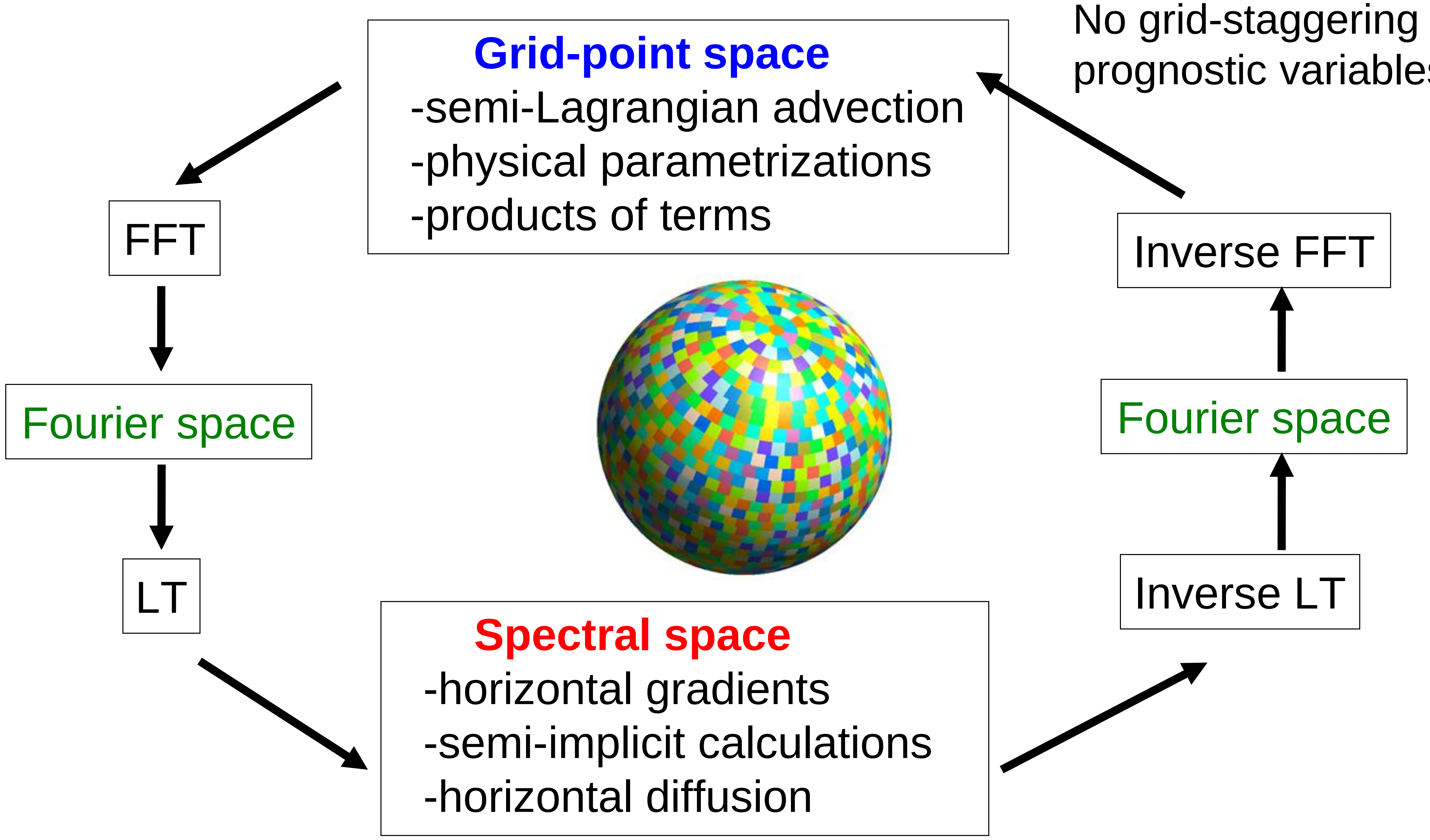
$$f(\phi, \lambda) = \Re\left(\sum_{m=0}^{M} \mathrm{e}^{im\lambda} \underbrace{\sum_{n=m}^{M} f_{m,n} P_n^m(\phi)}_{\text{Legendre transform}}\right)$$

Legendre transform

Fourier transform

ECMWF

time step in IFS



No grid-staggering of prognostic variables

FFT: Fast Fourier Transform, LT: Legendre Transform

hands-on session

for everyone: interactive web-app about spectral transform

open in a browser: anmrde.github.io/spectral

optional: Python course

open in Jupyterlab in your browser: /NMcourse/spectral/solution.ipynb

Exercises are getting more difficult. Feel free to skip exercises as you want. The full Python course is designed to fill 20 hours.

files: exercises.ipynb: Python notebook with exercises
solution.ipynb: notebook including sample solutions

ECMWF Jupyterhub (16GB of RAM) or personal Linux computer:
https://github.com/anmrde/spectral/tree/master/jupyter

aliasing

Issue: multiplication of two variables produces shorter waves than grid can handle

aliasing

wave generated in spectral space

Issue: multiplication of two variables produces shorter waves than grid can handle

ECMWF

aliasing

Issue: multiplication of two variables produces shorter waves than grid can handle

aliasing

Issue: multiplication of two variables produces shorter waves than grid can handle

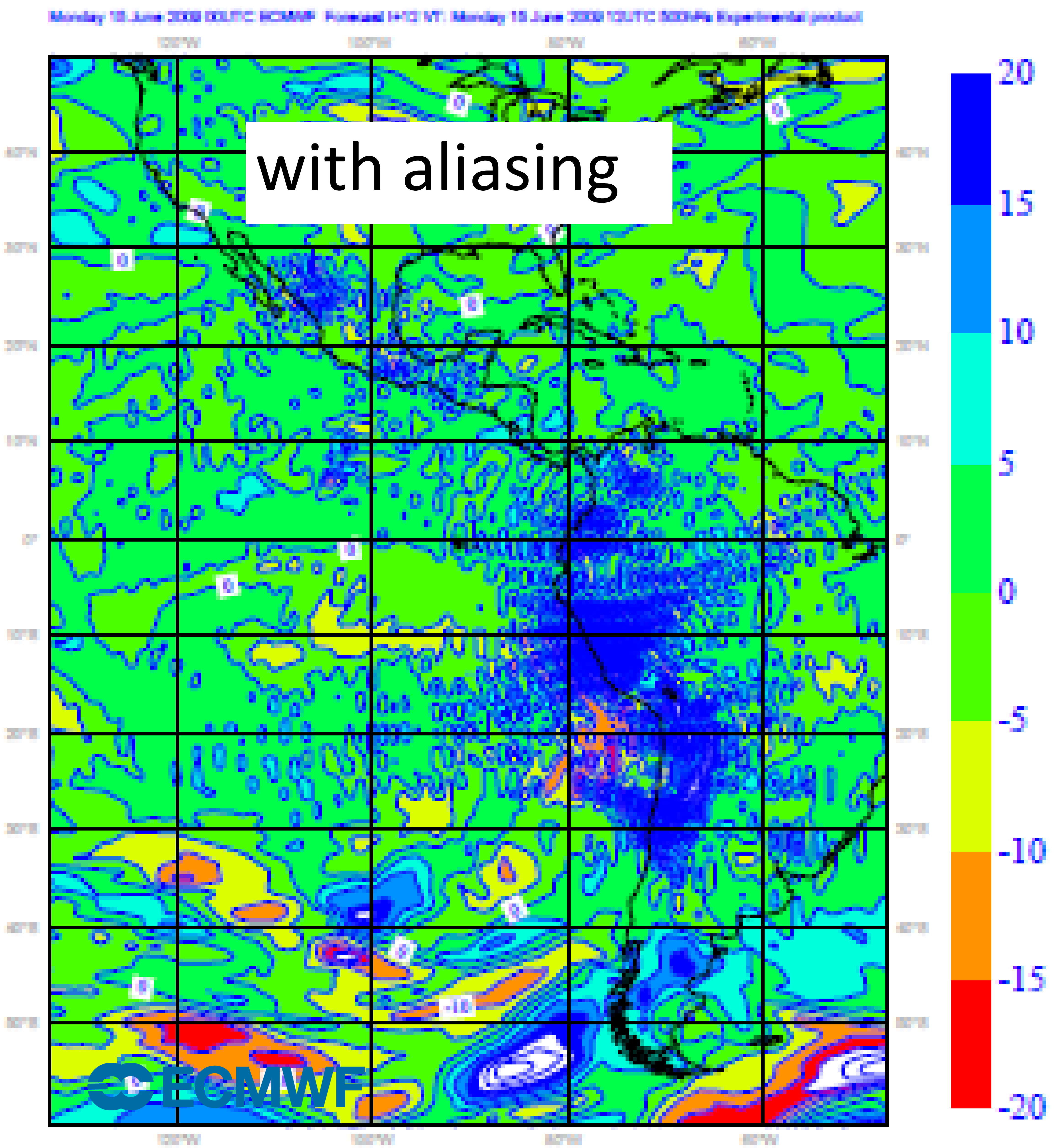
aliasing

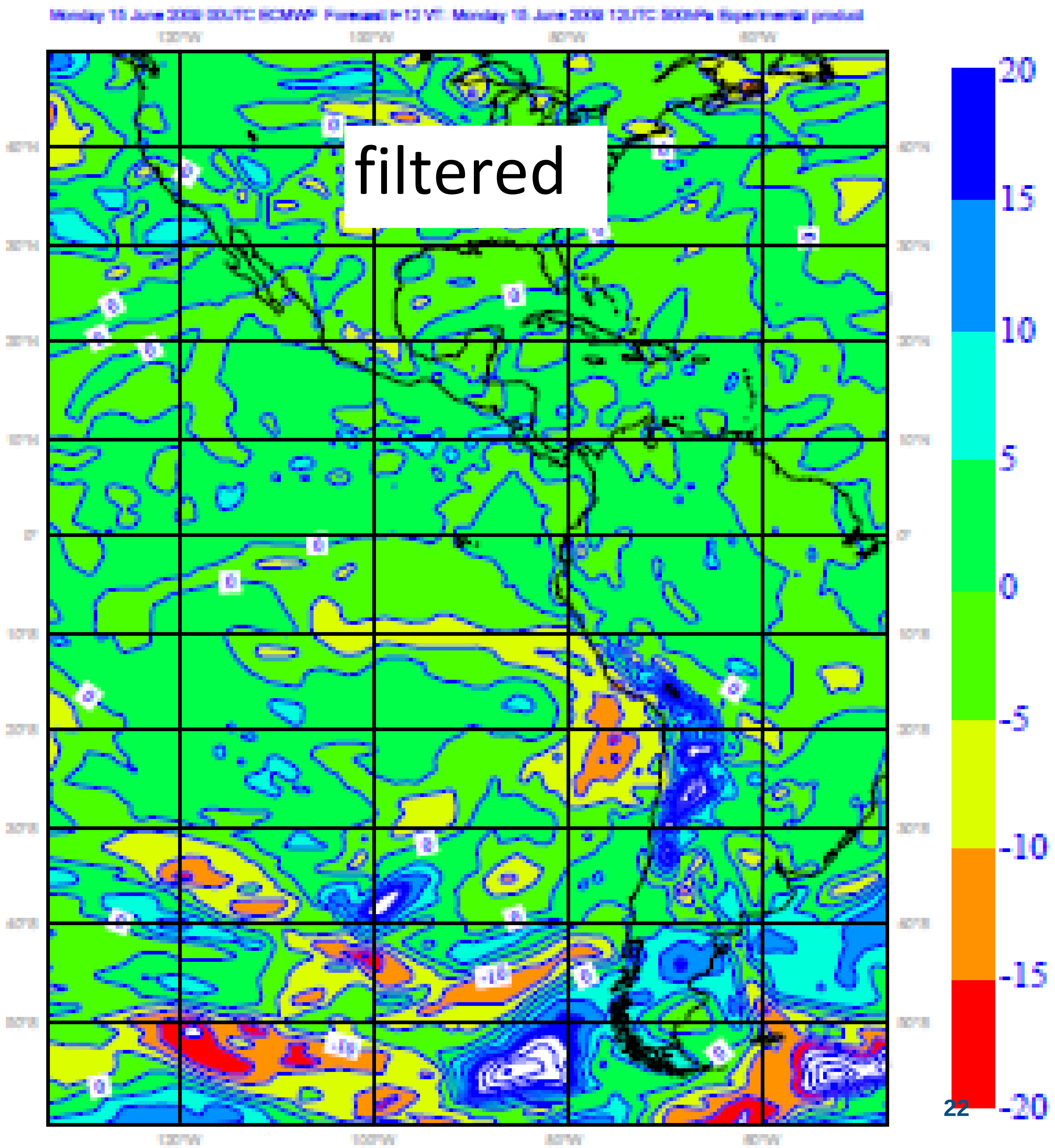
Issue: multiplication of two variables produces shorter waves than grid can handle

ECMWF

aliasing example

500hPa adiabatic zonal wind tendencies (T159)

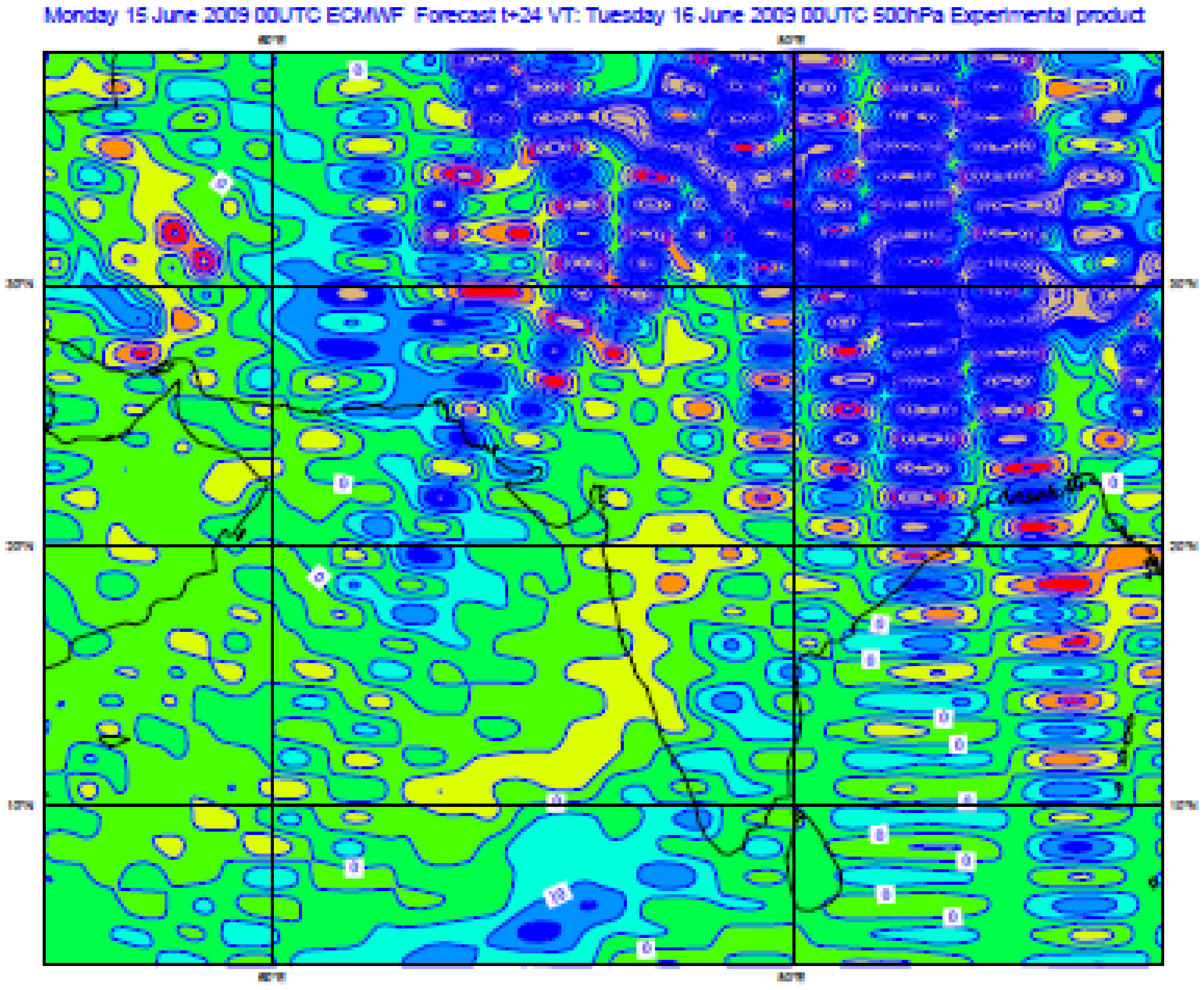




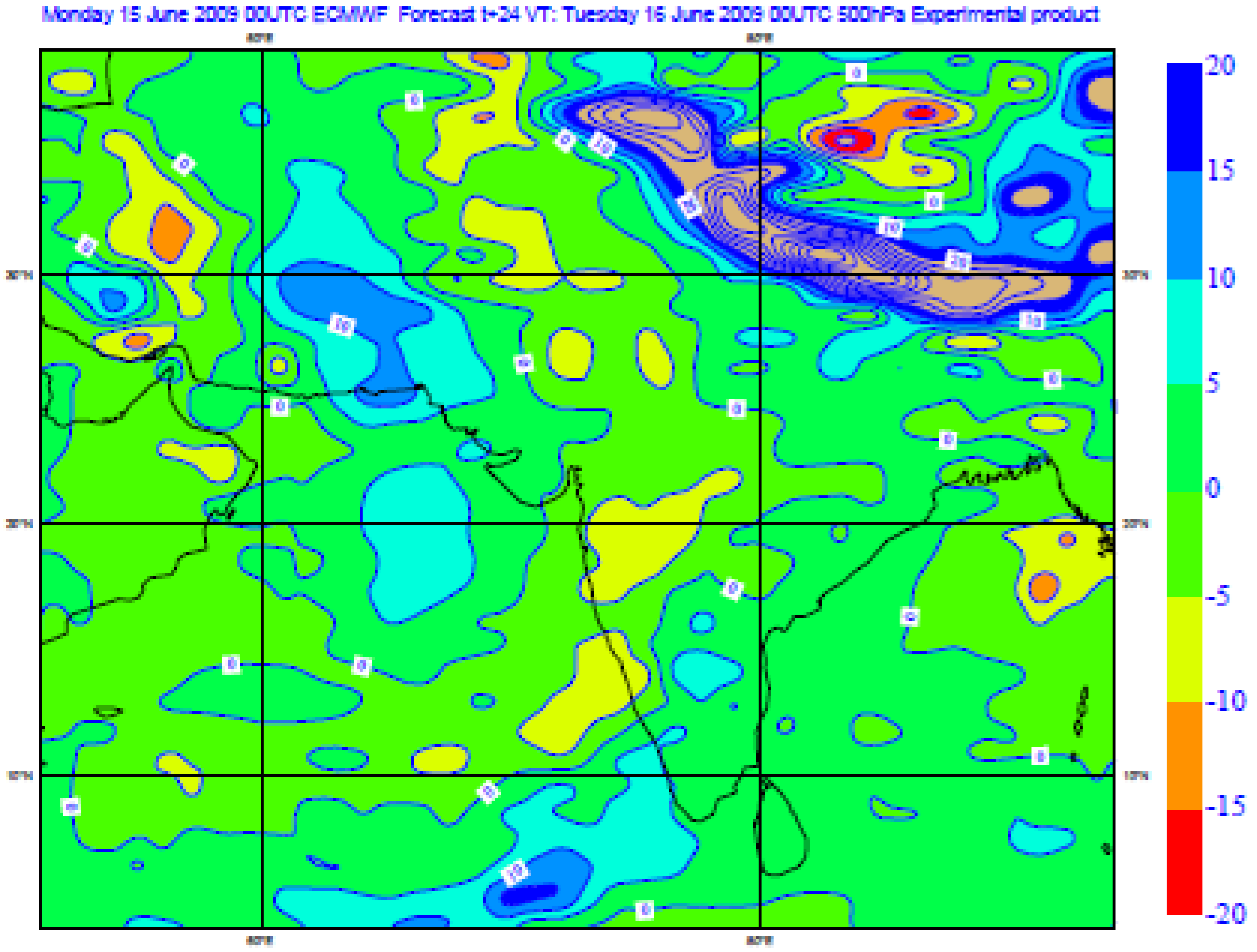
aliasing example

500hPa adiabatic meridional wind tendencies (T159)

with aliasing

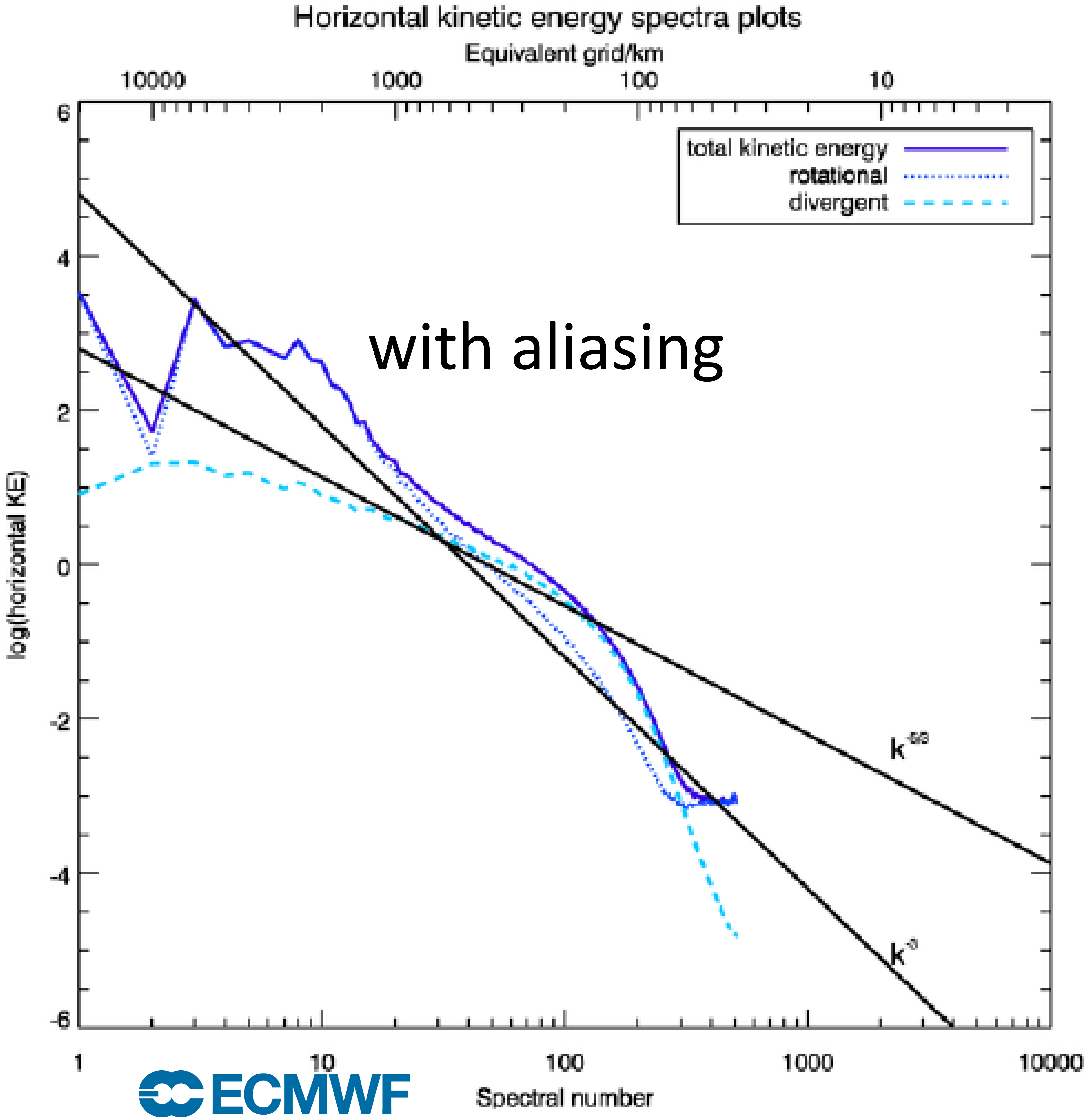


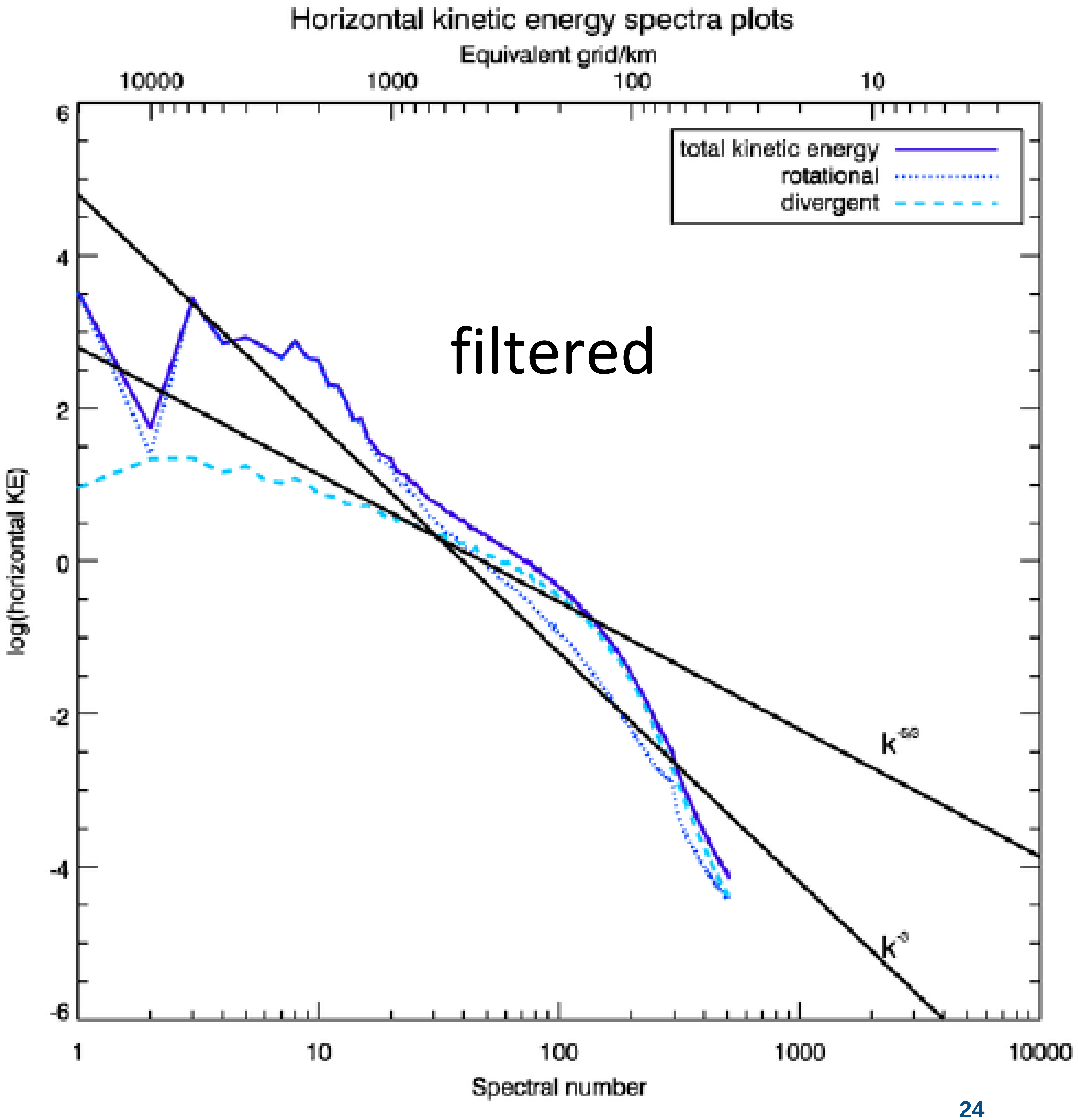
filtered



aliasing example

kinetic energy spectra, 100 hPa





alternatives to using a filter

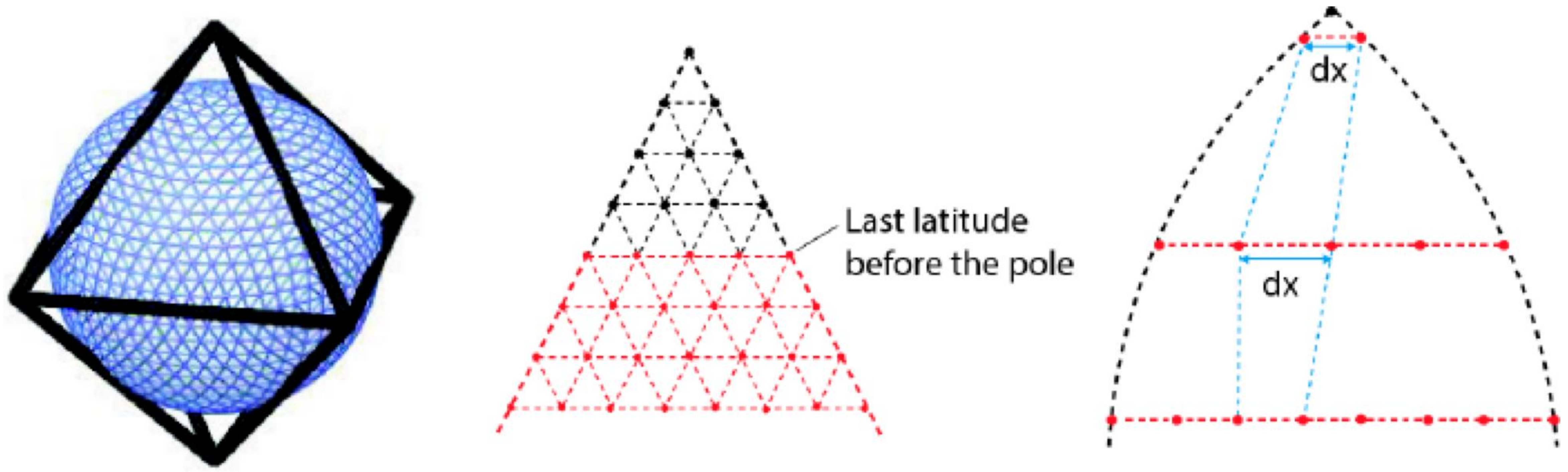
Idea: use more grid points than spectral coefficients

Orszag, 1971:

2N+1 gridpoints to N waves : linear grid	~ 1-2 Δ
3N+1 gridpoints to N waves : quadratic grid	~ 2-3 Δ
4N+1 gridpoints to N waves : cubic grid	~ 3-4 Δ
	Spatial filter range

(Wedi, 2014)

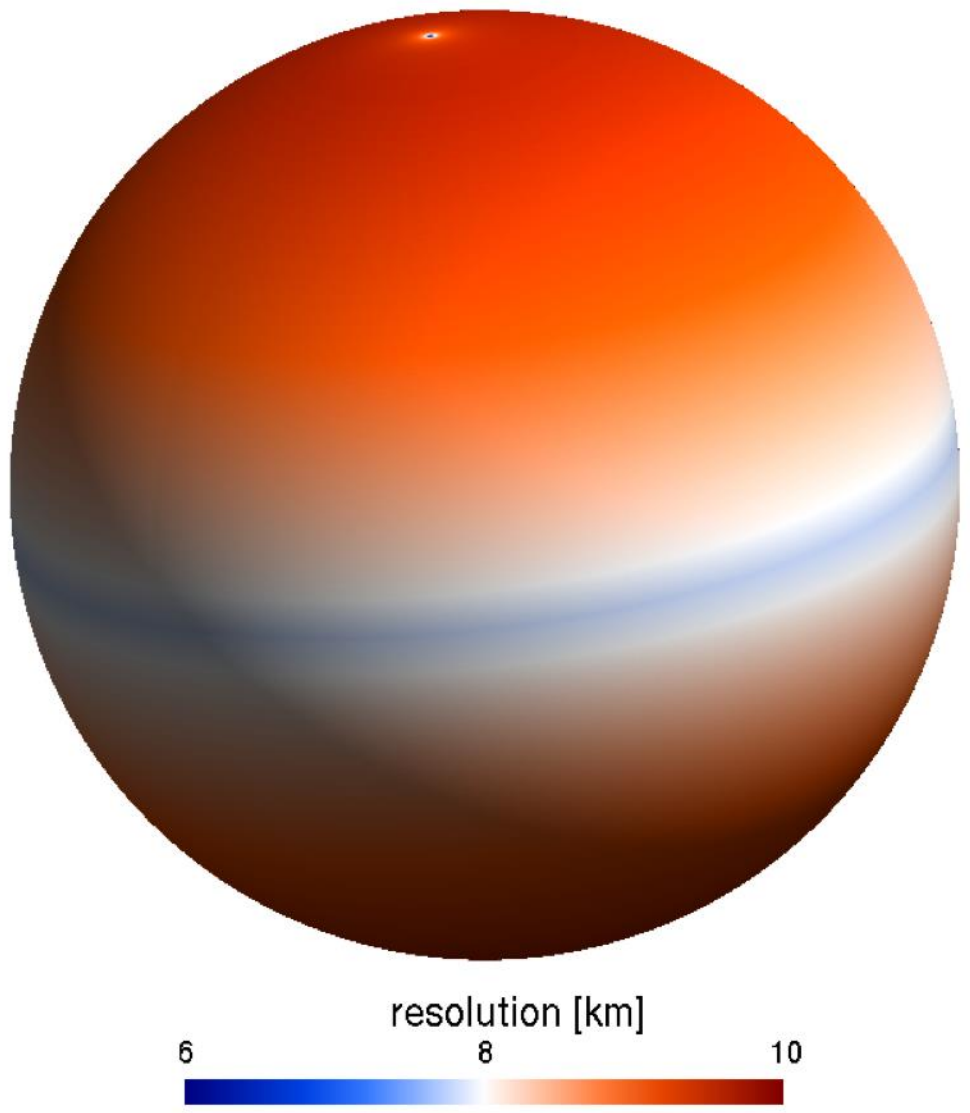
Cubic octahedral (Gaussian) grid of IFS



- No aliasing in nonlinear products
- Improved accuracy and mass conservation compared with linear grid
- Efficiency and scalability for large size problems: high grid-point resolution for a given spectral truncation i.e. expensive transforms become a smaller fraction of total computations

Collignon projection on the sphere: Number of points at latitude line i = 4 × i + 16, i = 1, . . . ,2N

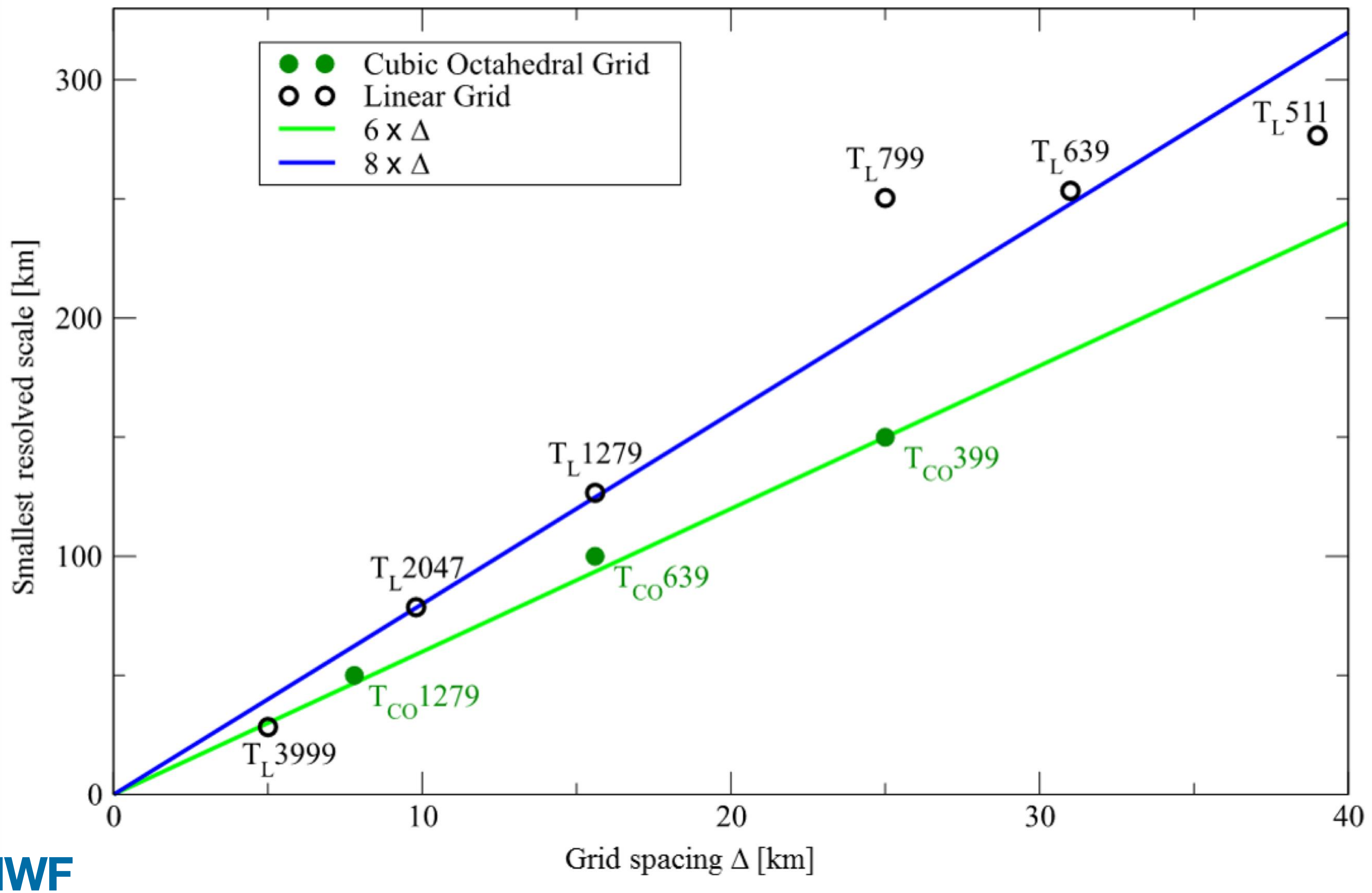
Variation of grid-point resolution with latitude



For a given spectral triangular truncation N the cubic reduced octahedral Gaussian grid has:

- 2N points between pole and equator which coincide with Gaussian latitudes
- 4N+16 east-west points along the equator
- 4N(N+9) points in total

effective resolution
of linear and cubic grids (Abdalla et al. 2013)



ECMWF

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

fastest index left (column-major order like in Fortran)

fields (variables, height levels)

real and imaginary part

wave numbers
m=0,…,N; n=0,…,N-m
(N: truncation)

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

m=0,…,N; n=0,…,N-m

even n

odd n

for each m:

$\mathbf{D}_{e,m}(f, \mathrm{i}, n)$

$\mathbf{D}_{o,m}(f, \mathrm{i}, n)$

ECMWF

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

m=0,…,N; n=0,…,N-m

even n

odd n

P: precomputed Legendre polynomials

for each m:

$$\mathbf{S}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{e,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{e,m}(n, \phi), \quad \mathbf{A}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{o,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{o,m}(n, \phi)$$

matrix multiplications

ECMWF

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

m=0,…,N; n=0,…,N-m

even n

odd n

P: precomputed Legendre polynomials

for each m:

$$\mathbf{S}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{e,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{e,m}(n, \phi), \quad \mathbf{A}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{o,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{o,m}(n, \phi)$$

matrix multiplications

$$\phi > 0: \quad \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, \phi) + \mathbf{A}_m(f, \mathrm{i}, \phi)$$

$$\phi < 0: \quad \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, -\phi) - \mathbf{A}_m(f, \mathrm{i}, -\phi)$$

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

m=0,…,N; n=0,…,N-m

even n

odd n

P: precomputed Legendre polynomials

for each m:

$$\mathbf{S}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{e,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{e,m}(n, \phi), \quad \mathbf{A}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{o,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{o,m}(n, \phi)$$

matrix multiplications

$$\phi > 0: \quad \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, \phi) + \mathbf{A}_m(f, \mathrm{i}, \phi)$$

$$\phi < 0: \quad \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, -\phi) - \mathbf{A}_m(f, \mathrm{i}, -\phi)$$

for each ϕ,f:

$$\mathbf{G}_{\phi,f}(\lambda) = \mathrm{FFT}(\mathbf{F}_{\phi,f}(\mathrm{i}, m))$$

FFT: Fast Fourier Transform

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

m=0,…,N; n=0,…,N-m

P: precomputed Legendre polynomials

even n

odd n

for each m:

$$\mathbf{S}_m(f,\mathrm{i},\phi) = \sum_n \mathbf{D}_{e,m}(f,\mathrm{i},n) \cdot \mathbf{P}_{e,m}(n,\phi), \quad \mathbf{A}_m(f,\mathrm{i},\phi) = \sum_n \mathbf{D}_{o,m}(f,\mathrm{i},n) \cdot \mathbf{P}_{o,m}(n,\phi)$$

matrix multiplications

$$\phi > 0: \quad \mathbf{F}(\mathrm{i},m,\phi,f) = \mathbf{S}_m(f,\mathrm{i},\phi) + \mathbf{A}_m(f,\mathrm{i},\phi)$$

$$\phi < 0: \quad \mathbf{F}(\mathrm{i},m,\phi,f) = \mathbf{S}_m(f,\mathrm{i},-\phi) - \mathbf{A}_m(f,\mathrm{i},-\phi)$$

for each ϕ,f: $\mathbf{G}_{\phi,f}(\lambda) = \mathrm{FFT}(\mathbf{F}_{\phi,f}(\mathrm{i},m))$

FFT: Fast Fourier Transform

grid point data: $\mathbf{G}(f,\lambda,\phi)$

ECMWF

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

spectral space

even n

odd n

for each m:

$$\mathbf{S}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{e,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{e,m}(n, \phi),$$

$$\mathbf{A}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{o,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{o,m}(n, \phi)$$

$$\phi > 0 : \ \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, \phi) + \mathbf{A}_m(f, \mathrm{i}, \phi)$$

$$\phi < 0 : \ \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, -\phi) - \mathbf{A}_m(f, \mathrm{i}, -\phi)$$

inverse Legendre transform

inverse Fourier transform

for each ϕ,f: $\mathbf{G}_{\phi,f}(\lambda) = \mathrm{FFT}(\mathbf{F}_{\phi,f}(\mathrm{i}, m))$

grid point data: $\mathbf{G}(f, \lambda, \phi)$

grid point space

ECMWF

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$ — spectral space — m,n

even n — odd n

parallelisation over these indices

for each m:

$$\mathbf{S}_m(f,\mathrm{i},\phi) = \sum_n \mathbf{D}_{e,m}(f,\mathrm{i},n) \cdot \mathbf{P}_{e,m}(n,\phi),$$
$$\mathbf{A}_m(f,\mathrm{i},\phi) = \sum_n \mathbf{D}_{o,m}(f,\mathrm{i},n) \cdot \mathbf{P}_{o,m}(n,\phi)$$

$$\phi > 0: \ \mathbf{F}(\mathrm{i},m,\phi,f) = \mathbf{S}_m(f,\mathrm{i},\phi) + \mathbf{A}_m(f,\mathrm{i},\phi)$$
$$\phi < 0: \ \mathbf{F}(\mathrm{i},m,\phi,f) = \mathbf{S}_m(f,\mathrm{i},-\phi) - \mathbf{A}_m(f,\mathrm{i},-\phi)$$

inverse Legendre transform — m,f

for each ϕ,f: $\mathbf{G}_{\phi,f}(\lambda) = \mathrm{FFT}(\mathbf{F}_{\phi,f}(\mathrm{i},m))$ — inverse Fourier transform — ϕ,f

grid point data: $\mathbf{G}(f,\lambda,\phi)$ — grid point space — ϕ,λ

inverse spectral transform

spectral data: $\mathbf{D}(f, \mathrm{i}, n, m)$

even n — odd n

for each m:

$$\mathbf{S}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{e,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{e,m}(n, \phi),$$

$$\mathbf{A}_m(f, \mathrm{i}, \phi) = \sum_n \mathbf{D}_{o,m}(f, \mathrm{i}, n) \cdot \mathbf{P}_{o,m}(n, \phi)$$

$$\phi > 0: \ \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, \phi) + \mathbf{A}_m(f, \mathrm{i}, \phi)$$

$$\phi < 0: \ \mathbf{F}(\mathrm{i}, m, \phi, f) = \mathbf{S}_m(f, \mathrm{i}, -\phi) - \mathbf{A}_m(f, \mathrm{i}, -\phi)$$

for each ϕ,f: $\mathbf{G}_{\phi,f}(\lambda) = \mathrm{FFT}(\mathbf{F}_{\phi,f}(\mathrm{i}, m))$

grid point data: $\mathbf{G}(f, \lambda, \phi)$

spectral space	m,n
inverse Legendre transform	m,f
inverse Fourier transform	ϕ,f
grid point space	ϕ,λ

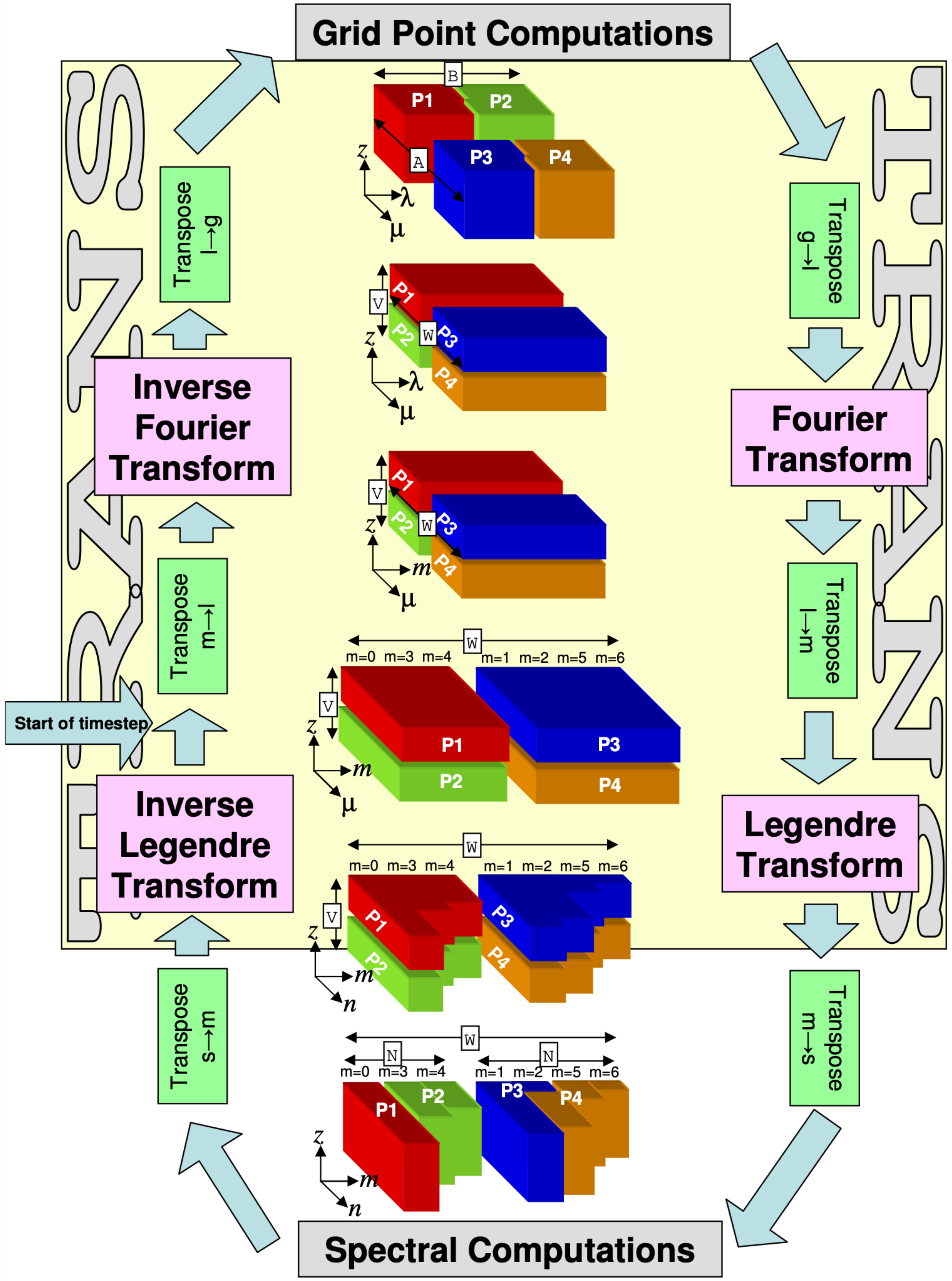
parallelisation over these indices

lots of MPI communication

ECMWF

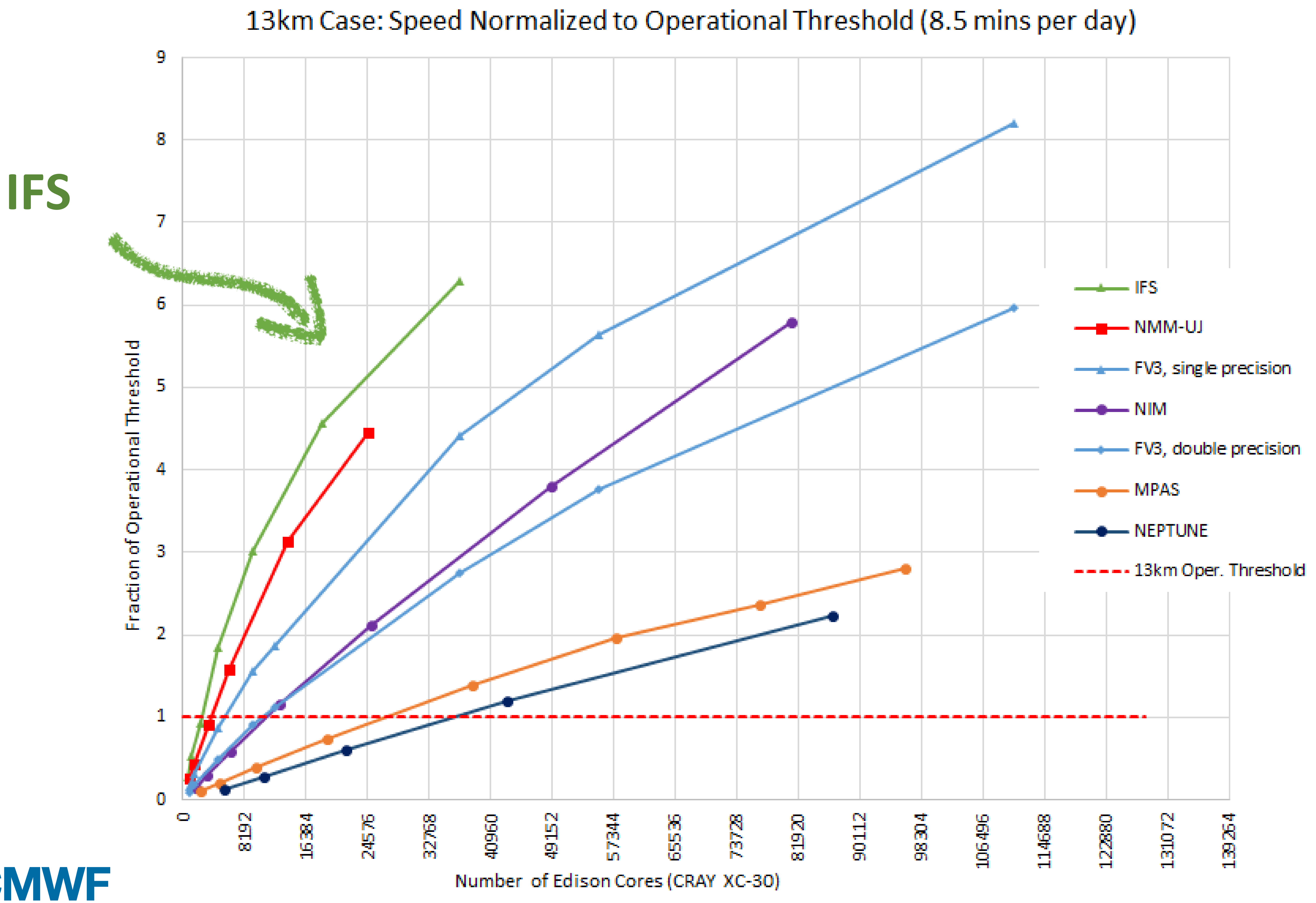
direct spectral transform

- same like inverse spectral transform
- reverse order
- multiply data with Gaussian quadrature weights before Legendre transform



ECMWF

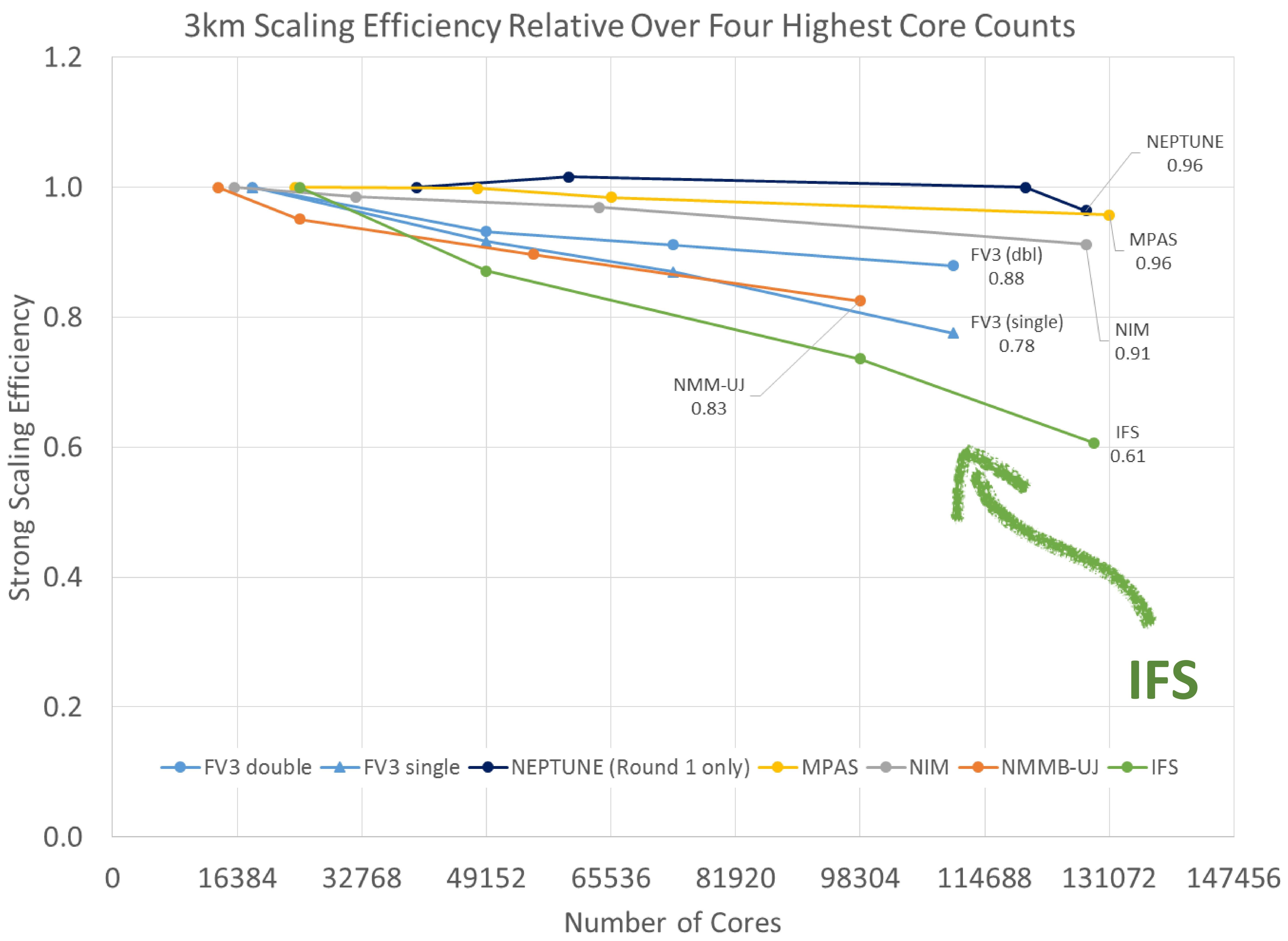
performance comparison of IFS with other models



(Michalakes et al, NGGPS AVEC report, 2015)

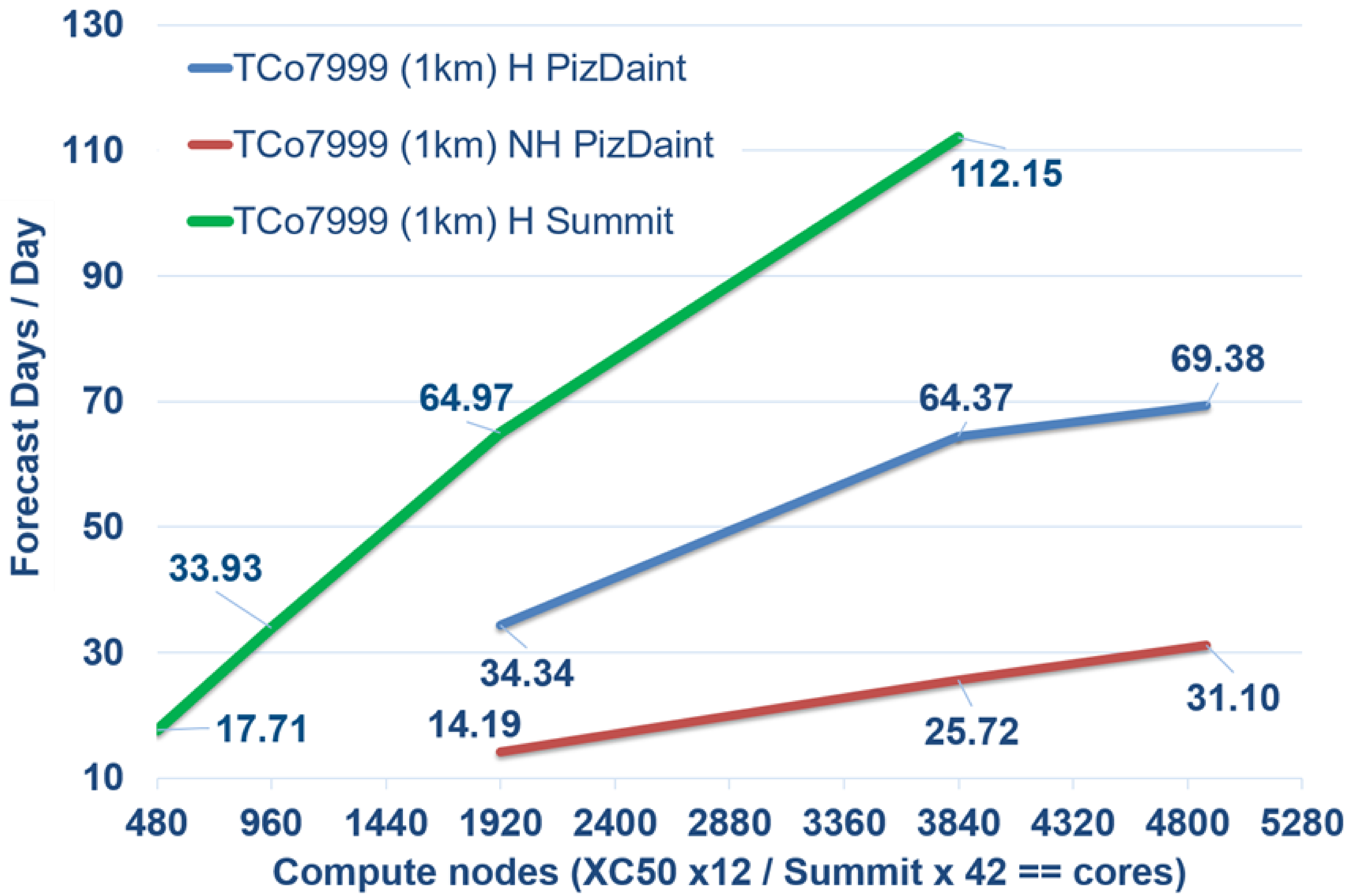
ECMWF

scalability comparison of IFS with other models



(Michalakes et al, NGGPS AVEC report, 2015)

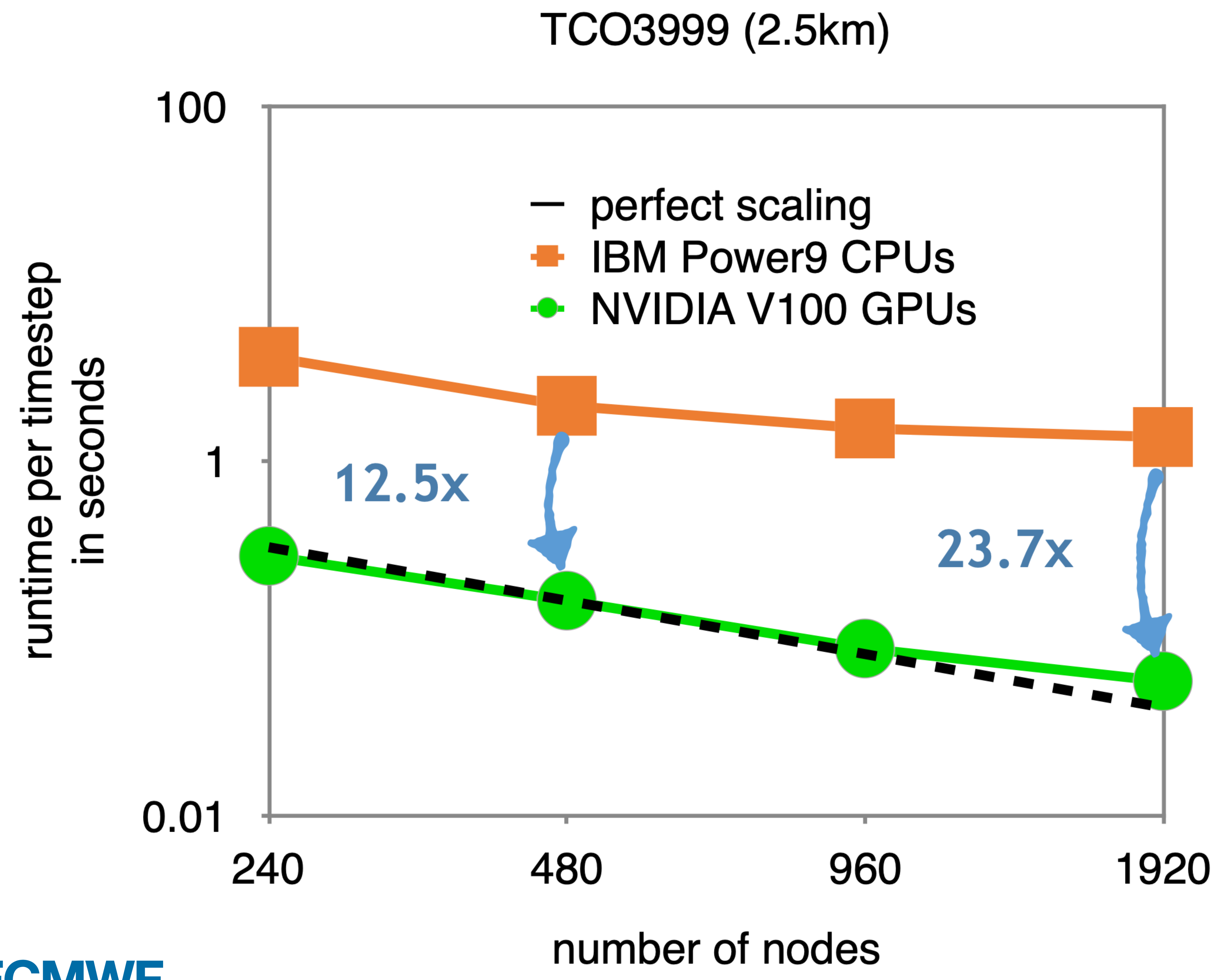
IFS scaling on Summit and PizDaint (CPU only)



ECMWF

GPUs vs CPUs on Summit

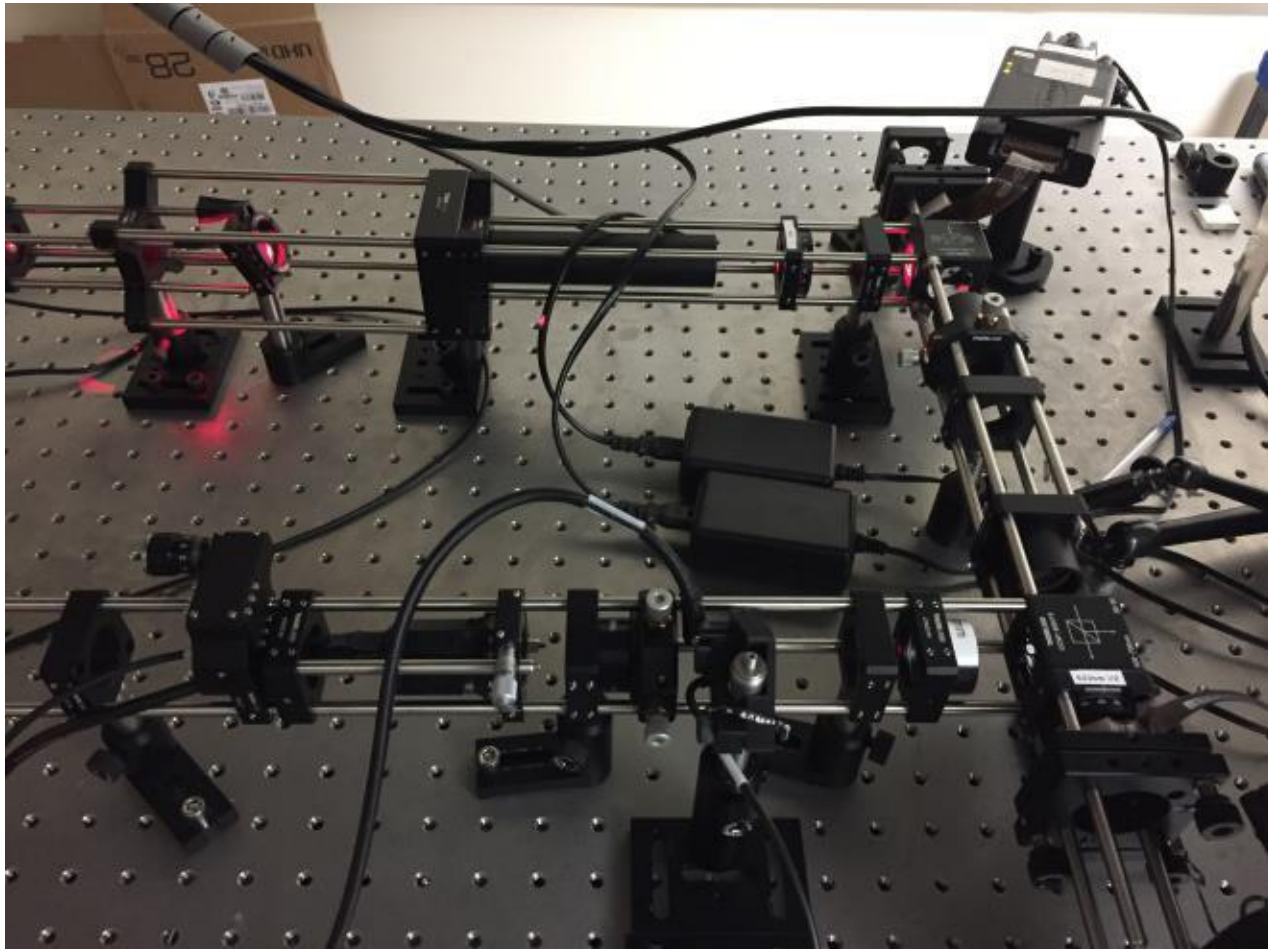
spectral transform only

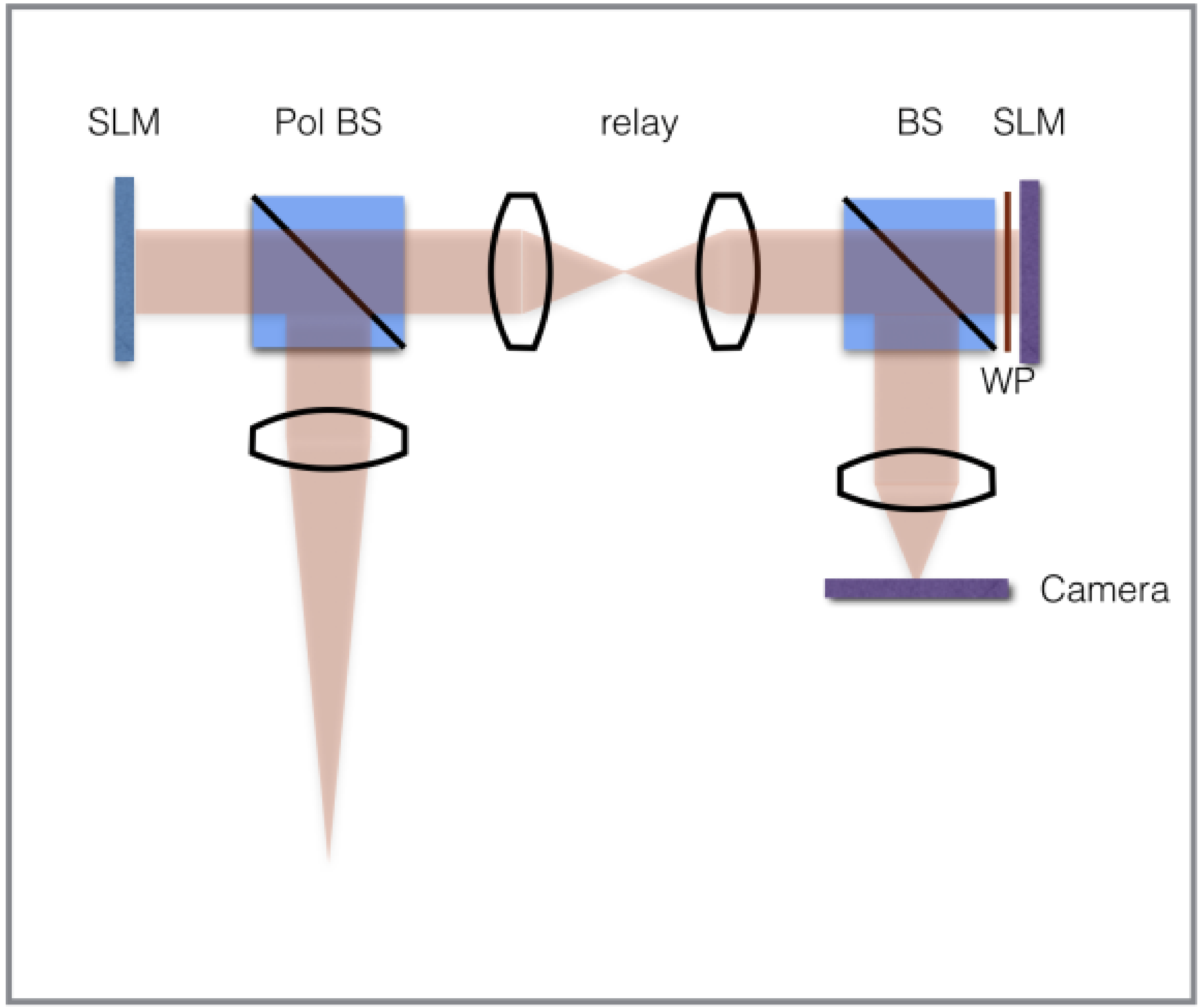


At 2.5km resolution, less than 1s per time-step fits operational needs.

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE office of Science User Facility supported under contract DE-AC05-00OR22725.

Optalysys: optical processor for spectral transform





Figures used with permission from Optalysys, 2017

ECMWF

Questions?