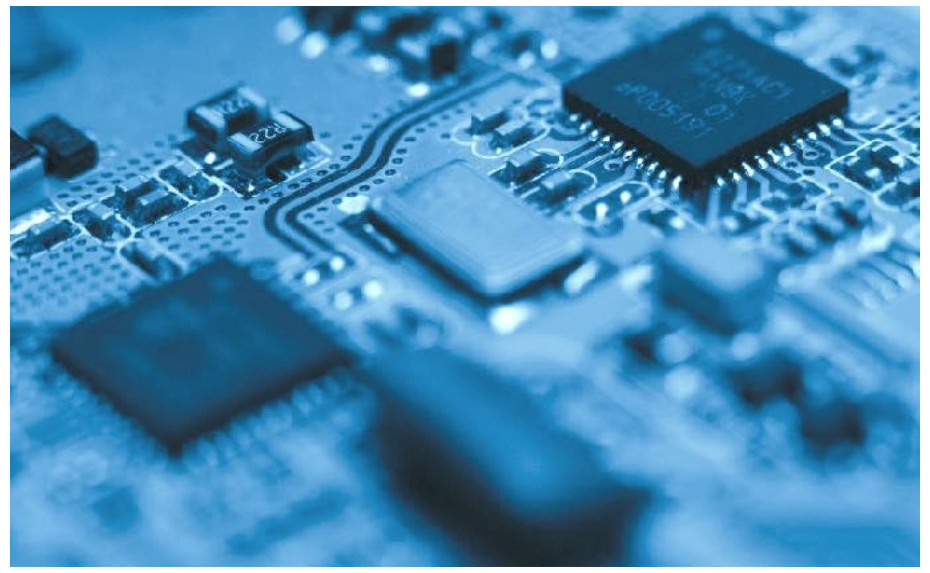
Funded by the European Union

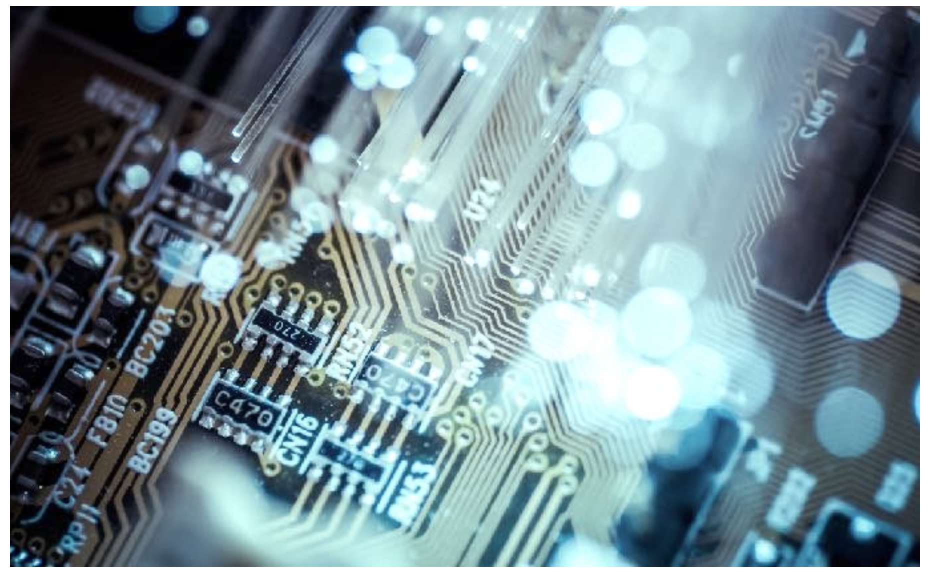
Co-ordinated by ECMWF

ESCAPE 2

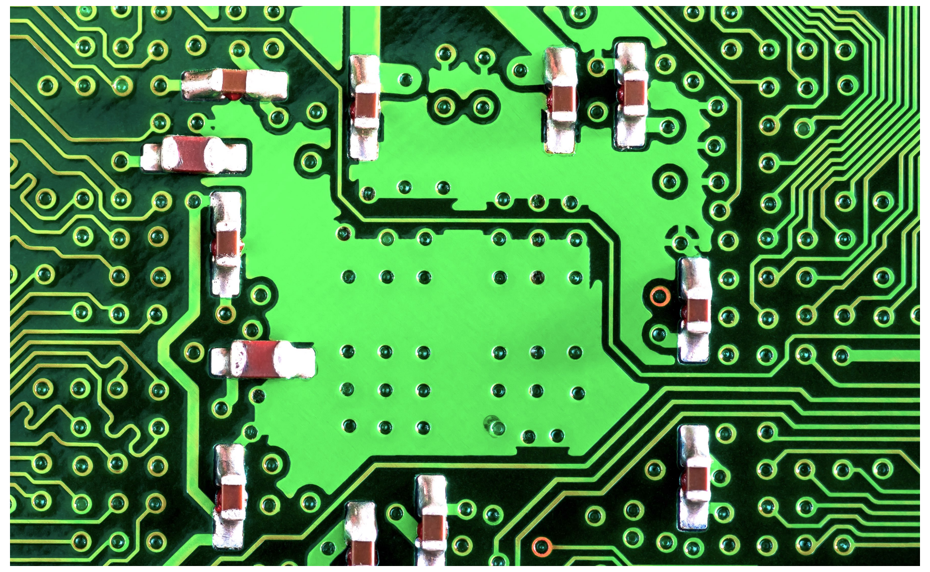
Funded by the European Union

Co-ordinated by ECMWF



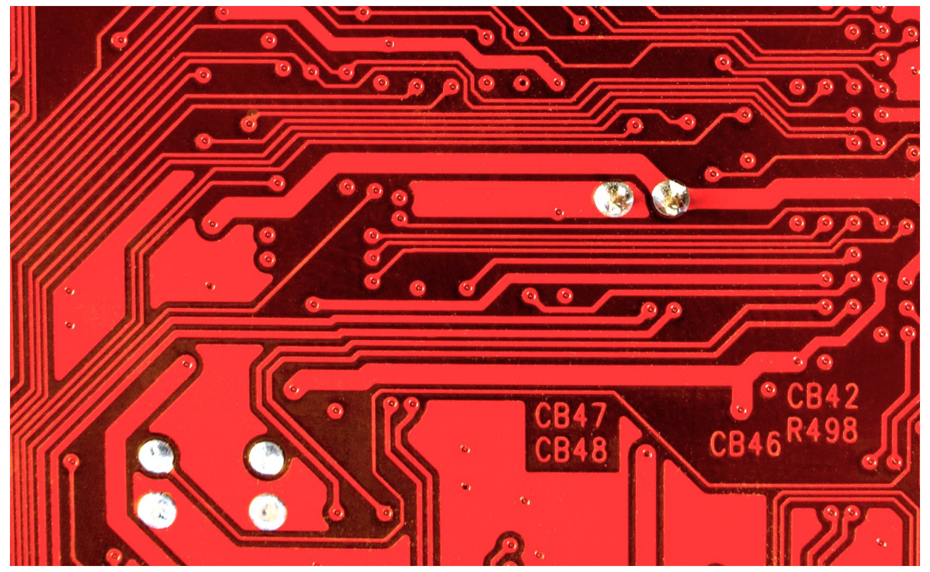


ESCAPE 2

Spectral Transform

Andreas Mueller




ECMWF DKRZ Max-Planck-Institut für Meteorologie MeteoSwiss Barcelona Supercomputing Center cea Loughborough University RMI POLITECNICO MOX DMI cmcc Bull atos technologies

The ESCAPE-2 project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800897

ESCAPE 2

Funded by the European Union

ESCAPE: Energy-efficient Scalable Algorithms for Weather Prediction at Exascale

ECMWF · DKRZ Deutsches Klimarechenzentrum · Max-Planck-Institut für Meteorologie · MeteoSwiss · BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación · cea · Loughborough University · RMI · Politecnico Milano 1863 · MOX · DMI · cmcc Centro Euro-Mediterraneo sui Cambiamenti Climatici · Bull atos technologies

Overview

Funded by the European Union

10 minutes	• Fourier transform • Spectral transform
60 minutes	• hands-on exercises with Python • coffee break and group photo in between
30 minutes	• aliasing • parallelization • performance • Fast Legendre Transform

ESCAPE 2

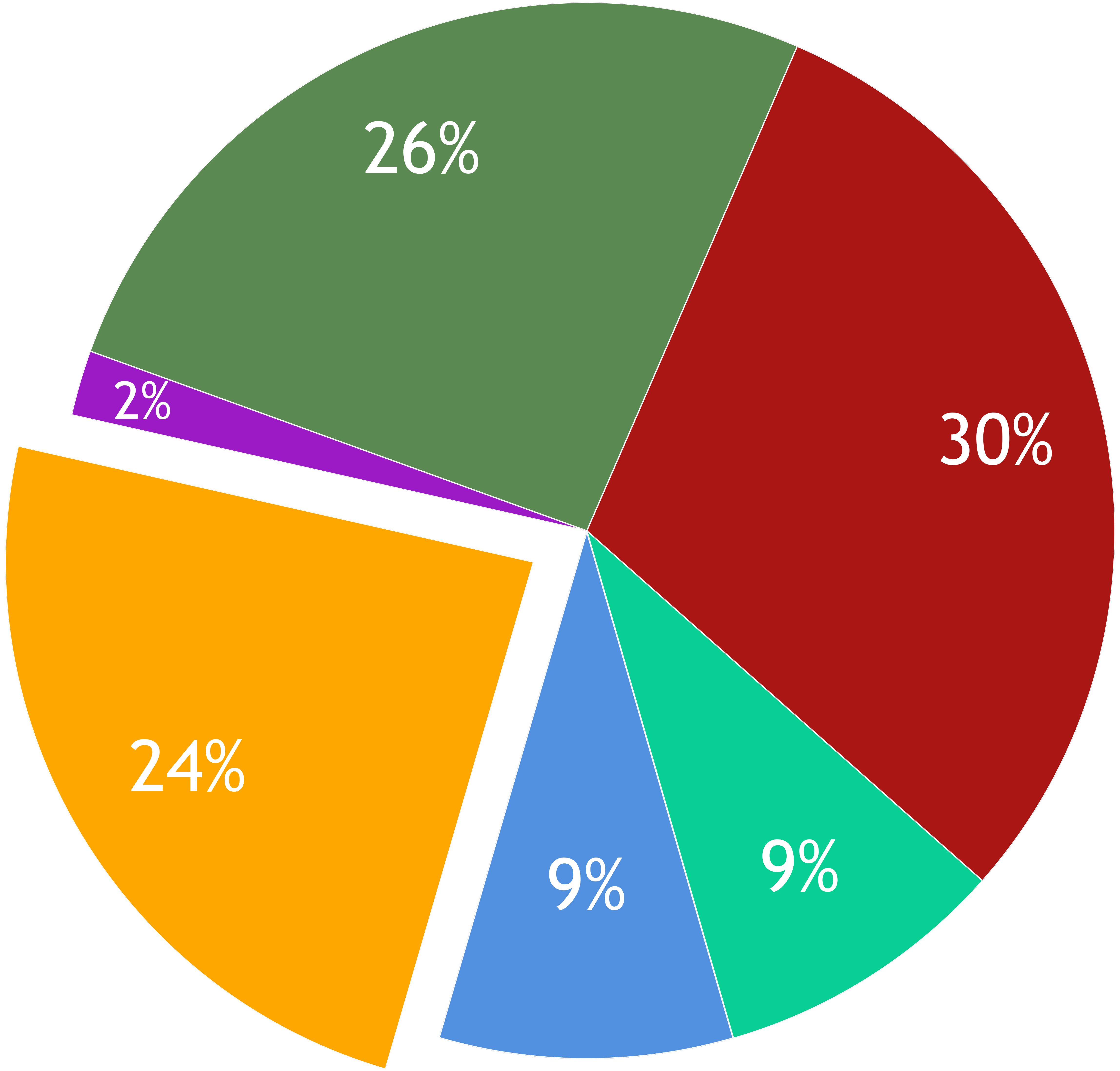
Funded by the European Union

IFS (Integrated Forecast System)

technology applied at ECMWF for the last 30 years

- spectral transform
- semi-Lagrangian
- semi-implicit

pie chart: % of runtime in 9km operational forecast



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

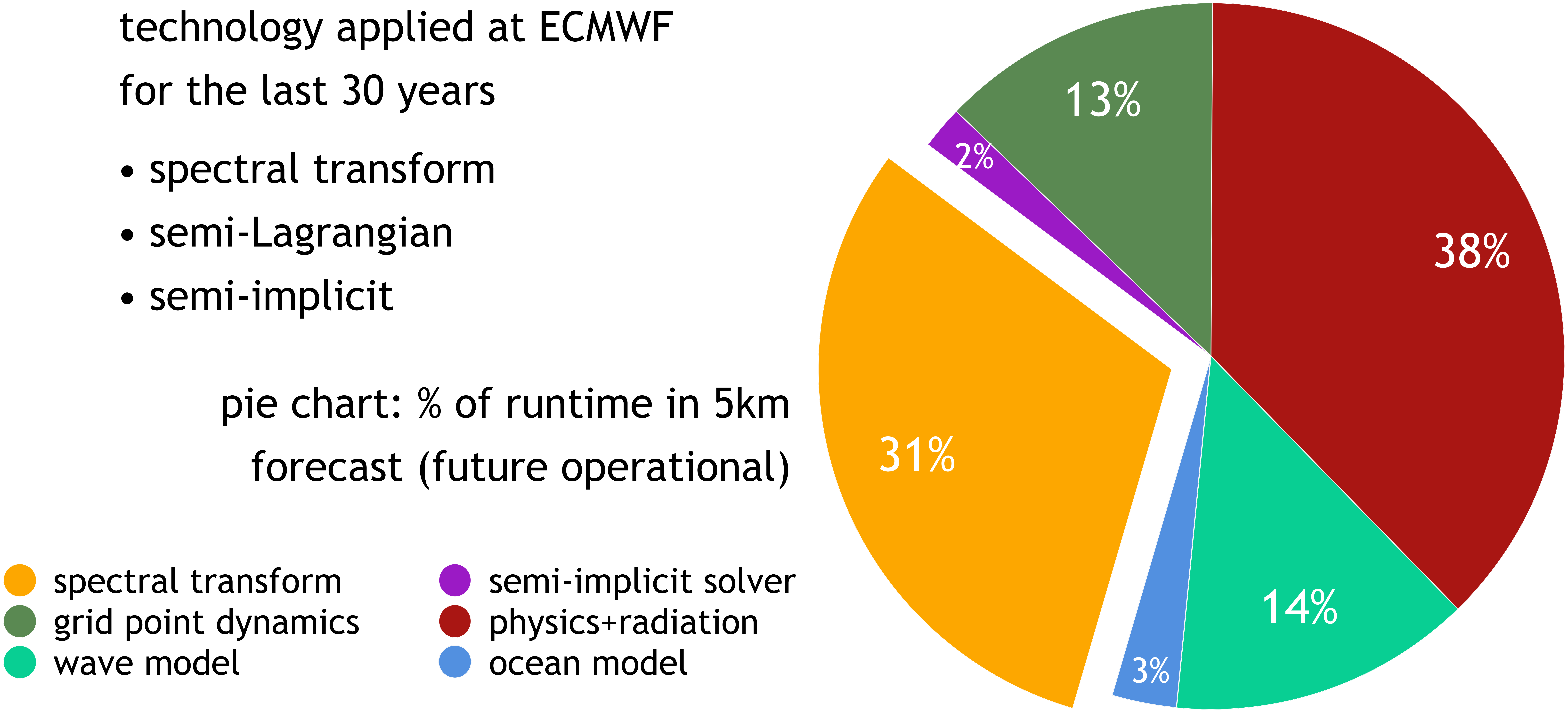
IFS (Integrated Forecast System)

Funded by the European Union

technology applied at ECMWF for the last 30 years

- spectral transform
- semi-Lagrangian
- semi-implicit

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

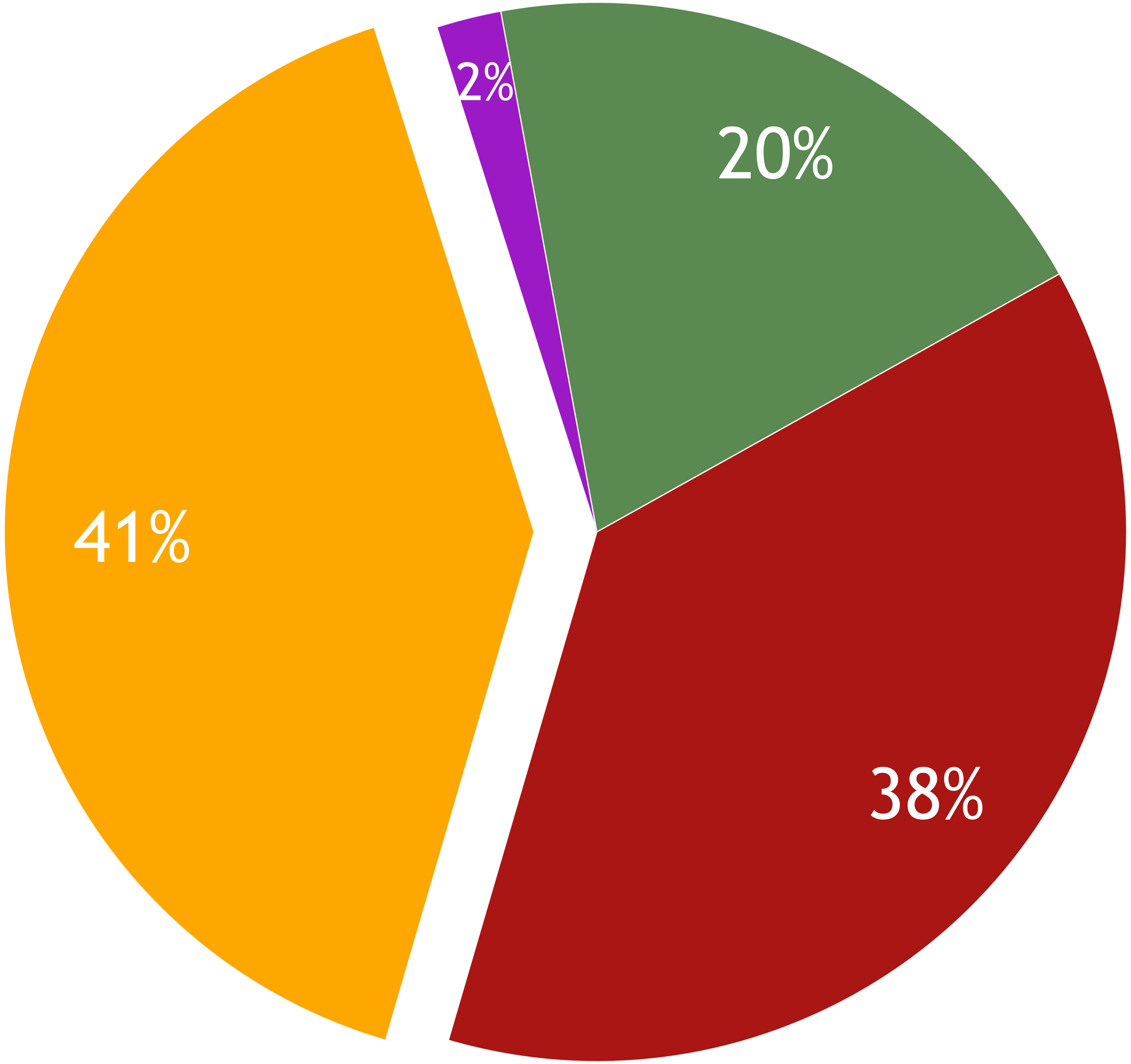


IFS (Integrated Forecast System)

technology applied at ECMWF for the last 30 years

- spectral transform
- semi-Lagrangian
- semi-implicit

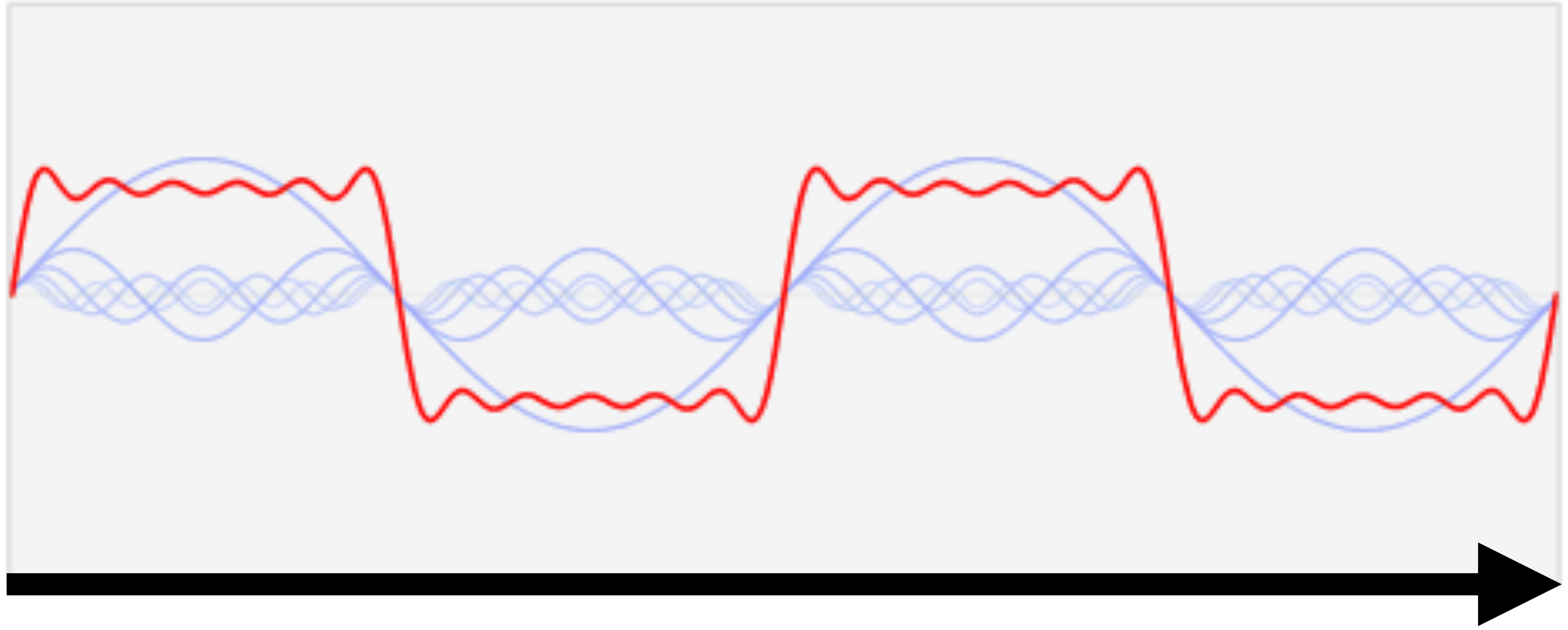
pie chart: % of runtime in 1.25km forecast (experiment, no ocean)



Funded by the European Union

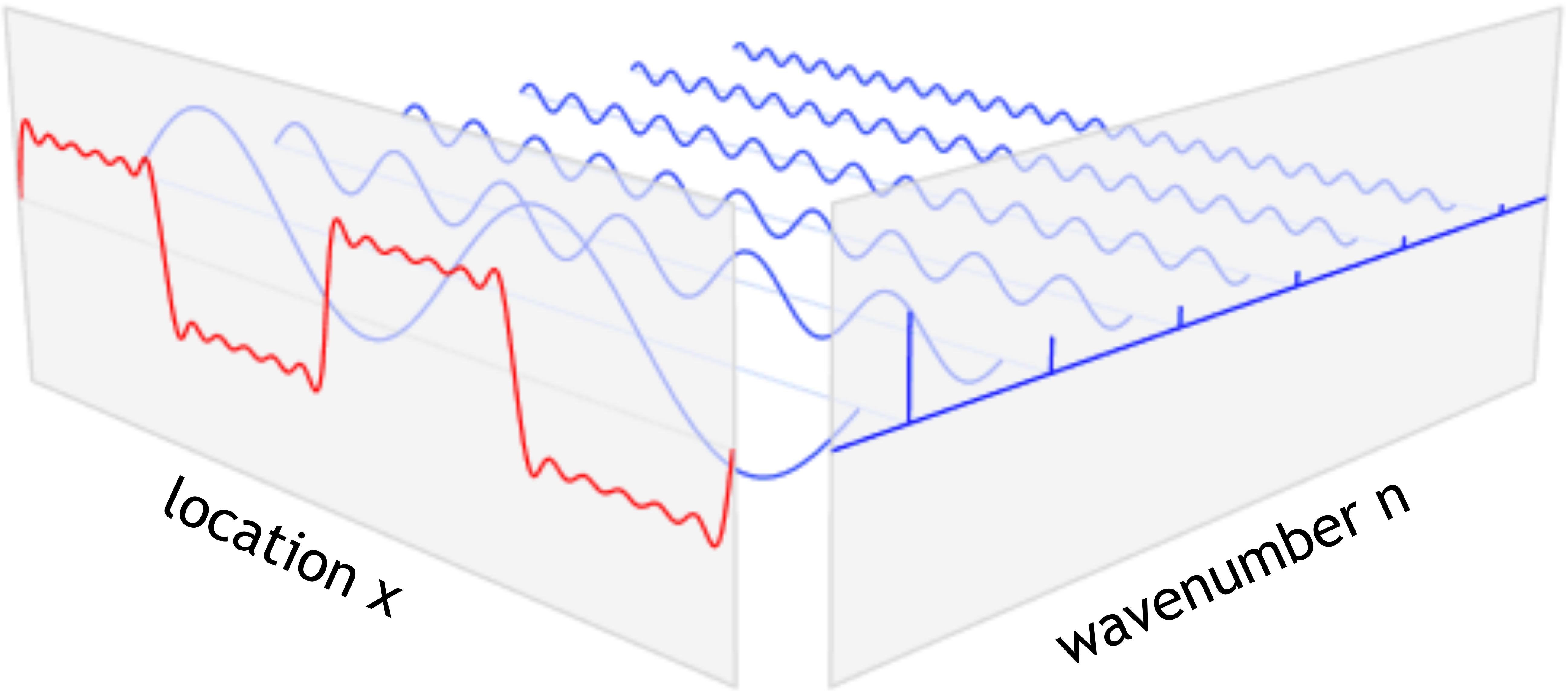
Fourier transform

Fourier transform = Spectral transform in 1D



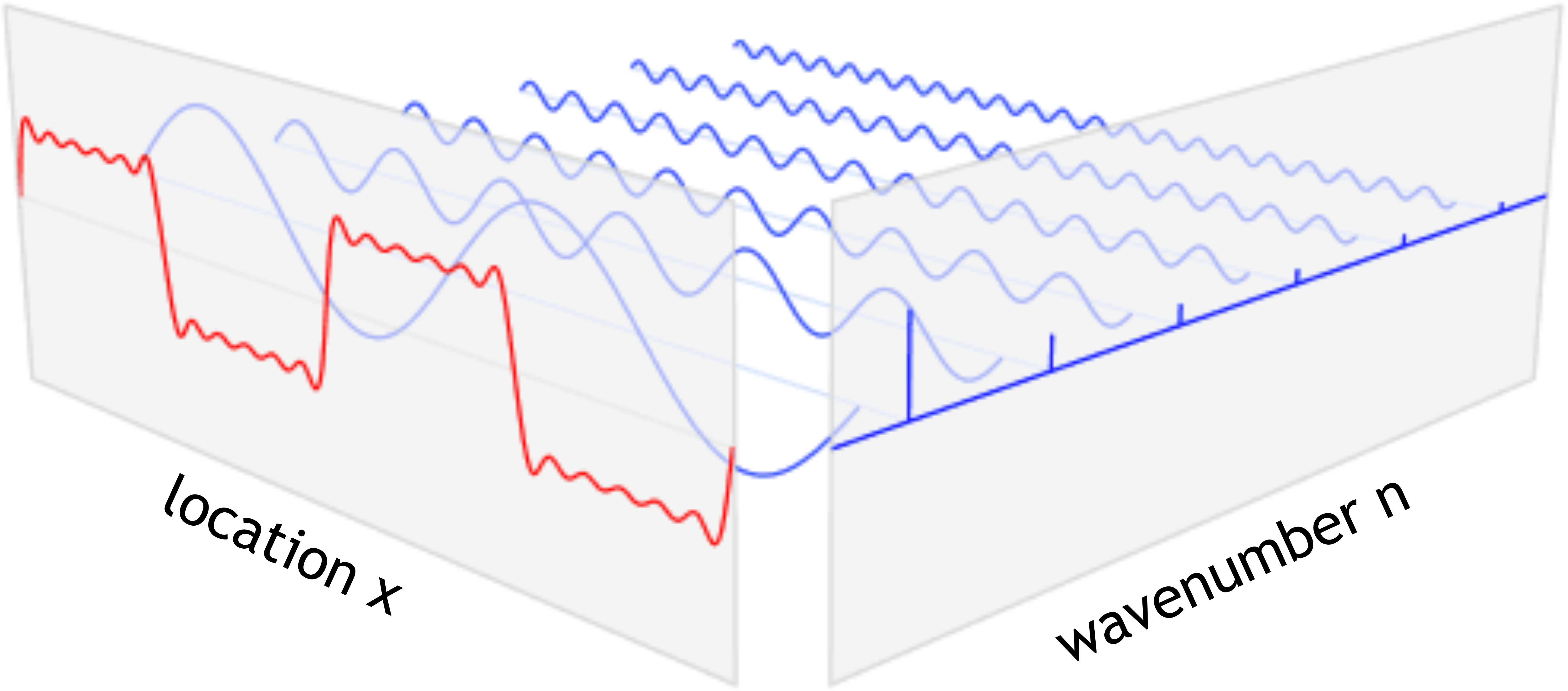
Fourier transform

Fourier transform = Spectral transform in 1D



grid point space **Fourier space**

Fourier transform



function in grid point space

Fourier coefficients

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

Funded by the European Union

Funded by the European Union

Fourier transform

function in grid point space

Fourier coefficients

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

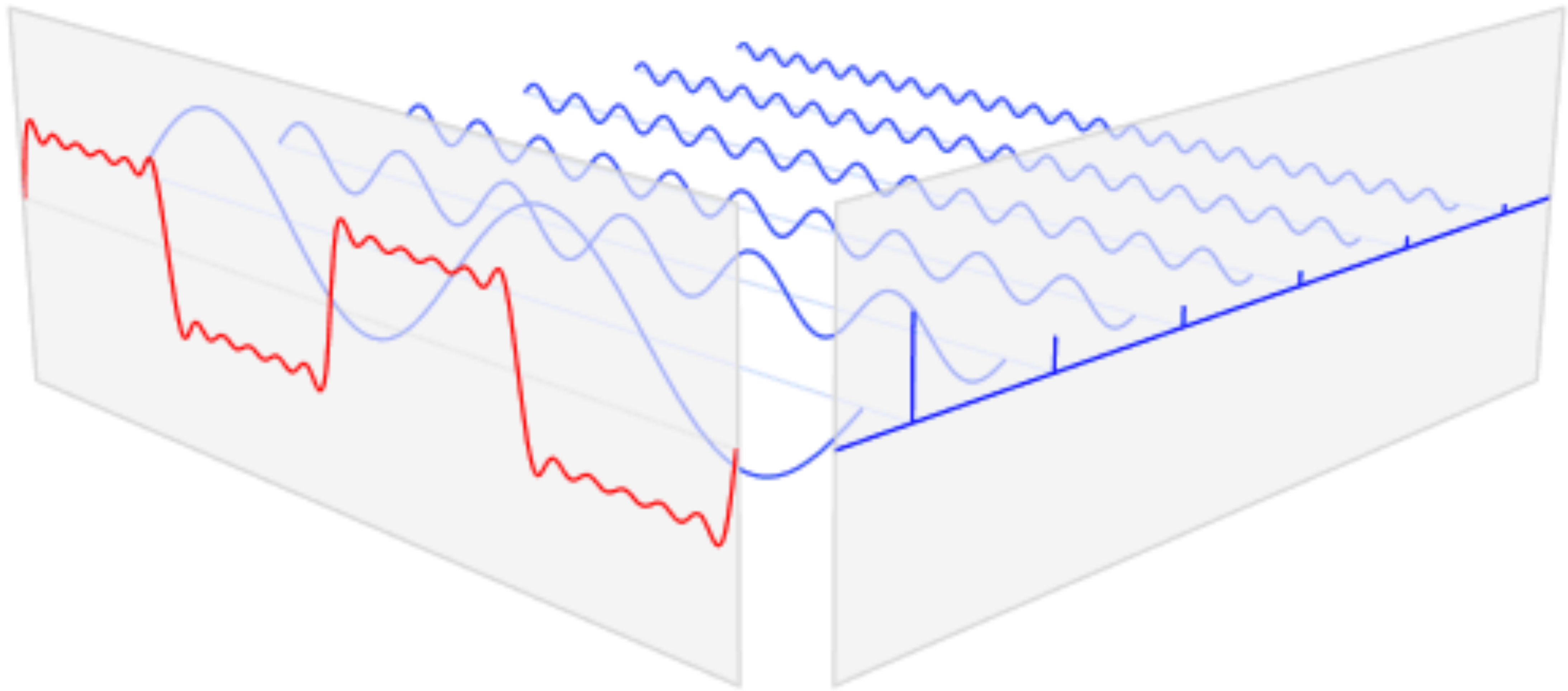
differentiation

simple multiplication

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

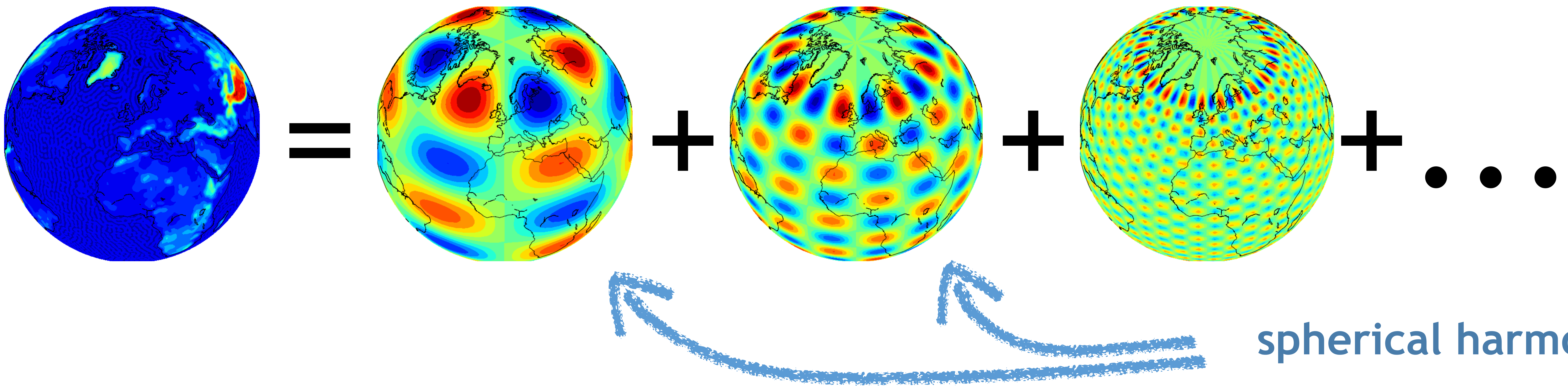
Funded by the European Union

on the sphere: spectral transform



grid point space

spectral space



spherical harmonics

ESCAPE 2

Funded by the European Union

on the sphere: spectral transform

Spectral coefficients

Latitude

Longitude

Spherical harmonics

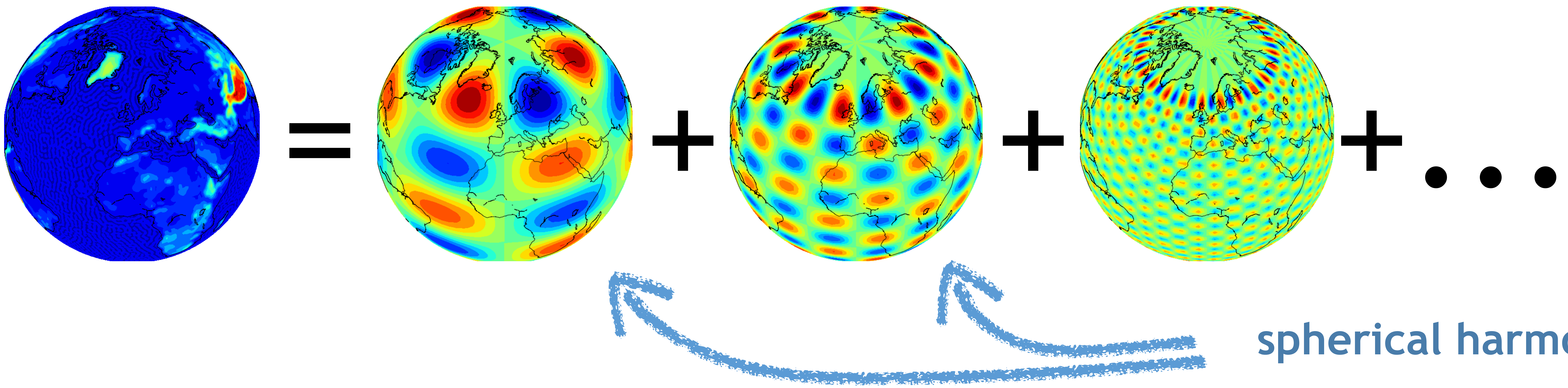
Grid point variable

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

m: zonal wavenumber
n: total wavenumber
M: truncation

grid point space

spectral space



spherical harmonics

Funded by the European Union

on the sphere: spectral transform

Spectral coefficients

Latitude

Longitude

Spherical harmonics

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

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

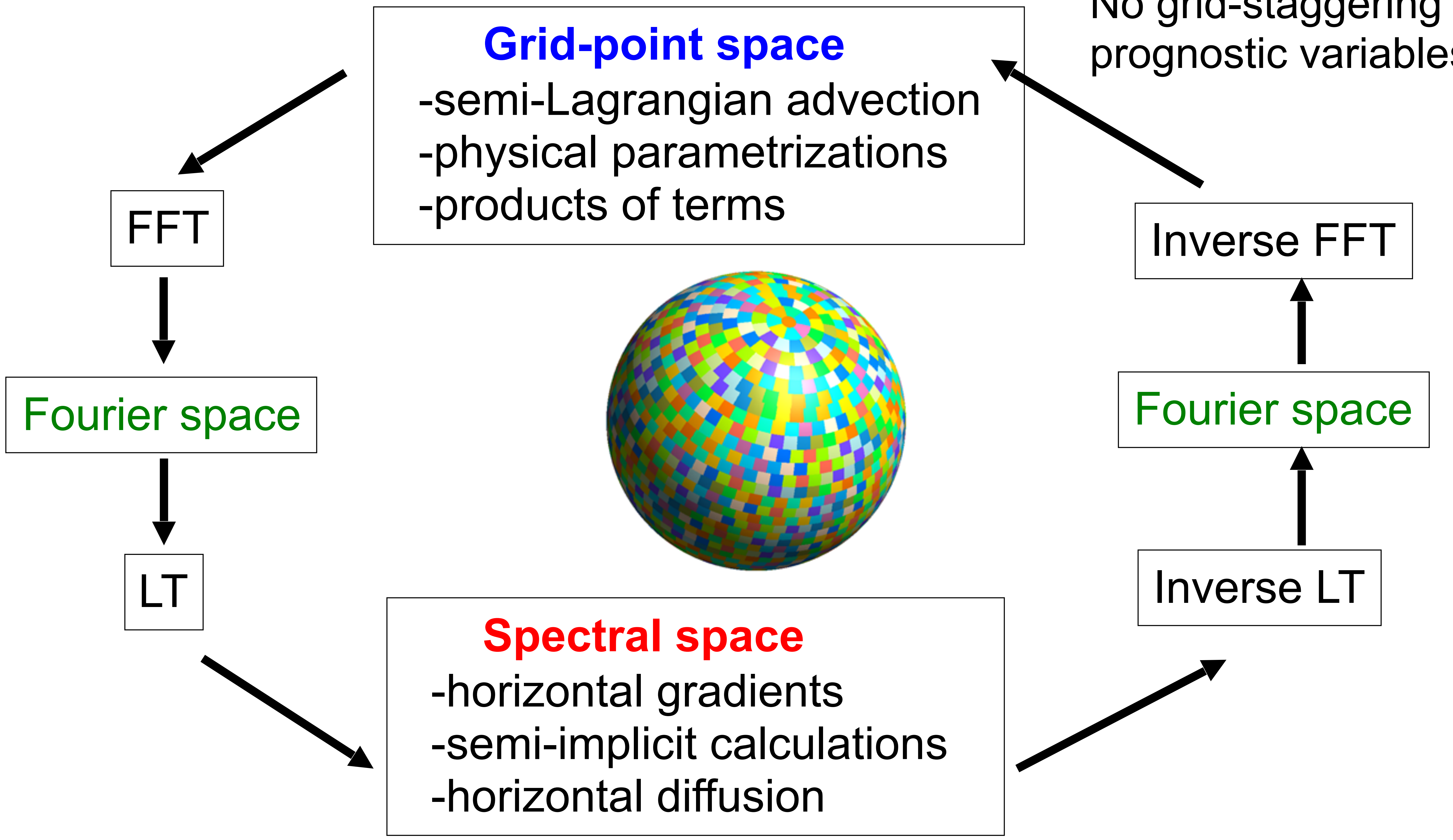
Legendre transform

Fourier transform

time step in IFS

Funded by the European Union

No grid-staggering of prognostic variables



FFT: Fast Fourier Transform, LT: Legendre Transform

Funded by the European Union

hands-on session

on the classroom computers:

run in the terminal:
/home/ectrain/trx/NM_TC2019/copyspectral.sh

in the cloud (Microsoft):

https://notebooks.azure.com/anmrde/libraries/tcnm2019
click on clone

files:

TCNM2019.ipynb: Python notebook with exercises
TCNM2019solution.ipynb: notebook including sample solutions

aliasing

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

Funded by the European Union

ESCAPE 2

Funded by the European Union

aliasing

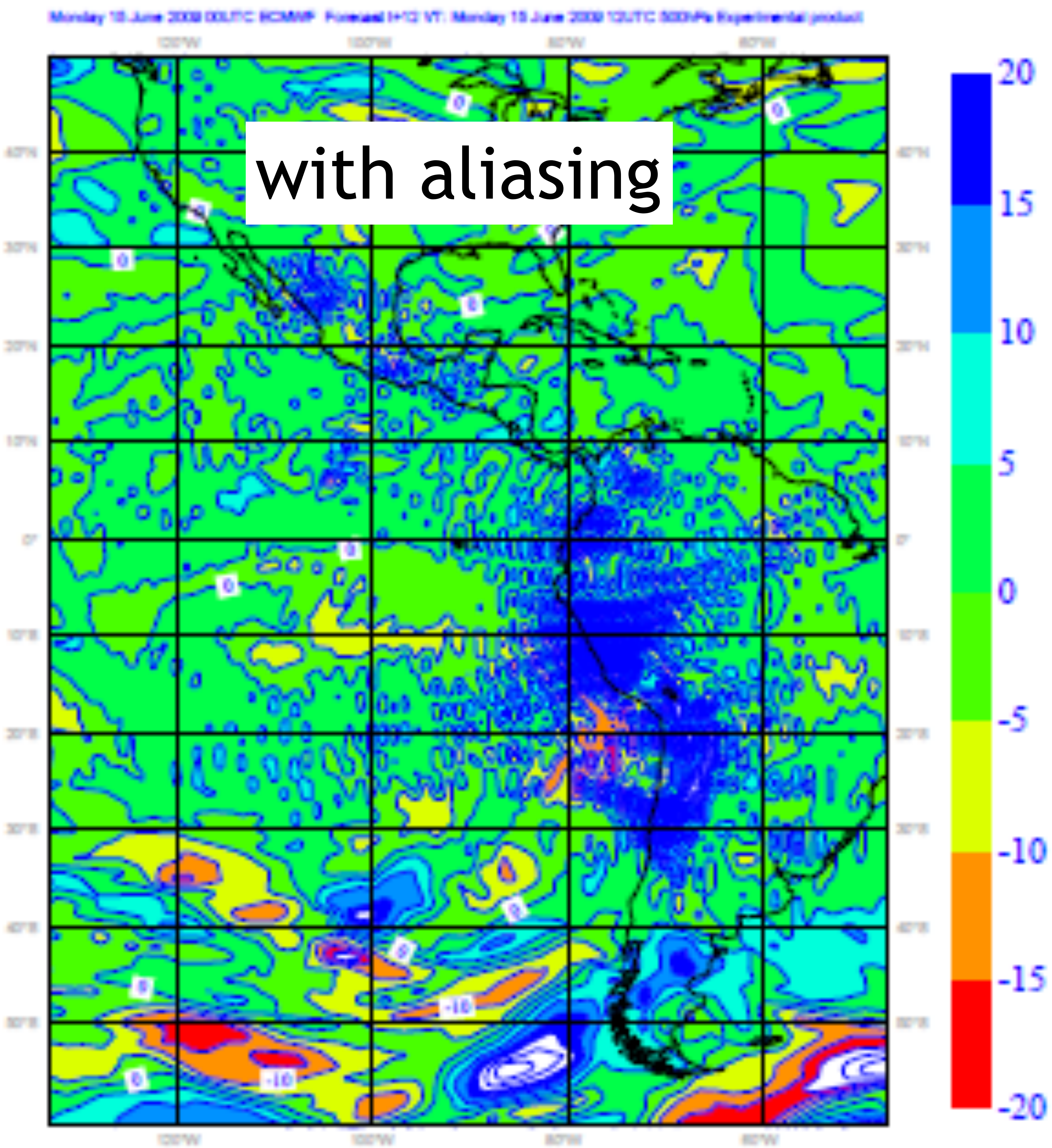
Issue: multiplication of two variables produces shorter waves than grid can handle

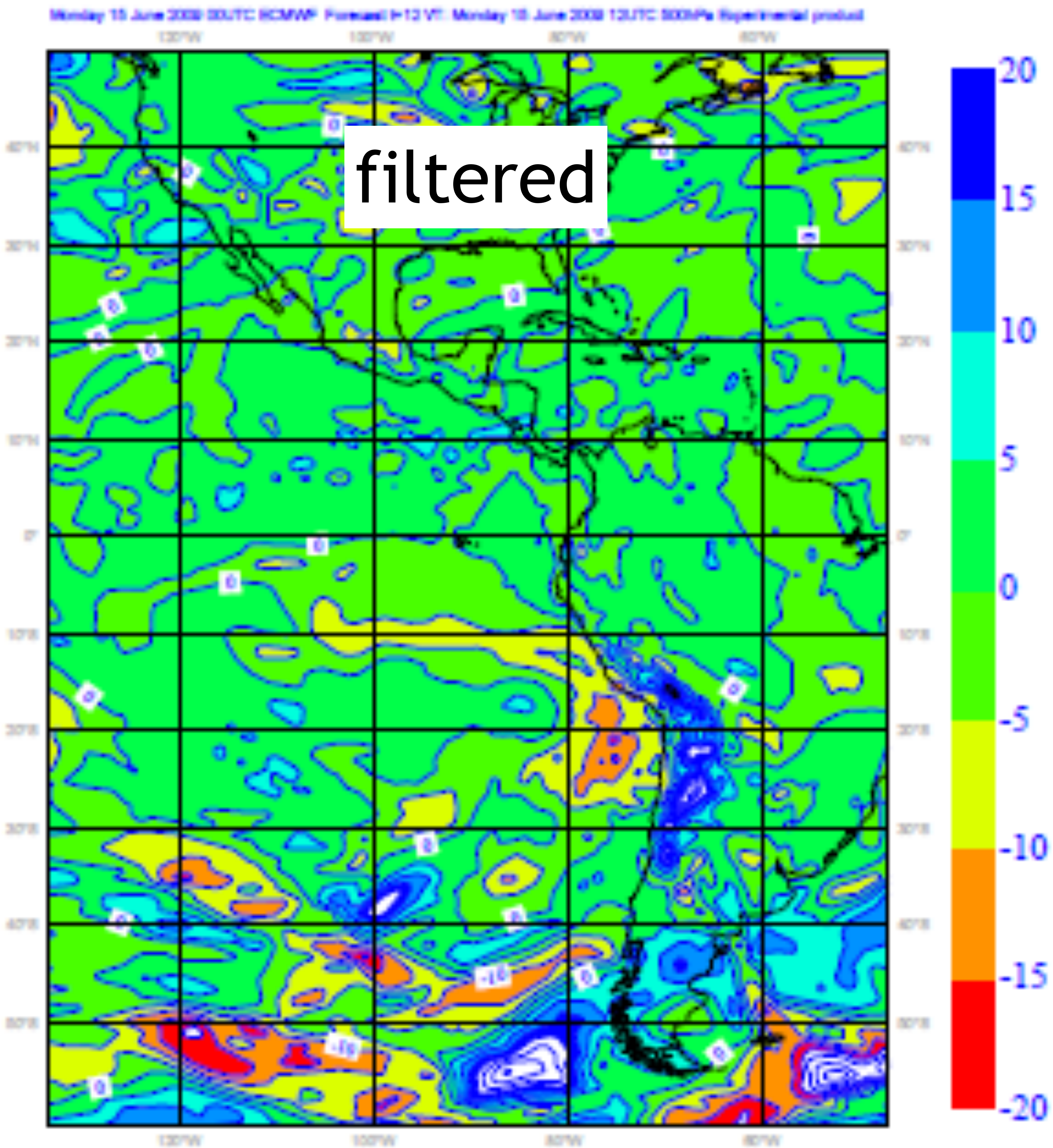
ESCAPE 2

Funded by the European Union

aliasing example

500hPa adiabatic zonal wind tendencies (T159)



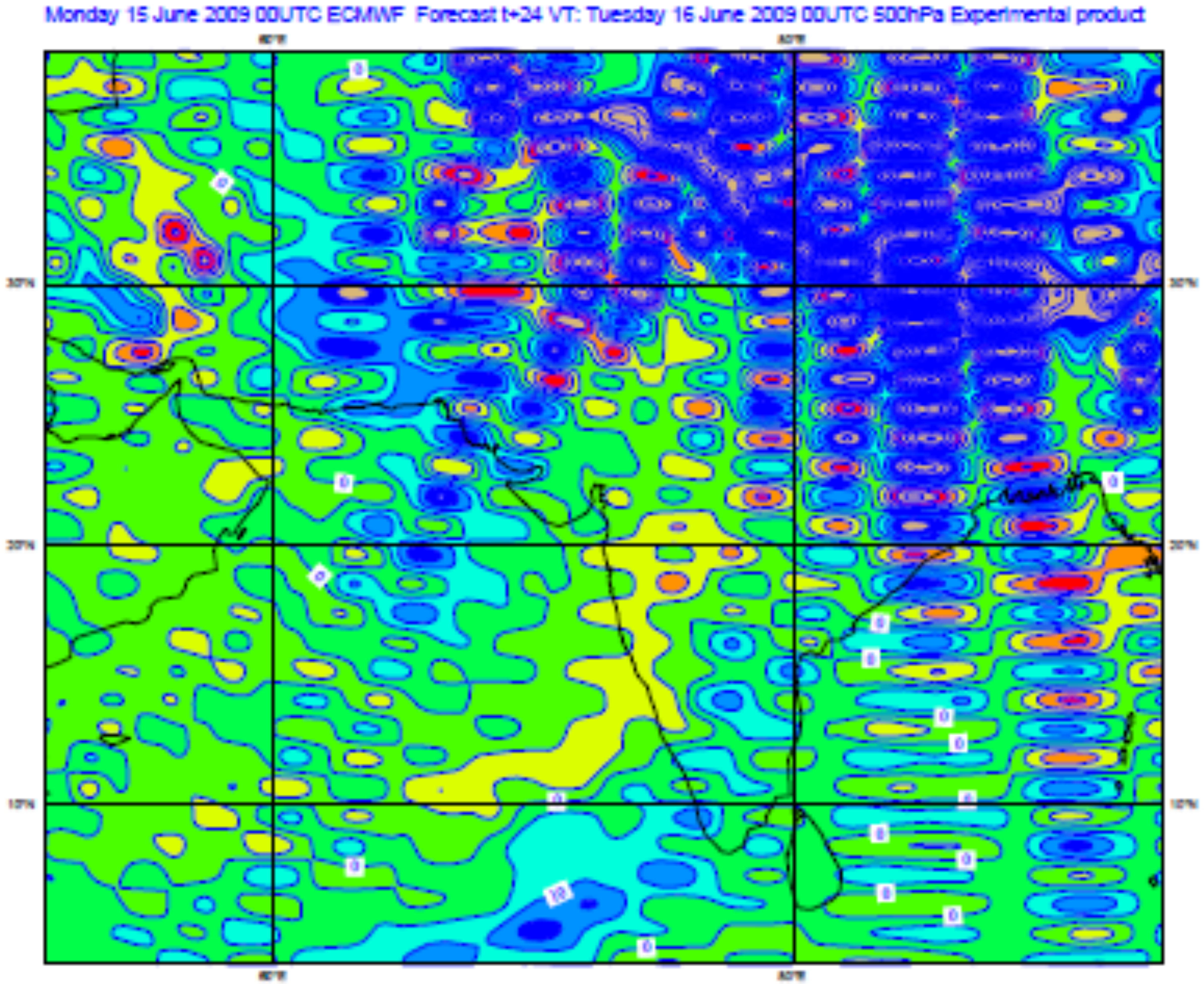


aliasing example

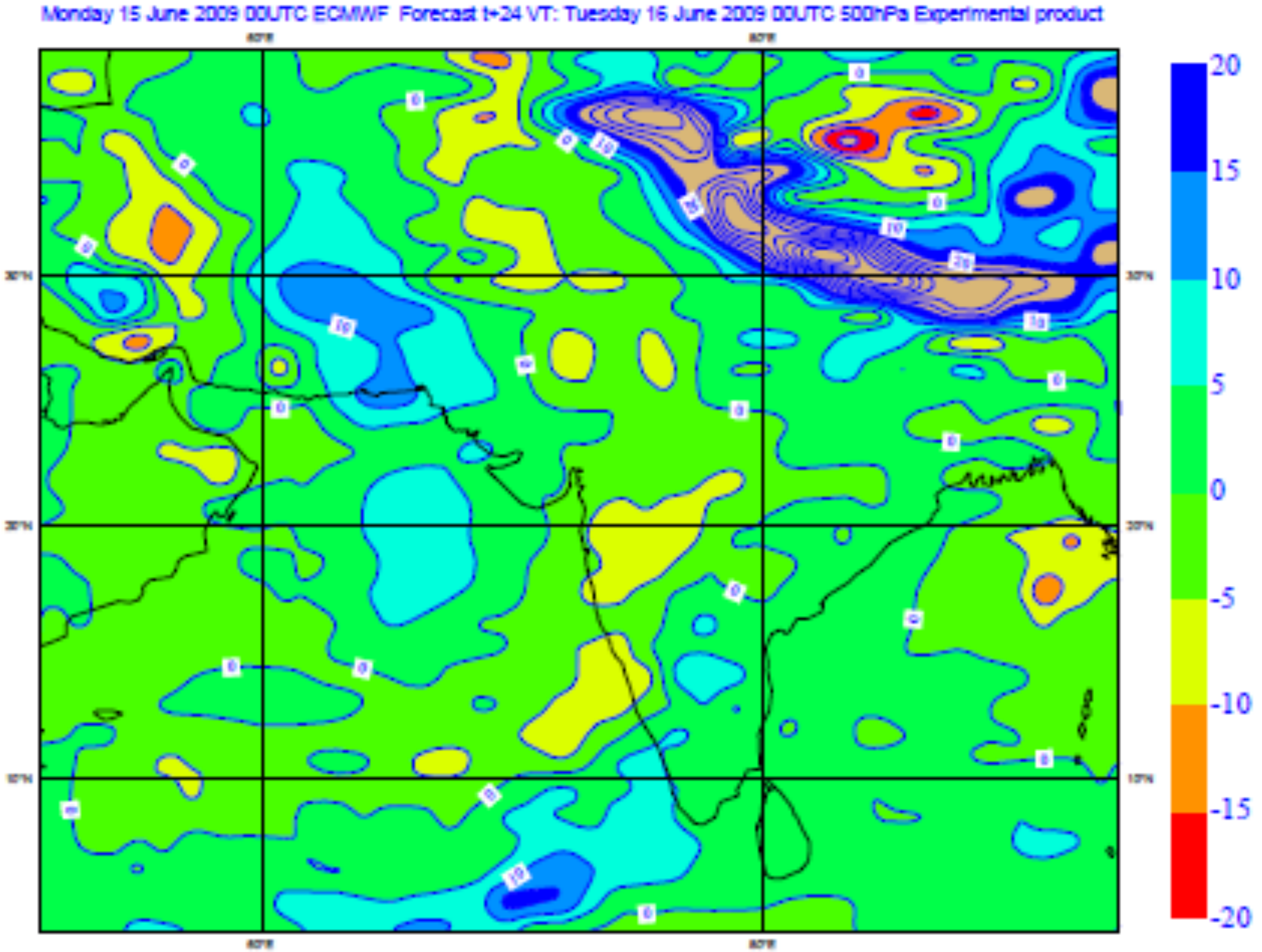
500hPa adiabatic meridional wind tendencies (T159)

Funded by the European Union

with aliasing



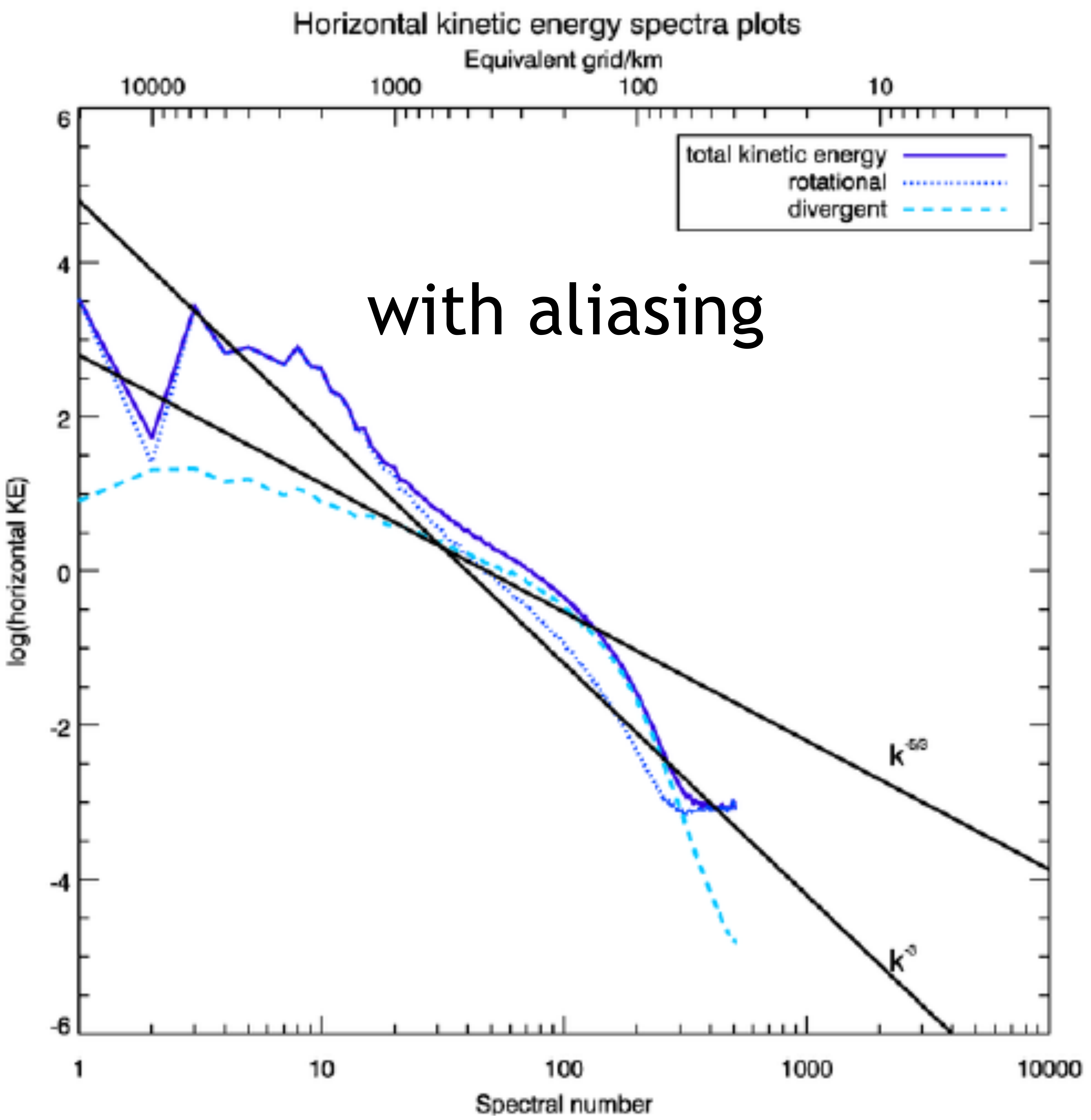
filtered

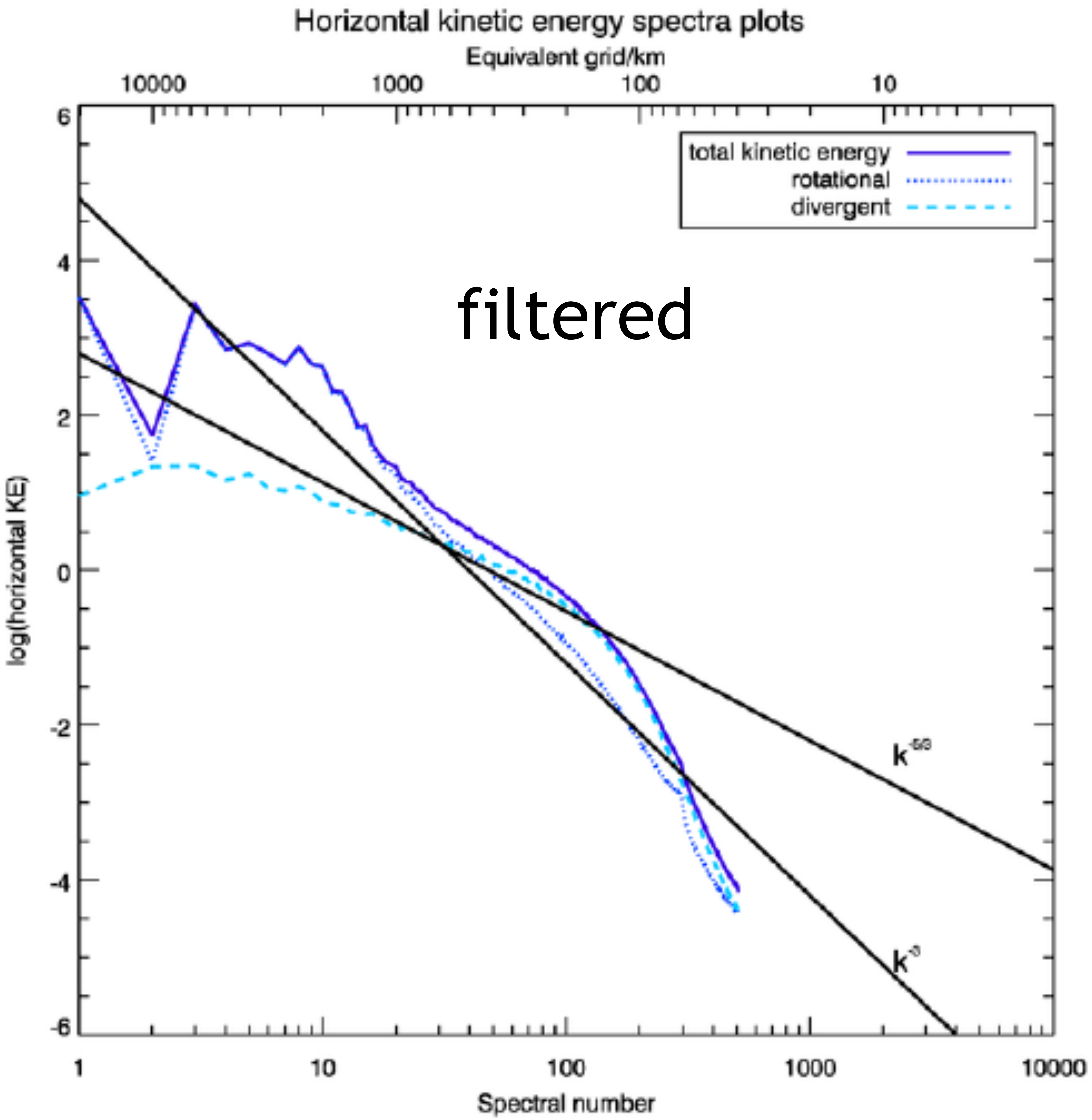


Funded by the European Union

aliasing example

kinetic energy spectra, 100 hPa





Funded by the European Union

alternatives to using a filter

Idea: use more grid points than spectral coefficients

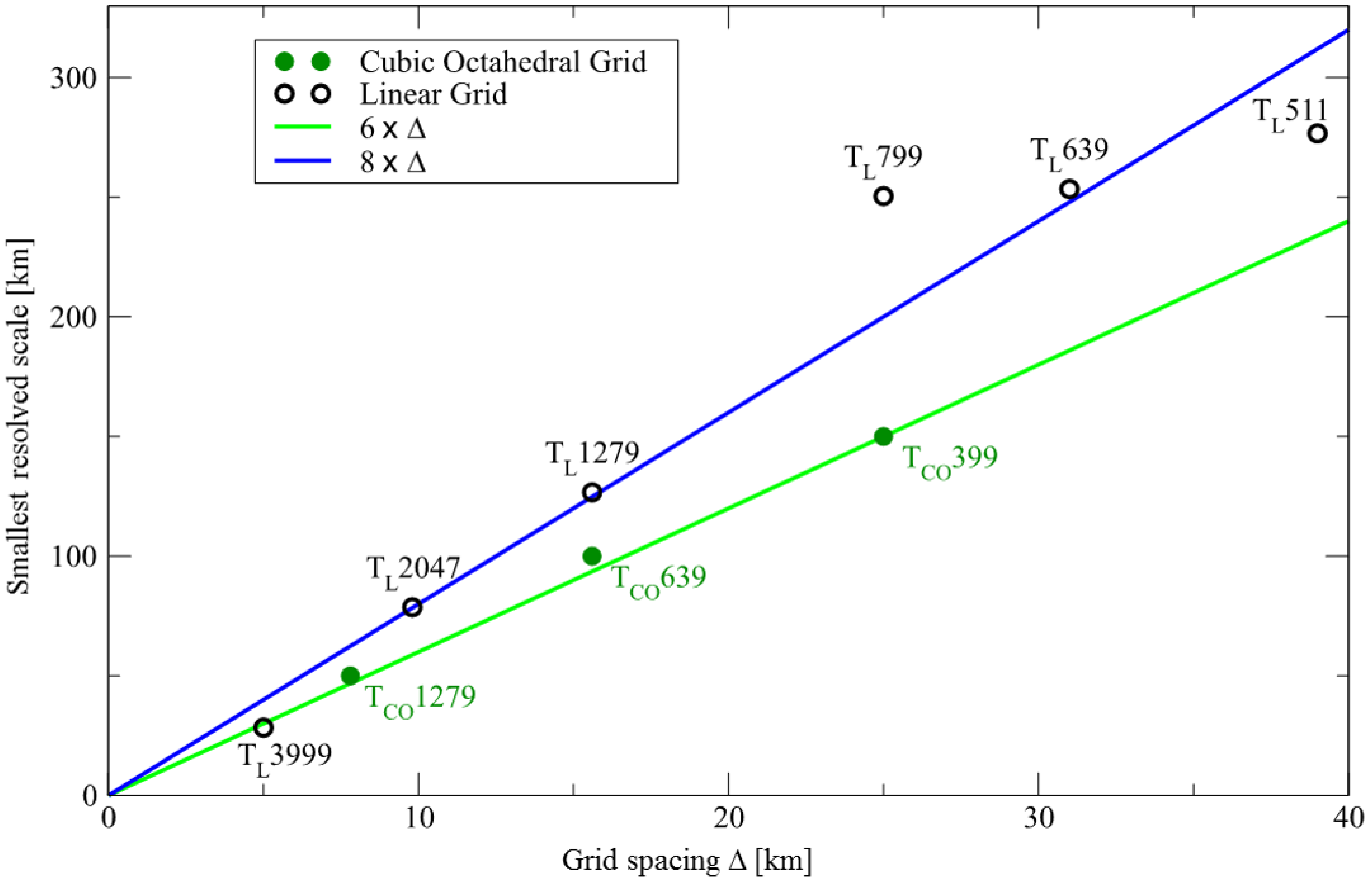
Orszag, 1971:

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 *(Wedi, 2014)*
	Spatial filter range

Funded by the European Union

effective resolution

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



ESCAPE 2

Funded by the European Union

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

Funded by the European Union

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

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

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

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

FFT: Fast Fourier Transform

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

Funded by the European Union

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

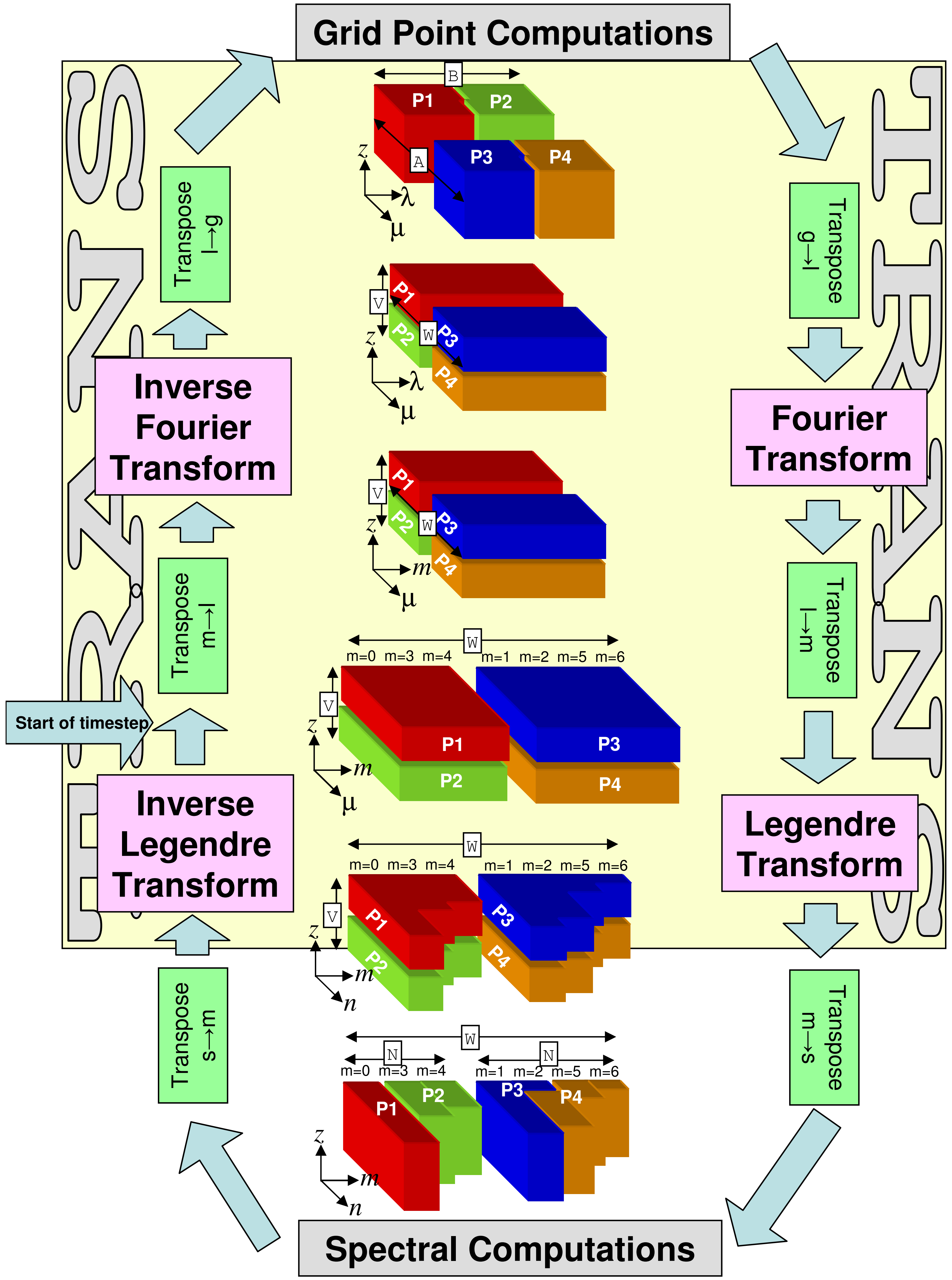
lots of MPI communication

ESCAPE 2

Funded by the European Union

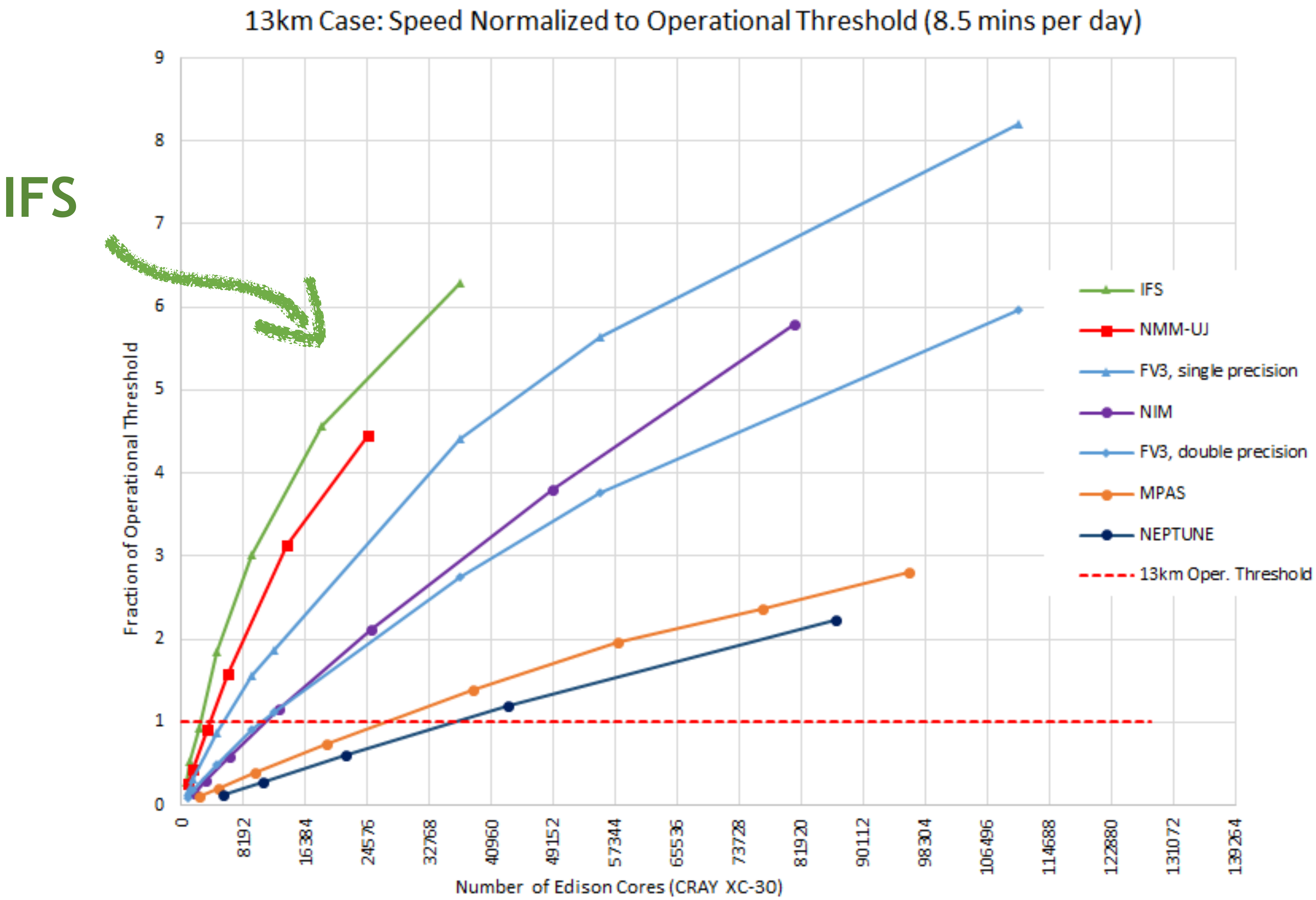
direct spectral transform

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



performance comparison
of IFS with other models

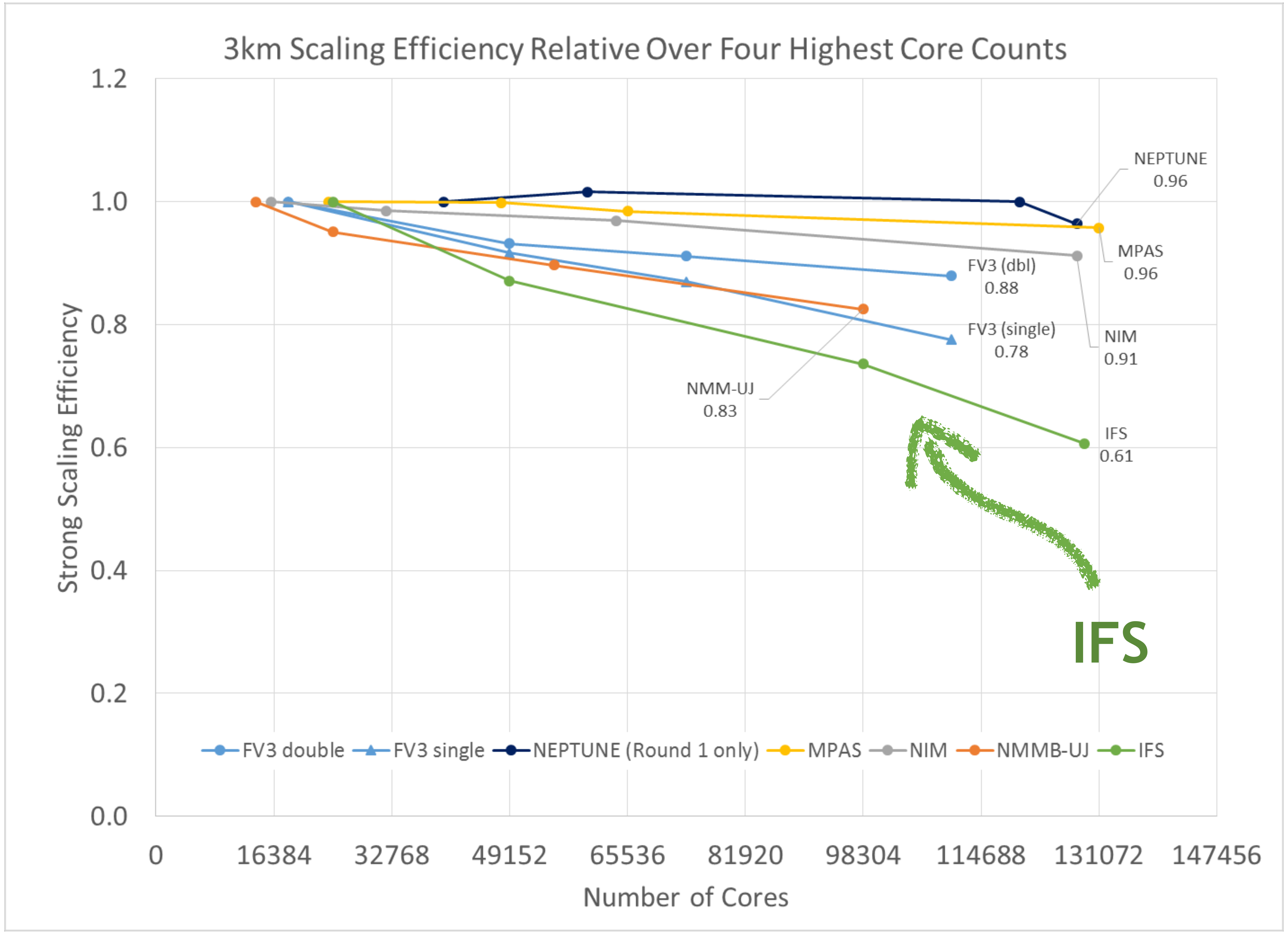
Funded by the European Union



(Michalakes et al, NGGPS AVEC report, 2015)

scalability comparison
of IFS with other models

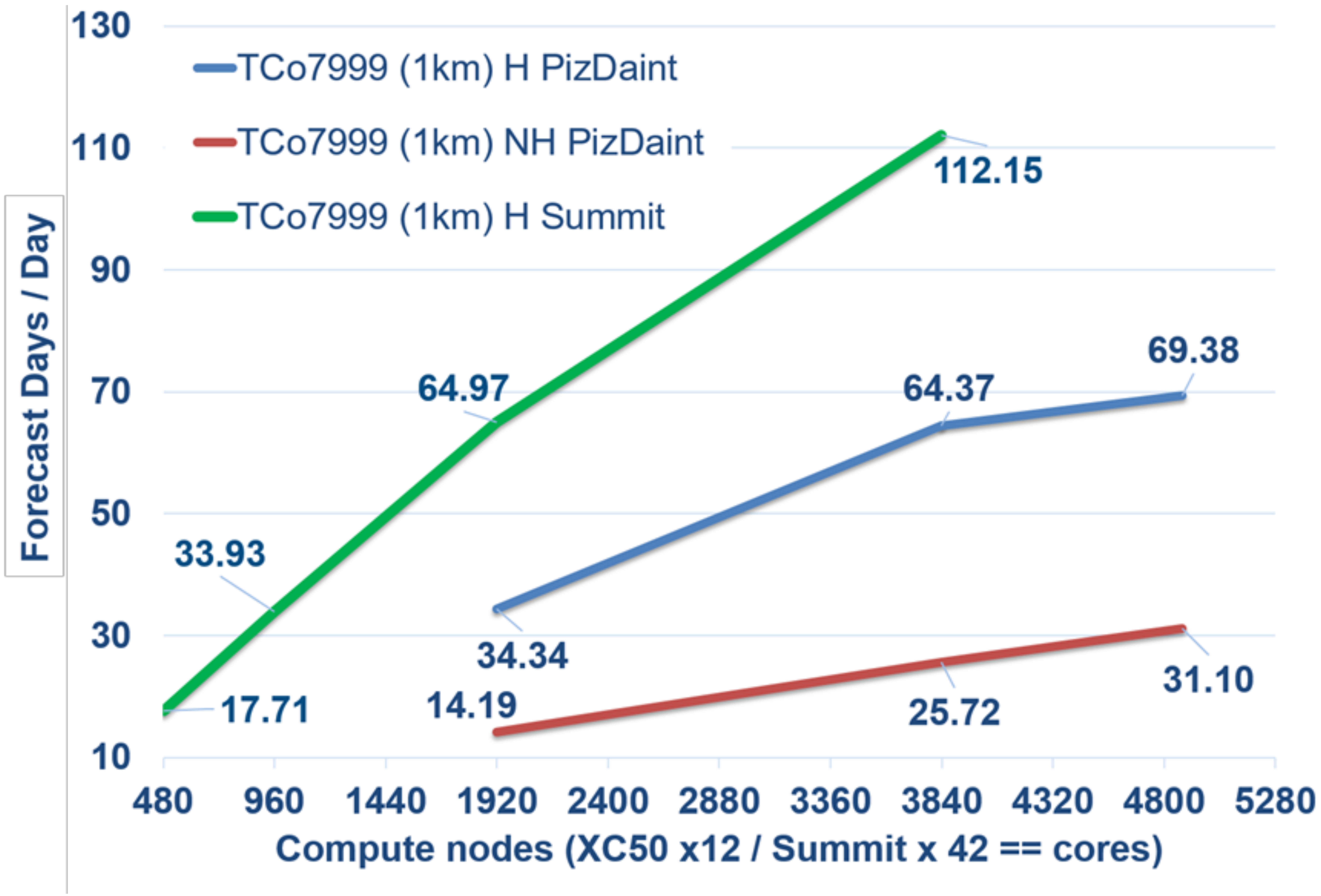
Funded by the European Union



(Michalakes et al, NGGPS AVEC report, 2015)

IFS scaling on Summit and PizDaint (CPU only)

Funded by the European Union



spectral transform vs discontinuous Galerkir

projected for 5km 2-day forecast

Funded by the European Union

	DG, horizontally explicit => 4s time-step, almost no communication	IFS (spectral transform): 240s time-step, lots of communication	DG (like on the left)
communication volume:	**34 TB on 2880 MPI procs**	**427 TB on 2880 MPI procs**	**689 TB on 57600 MPI procs**
time to solution:	**4 hours**	**12 minutes**	**12 minutes**

optimisations by NVIDIA in ESCAPE

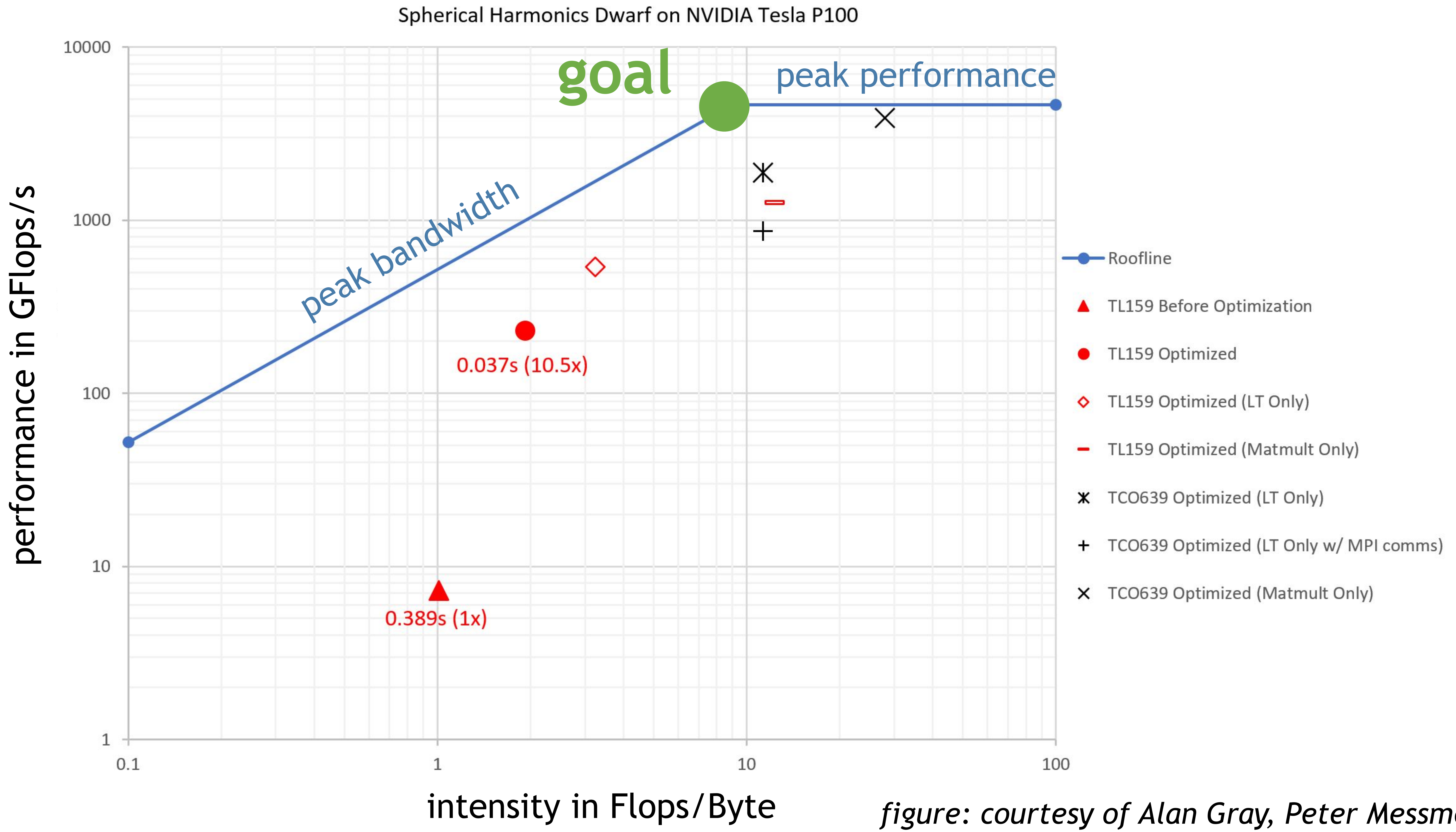


figure: courtesy of Alan Gray, Peter Messmer (NVIDIA)

Funded by the European Union

optimisations by NVIDIA in ESCAPE

Funded by the European Union

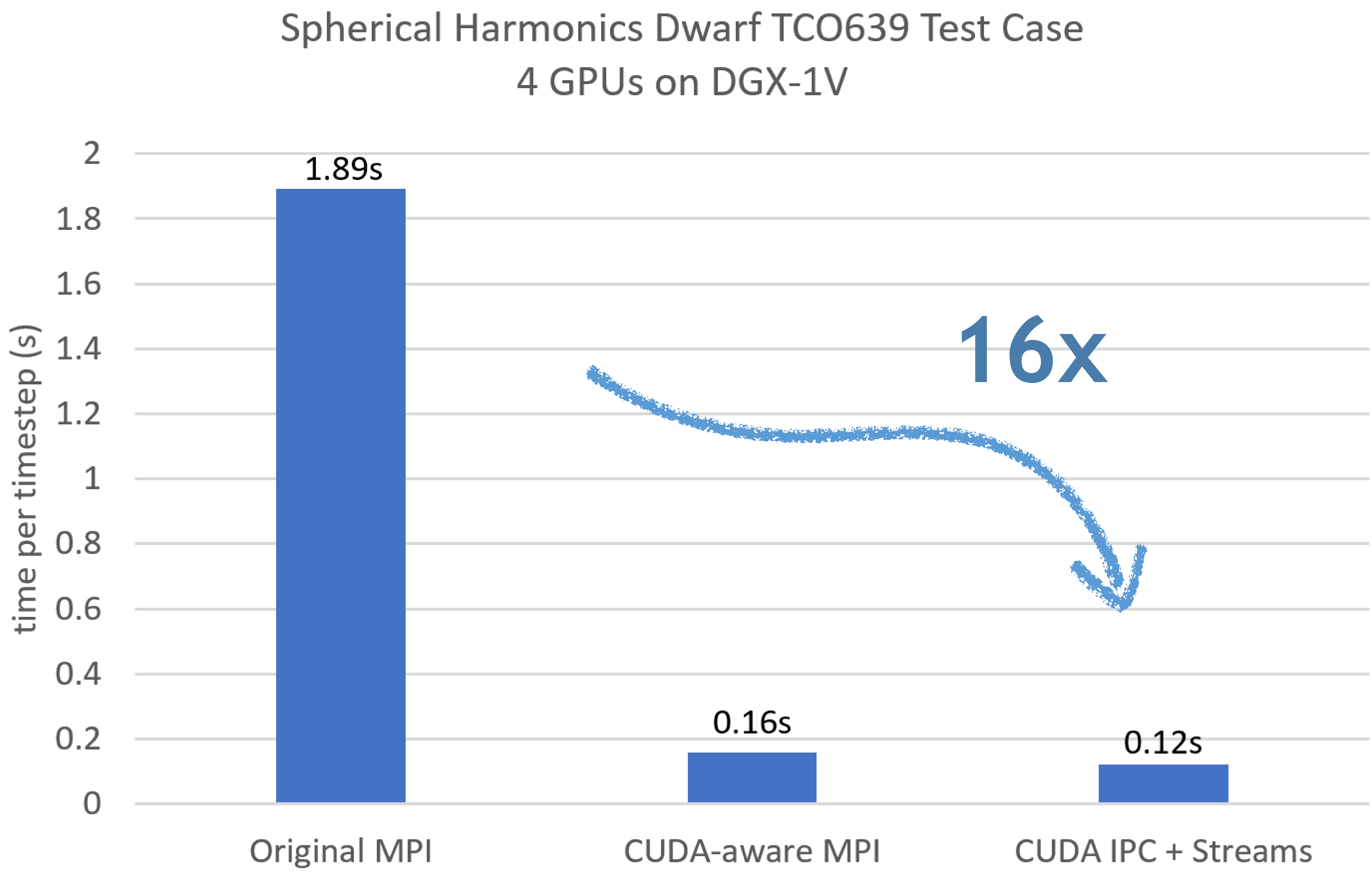
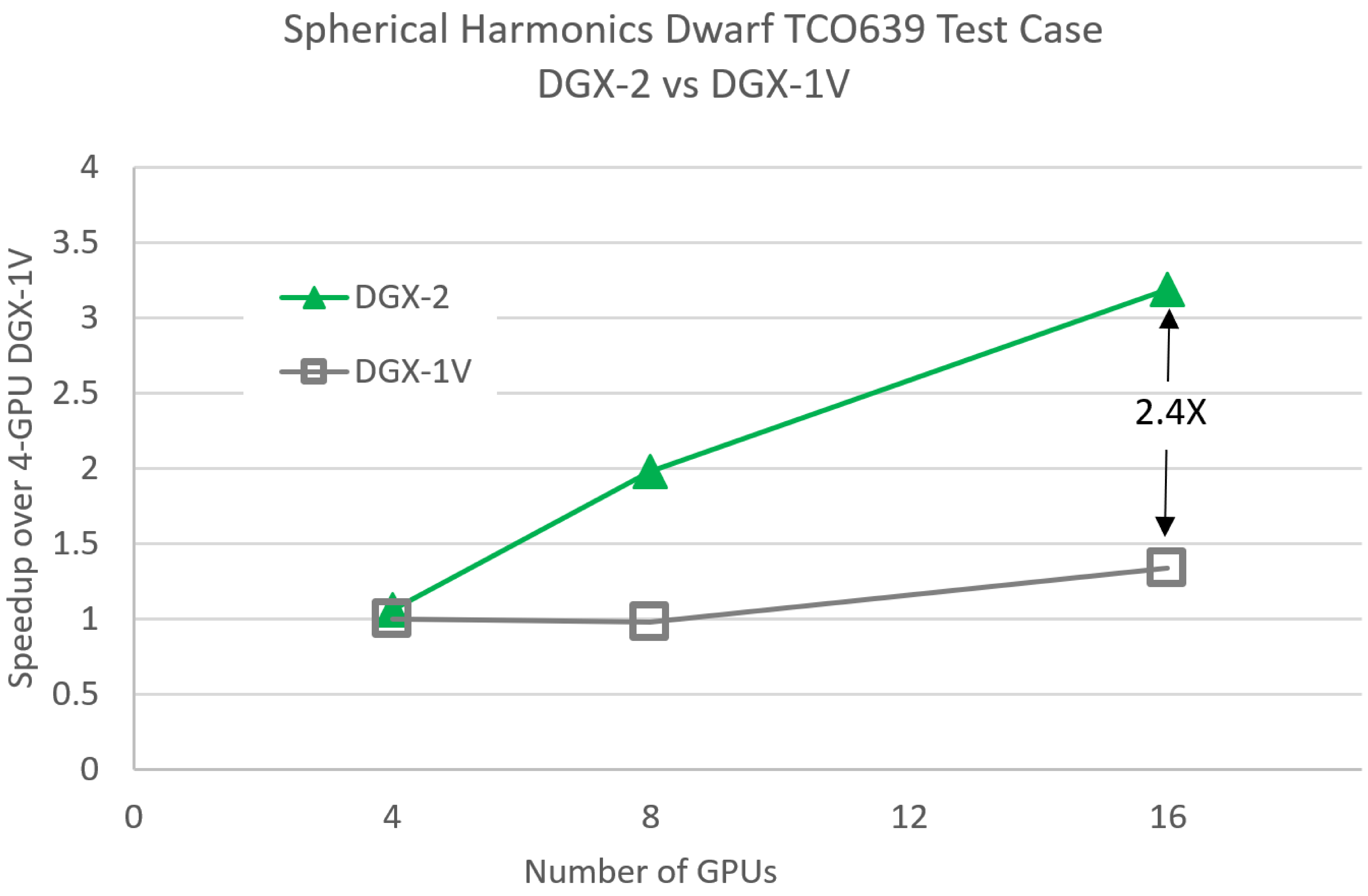


figure: courtesy of Alan Gray, Peter Messmer (NVIDIA)

optimisations by NVIDIA in ESCAPE

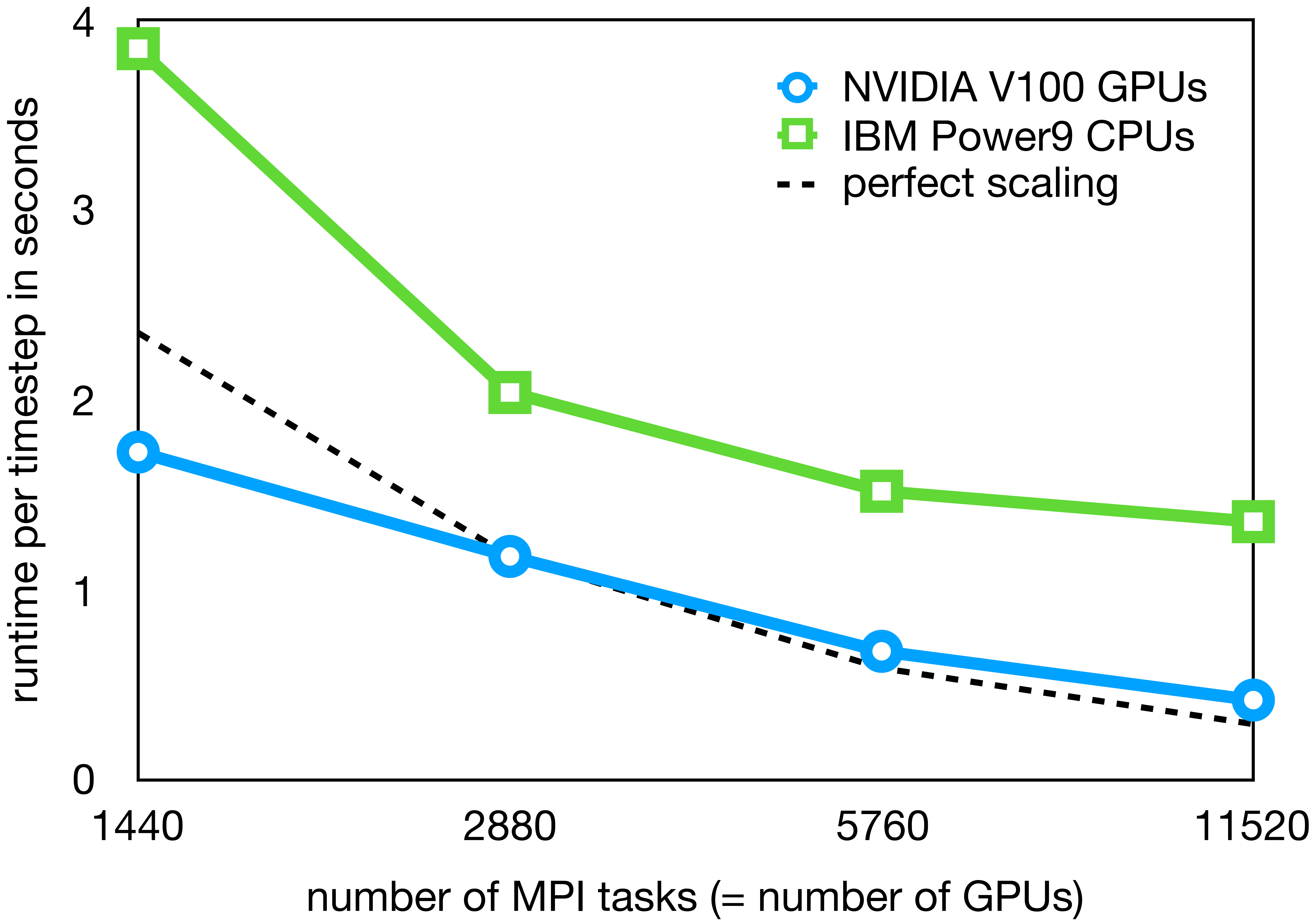


DGX-1V uses MPI for >=8 GPUs (due to lack of AlltoAll links), all others use CUDA IPC.
DGX-2 results use pre-production hardware.

figure: courtesy of Alan Gray, Peter Messmer (NVIDIA)

Funded by the European Union

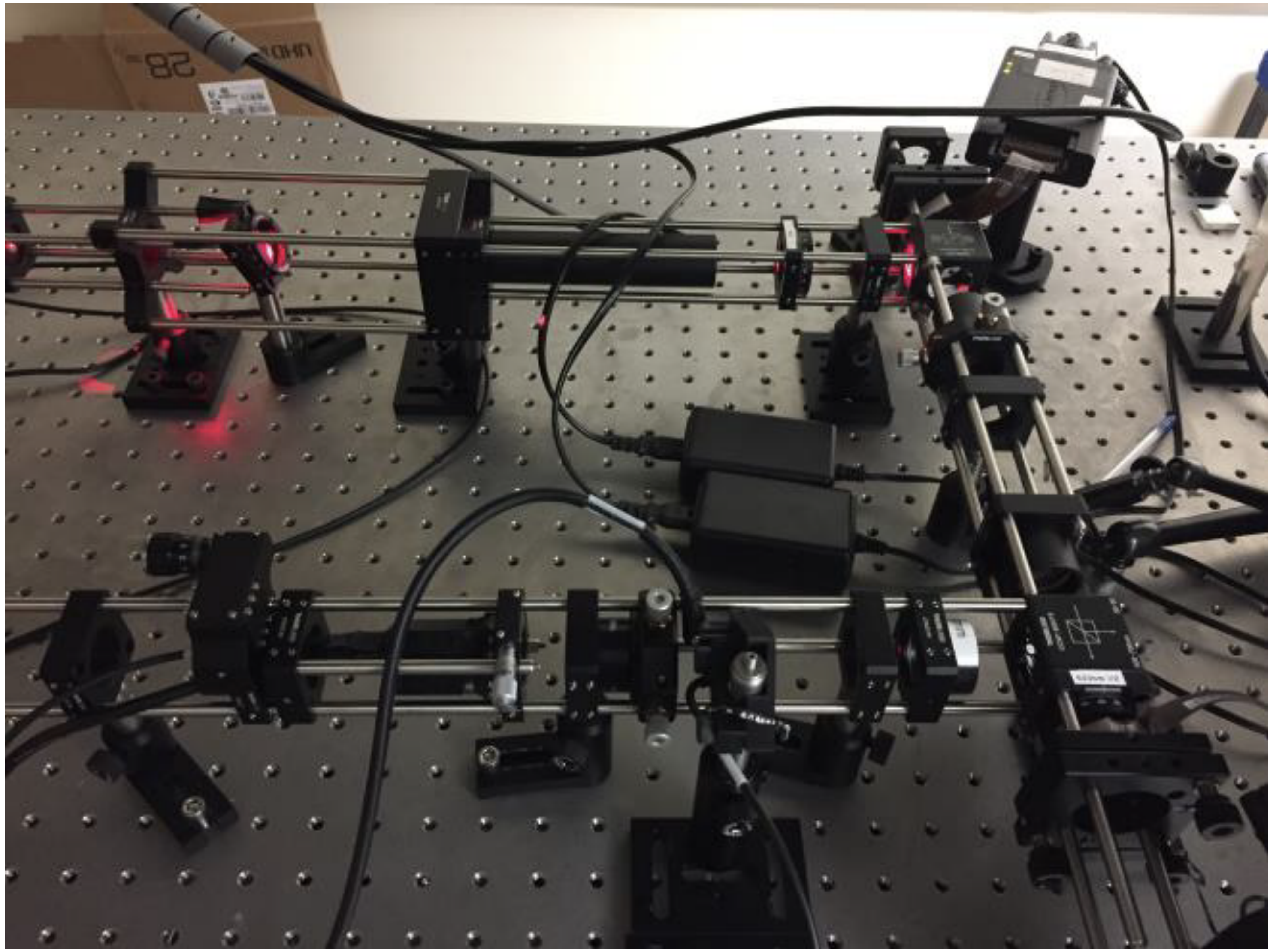
GPUs vs CPUs on Summit

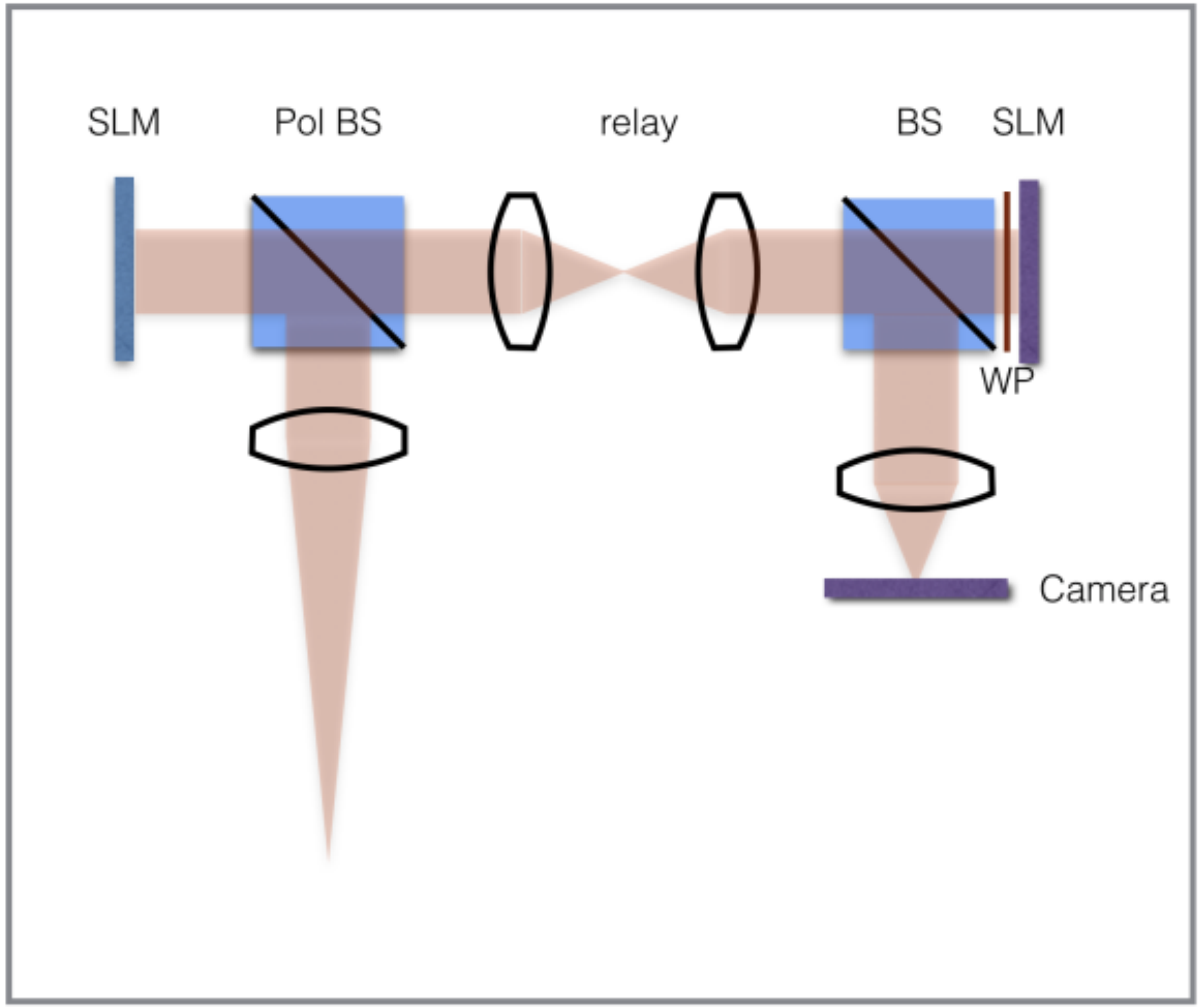


Funded by the European Union

Optalysys: optical processor for spectral transform

Funded by the European Union





Figures used with permission from Optalysys, 2017

ESCAPE 2

Fast Legendre Transform

Funded by the European Union

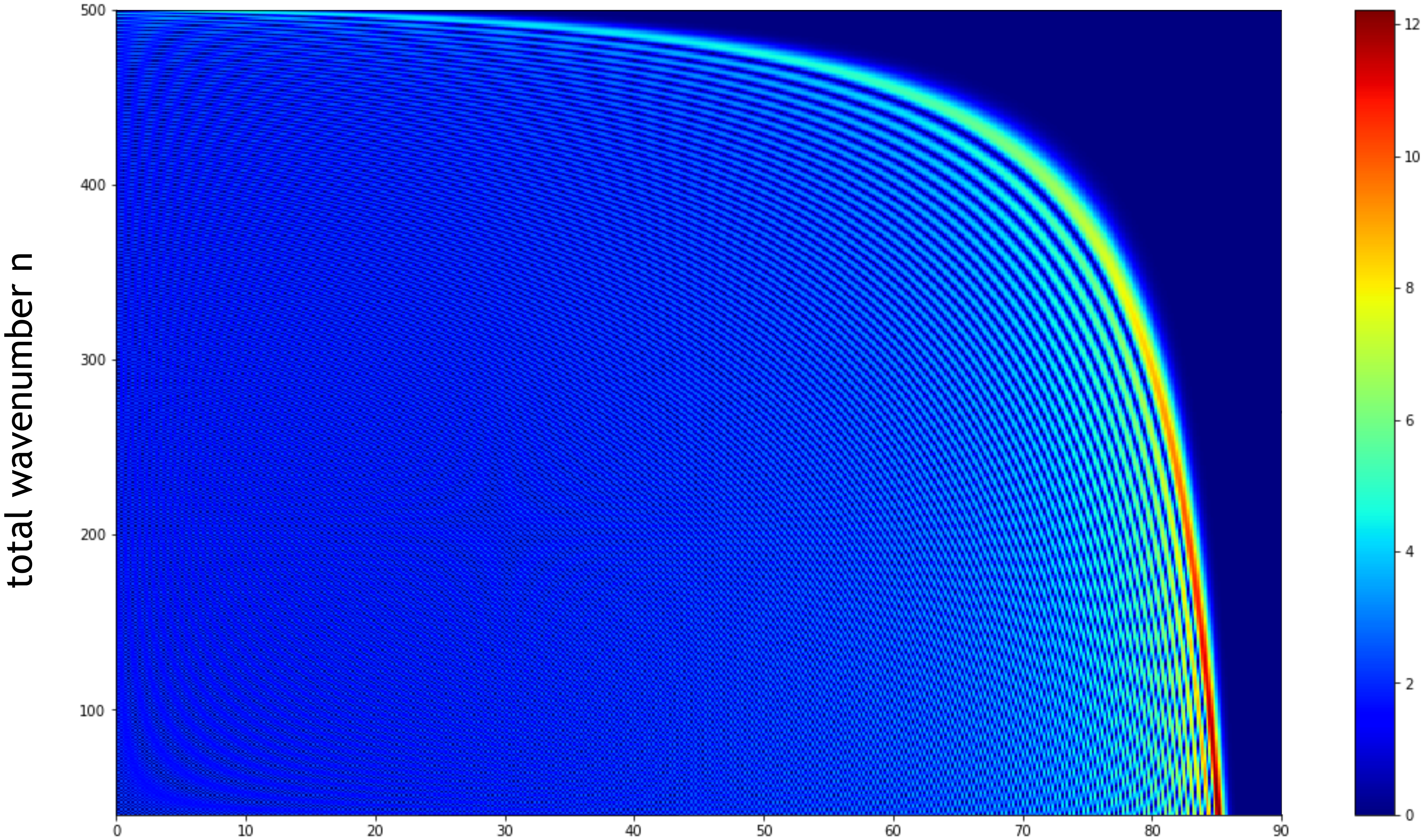
matrix of Legendre polynomials

truncation N=500, zonal wavenumber m=40

FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns



equator latitude ϕ pole

Fast Legendre Transform

ESCAPE 2

Funded by the European Union

matrix of
Legendre polynomials

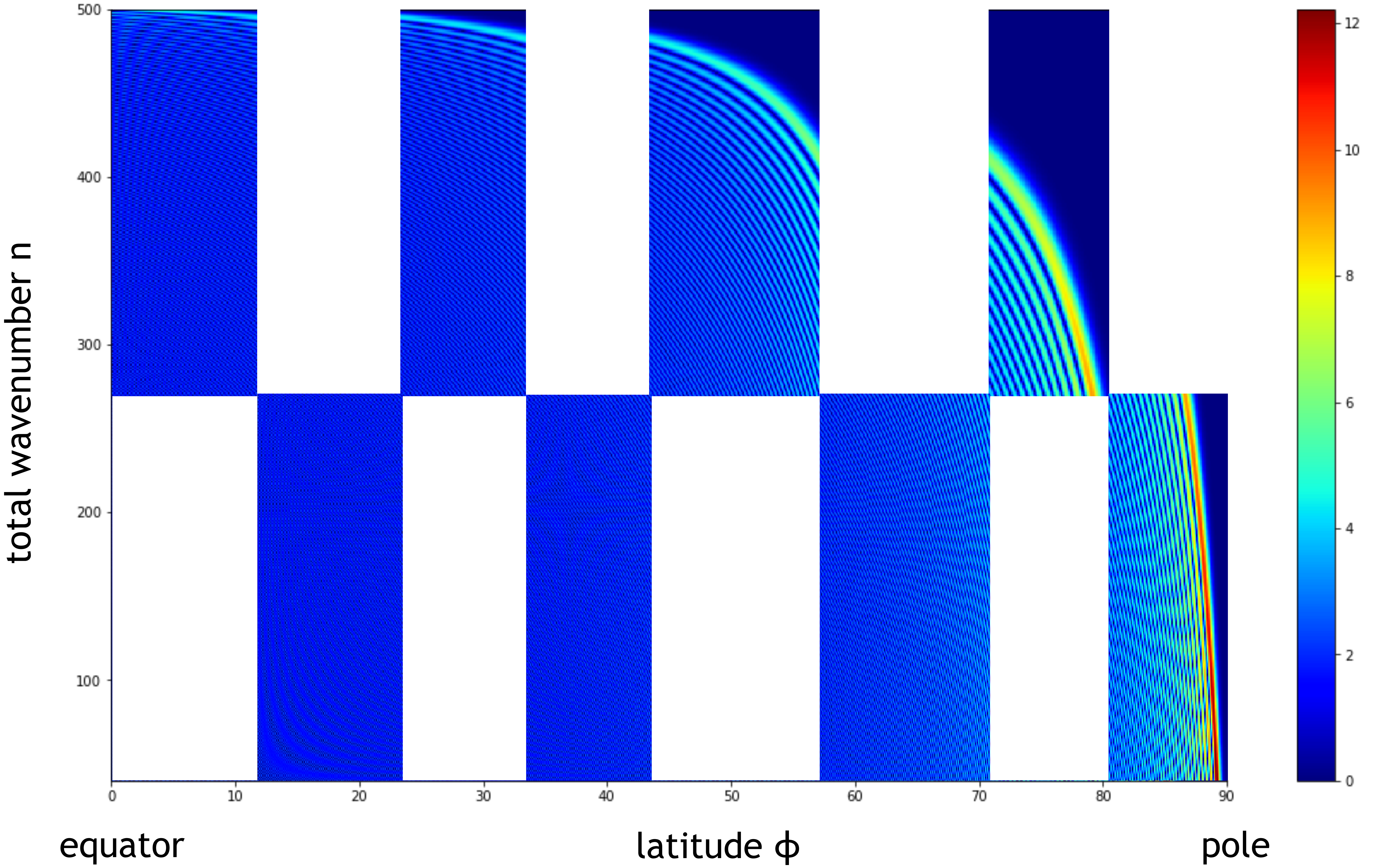
truncation N=500,
zonal wavenumber
m=40

FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns

step 3: reorder columns



Fast Legendre Transform

Funded by the European Union

matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

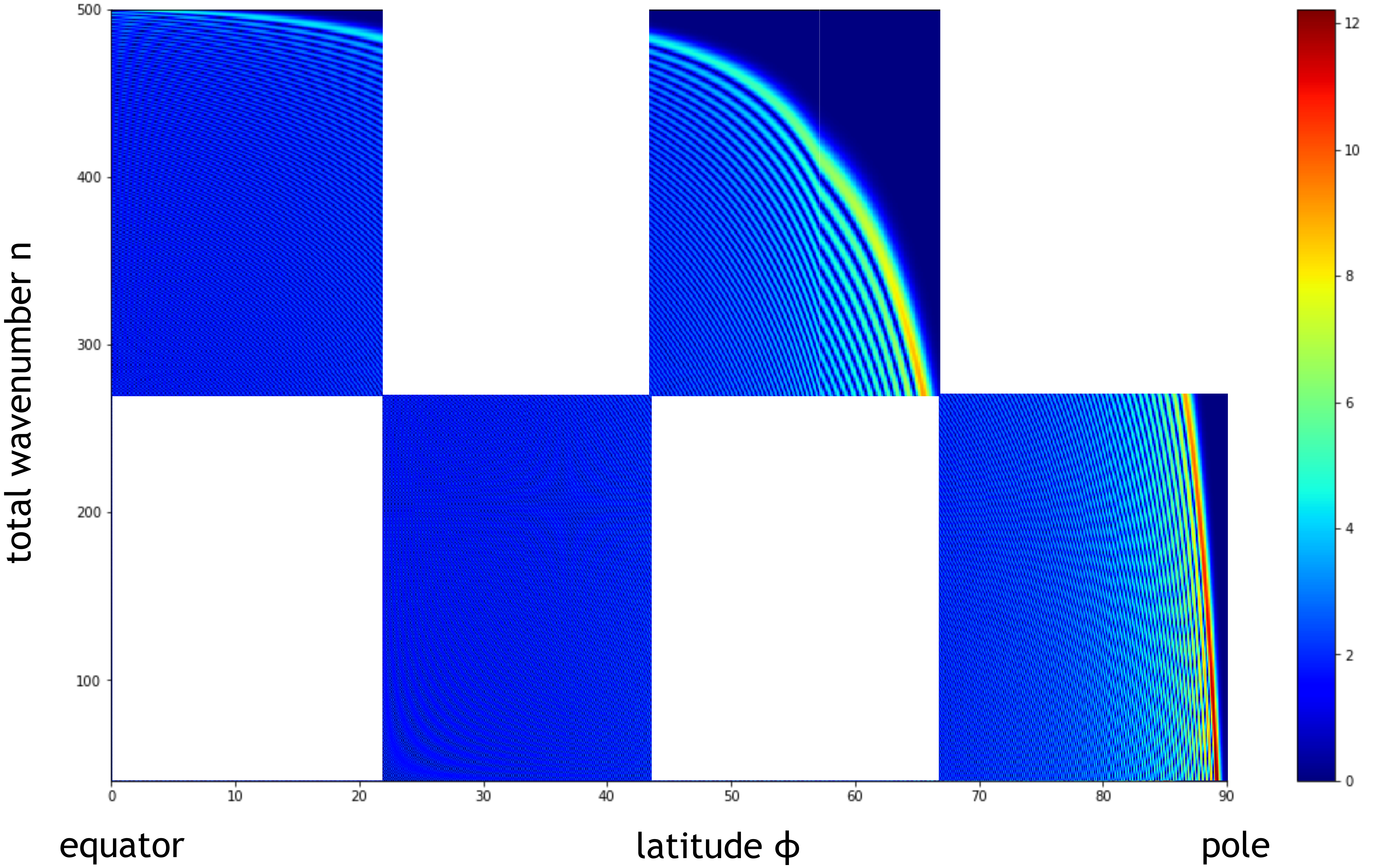
FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns

step 3: reorder columns

step 4: apply to each block recursively



Fast Legendre Transform

Funded by the European Union

matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

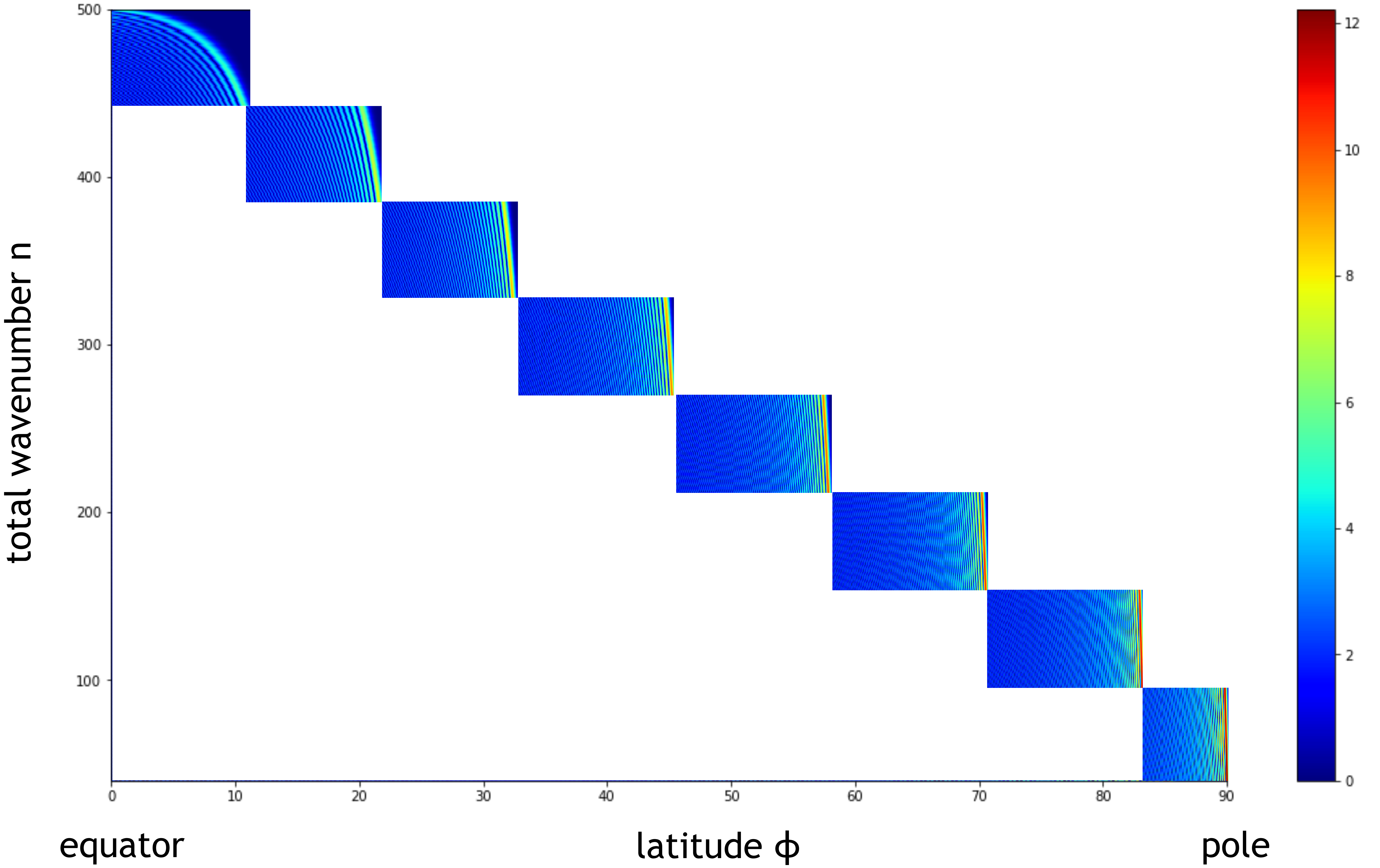
FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns

step 3: reorder columns

step 4: apply to each block recursively



Fast Legendre Transform

matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

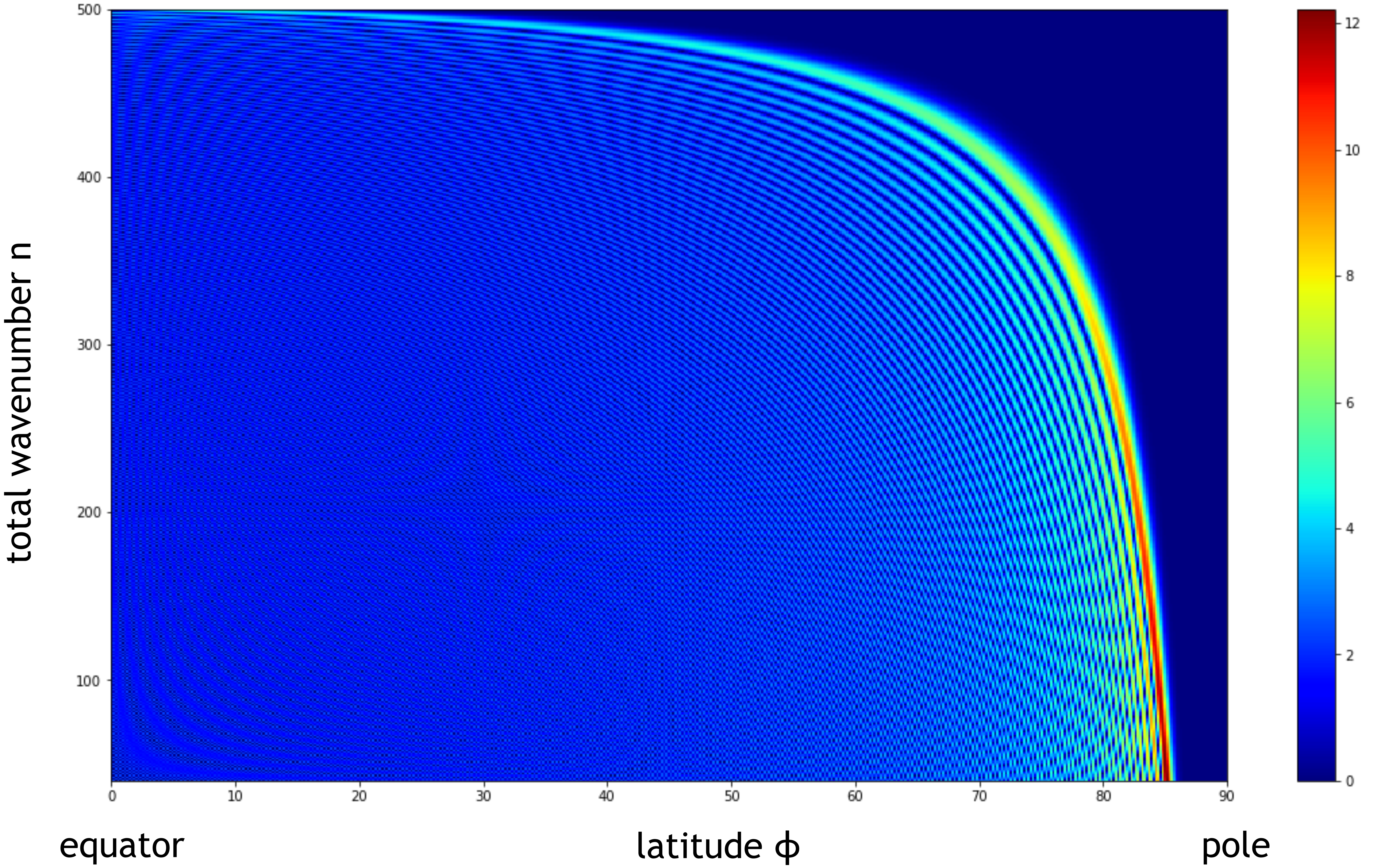
FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns

step 3: reorder columns

step 4: apply to each block recursively



Fast Legendre Transform

Funded by the European Union

matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=100

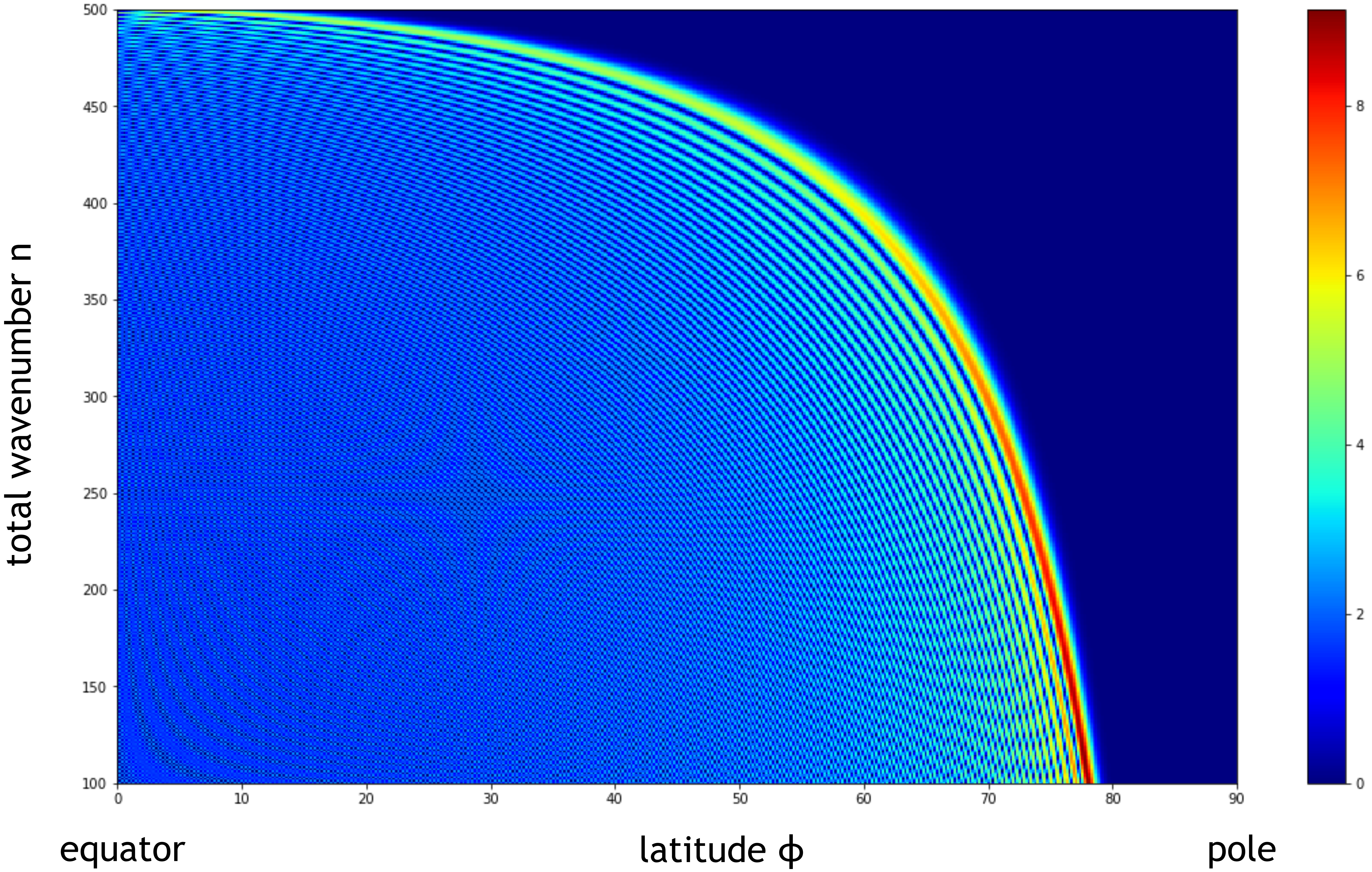
FLT:

step 1: split matrix into two rows

step 2: use interpolation to empty half of the columns

step 3: reorder columns

step 4: apply to each block recursively



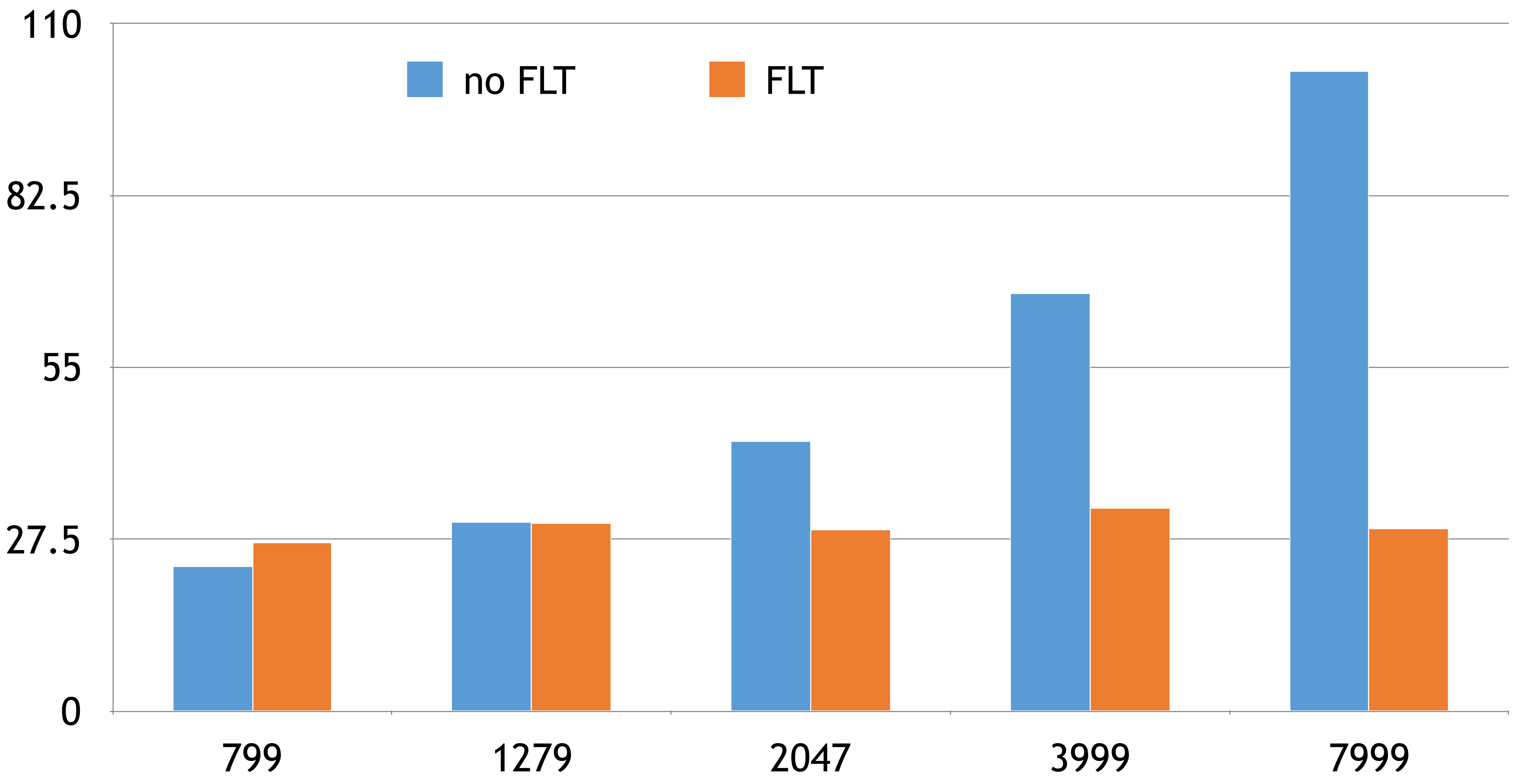
ESCAPE 2

Funded by the European Union

Fast Legendre Transform

floating point operations

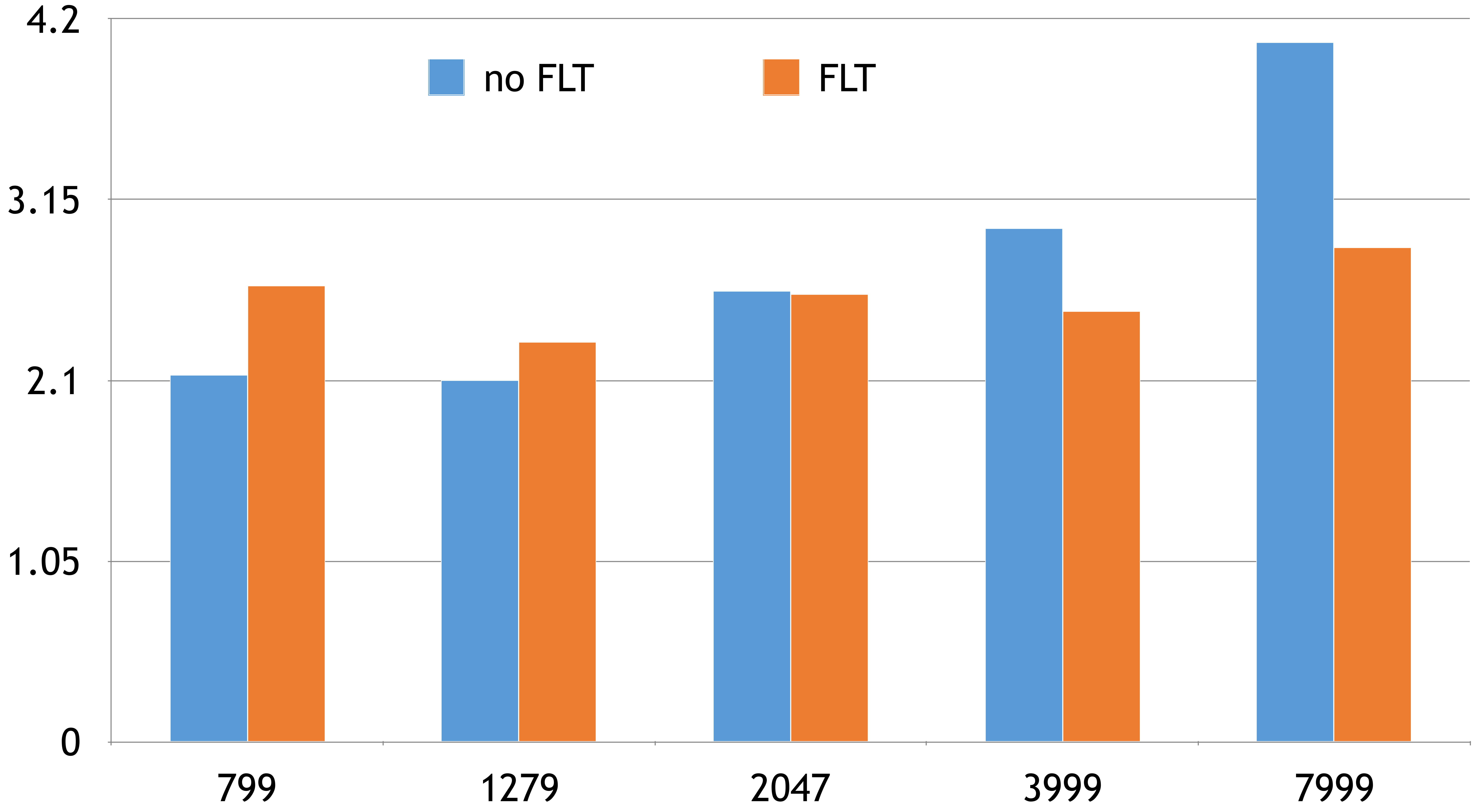
Number of floating point operations for direct or inverse spectral transforms of a single field, scaled by $N^2 \log^3 N$



Fast Legendre Transform

wallclock time

Average wall-clock time compute cost of 10^7 spectral transforms scaled by $N^2 log^3 N$



Funded by the European Union

- Images on slide 2 used under license from shutterstock.com