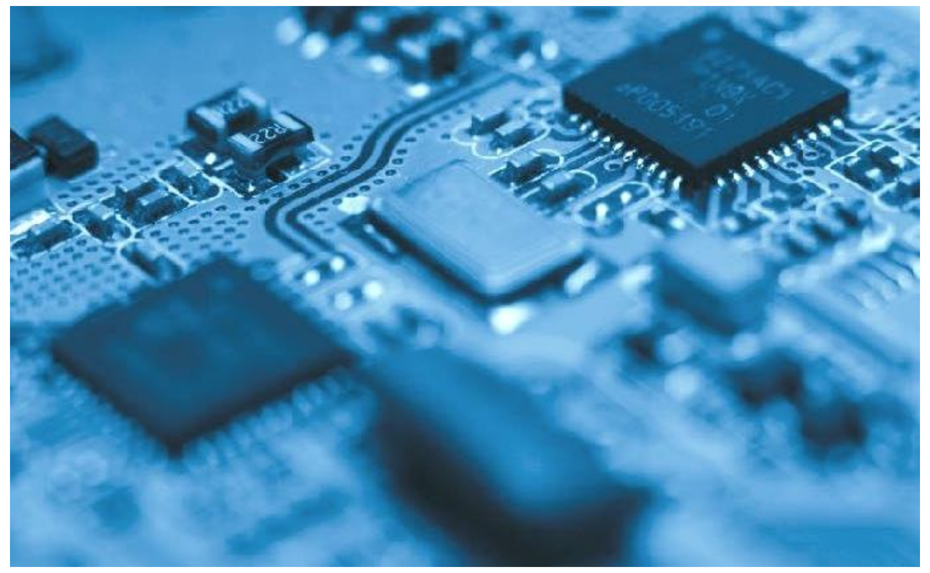
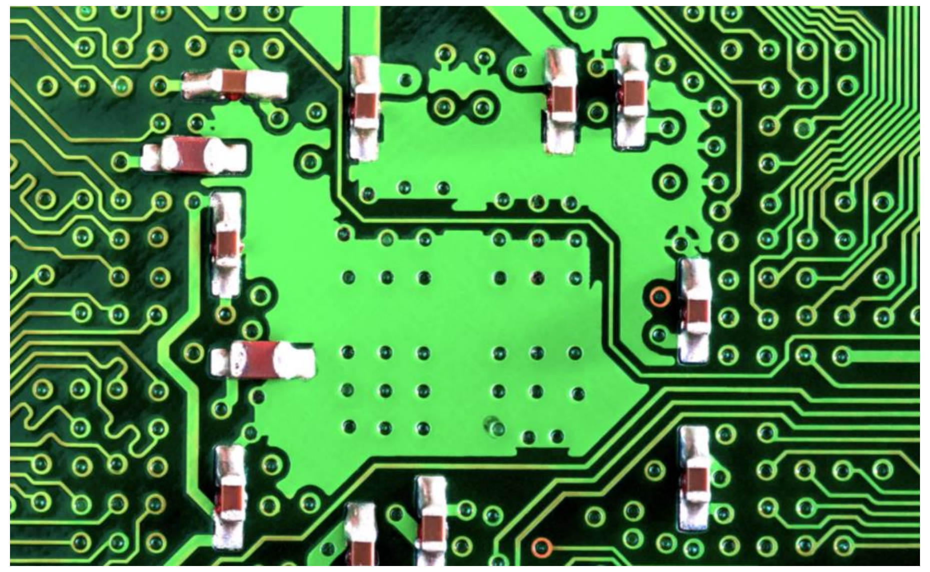
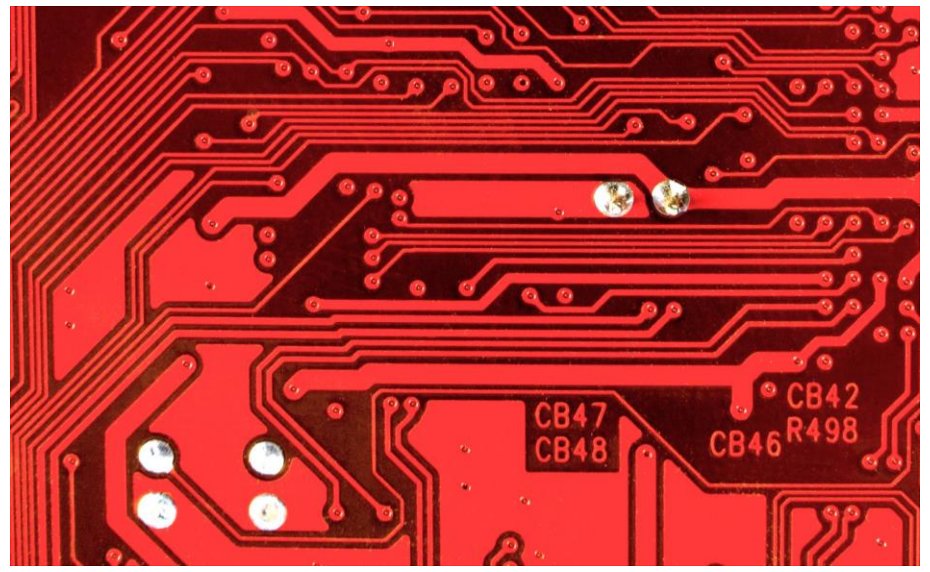
Funded by the European Union

Co-ordinated by ECMWF



ESCAPE 2

# Spectral Transform

***Michail Diamantakis***

based on the lecture slides by **Andreas Mueller**

ECMWF DKRZ Max-Planck-Institut für Meteorologie MeteoSwiss Barcelona Supercomputing Center cea Loughborough University RMI Politecnico Milano 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

# IFS (Integrated Forecast System)

ESCAPE 2

Funded by the European Union

technology applied at ECMWF for
the last 30 years

- spectral transform
- semi-Lagrangian
- semi-implicit

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

https://www.ecmwf.int/escape

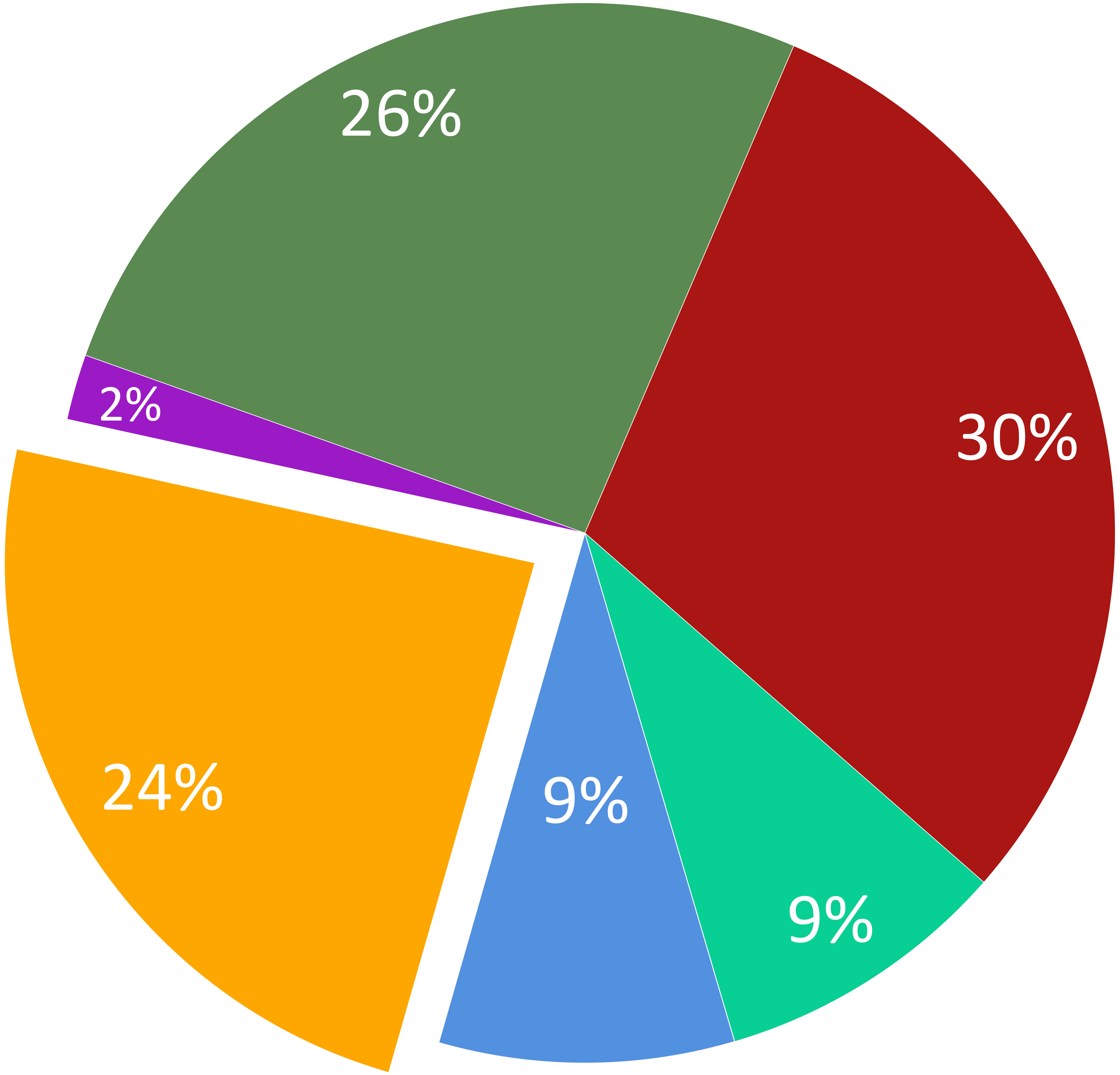
*"ESCAPE aimed to develop world-class, extreme-scale computing capabilities for European operational numerical weather prediction (NWP) and future climate models."*

# 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 | semi-implicit solver
grid point dynamics | physics+radiation
wave model | ocean model

ESCAPE 2

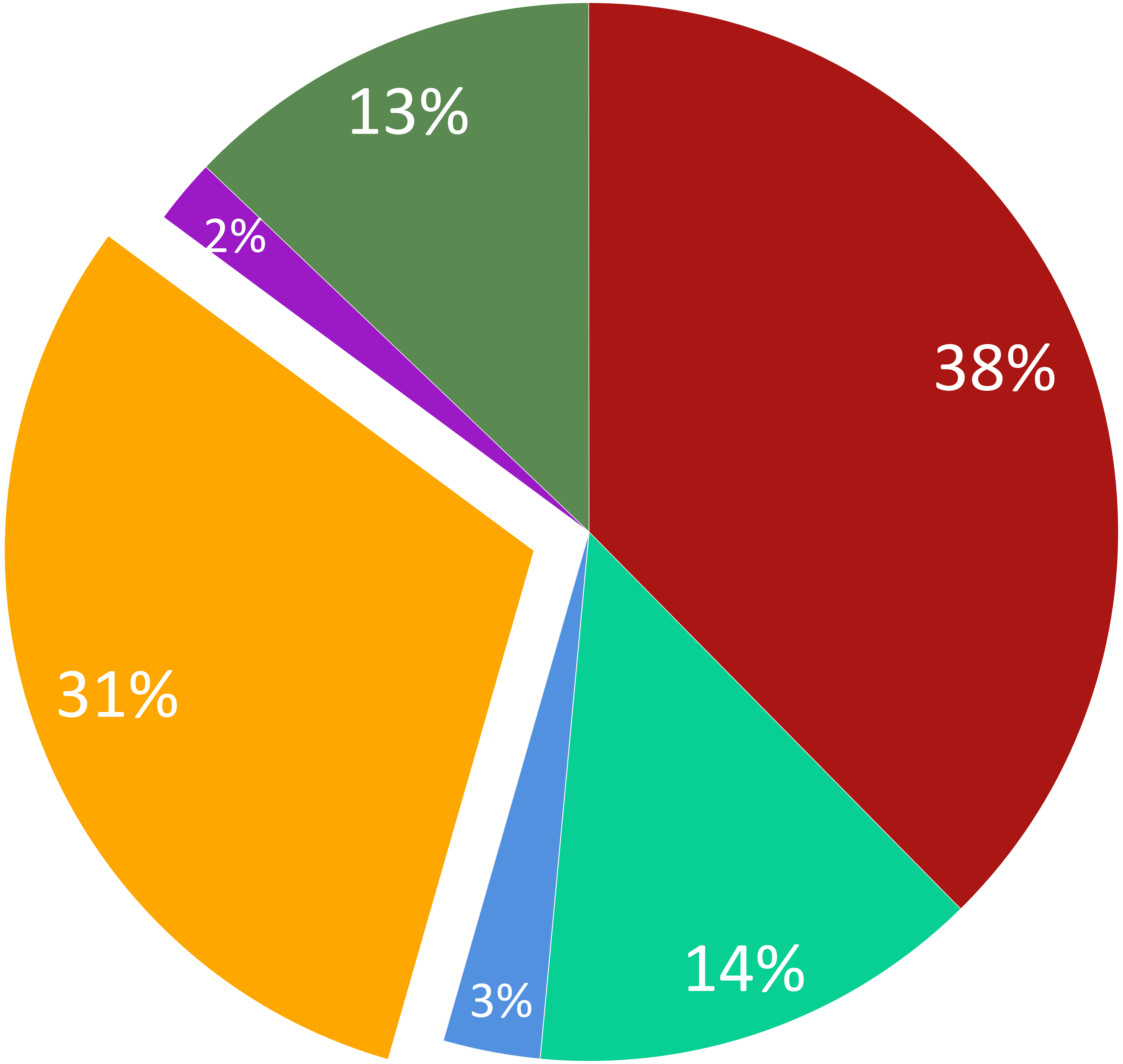
Funded by the European Union

# IFS (Integrated Forecast System)

technology applied at ECMWF for the last 30 years

- spectral transform
- semi-Lagrangian
- semi-implicit

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

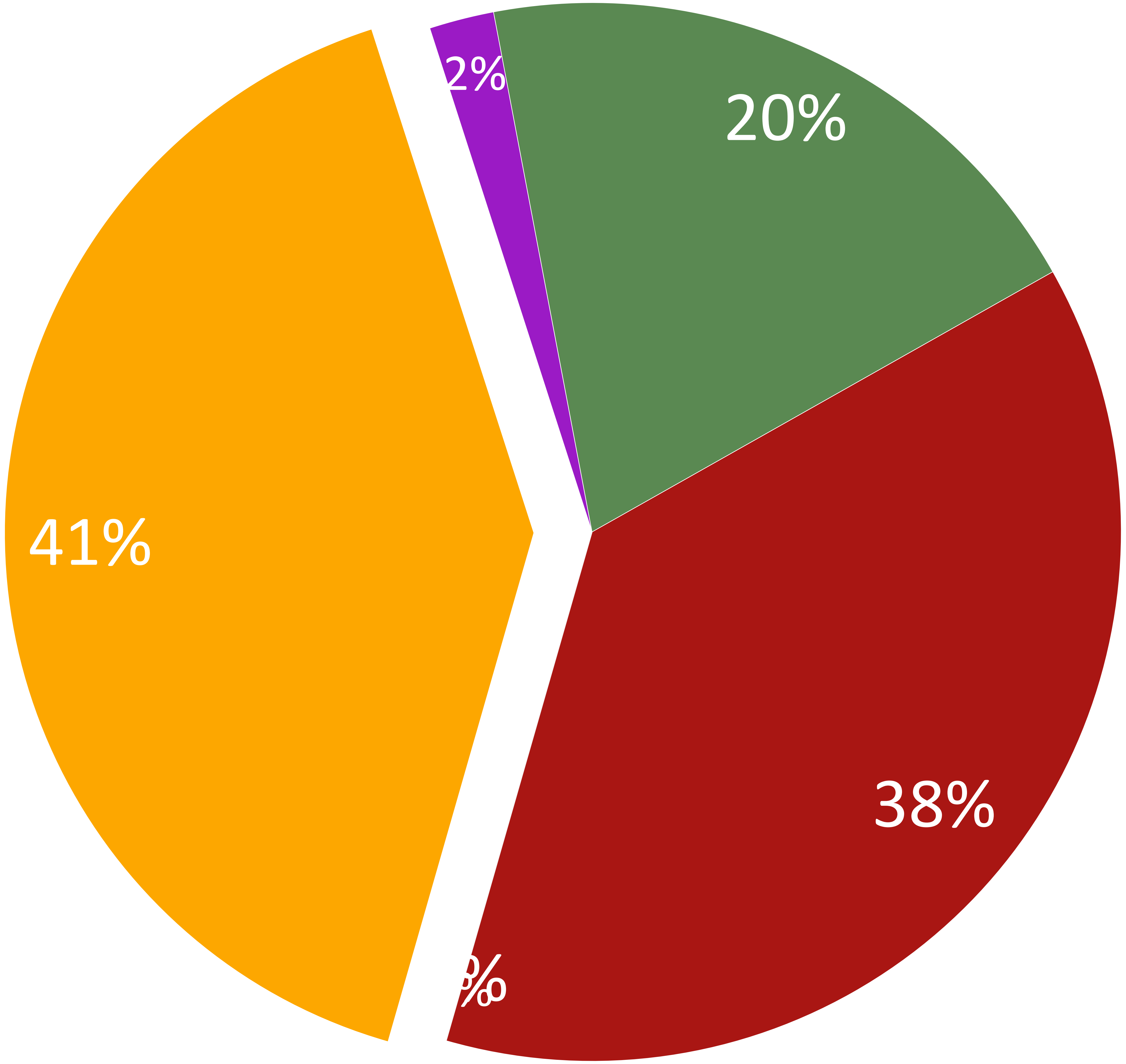


# IFS (Integrated Forecast System)

technology applied at ECMWF for the last 30 years

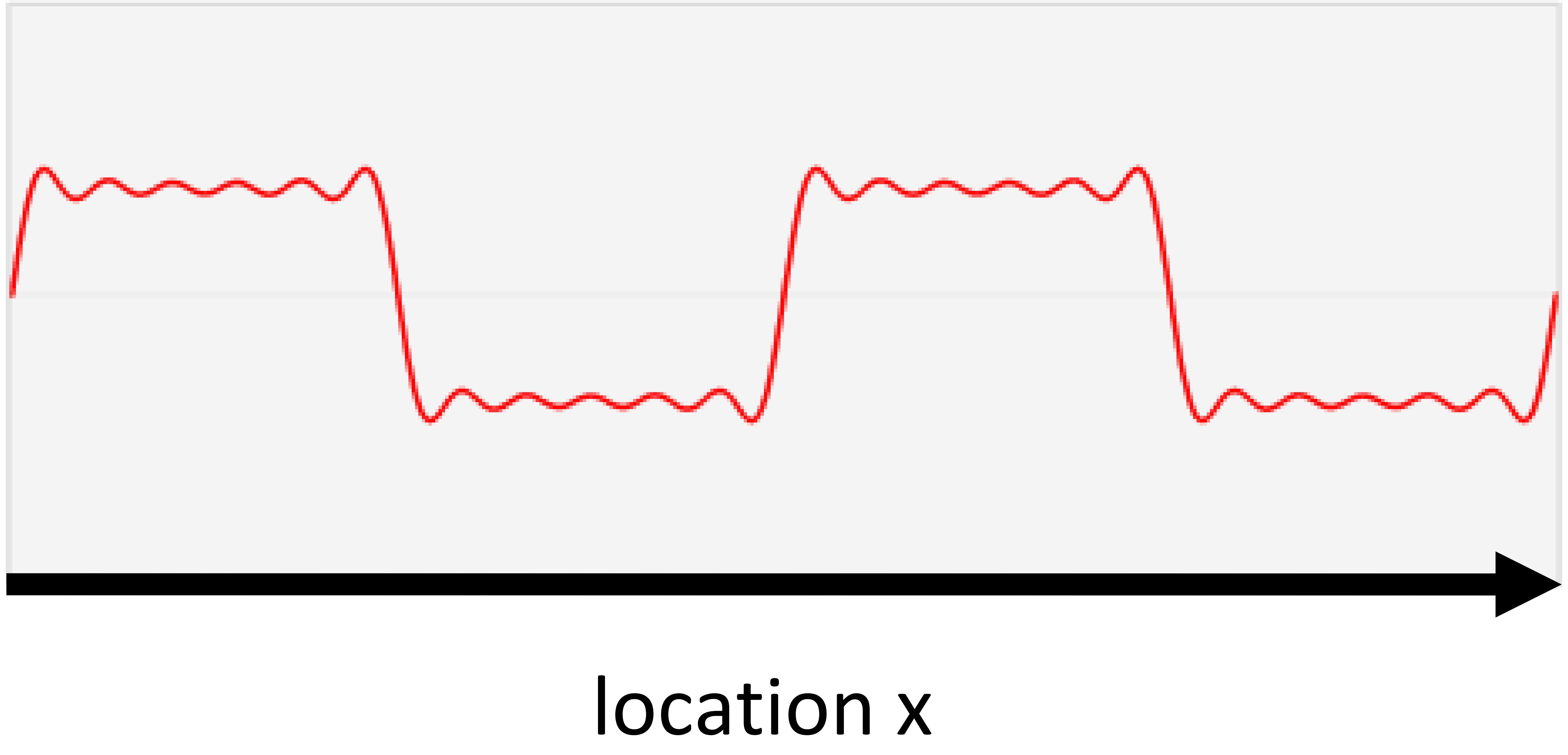
- spectral transform
- semi-Lagrangian
- semi-implicit

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



# Fourier transform

Fourier transform = Spectral transform in 1D

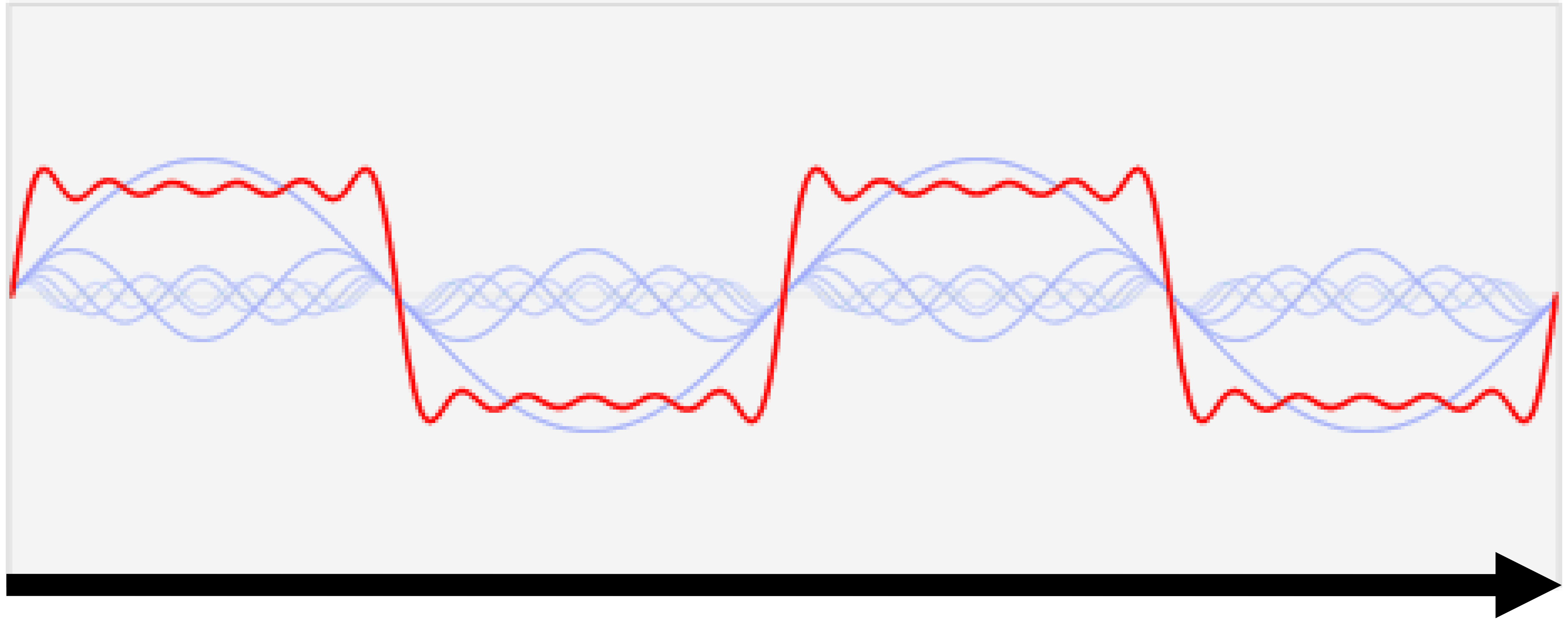


ESCAPE 2

Funded by the European Union

# Fourier transform

Fourier transform = Spectral transform in 1D



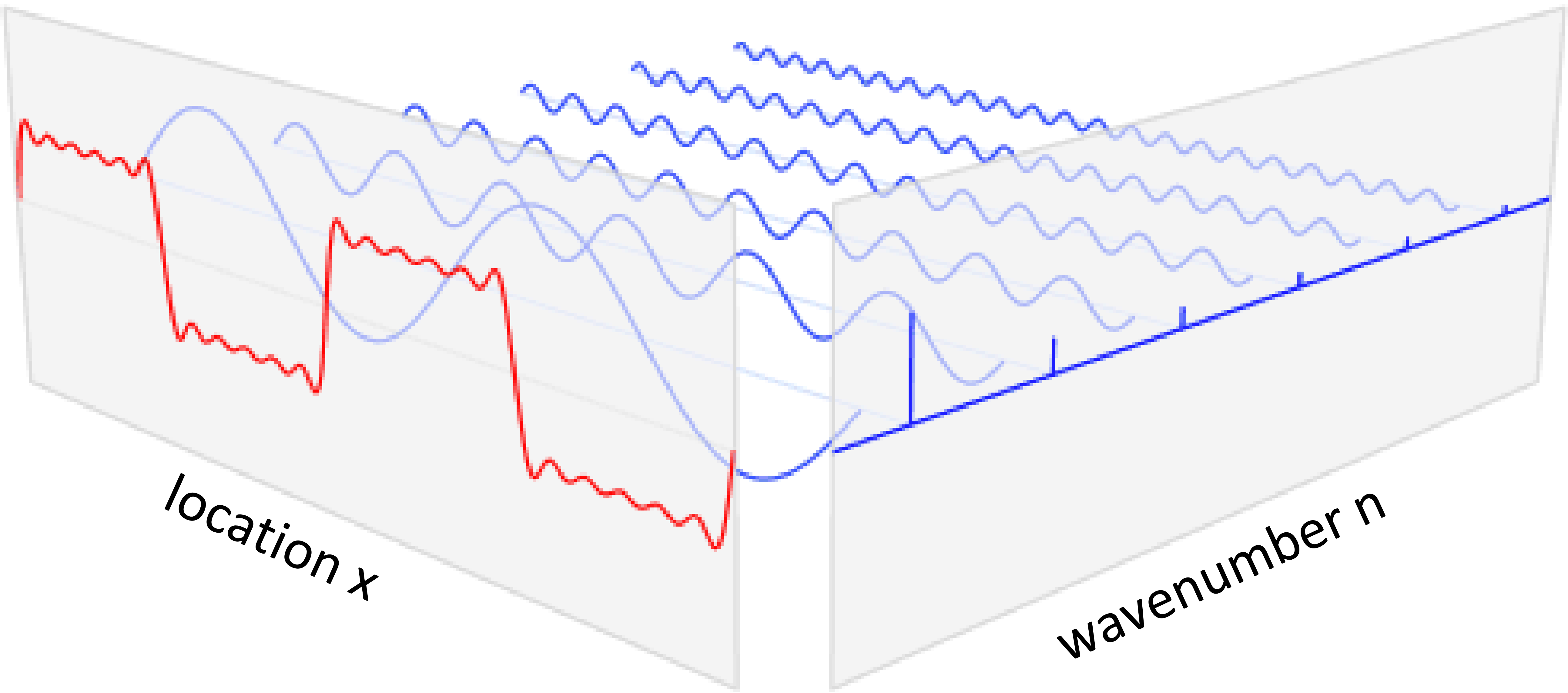
location x

ESCAPE 2

Funded by the
European Union

# Fourier transform

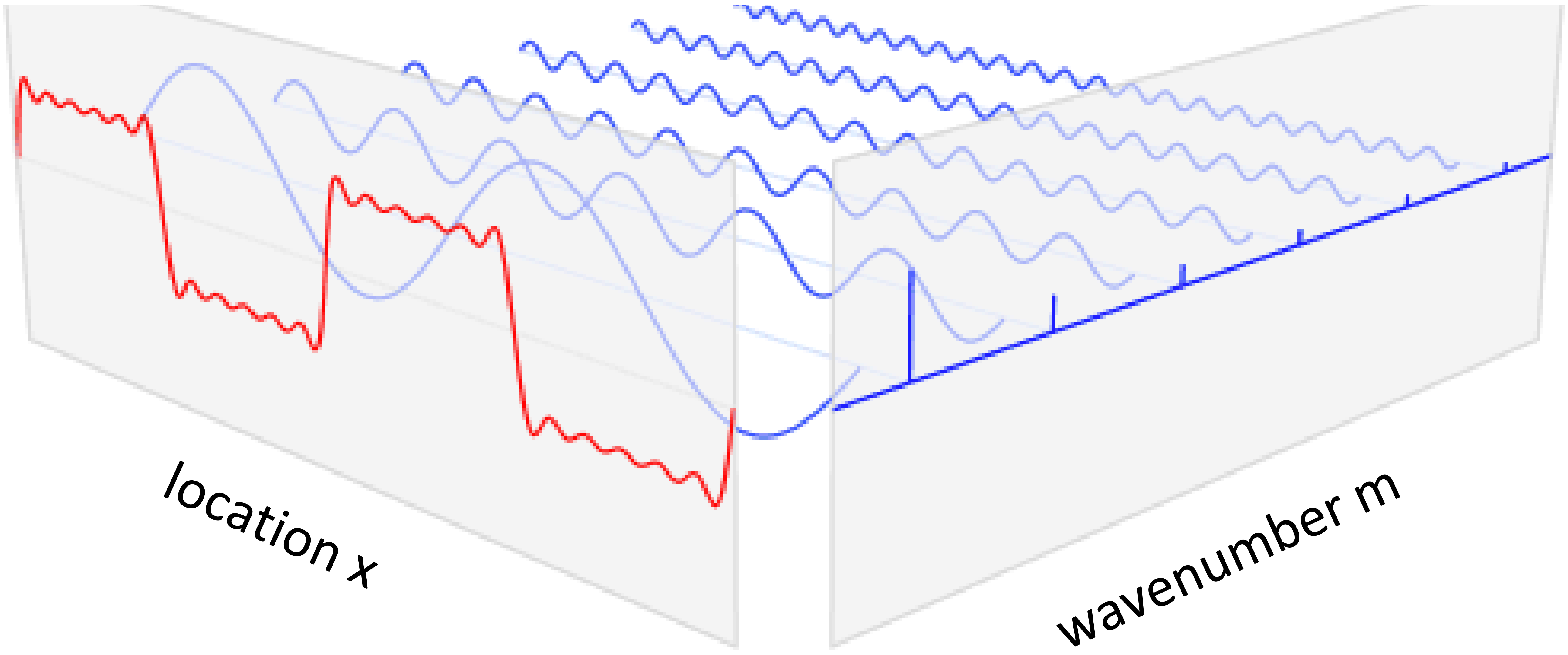
Fourier transform = Spectral transform in 1D




**grid point space** **Fourier space**

Funded by the European Union

# Fourier transform and its inverse




function of a real variable x

Fourier coefficients

$$f(x)=\sum_{m=-\infty}^{\infty} f_m e^{imx}, \quad x\in[0,2\pi]$$

Inverse Fourier transform

where,

$$f_m=\frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-imx}dx$$

Fourier transform

In practice these transforms are discrete in nature transforming grid-point functions (fields) to a finite number of discrete Fourier coefficients and vice versa.

The Fast Fourier Transform (FFT) is the standard way of performing this operation.

Funded by the European Union

# Spatial derivatives and Fourier representation

function of x

Fourier coefficients

$$f(x) = \sum_{m=-\infty}^{\infty} f_m e^{imx}, \quad x \in [0, 2\pi]$$

simple multiplication

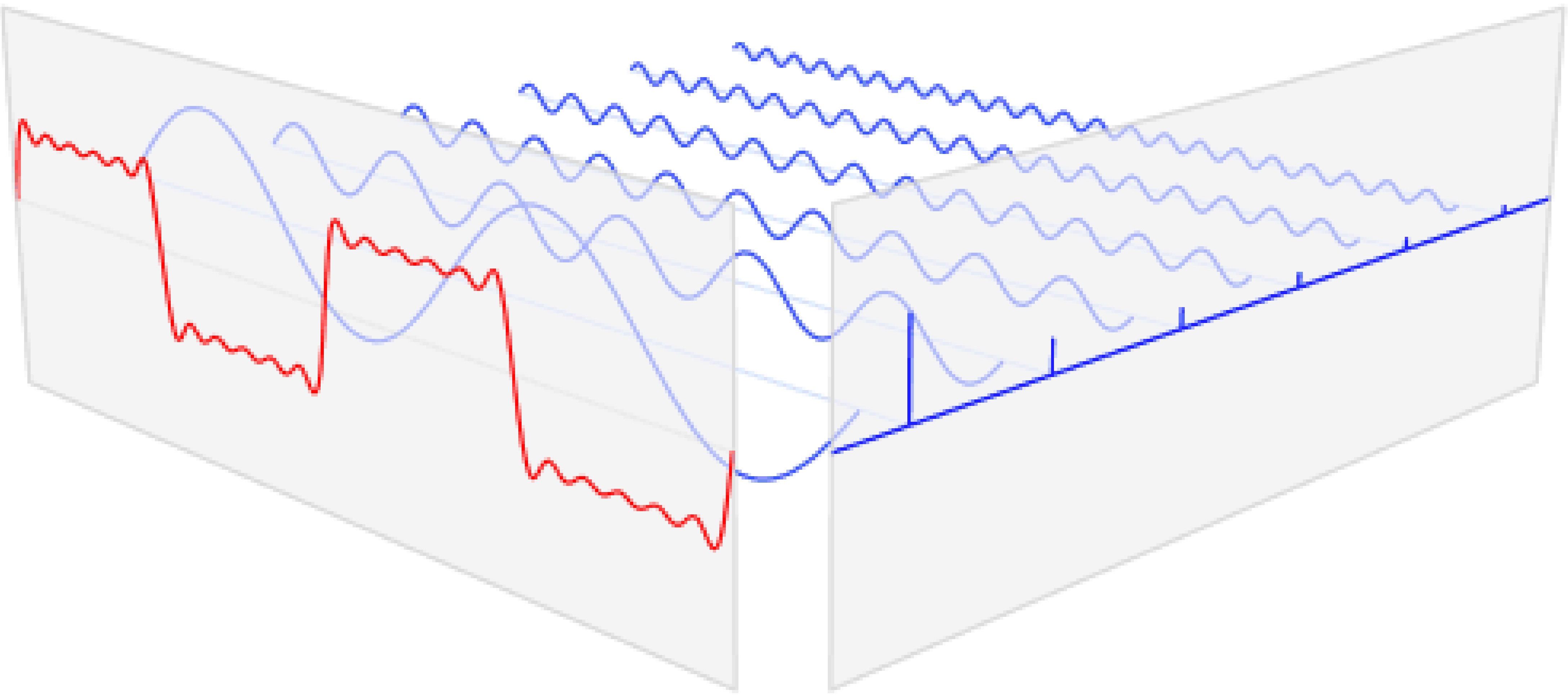
derivative

$$\frac{df(x)}{dx} = \sum_{m=-\infty}^{\infty} i m f_m e^{imx}$$

ESCAPE 2

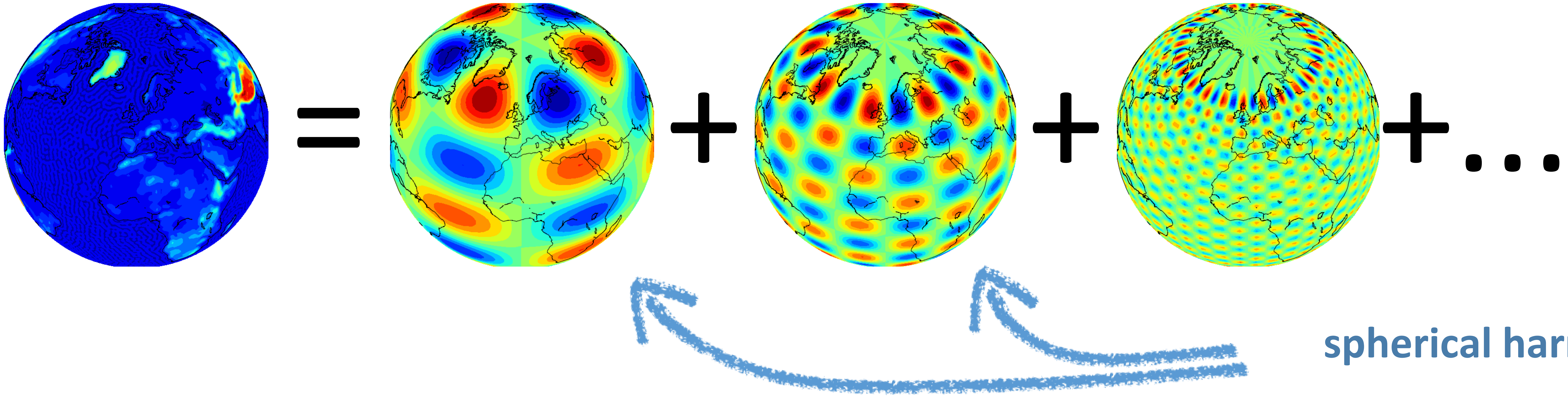
Funded by the European Union

# on the sphere: spectral transform



grid point space

spectral space



spherical harmonics

ESCAPE 2

Funded by the European Union

# Truncated spectral transform series

Spectral coefficient — longitude — latitude — Spherical harmonics

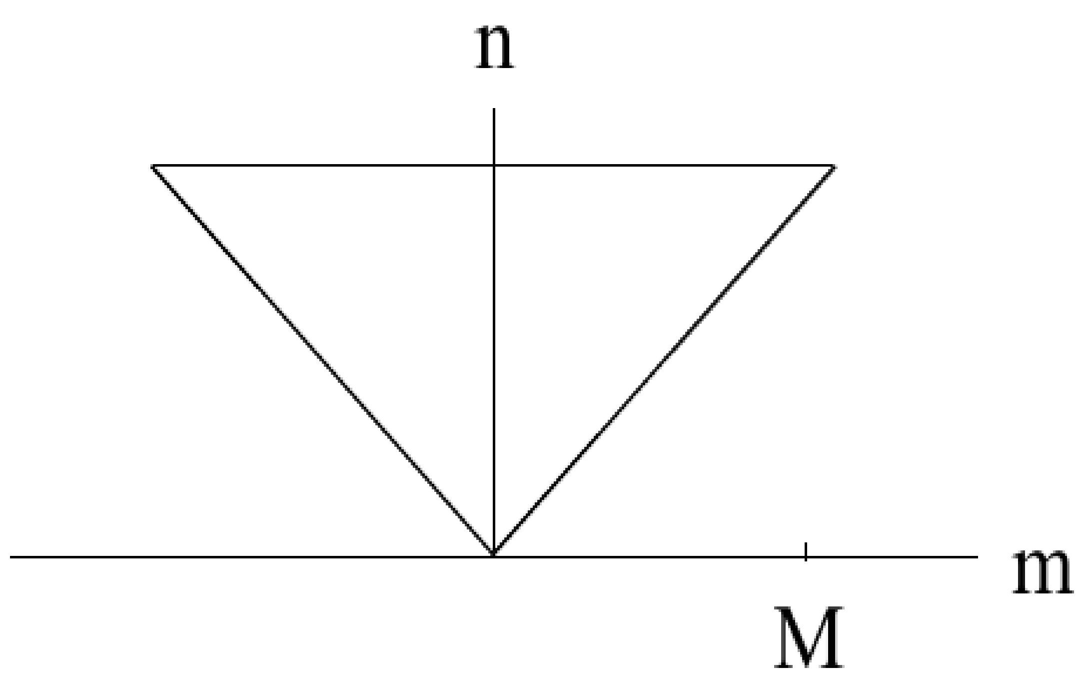
$$f(\lambda,\phi)=\sum_{m=-\infty}^{\infty}\sum_{n=|m|}^{\infty} f_n^m Y_n^m(\lambda,\phi),\quad Y_n^m(\lambda,\phi)=P_n^m(\sin\phi)\,e^{im\lambda}$$

m: zonal wavenumber
n: total wavenumber

Associated Legendre Polynomials (normalised)

Truncated series:

$$f(\lambda,\phi)=\sum_{m=-M}^{M}\sum_{n=|m|}^{M} f_n^m Y_n^m(\lambda,\phi),\quad Y_n^m(\lambda,\phi)=P_n^m(\mu)\,e^{im\lambda},\quad \mu=\sin\phi$$




Triangular truncation: (n,m) indices lie within a triangle.

Uniform resolution over entire surface of the sphere

# Spherical harmonics: where do they come from?

Funded by the European Union

Consider Laplace's equation on the sphere, assuming a solution (separation of variables, see book by Krishnamurti et al) of the form:

$$Y(\lambda, \mu) = L(\lambda)P(\mu), \quad \lambda: \text{ longitude}, \mu = \sin\phi$$

then, we obtain two ODEs:

$$\frac{d^2 L}{d\lambda^2} + m^2 L = 0, \qquad \frac{1-\mu^2}{P}\frac{d}{d\mu}\left(\left(1-\mu^2\right)\frac{dP}{d\mu}\right) + n(n+1)\left(1-\mu^2\right) = m^2$$

Solving for L, P the above we find that the solution is the *spherical harmonics function*:

$$Y_n^m(\lambda, \mu) = \underbrace{e^{im\lambda}}_{L(\lambda):\text{ Fourier mode}} \cdot \underbrace{P_n^m(\mu)}_{P(\mu):\text{ associated Legendre poly}}$$

ESCAPE 2

Funded by the European Union

# Properties of spherical harmonics

- Derivatives can be accurately, cheaply and trivially computed:

$$\frac{\partial Y_n^m}{\partial \lambda} = imY_n^m$$

$$\left(1-\mu^2\right)\frac{\partial Y_n^m}{\partial \mu} = -n\varepsilon_{n+1}^m Y_{n+1}^m + (n+1)\varepsilon_n^m Y_{n-1}^m, \quad \varepsilon_n^m = \sqrt{\frac{n^2-m^2}{4n^2-1}}$$

- Spherical harmonics are the Eigenfunctions of the horizontal Laplace operator and they are orthogonal (due to orthogonality of Legendre polynomials)

$$\nabla^2 Y_n^m = \frac{-n(n+1)}{a^2} Y_n^m, \quad a: \text{ Earth radius}$$

- Thus, elliptic equations are easy and cheap to solve ⇨ important for semi-implicit time-stepping

- Spectral transform methods do not suffer from pole singularities and have uniform spatial resolution over entire sphere with triangular truncation for m, n (used in these notes)

# Continuous transforms in space-time

$$f(\lambda,\mu,z,t) = \sum_{m=-\infty}^{\infty} \sum_{n=|m|}^{\infty} f_n^m(z,t) Y_n^m(\lambda,\mu)$$

$$f_m(\mu,z,t) = \frac{1}{2\pi} \int_0^{2\pi} f(\lambda,\mu,z,t) e^{-im\lambda} d\lambda$$

$$f_n^m(z,t) = \frac{1}{2} \int_{-1}^{1} f_m(\mu,z,t) P_n^m(\mu) d\mu$$

**Continuous spectral transform for a 4-dimensional equation model (space-time)**

**Continuous Fourier transform in longitude**

**Continuous Legendre transform in latitude**

Funded by the European Union

# Discrete transforms in space-time

ESCAPE 2
Funded by the European Union

$$f_m(\mu_k, z, t) = \frac{1}{L}\sum_{j=1}^{L} f(\lambda_j, \mu_k, z, t) e^{-im\lambda_j}$$

Fourier transform at latitude $\phi_k$: computed using a FFT

$$f_n^m(z,t) = \frac{1}{2}\sum_{k=1}^{K} w_k \; f_m(\mu_k, z, t) P_n^m(\mu_k)$$

Legendre transform: a Gaussian quadrature exact for all polynomials of degree 2K-1

$$\left.\begin{array}{l} w_k : \text{Gaussian weights} \\ \mu_k : \text{Gaussian quadrature points} \end{array}\right\}, \quad k = 1, 2, \ldots, K$$

$$f_m(\mu_k, z, t) = \sum_{n=|m|}^{M} f_n^m(z,t) P_n^m(\mu_k)$$

Inverse Legendre transform

$$f(\lambda_j, \mu_k, z, t) = \sum_{m=-M}^{M} f_m(\mu_k, z, t) e^{im\lambda_j}$$

Inverse Fourier transform

**For accurate LTs a Gaussian grid must be used: grid-point latitudes coincide with the latitude of Gaussian quadrature points (roots of Legendre polynomials)**

# aliasing

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

# aliasing

wave generated in spectral space

**Issue**: multiplication of two waveform variables produces a new variable with shorter wavelength than the one the grid can handle

# aliasing


**Issue**: multiplication of two waveform variables produces a new variable with shorter wavelength than the one the grid can handle

Funded by the European Union

# aliasing

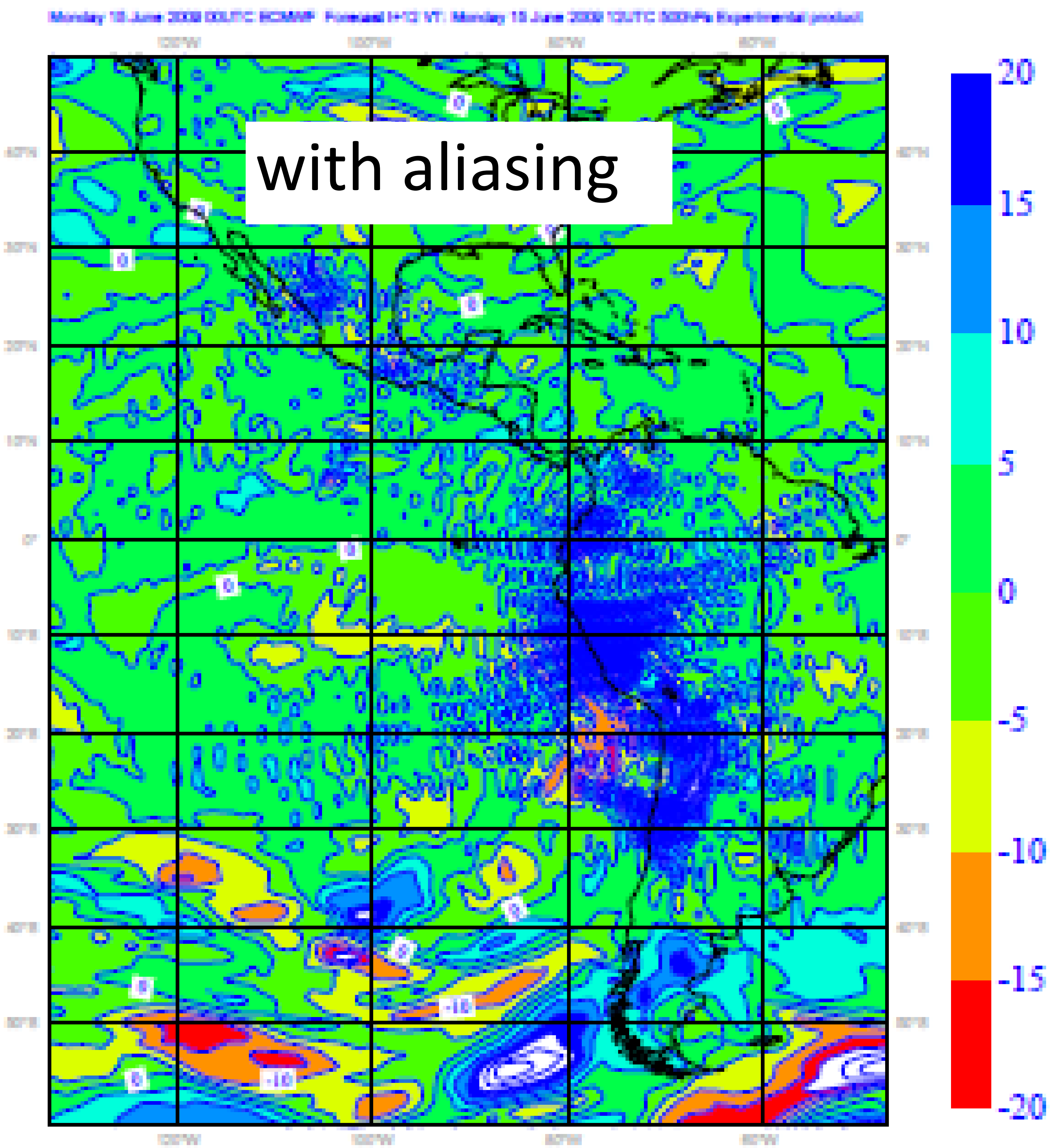

**Issue**: multiplication of two waveform variables produces a new variable with shorter wavelength than the one the grid can handle

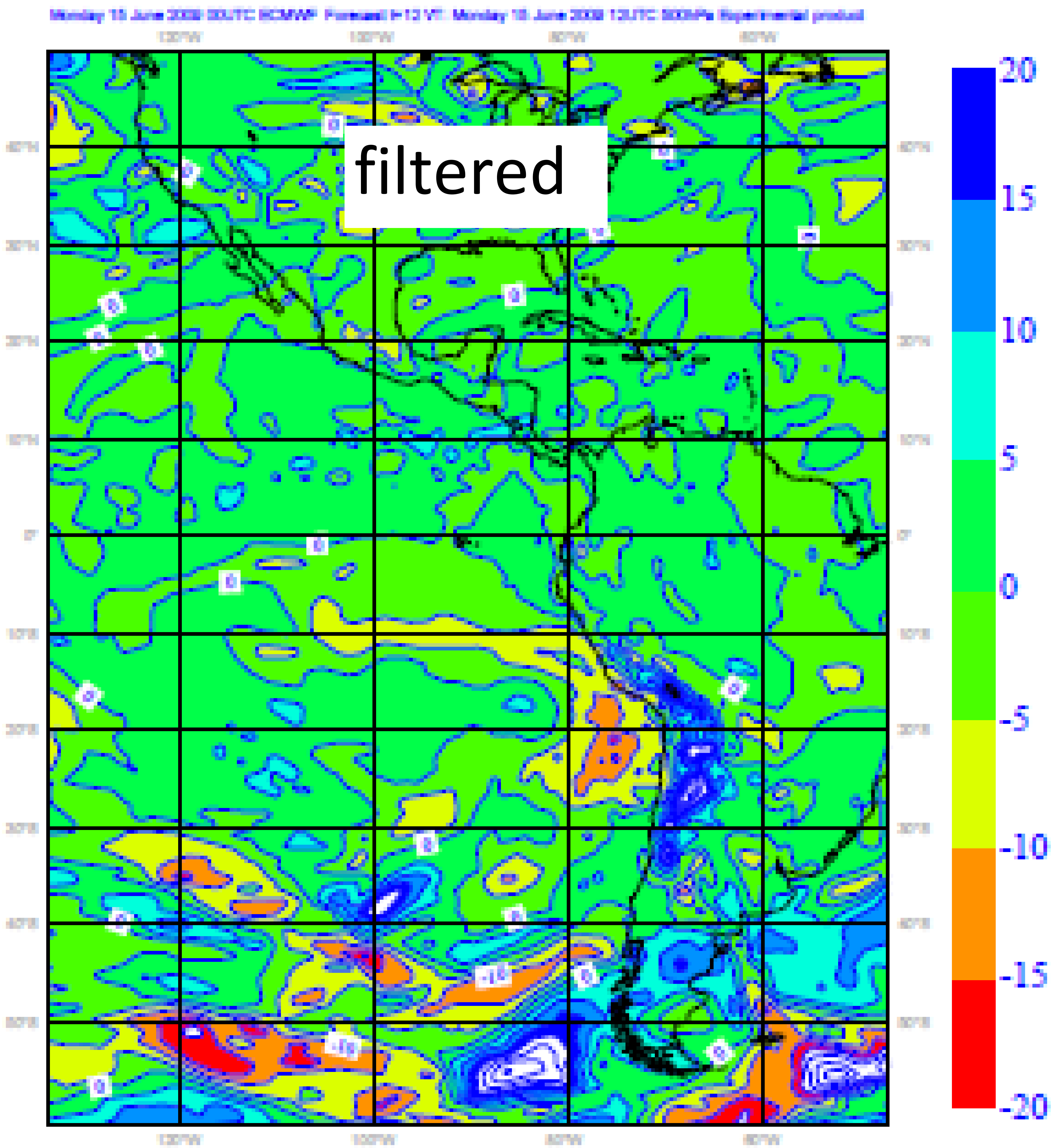
# aliasing


**Issue**: multiplication of two waveform variables produces a new variable with shorter wavelength than the one the grid can handle

# aliasing example

## 500hPa adiabatic zonal wind tendencies (T159)



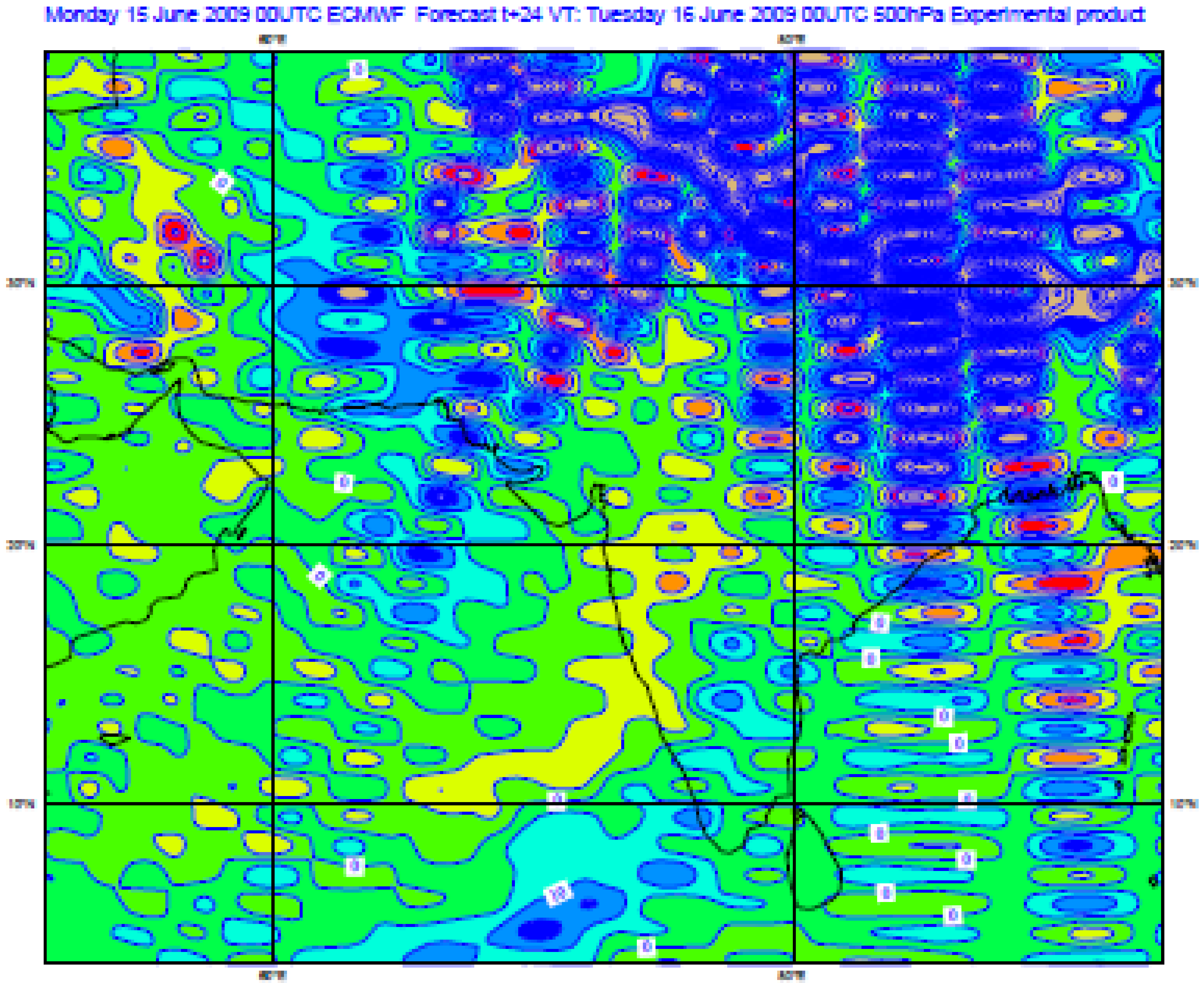

Funded by the European Union

# aliasing example

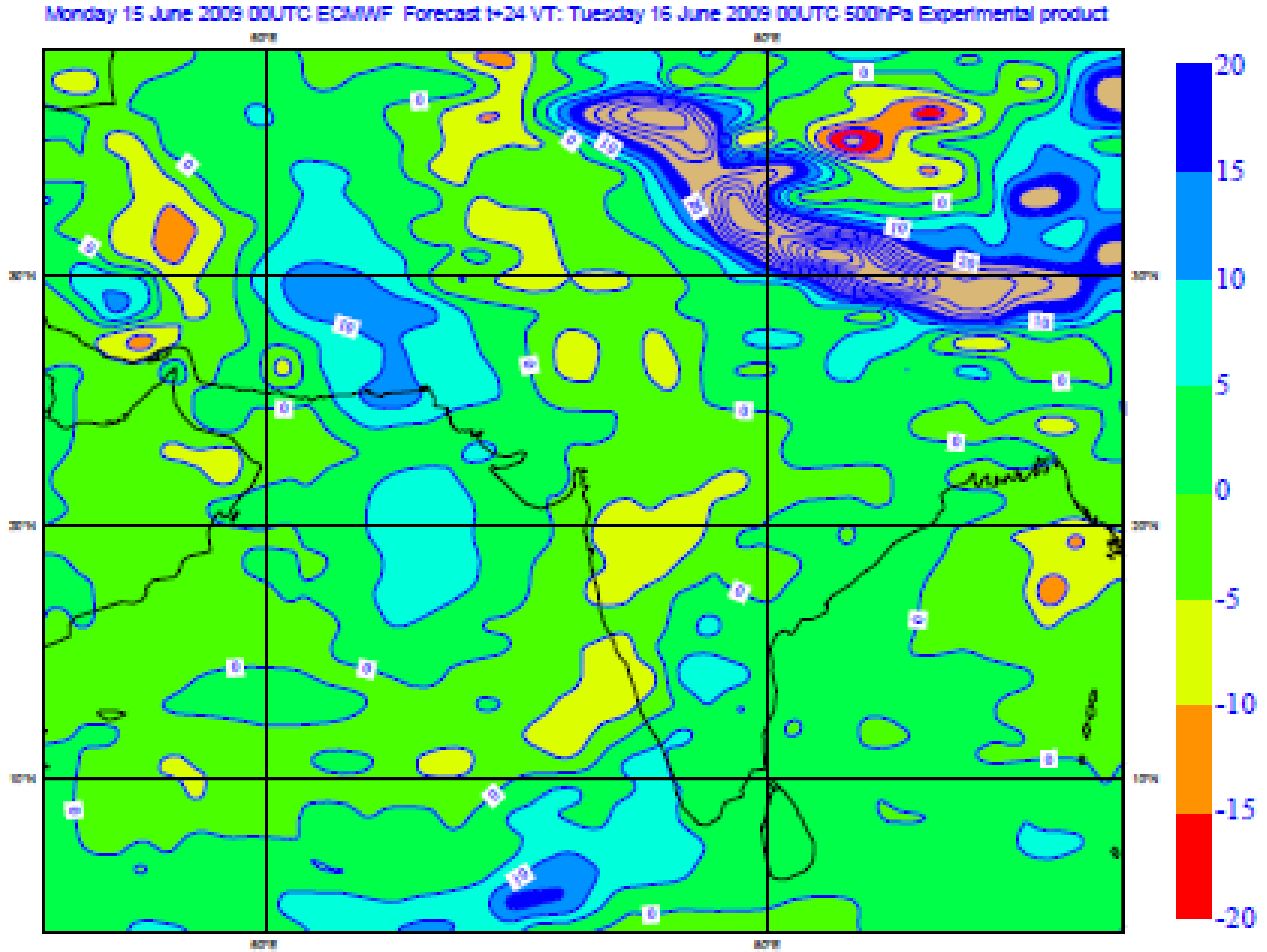
## 500hPa adiabatic meridional wind tendencies (T159)

Funded by the European Union

### with aliasing




### filtered



Funded by the European Union

# alternatives to using a filter

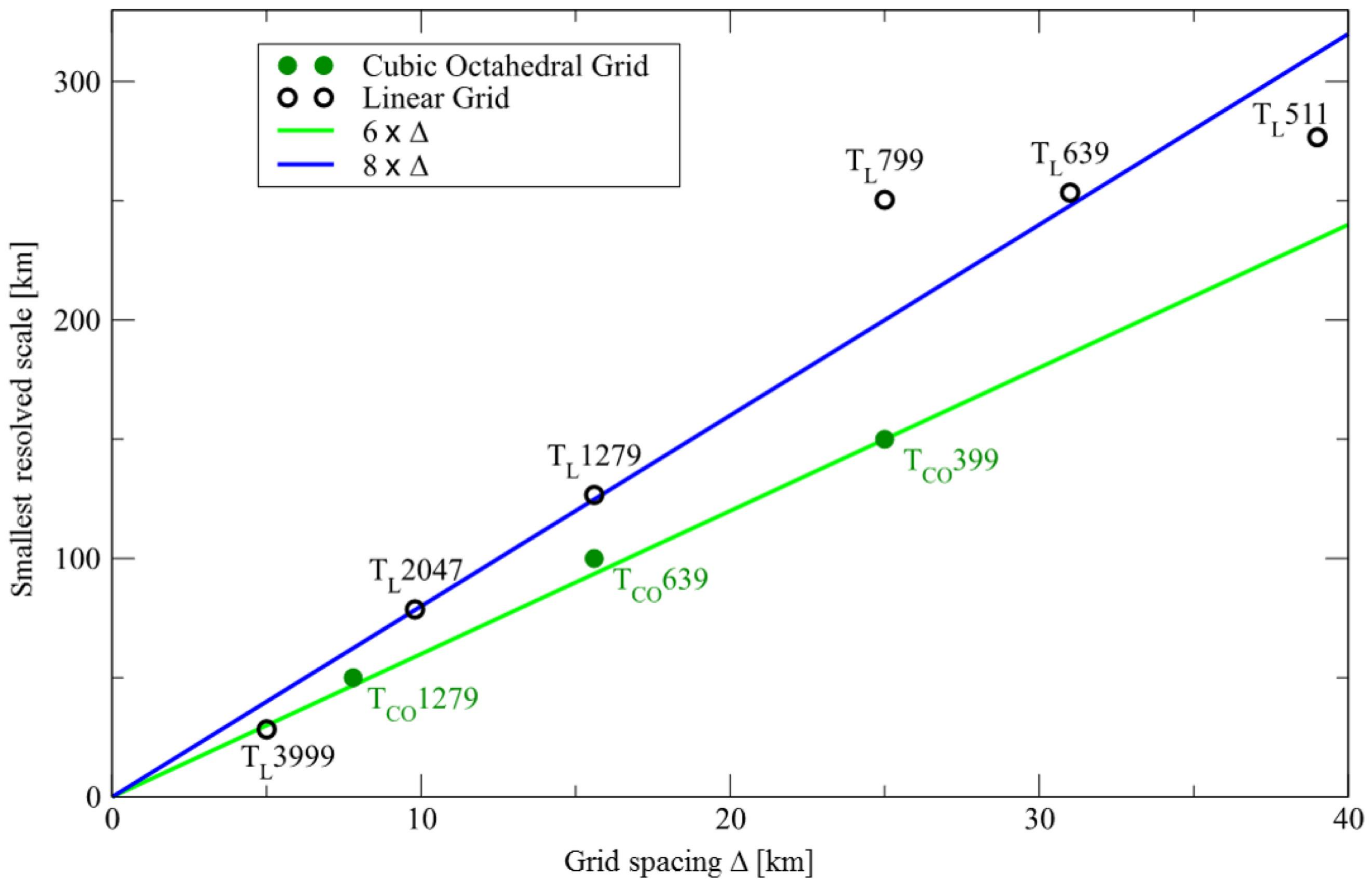
**Idea**: use more grid points than spectral coefficients

| Orszag, 1971: | Spatial filter range<br>Δ : grid-length<br>*(Wedi, 2014)* | Equation terms accurately represented without aliasing |
|---|---|---|
| 2N+1 gridpoints to N waves : linear grid | ~ 1-2 Δ | Linear |
| 3N+1 gridpoints to N waves : quadratic grid | ~ 2-3 Δ | Quadratic |
| 4N+1 gridpoints to N waves : cubic grid | ~ 3-4 Δ | Cubic |

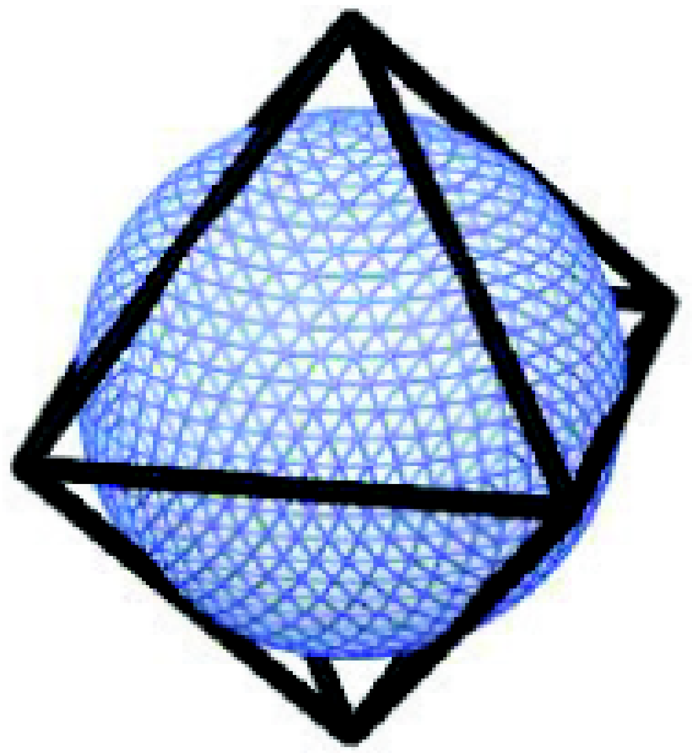
- Cubic grid filters 3-4 grid-length oscillations therefore no need to apply an extra de-aliasing filter as in the linear grid
- The smallest wavelength 2πα/N is resolved by 2,3,4 points by the linear, quadratic, cubic grids

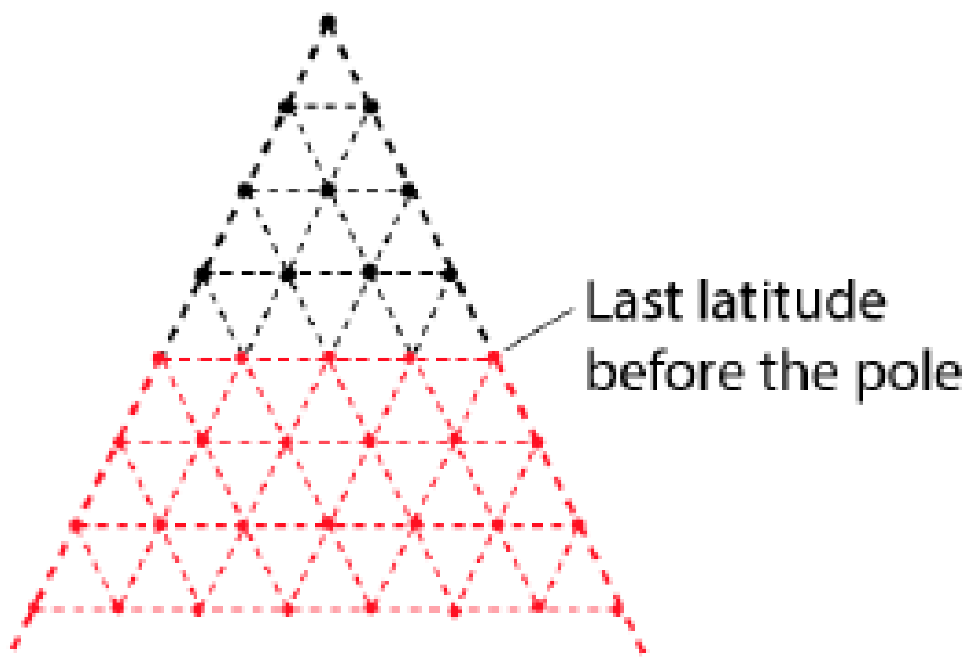
# effective resolution

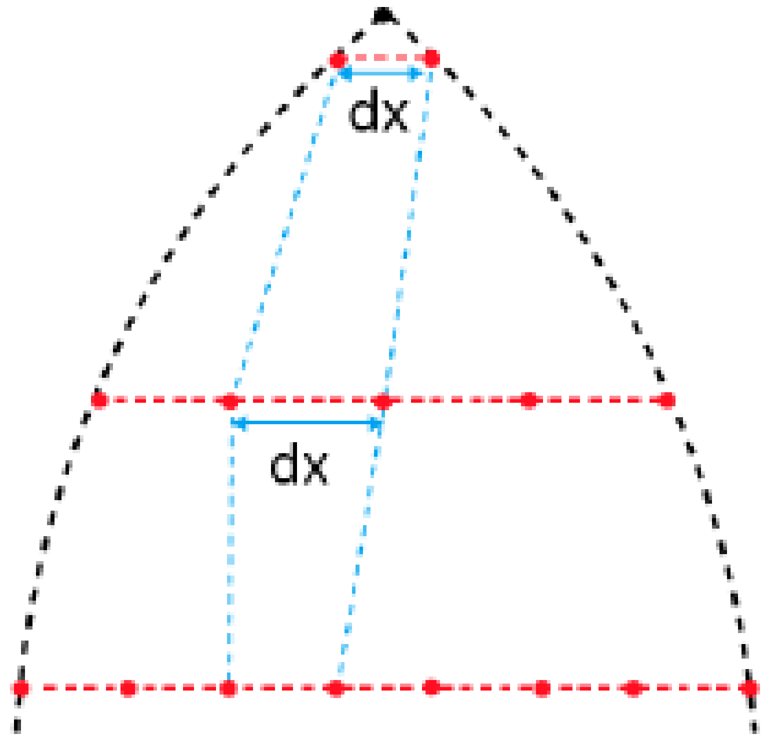
of linear and cubic grids (Abdalla et al. 2013)



# Cubic octahedral (Gaussian) grid of IFS

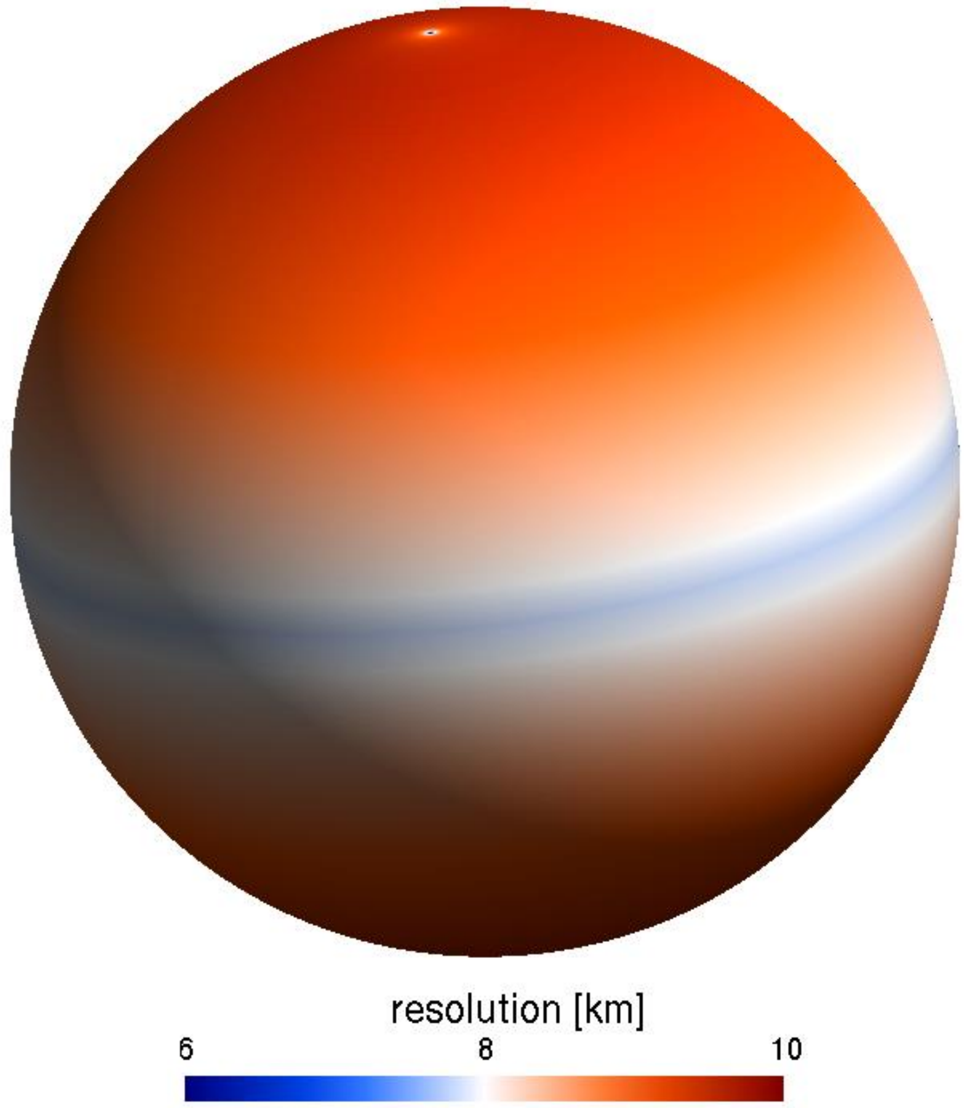





- No aliasing in nonlinear products
- Improved accuracy and mass conservation compared with linear grid
- Efficiency and scalability for large size problems: high grid-point resolution for a given spectral truncation i.e. expensive transforms become a smaller fraction of total computations

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



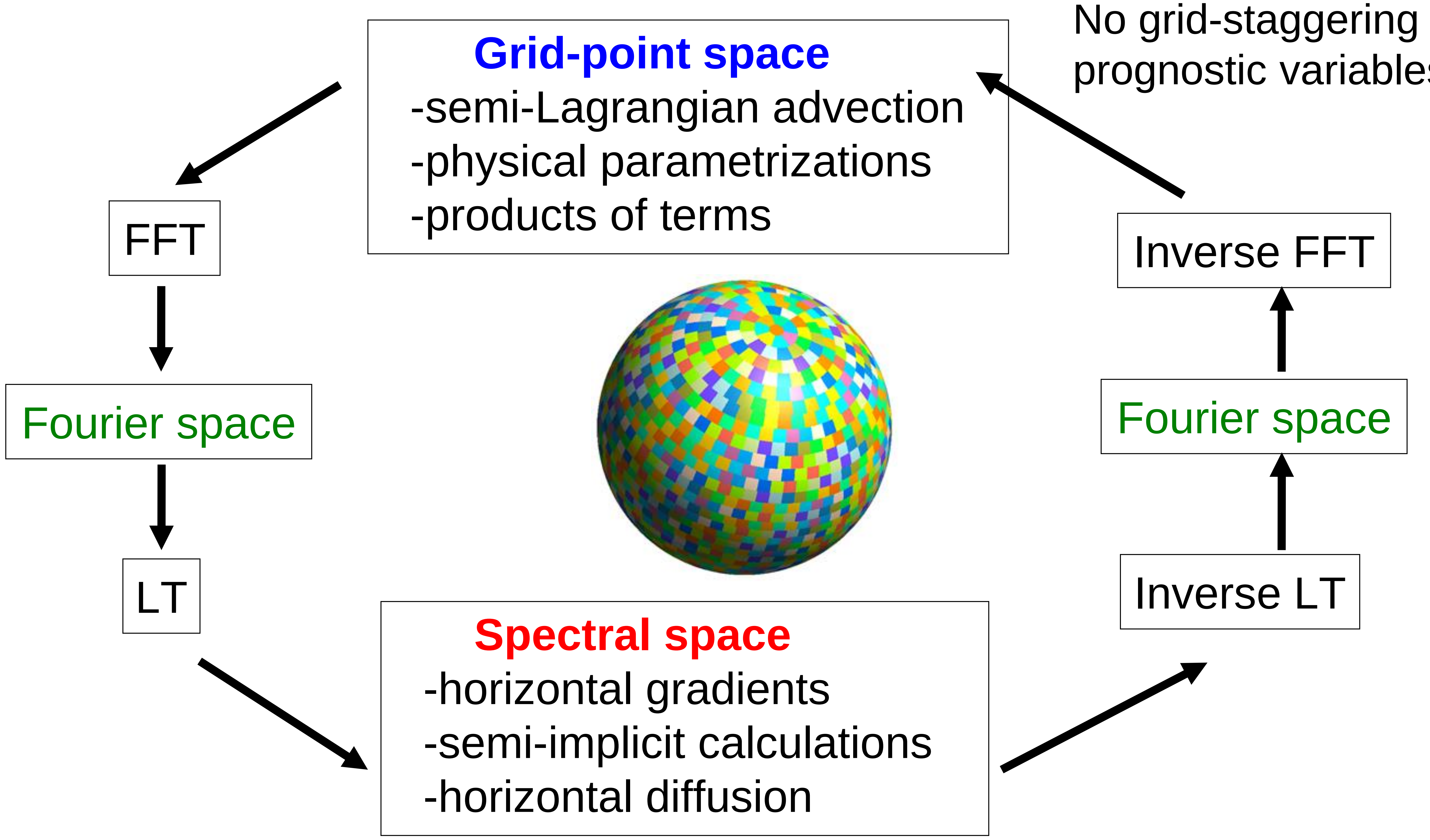

Variation of grid-point resolution with latitude

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

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

# time step in IFS

No grid-staggering of prognostic variables




FFT: Fast Fourier Transform, LT: Legendre Transform

Funded by the European Union

# Inverse transforms: use of symmetry properties of Legendre polynomials in computation

Funded by the European Union

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

spectral space

**even n** | **odd n**

for each m:

Normalised associated Legendre polynomial

**parallelisation over m, n indices**

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

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

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

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

**lots of MPI communication**

inverse Legendre transform

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

inverse Fourier transform

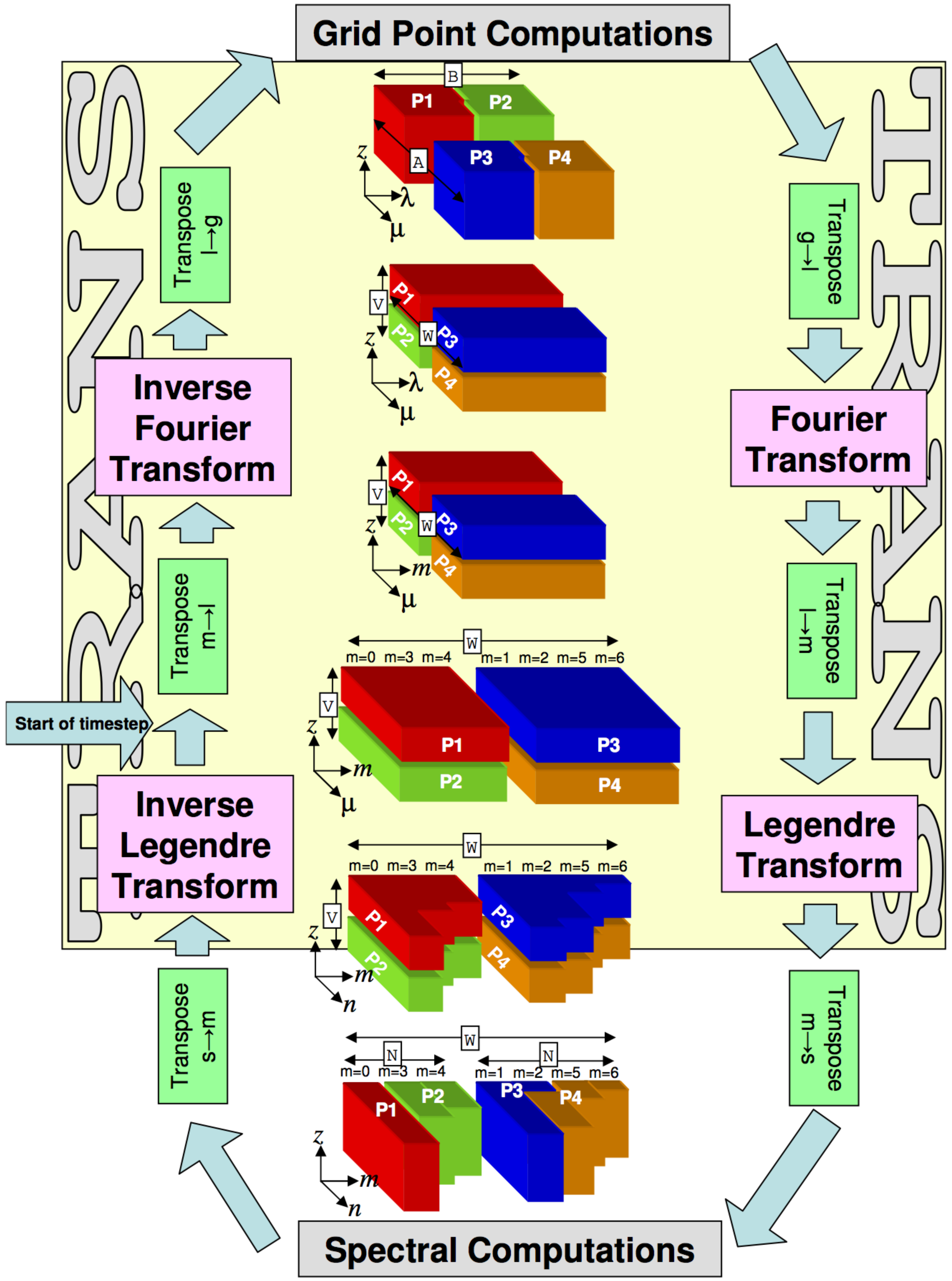
grid point data: $\mathbf{G}(f, \lambda, \phi)$

grid point space

# Domain decomposition in spectral transform semi-implicit solver

Direct Legendre transform:

- multiply data with Gaussian quadrature weights
- Same as an inverse transform but in reverse order
- The data transpositions needed by spectral transforms imply heavy communication load





**References**:
*Foster et al, Parallel Algorithms for the Spectral Transform Method, SIAM J Sci Com, 1997*
*Baros et al, The IFS model: A parallel production weather code, Parallel computing 21*

# Matrix-matrix multiply in a LT

Funded by the European Union

Legendre and inverse Legendre transforms are expressed as a matrix-matrix multiply for each wavenumber m (Wedi et al, MWR 2013)

$$f_n^m(z,t) = \frac{1}{2}\sum_{k=1}^{K} w_k \; f_m(\mu_k, z, t) P_n^m(\mu_k)$$

$$f_m(\mu_k, z, t) = \sum_{n=|m|}^{M} f_n^m(z,t) P_n^m(\mu_k)$$

**Left: LT**
**Right: Inverse LT**

$$\left[\; f^m(\mu_k)_l \;\right] = \left[\; P_n^m(\mu_k)_l \;\right]\left[\; f_{n,l}^m \;\right]$$

A matrix-matrix multiply for each zonal wavenumber m across all Gaussian Latitudes and vertical levels

Gaussian latitude

Vertical level

Fast LT: apply "butterfly" compression (Tygart 2010)

Spectral coefficient at vertical level l

Funded by the European Union

# Interpolative decomposition ("butterfly compression")

The ***left hand-side matrix*** in a LT transform (matrix-matrix multiply) ***remains the same*** *regardless the timestep*. It can be compressed and approximated in a form that accelerates computation

$$A_{m\times n} \approx R_{m\times k} S_{k\times n}$$

- A: rank deficient matrix
- R: contains blocks with full rank
- S: "interpolation matrix". A subset of its columns makes up the identity matrix -cannot be further compressed so must be saved
- The approximation is valid within a tolerance selected so that it doesn't change significant the results

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} R_1 \cdot S_1 \\ R_2 \cdot S_2 \end{bmatrix}$$

Split matrix in two halves applying decomposition

$$\begin{bmatrix} R_1 \cdot S_1 & 0 \\ 0 & R_2 \cdot S_2 \end{bmatrix} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix}$$

Rewrite matrix as the product of two block matrices emptying half of each column

The above algorithm can be repeated until the residual block matrix contains a single diagonal of full-rank blocks

# Fast Legendre Transform

Funded by the European Union

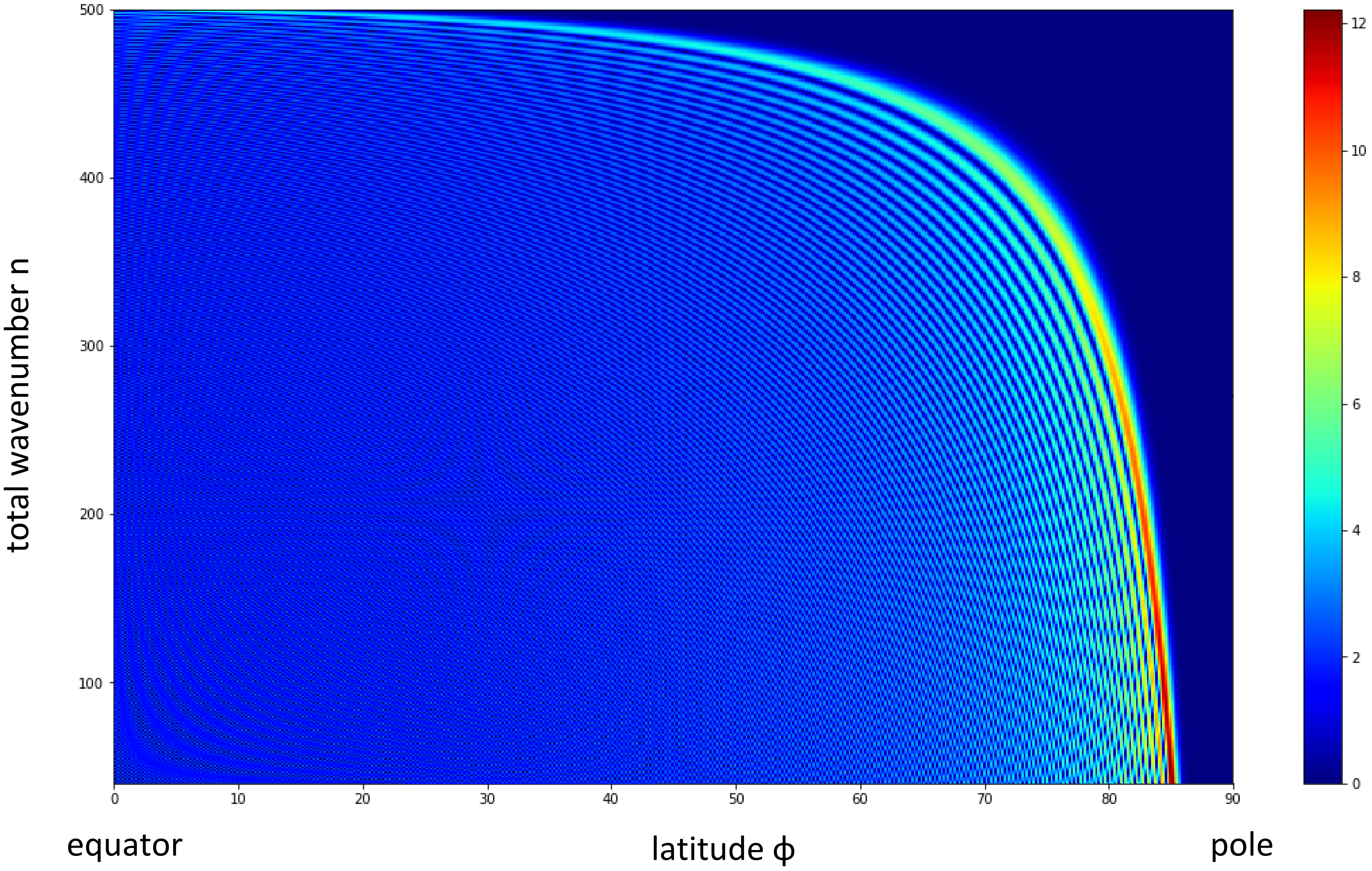
Matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition”



# Fast Legendre Transform

Funded by the European Union

Matrix of
Legendre polynomials

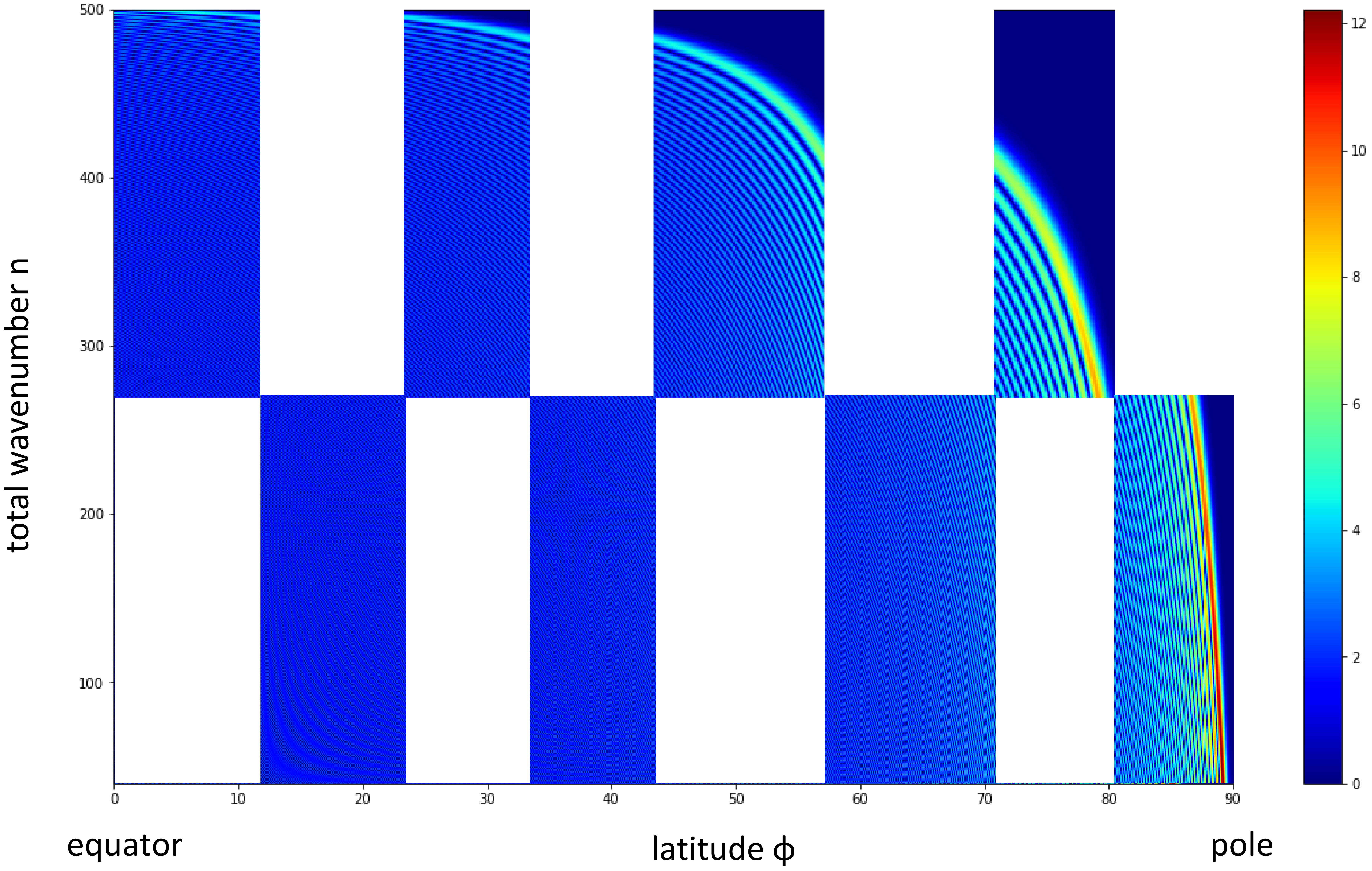
truncation N=500,
zonal wavenumber
m=40

**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition"

**step 3**: reorder columns



# Fast Legendre Transform

Funded by the European Union

Matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

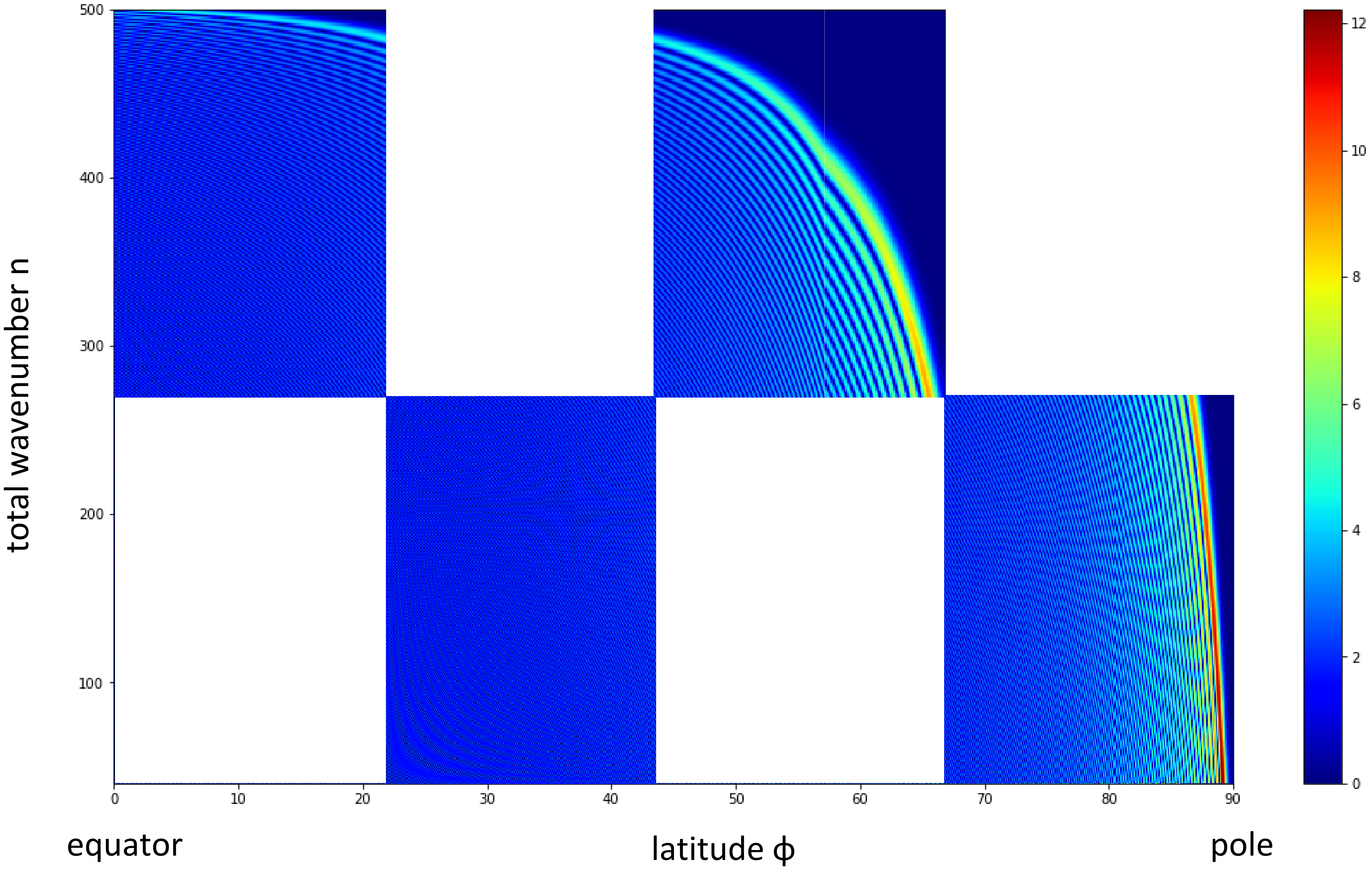
**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition”

**step 3**: reorder columns

**step 4**: apply to each block recursively



Funded by the European Union

# Fast Legendre Transform

Matrix of
Legendre polynomials

truncation N=500,
zonal wavenumber
m=40

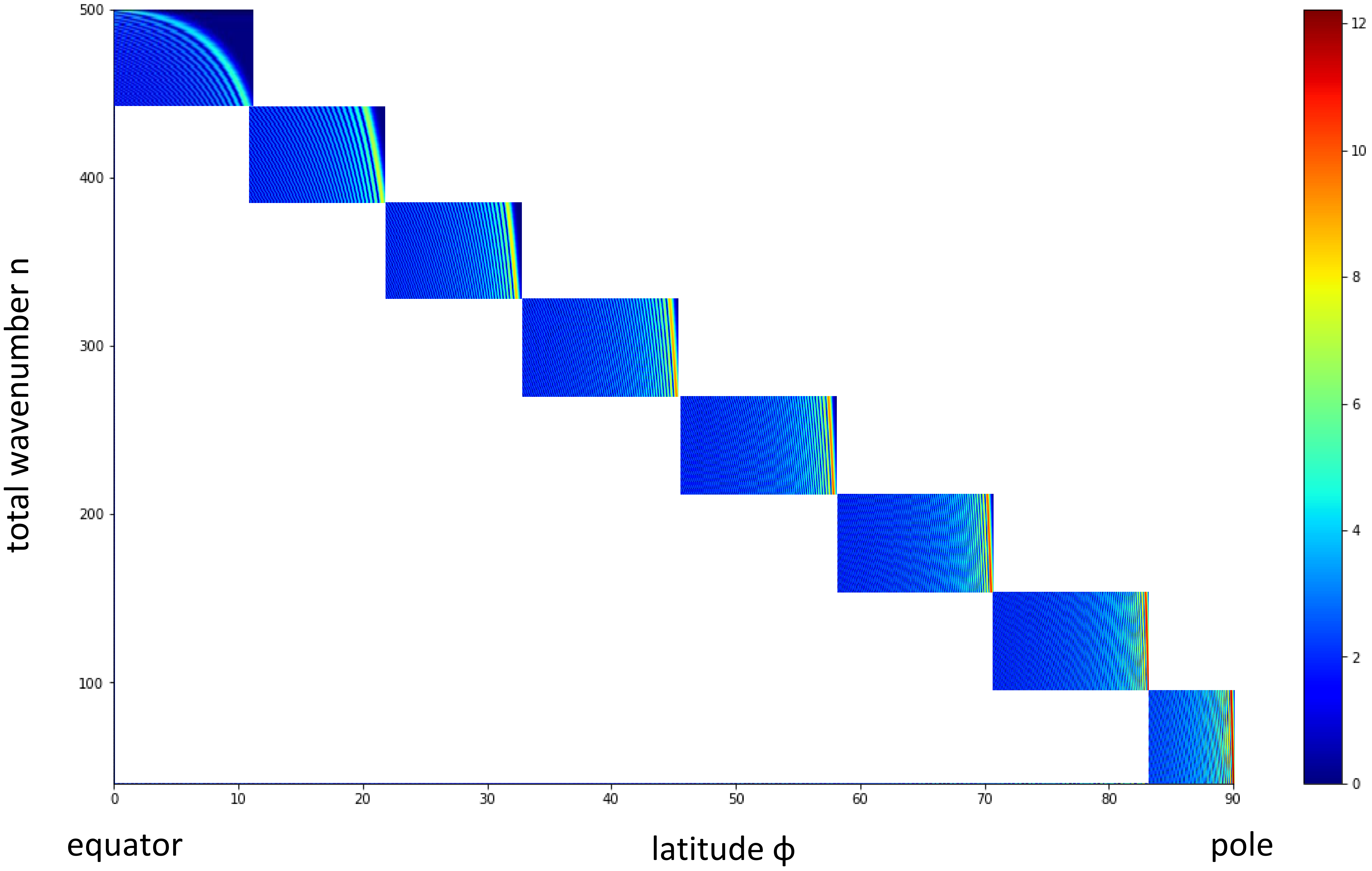
**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition"

**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

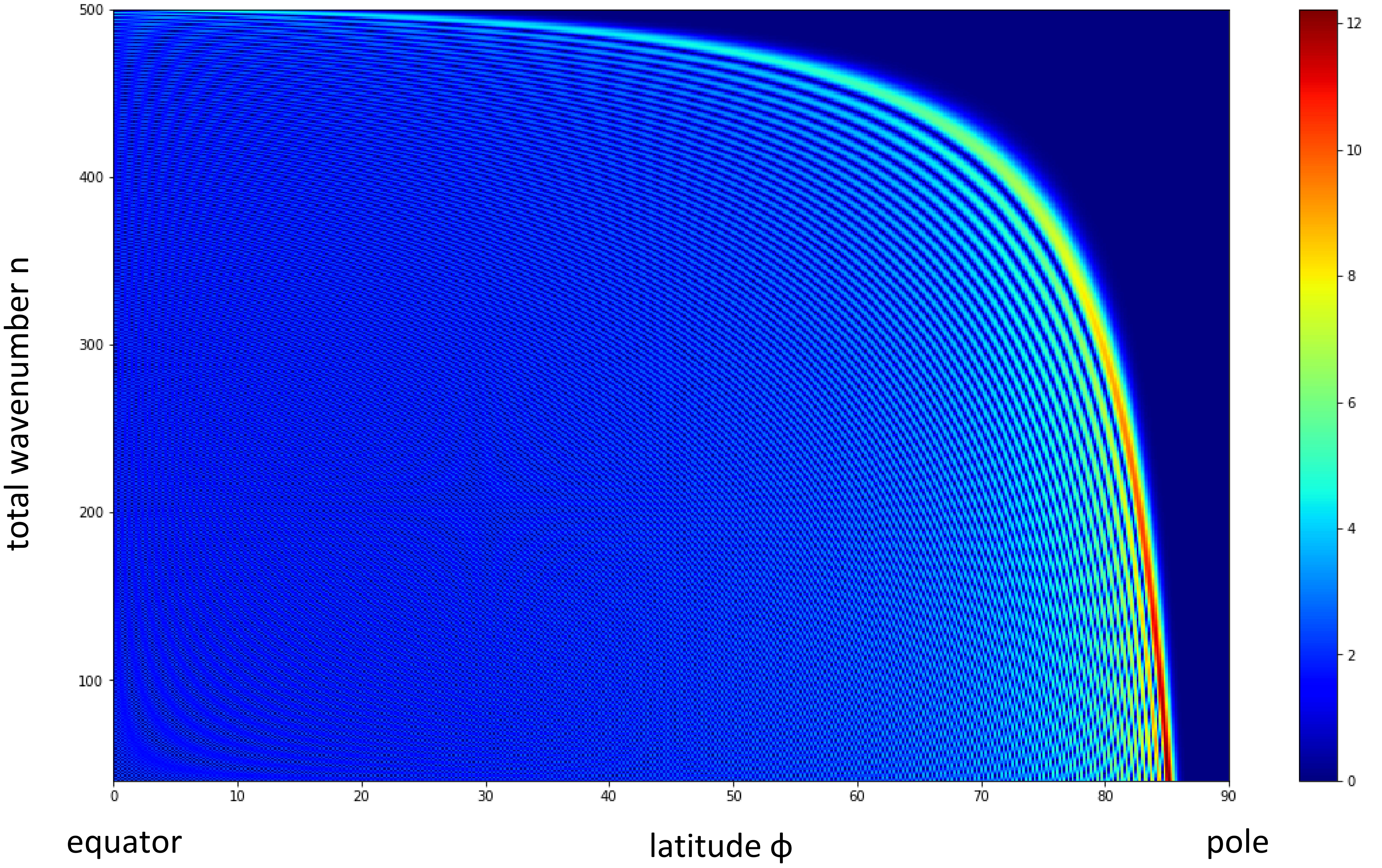
**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition"

**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

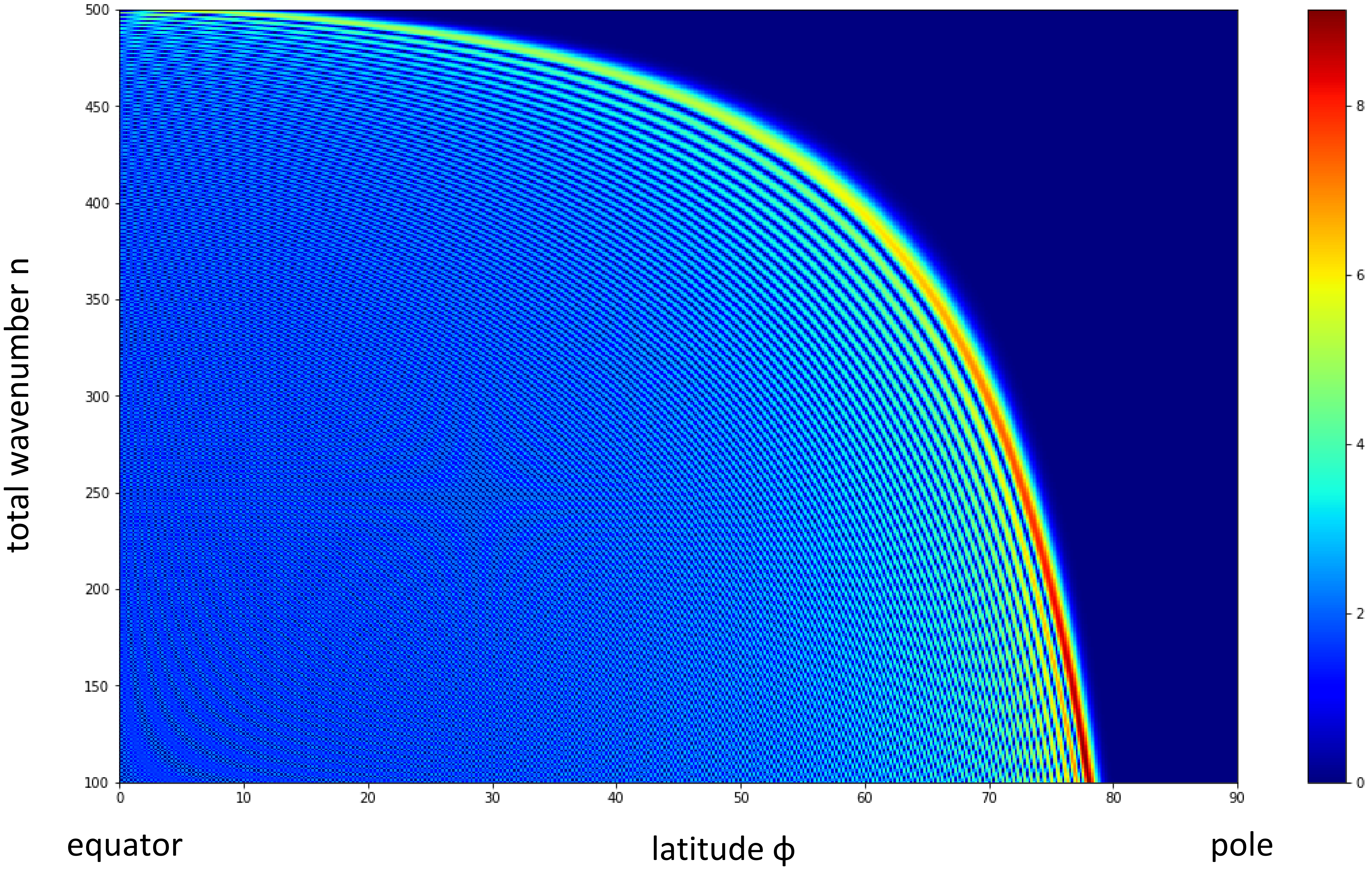
**FLT:**

**step 1**: split matrix into two halves

**step 2**: empty half of each column and apply "interpolative decomposition”

**step 3**: reorder columns

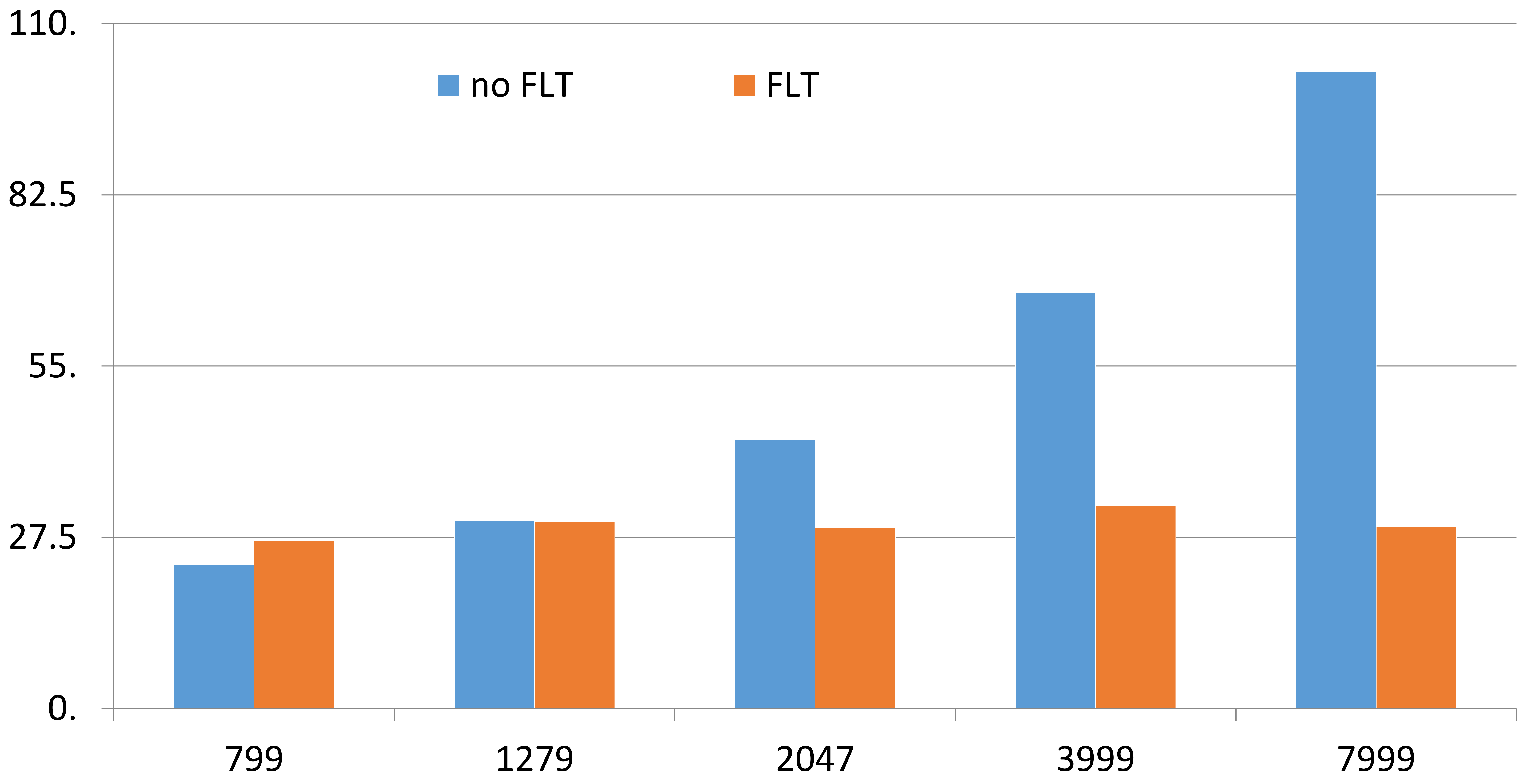
**step 4**: apply to each block recursively



# Fast Legendre Transform

## floating point operations

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

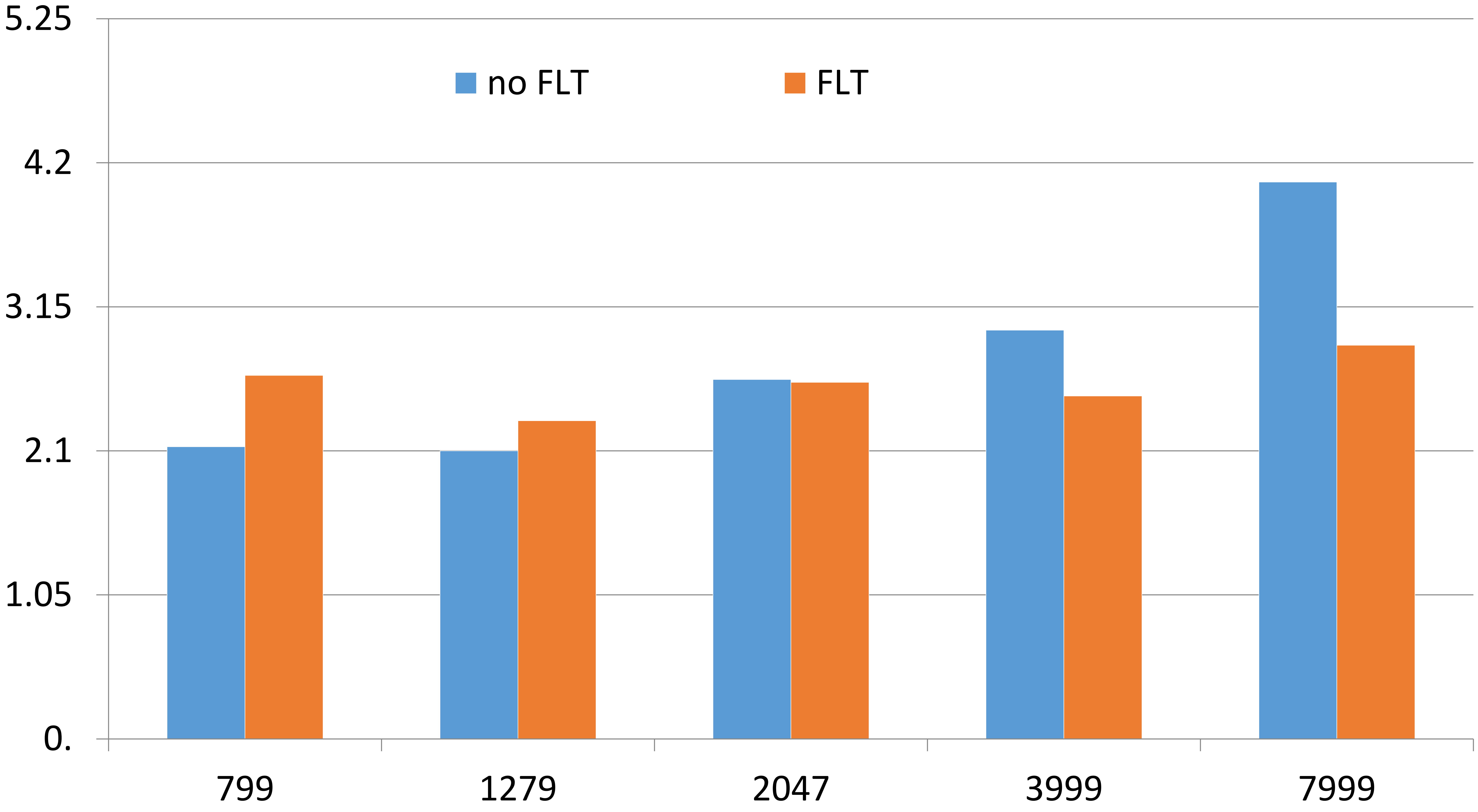



Funded by the European Union

# Fast Legendre Transform

wallclock time

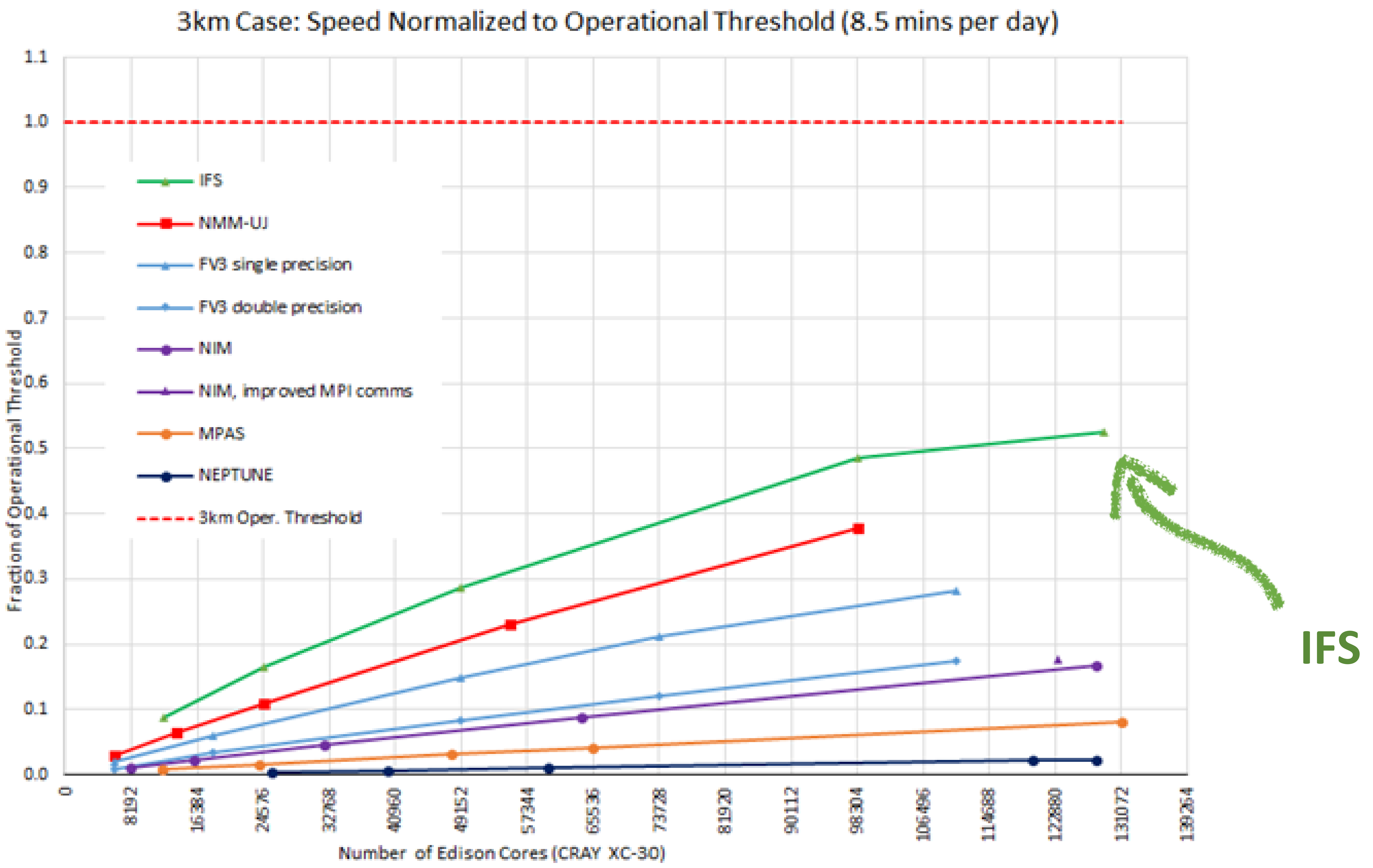
Average wall-clock time compute cost of $10^7$ spectral transforms scaled by $N^2 \log^3 N$




Funded by the European Union

# performance comparison

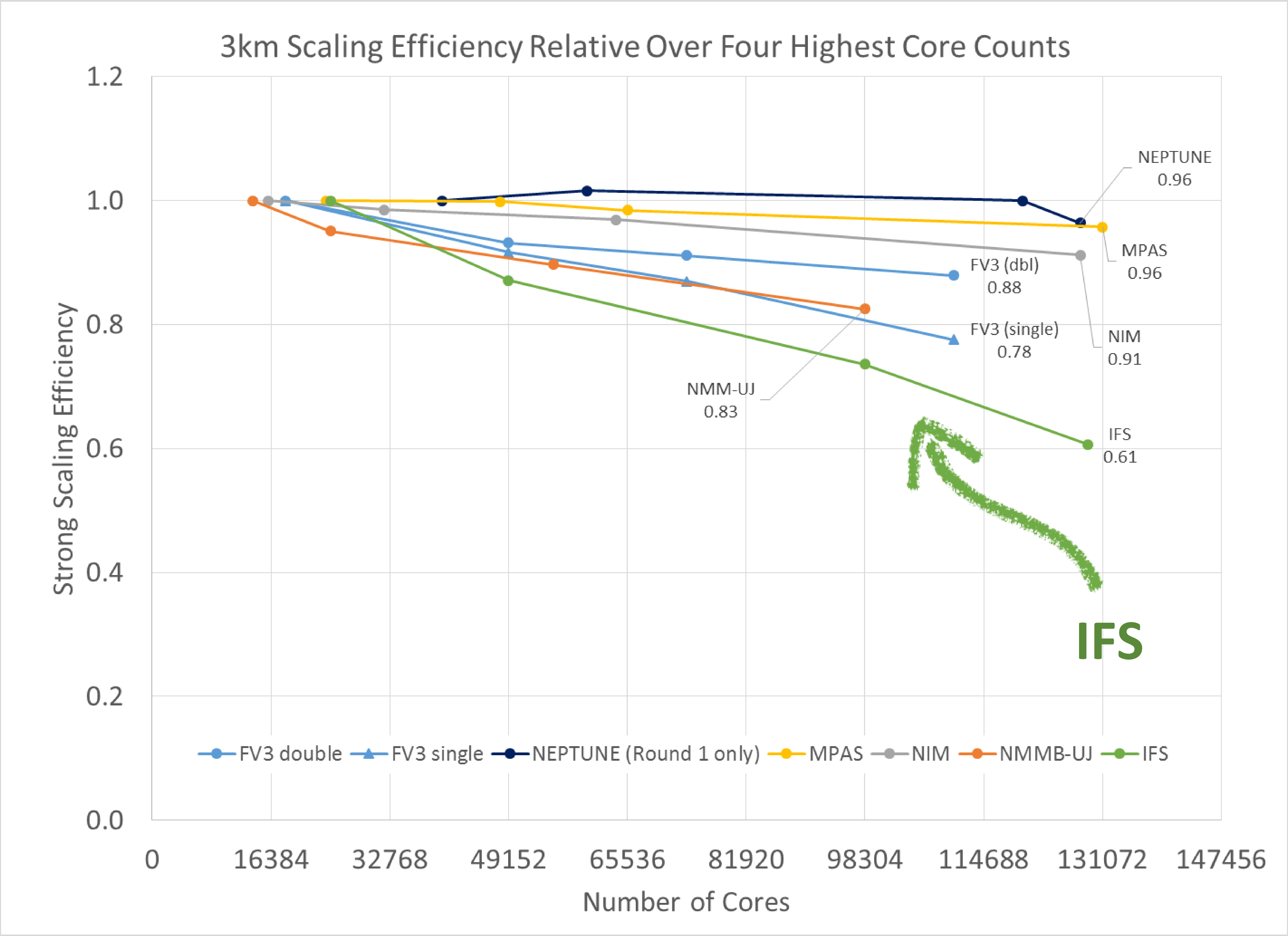
## of IFS with other models




*(Michalakes et al, NGGPS AVEC report, 2015)*

# scalability comparison

## of IFS with other models



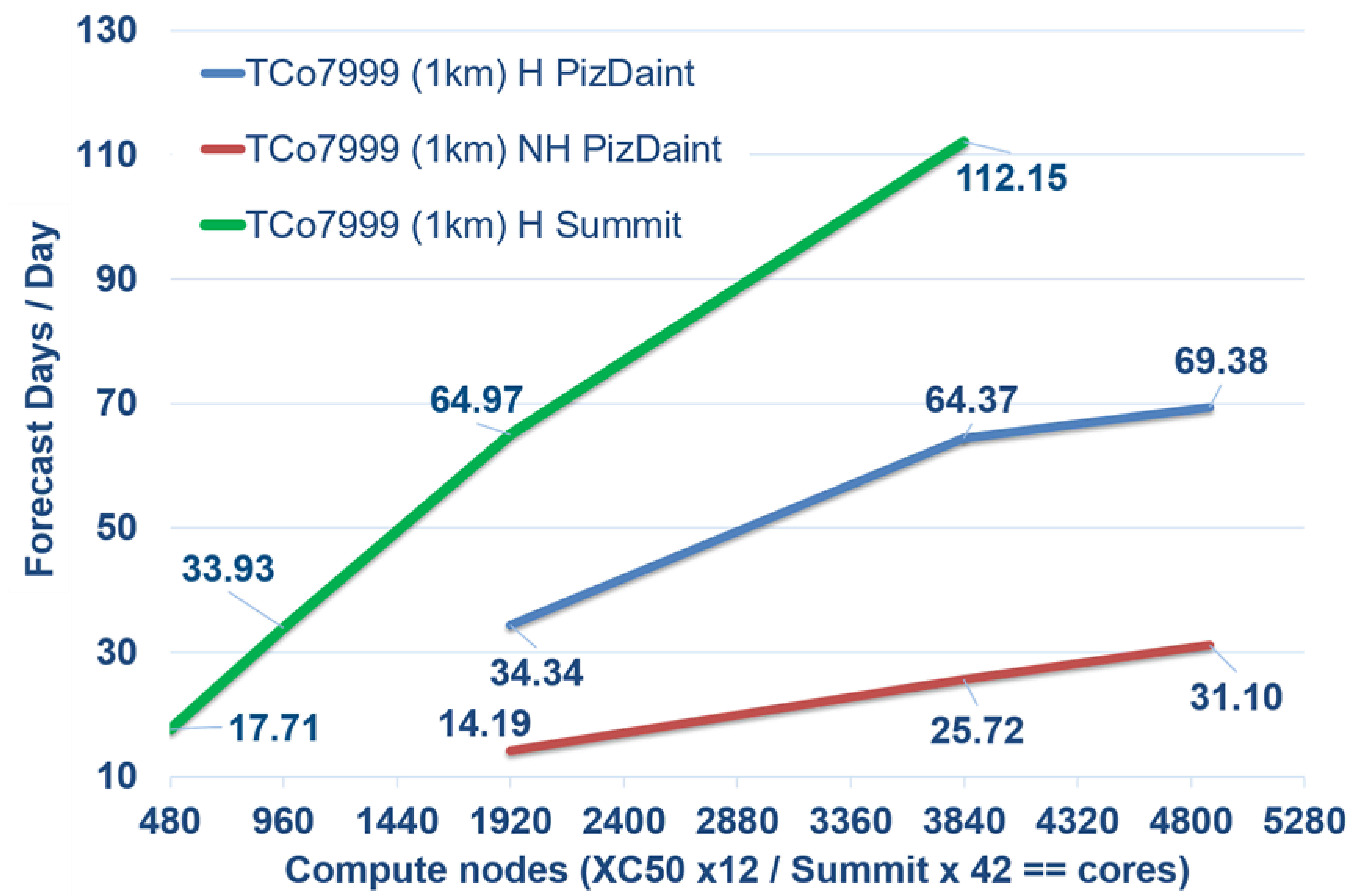

*(Michalakes et al, NGGPS AVEC report, 2015)*

Funded by the European Union

ESCAPE 2

# IFS scaling on Summit and PizDaint (CPU only)

Funded by the European Union



# optimisations by NVIDIA in ESCAPE

Funded by the European Union

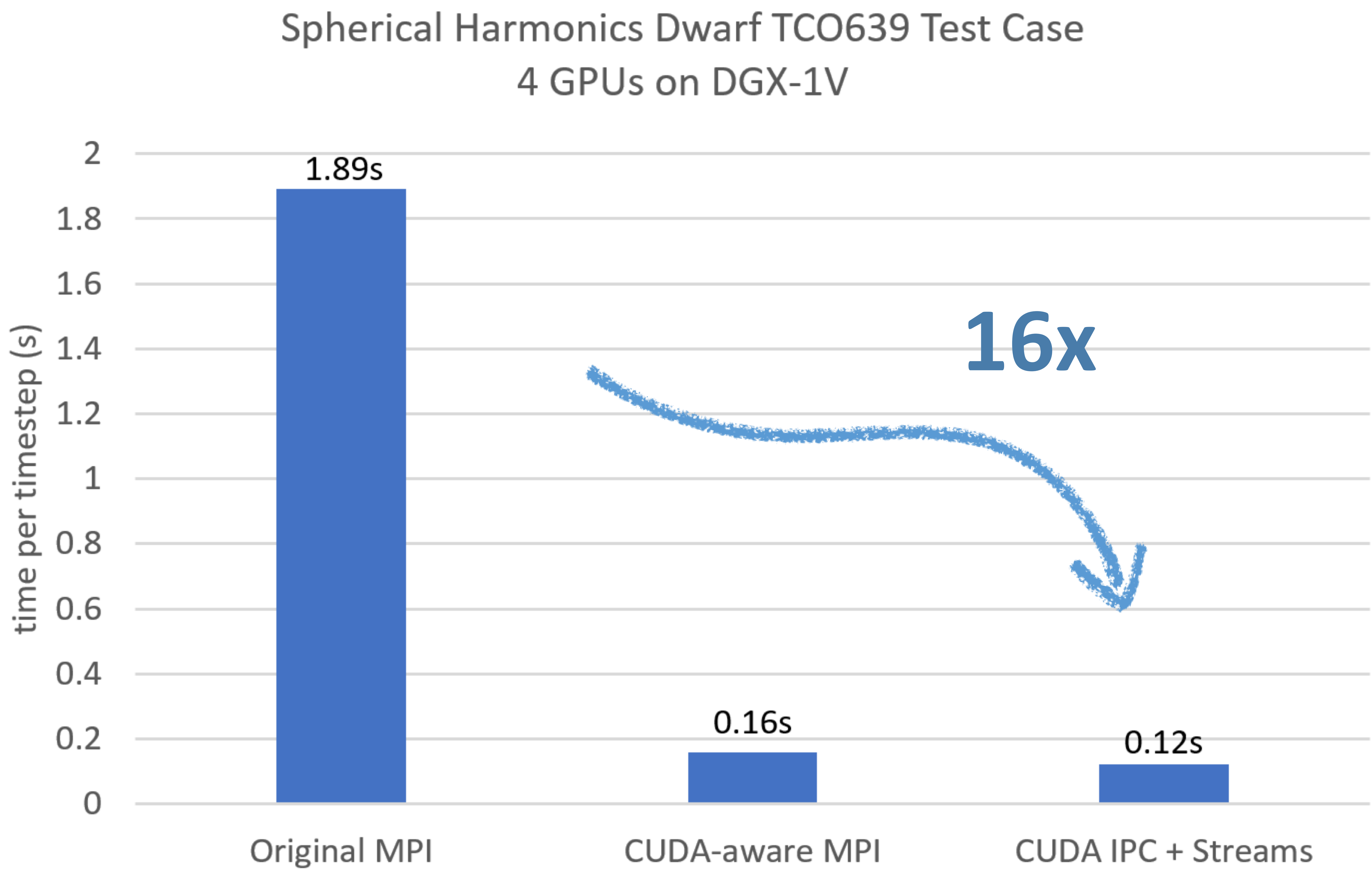
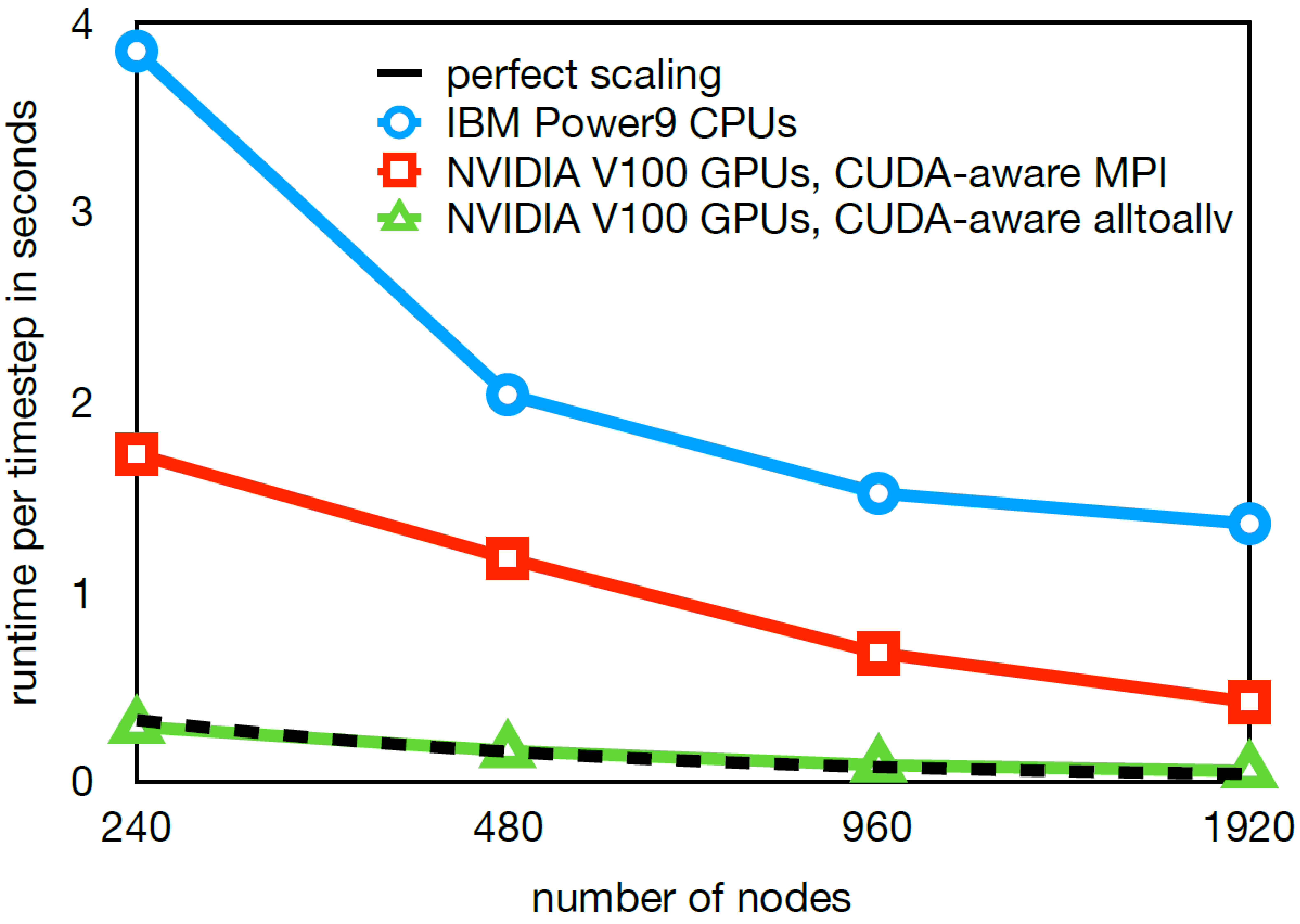



*figure: courtesy of Alan Gray, Peter Messmer (NVIDIA)*

# GPUs vs CPUs on Summit

Funded by the European Union



# References & Acknowledgements

*Seljebotn 2012, WAVEMOST-fast spherical transforms by butterfly matrix compression (Astrophysical Journal Supplement, vol 199)*

*Tygert 2010, Fast algorithms for spherical harmonic expansions III (JCP vol 229)*

*Wedi et al 2013, A fast spherical harmonics transform for global NWP and climate models (MWR vol 141)*

*Krishnamurti, T.N., Bedi, H.S., Hardiker, V., Watson-Ramaswamy, L., An Introduction to Global Spectral Modeling, 2006*

The ESCAPE 2 project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 800897. This material reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

**Image credits:**

- Images on slide 1 used under license from shutterstock.com

# Some practice …

**interactive web-app by Andreas Müller about spectral transform**

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

Funded by the European Union