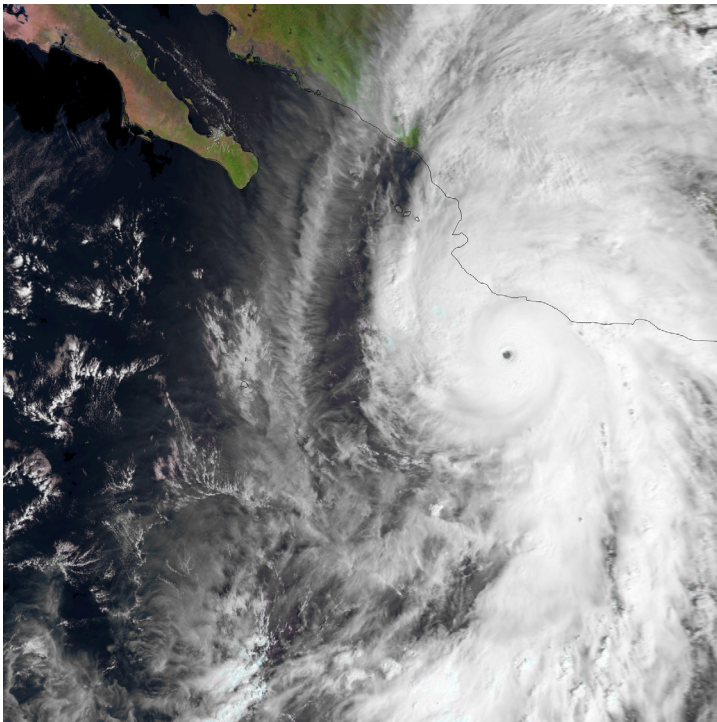


COMPUTING

The new ECMWF interpolation package MIR

Hurricane Patricia off the coast of Mexico on 23 October 2015 (© 2015 EUMETSAT)



This article appeared in the Computing section of ECMWF Newsletter No. 152 – Summer 2017, pp. 36-39.

The new ECMWF interpolation package MIR

Pedro Maciel, Tiago Quintino, Umberto Modigliani, Paul Dando, Baudouin Raoult, Willem Deconinck, Florian Rathgeber, Cristian Simarro

To use ECMWF forecasts, users typically need to either transform the data from its original spherical harmonics representation (spectral space) into grid space (physical space) or map the data from the model's Gaussian grid to a grid adapted to their needs. These operations are mostly carried out in user requests to the ECMWF real-time product generation system, to the MARS archiving and retrieval system, or within tools such as Metview and the ECMWF Web API.

The interpolation software used until now to perform these transformations, which is part of the EMOSLIB package, has progressively outgrown its original design and implementation limitations. As a result, it has become hard to extend and maintain. ECMWF is about to replace it with the new Meteorological Interpolation and Regridding (MIR) software package developed in-house. The primary goal of MIR is to replace the current EMOSLIB interpolation functions. Beyond this, MIR's flexible design facilitates scalability improvements and additional features. These include efficiency gains, a high degree of user configurability, and support for a wider range of grids than in the current package.

MIR design

The EMOSLIB interpolation package was written in the 1980s and much has changed since then: the model grid resolution has steadily increased, a variety of grid types have been introduced (Malardel et al., 2016), and many new parameters have been added over the years, often associated with different processing requirements. Both software and hardware technologies, such as programming languages, design paradigms, supporting libraries and hardware architectures, have evolved significantly. Together with new numerical methods and ECMWF's improved understanding of user requirements, all these aspects have prompted ECMWF to design the new, extensible and easy-to-maintain MIR package.

MIR features configurable operations and comes with reasonable defaults; it can be extended and configured by users; and it uses Atlas, ECMWF's framework for the development of efficient numerical weather prediction (NWP) and climate applications, to implement its data structures and numerical techniques. Compute-intensive numerical operations are delegated to a linear algebra interface with backend libraries able to provide specialised support for GPUs, accelerator cards and other specific hardware.

MIR builds on past experience and is aligned with the new possibilities enabled by advances in high-performance computing. Ideas and code contributions from multiple teams at ECMWF are continuously improving the different code bases. MIR uses the spectral transforms library in the Integrated Forecasting System (IFS), the Atlas, ecKit and MARS libraries, and well established standard libraries (FFTW, Eigen) and their hardware or vendor-specific optimised implementations (array operations, GPU, multi-core, etc.). Building on well-defined interfaces and consistent behaviour, the development effort is being shared and re-used across different projects, and MIR's functionality is provided downstream to a wide range of the Centre's services.

Under the hood

MIR is implemented in C++, a programming language appropriate for expressing object-oriented (OO) designs. C++ is also suitable for high-performance computing (HPC) and portable to different hardware architectures. The right balance of language features, such as dynamic polymorphism (inheritance) and static polymorphism (including generic programming), helps to keep the code very concise, readable and extensible. All this makes the code easy to maintain.

Through the MIR API, users (or client software) define an interpolation task by providing both the description of the required interpolation, such as the characteristics of the target grid and the method to use, and the source fields to interpolate from. Together with a third aspect, the default interpolation settings, MIR prepares an 'action plan' to execute the request in order to provide the user with the desired result. The fundamental concept in MIR is the data processing pipeline. It is composed of interpolation operators to process input data with deferred execution. The preparation of the action plan involves decisions on which actions to use and how they are parametrized. The preparation of the action plan can be specific to a 'style', of which two are being tested: 'MARS' and 'Dissemination'. These styles mimic

the respective operational services and implement their characteristics whilst ensuring consistent results across the services.

Figure 1 shows an example flowchart for the preparation of the action plan, effectively a data processing pipeline. Such flowcharts can be interpreted as directed acyclic graphs (DAG). In an environment that processes requests in parallel, such as product generation, processing pipelines are combined to construct a data processing tree (arborescence), exemplified in Figure 2. Operations have to match in both type and parametrization to be merged. This approach effectively minimises the number of total operations and thus runtime and input/output (I/O).

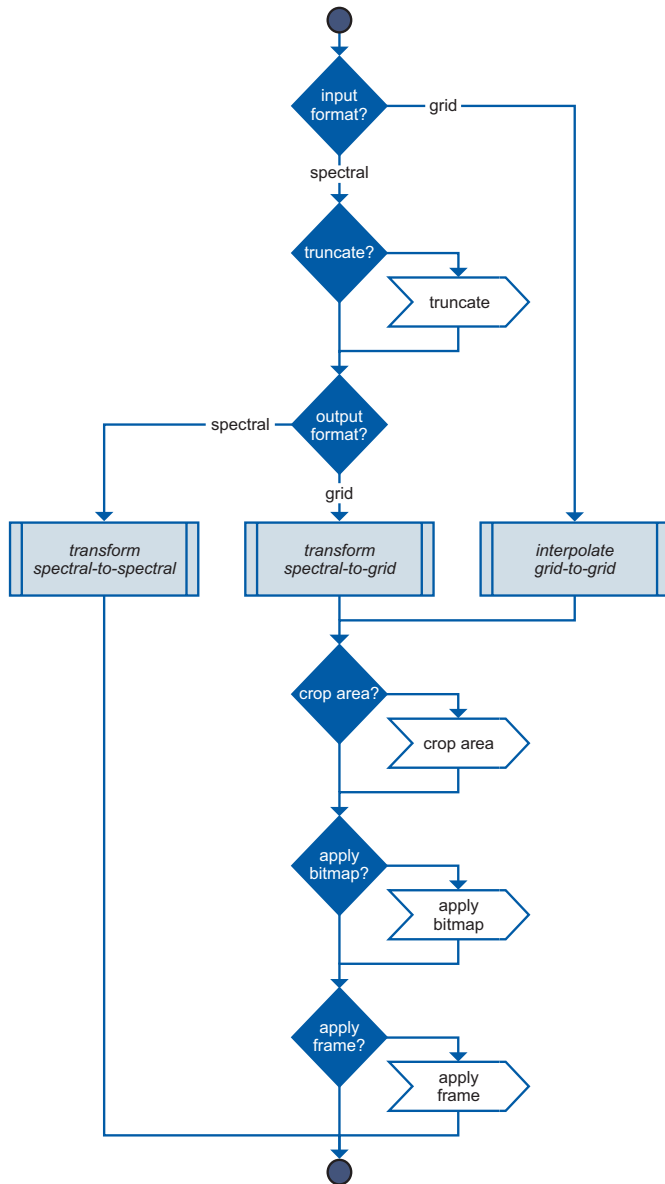


Figure 1 This example flowchart shows how a MIR action plan is prepared.

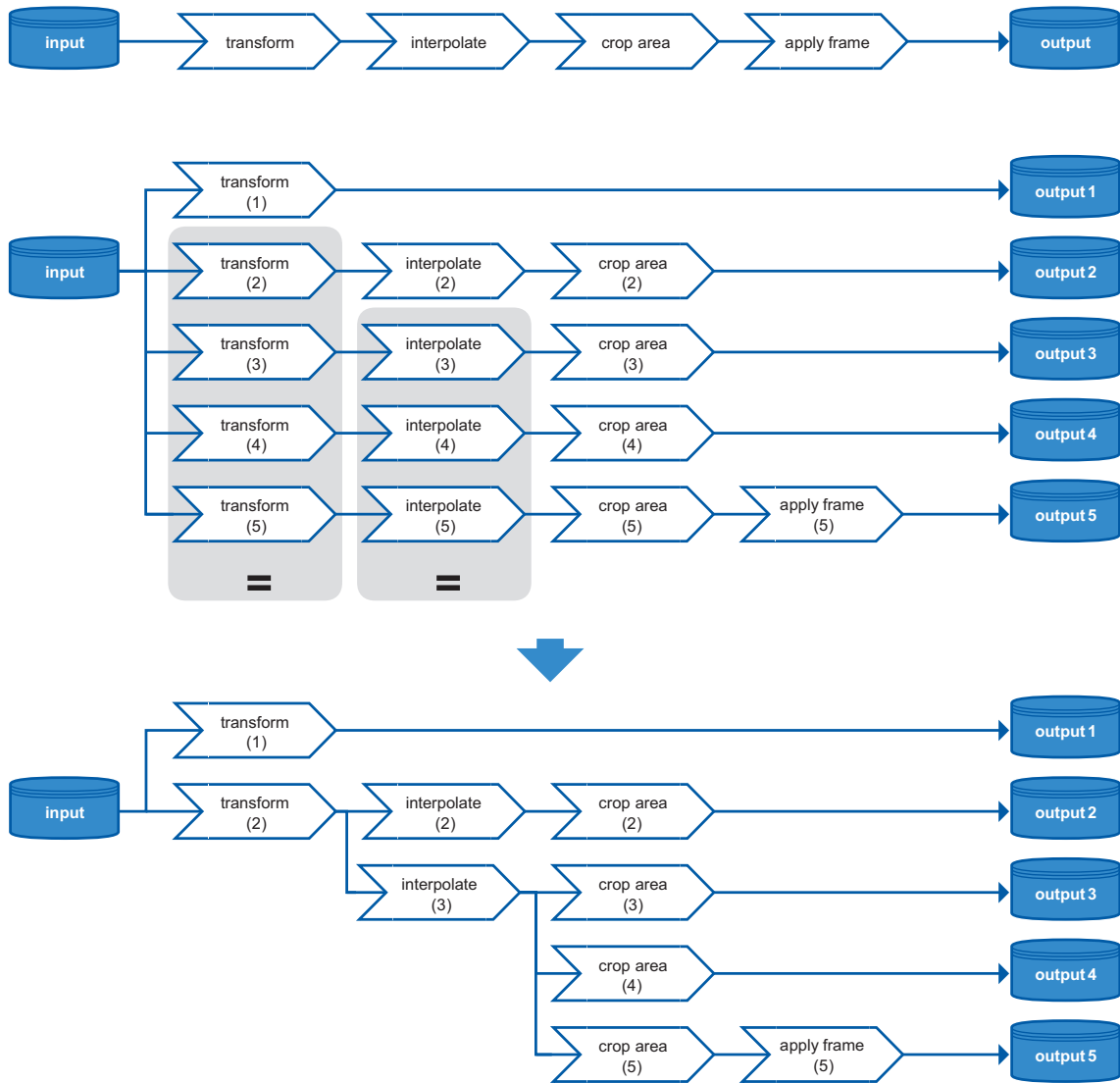


Figure 2 Two types of processing pipelines (action plans): a serial execution pipeline (top) and two parallel execution processing trees (middle and bottom). The middle pipeline has identically parametrized operations, identified by numbers, which are merged into single operations in the bottom pipeline, thus streamlining the process.

The most performance-critical of these operations are additionally supported by caching mechanisms, which enables performance gains for repeated operations. Caching not only expedites the usual spectral transform coefficients calculation but also the calculation of the matrix used in grid-to-grid interpolations and area crop mappings. Actions range from interpolations (grid-to-grid), transformations (spherical harmonics to grid space) and corrections (adjusting wind direction) to generic post-processing (such as truncation, area cropping or bitmapping). Each action is applied to input data whose output acts as input to the next action in the plan. Actions are described in an OO logical and extensible hierarchy. The two most relevant interpolation action classes are described below.

Grid-to-grid interpolation

Actions of this class interpolate between two grids using a ‘method’. The method is responsible for assembling a linear system matrix (say **W**) which relates the input data vector (field data **a**) to the output data vector **b = Wa**. The matrix entries can be corrected for missing values, the presence of a land sea mask or other shape preserving corrections.

The linear system abstraction makes it possible to easily vary the method. For instance, a mass conserving method can be expressed in the same system: $\mathbf{b} = \mathbf{M}_B \mathbf{W} \mathbf{M}_A \mathbf{a}$. The matrices \mathbf{M}_A and \mathbf{M}_B are independent of the construction of \mathbf{W} and relate the surface meshes built from the input and output grids (respectively) to each other. Another example is the support for circular quantities (such as angles or directions) expressed in the Cartesian components of the input and output vectors $[\mathbf{b}_x \ \mathbf{b}_y] = \mathbf{W} [\mathbf{a}_x \ \mathbf{a}_y]$, providing the Cartesian de- and re-compositions $[\mathbf{a}_x \ \mathbf{a}_y] = f(\mathbf{a})$ and $\mathbf{b} = f^{-1}([\mathbf{b}_x \ \mathbf{b}_y])$.

The typical interpolation case uses a Finite Element (FE)-based method: each matrix row represents an output point projected from the Earth's centre to intersect a 3D input surface mesh element (similar to a ray tracing technique). From this intersection the input point interpolation weights are obtained, and hence the equation for the output point. Interpolating in 3D space employs the Earth-Centred, Earth-Fixed (ECEF) coordinate system, which avoids two-dimensional coordinate system singularities and planar projection distortions and is the natural representation of the physical space.

The FE family of methods includes bilinear and linear interpolations, which differ in the supporting mesh element (quadrilaterals and triangles, respectively). Since this general approach operates on spherical surface meshes, it is important to consider the most adequate surface element which will drive the interpolation weights calculation. In the case of linear interpolations, the natural choice for supporting triangle elements is the Lagrange linear shape function, and for quadrilaterals the bilinear interpolation. Both interpolate linearly within the element, and these two options are mapped to the 'linear' and 'bilinear' methods, respectively. The octahedral reduced Gaussian grid currently used in the IFS has a very suitable structure for triangular tessellation as it is constructed from a hierarchical triangulation of a regular octahedron.

Another typical interpolation case is the 'k-nearest neighbours' method, with a configurable choice of k neighbours. A common choice is to use the 'nearest neighbour' method ($k = 1$). To retrieve each output point's neighbours, a supporting k-dimensional tree structure makes the search very efficient. Consistent with the FE family of methods described above, the 3D distance of the output point to these neighbour input points is used to determine the interpolation weights and thus to build the matrix used in the linear system.

Since FE-based interpolation methods are driven by the elements composing the mesh, the methods are not necessarily confined to structured meshes. Unstructured meshes can be built for point clouds using Atlas and interpolated to and from other structured or unstructured meshes. In a different way, the k-nearest neighbours methods also naturally support arbitrary point clouds because they are not mesh-based. Other methods can be conceived, for example, exploiting the structured nature of the grid for significant speed gains and lower memory requirements. The implementation of these numerical techniques – interpolation methods and mesh generation – is supported by Atlas. Figure 3 shows a remapping from a temperature field point cloud and associated unstructured mesh to a regular grid. Figure 4 shows a geopotential height field remapped from high resolution TCo1279 (HRES operational 'cubic-octahedral' grid) to a very coarse TCo31.

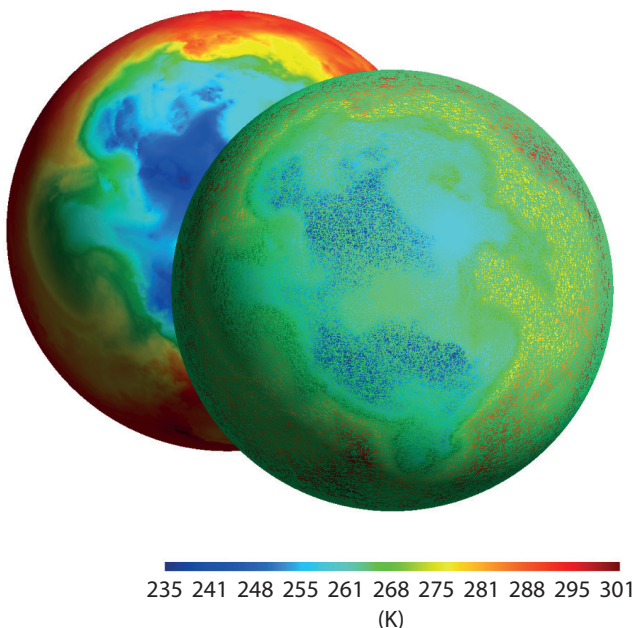


Figure 3 Upper atmosphere temperature field in grid space (back) interpolated from a point cloud (an arbitrary set of data points not following a structure) and associated unstructured mesh (front). MIR can perform transformations from one into the other.

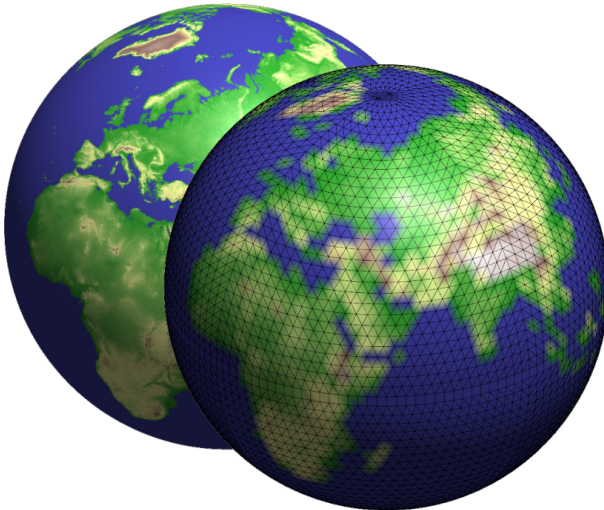


Figure 4 Surface geopotential height field (in $\text{m}^2 \text{s}^{-2}$) interpolated from an octahedral reduced Gaussian grid TCo1279 (HRES operational grid, back) to low resolution TCo31 (front). The colour palette is qualitative, for illustrative purposes only.

Spherical harmonics to grid space

The spectral transform which converts a field between spectral space and grid space is handled by Trans (the spectral transforms library in the IFS). It combines the Fourier and Legendre transforms when going from physical to spectral space (forward transform) and the inverse Legendre and inverse Fourier transforms when going from spectral to physical space (backward transform). The latter direction is the one chiefly of interest for MIR.

MIR uses Trans to transform data from spectral space to grid space: scalar fields and wind vorticity/divergence are transformed into Cartesian components. It also uses Trans to convert vorticity/divergence into Cartesian components within spectral space. All these operations share the implementation used in the IFS. Recent improvements in Trans introduced the more computationally efficient fast Legendre transform (FLT), from which MIR also benefits.

Spectral transformations are computationally very expensive. MIR includes a caching mechanism to handle the pre-calculated coefficients that speed up successive calculations, after the initial transform coefficient calculation. This mechanism can handle cache files both on disk and in memory. The user is also given control of these operations by defining the truncation (if any is required) to apply to the coefficients. MIR always defaults to a truncation choice that is consistent with the grid resolution of the operational data.

Validation

To substitute the current operational functionality based on EMOSLIB, a thorough validation of MIR results started in late 2016 and is scheduled to finish later this year. There are several aspects to the validation, ranging from the data the IFS handles and data from product generation to data in the MARS archive. There are also many additional sources of data included in the validation, such as HRES/ENS, ERA projects, Copernicus, etc. To efficiently test the variety of data handled for post-processing, four parameter interpolation classes have been planned:

- Upper atmosphere parameters, testing the correctness of spectral truncations and transforms; this includes transforms of scalar parameters, such as temperature or geopotential height, and of wind vorticity/divergence to Cartesian components (vector)
- Wave model parameters, testing the correct handling of special grid formats, application of directional interpolations, non-interpolation of spectra and associated parameters, and correctness near the coast
- Surface parameters, testing a relevant sample of surface parameters and associated specialised treatments, such as the role of land–sea masks and parameter-specific corrections
- Special cases such as precipitation threshold, sub-area handling, Copernicus new parameters, shifted grids, simulated satellite images, etc.

A validation suite, which encompasses a large and growing set of tests, has been created to cover the testing sections above. A substantial amount of testing has already been carried out and the first two parameter interpolation classes in the list above are very close to completion. Testing has so far identified issues in both MIR and the current operational interpolation software. As a result, it has been possible to improve the quality of both. In addition, using the validation suite will increase confidence when making any changes to MIR in the future.

The validation process is following a very agile process: much of the testing suite runs in parallel and on different platforms. Testing is usually carried out within one of the classes above, or it is restricted to a specific grid type or parameter. When an issue is identified, such as a significant difference when compared against reference results, it is submitted to the issue-tracking system.

What will change for users?

The aim is to replace the existing operational interpolation software in all its uses. Most users interact with the interpolation package through their MARS requests and Metview macros, or indirectly with product generation requests, and in these cases the changes for users will be minimal.

With the new interpolation closely linked to the IFS's Atlas library, it is ensured that ECMWF software and operational services can react fast to new research developments. This will be especially important in the coming years when testing innovative grids and numerical methods.

Test versions of MIR will be made available to all users later this year. The first operational use of MIR as part of the MARS client and product generation system used at ECMWF is expected for next year. After that a new API will be developed and released. MIR's flexible design will allow the addition of extra features, such as additional interpolation features, e.g. mass conservation.

Further reading

Malardel, S., N. Wedi, W. Deconinck, M. Diamantakis, C. Kühnlein, G. Mozdzyński, M. Hamrud & P. Smolarkiewicz, 2016: A new grid for the IFS, *ECMWF Newsletter No. 146*, 23–28.

© Copyright 2017

European Centre for Medium-Range Weather Forecasts, Shinfield Park, Reading, RG2 9AX, England

The content of this Newsletter is available for use under a Creative Commons Attribution-Non-Commercial-No-Derivatives-4.0-Unported Licence. See the terms at <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

The information within this publication is given in good faith and considered to be true, but ECMWF accepts no liability for error or omission or for loss or damage arising from its use.