MIR

ECMWF’s New Interpolation Package

P. Maciel, T. Quintino, B. Raoult, S. Siemen

ECMWF

tiago.quintino@ecmwf.int
Interpolation is **pervasive**:

- Product generation
- Access to data archive (MARS)
- Visualisation of products
- Web services

*Used by many operational systems at ECMWF*
MIR

Key Features

• **Flexible** and maintainable design
  – Configuration driven
  – Plugin based (users can extend)
  – Share **data-structures** with future IFS dynamical core

• **Any-to-Any** Grid algorithm
  – There is *always* a default algorithm

• Kernel based on linear **Interpolation Operators**
  – Enabling caching of operators
  – **Linear Algebra backend** support for GPU’s & Accelerator cards (Intel Phi)
Flexible Design
Architecture

Construct an Action Plan

- Decode
- Transform
- Interpolation
- Filter
- Encode

GRIB
NetCDF (basic)
Raw (mem. buffer)
Architecture

Construct an Action Plan

Decode → Transform → Interpolation → Filter → Encode

SH to SH, VOD-2-UV

SH to Grid
Construct an Action Plan

- Decode
- Transform
- Interpolation
- Filter
- Encode

Compute Interpolation Operator
Caching of operators
Linear algebra kernel
Construct an Action Plan
Architecture

Construct an Action Plan

Decode → Transform → Interpolation → Filter → Encode

GRIB
Accuracy
Compression
Atlas Library

- Framework for parallel, dynamic data structures
- Supporting multiple types of grids
- Fully written in C++ (Fortran 2003 interfaces)
- Basis to develop scalable dynamical core

Most importantly: co-developed with the Research Dep.
Any-to-Any Grid
A Myriad of Grids

- Lat-Lon
- Reduced Gaussian
- Reduced Gaussian Octahedral
A Myriad of Grids

Cubed Sphere

Yin Yang

Icosahedral
How to support *all* this in one algorithm?
Issues with latitude/longitude coordinate system

These points are close to each other

What about the poles?
These points are far away in lat/lon (cylindrical projection)

They are very close in this projection
But the earth is (almost) a 3D sphere...

- The poles are not special
- The anti-meridian is not special
- Use \((X, Y, Z)\) instead of latitude/longitude

**Distances are computed in 3D space**
How does it work?

- Input points
- Output points
Tessellation

Build a Finite-Element discretisation space
Projection

Use very efficient **Ray-Tracing** algorithms...
Tesselation

Support fully unstructured grids
3D Projection
+ Interpolation

Algorithm choice:
• Optimised if available
• Unstructured fallback
Interpolation Operators
Interpolation weights

Weight $W_j$ is proportional to area $A_j$
Interpolation Weights

\[ W_1 = (1-u)(1-v) \]
\[ W_2 = (1-u)v \]
\[ W_3 = uv \]
\[ W_4 = u(1-v) \]
Interpolation Operator

\[ F_i = \sum w_{ij} G_j \]

\( w_{ij} \) only depends on:
- Input grid, output grid, interpolation method, LSM

Can be cached!
Matrix multiplication: Linear Operator
Matrix multiplication: Batch Interpolation

CPU Cache and GPU friendly => SPEED
Features
2t: N640 ▶ LL 0.25/0.25 + rotation + crop + frame
2t N640 -> 1x1 + bitmap (MIR)
2t N640 -> 1x1 + crop (MIR)
2t N640 -> 1x1 + crop + frame (MIR)
2t: N640 ► LL 0.25/0.25 + rotation + crop + frame
ECMWF IFS MSL 18 May 2015 (N640)
Icosahedral (ICON) to Octahedral (IFS)

ICON grid treated as unstructured 2,949,120 points
Performance
Performance Comparison SH-to-Grid (preliminary)

3000+ fields from T1279 to reduced GG (auto truncation)

2x-3x single core performance
Performance Comparison Grid-to-Grid (preliminary)

3000+ fields from Reduced N640 to LL grid

4x single core performance
### Benchmarks

- Interpolation is driven by the size of the output grid

<table>
<thead>
<tr>
<th>Grid</th>
<th>N Points</th>
<th>Memory [GiB]</th>
<th>Wall Time [ms]</th>
<th>Speed [Mp/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N160</td>
<td>204 k</td>
<td>1.7</td>
<td>28.4</td>
<td>7.2</td>
</tr>
<tr>
<td>N256</td>
<td>524 k</td>
<td>1.8</td>
<td>33.0</td>
<td>15.9</td>
</tr>
<tr>
<td>N512</td>
<td>2097 k</td>
<td>1.8</td>
<td>51.2</td>
<td>40.9</td>
</tr>
<tr>
<td>LL 0.1/0.1</td>
<td>6483 k</td>
<td>2.6</td>
<td>99.9</td>
<td>64.9</td>
</tr>
<tr>
<td>N1024</td>
<td>8388 k</td>
<td>2.7</td>
<td>115.4</td>
<td>72.7</td>
</tr>
<tr>
<td>LL 0.05/0.05</td>
<td>25 927 k</td>
<td>6.1</td>
<td>252.2</td>
<td>102.8</td>
</tr>
</tbody>
</table>
Status

- Feature complete
- Going through validation
- Preparing Alpha release
- Seeking expert users feedback

Credits

- Pedro Maciel
- Tiago Quintino
- Baudouin Raoult
- Willem Deconinck
- Nils Wedi
- Mats Hamrud

All interpolations are wrong. Some are less wrong than others ...