ESCAPE

Preparing Forecasting Systems for the Next generation of Supercomputers

D4.4 Atlas extension for LAM use

Dissemination Level: public

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 67162

Funded by the European Union

Co-ordinated by ECMWF

ESCAPE

Energy-efficient Scalable Algorithms for Weather Prediction at Exascale

Author **Daan Degrauwe**

Date **30/03/2017**

Research and Innovation Action

H2020-FETHPC-2014

Project Coordinator: Dr. Peter Bauer (ECMWF)

Project Start Date: 01/10/2015

Project Duration: 36 month

Published by the ESCAPE Consortium

Version: 1.0

Contractual Delivery Date: 31/03/2017

Work Package/ Task: WP4/T4.4

Document Owner: Daan Degrauwe

Contributors: Daan Degrauwe (RMI), Willem Deconinck (ECMWF), Joris Vanbever (RMI)

Status: Final

Table of Contents

Figures

Tables

1 Executive Summary

ESCAPE stands for Energy-efficient Scalable Algorithms for Weather Prediction at Exascale. The project develops world-class, extreme-scale computing capabilities for European operational numerical weather prediction and future climate models. ESCAPE addresses the ETP4HPC Strategic Research Agenda 'Energy and resiliency' priority topic, promoting a holistic understanding of energy-efficiency for extreme-scale applications using heterogeneous architectures, accelerators and special compute units by:

- Defining and encapsulating the fundamental algorithmic building blocks underlying weather and climate computing;
- Combining cutting-edge research on algorithm development for use in extreme-scale, high-performance computing applications, minimising time- and cost-to-solution;
- Synthesising the complementary skills of leading weather forecasting consortia, university research, high-performance computing centres, and innovative hardware companies.

ESCAPE is funded by the European Commission's Horizon 2020 funding framework under the Future and Emerging Technologies - High-Performance Computing call for research and innovation actions issued in 2014.

Numerical weather models can be categorised into global models, that model the atmosphere of the entire planet, and limited-area models (LAMs), that are restricted to a confined geographical region.

The ESCAPE project encompasses both types of models: some partners run a global model (ECMWF, DWD, MeteoFrance), while others focus more on limited-area models (DMI, RMI, MeteoSwiss).

One of the cornerstones of the developments in the ESCAPE project is the Atlas software, which acts as a common framework for the contributions of the different partners.

Atlas was originally developed for global models. The deliverable 4.4 deals with the extension of Atlas with features required for limited-area modeling. The developments for this deliverable were merged with other contributions to Atlas in deliverable D1.3, and consolidated in the git branch “D1.3” of Atlas on the ESCAPE software collaboration platform BitBucket.

2 Introduction

2.1 Background

Atlas is the framework developed at ECMWF to streamline dataflows in atmospheric models running in a distributed environment. As such, it is (among other things) responsible for the following tasks:

- the generation of a set of gridpoints, represented as a `Grid` object;
- the distribution of these gridpoints over different processors with a `Partitioner` object;
- establishing the connectivity between the gridpoints, i.e. a `Mesh` object;

- defining different types of approximations that are encountered in NWP models, such as point-value representations or spectral representations. These representations are defined within a so-called `FunctionSpace` object.
- setting up the communication patterns between the processors required for parallel algorithms using these `FunctionSpace` objects. Examples are global gather/scatter and the exchange of halo's.
- utilities such as handling configurations, logging, performing input/output to files, etc.

Atlas plays a central role in the ESCAPE project. In order to facilitate the porting (work package 2) to accelerator hardware such as GPU's and KNL, as well as their optimisation on these platforms (work package 3), the different NWP dwarfs (as defined in deliverable 1.1) are built on the common Atlas framework. The development of Atlas for these goals is carried out in deliverable 1.3.

Being developed at ECMWF, a Centre focusing on global weather forecasts, Atlas was originally targeted at global grids. Since many European countries run limited-area models, it is important that Atlas also supports such grids.

2.2 Scope of this deliverable

2.2.1 Objectives of this deliverable

The objective of this deliverable is to extend the Atlas library with features that are required for limited-area modeling.

2.2.2 Work performed in this deliverable

The following aspects of LAM grids have been introduced in Atlas:

- the notion of a geographic projection (section 3);
- the generation of LAM grids and their handling in a parallel computing environment (section 4);
- the generation of LAM meshes based on LAM grids (section 5).

2.2.3 Deviations and counter measures

Although the features listed in the previous section make it possible to define and use LAM grids in Atlas, some minor aspects of LAM modeling have not yet been introduced:

- The treatment of lateral boundary conditions and the notion of a periodisation zone (extension zone) for spectral LAM models. These features are (for the time being) considered to fall outside of the scope of the Atlas library. They can easily be dealt with at the level of the LAM model that is built on top of Atlas. One exception concerns the so-called periodic boundary conditions for spectral LAM models. These involve the connectivity between grid points, and should be treated at the level of the Atlas library. Therefore, periodic boundary conditions have been introduced in Atlas, as described in section 5 of this document.
- The effect of the geographic projection on numerical operators, such as the calculation of finite difference derivatives, has not yet been fully accounted for.

> The reason is that the development of these numerical operators in Atlas is still in an early stage. Therefore it was decided to revisit the effect of projections at a later time in the ESCAPE project. Section 6 already gives a brief outline of the work that needs to be done.

To make sure these issues will be dealt with in the course of the ESCAPE project, specific tasks were created on the ESCAPE JIRA platform.

3 Domains and projections

3.1 Domains

The first obvious feature of LAM grids is the fact that they do not cover the entire globe, but they only cover a limited region. This region is referred to as the domain of the grid. An abstract `Domain` class has been introduced, along with a few concrete classes: `GlobalDomain`, `ZonalBandDomain`, `RectangularDomain`. A Factory design pattern is to ensure flexibility in creating different domain types.

Most (if not all) structured LAM grids have a `RectangularDomain`. It should be noted that the domain is currently only used in Atlas during the *construction* of a grid. This means that grids with irregular domains (e.g. unstructured grids) can be defined as a collection of user-provided grid points, without the need for an explicit domain specification.

It should be noted that the domain is defined in projected coordinates, rather than in geographic coordinates (see section 3.2).

3.2 Projections

A less trivial feature of LAM grids is the mapping of the geographic domain (a region on the sphere) onto a plane, as shown in Figure 1. The geographic coordinate system is longitude-latitude *(λ, ϕ)*, the plane coordinate system is denoted with *(x, y)*. This mapping between both coordinate systems is called a geographic projection, and it is introduced in Atlas as a `Projection` class.

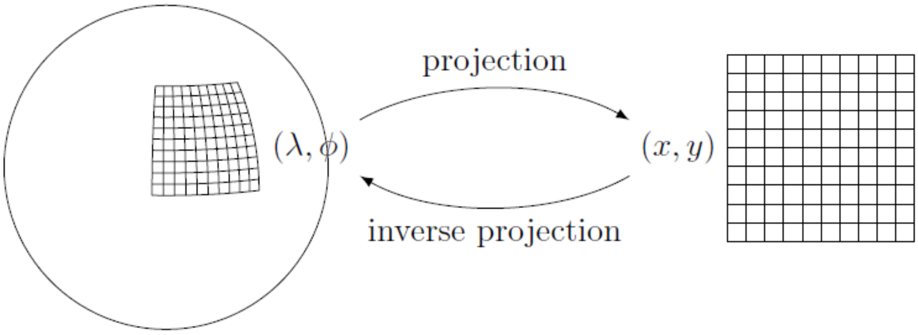


Figure 1: Projection between geographic coordinates (λ, ϕ) and grid coordinates (x, y).

The Projection class is abstract, and several concrete classes derive from it, again using the Factory design pattern to introduce flexibility to implement different projection types.

Some Projection classes internally allow to rotate and unrotate as part of the projection operation, which performs a rotation of geographic coordinates. This functionality is useful for several projections, such as the rotated longitude-latitude projection, or the rotated Mercator projection.

Table 1 gives an overview of the projections that are currently implemented in Atlas. Note that these projections cover the needs of the LAM grids in the ALADIN-HIRLAM system, as well as those of COSMO-LAM and UKMO's Unified Model. Adding additional projections is quite straightforward. For an elaborate overview of the mathematics of geographic projections, see Snyder (1987).

Atlas class	Remarks
LonLatProjection	Not really a projection, since $x=\lambda$, $y=\phi$
RotatedLonLatProjection	Longitude-latitude, but with a chosen Pole shifted to an arbitrary position
SchmidtProjection	Shifting of latitudes which allows for an increased resolution towards a Pole
RotatedSchmidtProjection	Schmidt, but with a shifted Pole
LambertProjection	Conformal projection on a secant or tangent cone
MercatorProjection	Conformal projection on a cylinder
RotatedMercatorProjection	Mercator, but with a shifted Pole

Table 1: Projections in Atlas.

Projection classes have the methods xy2lonlat and lonlat2xy to convert between geographic and grid coordinates.

3.3 Spacing

The concept of spacing has been introduced in Atlas to accommodate for the fact that grid points are not always equidistant. The most obvious example of this are Gaussian grids, for which the latitudes are distributed as the roots of Legendre polynomials. Another possible application is that a nonuniform spacing allows for a local increase of the resolution. An abstract Spacing class is implemented, along with a few concrete classes: LinearSpacing, GaussianSpacing and FocusSpacing.

The Spacing class contains a generate method, which generates coordinates according to a spacing.

4 Grids and grid partitioners

4.1 Grids

The original object-oriented inheritance between different grid types in Atlas was removed. In fact, only two classes of grids remain: `Structured` and `Unstructured`. Other grid types are specified through the concept of grid-interpretations. This means that instead of using separate classes for every grid type, the type of a grid is determined by its characteristics. For example, to determine if a grid is regular, it is verified if the number of grid points on every parallel is the same. Figure 2 shows the hierarchy between the grid types.

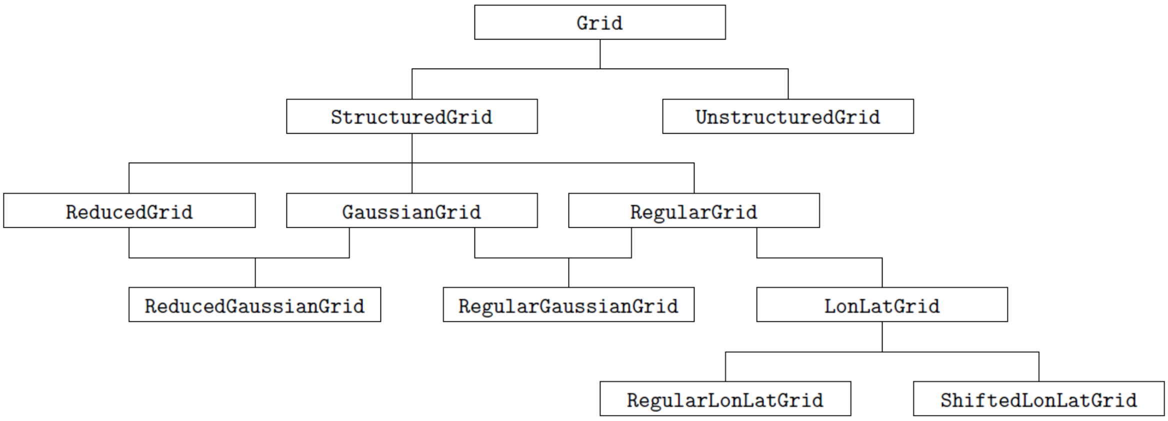


Figure 2: Grid types in Atlas.

The grid parameters (dimensions, domain, projection, etc.) are different for all grids, but are specified with an Atlas `Config` object. A gallery of some grids and the configurations to create them is presented in section 7.

A method `lonlat` has been added to the `Grid` class to retrieve the geographic coordinates of a gridpoint, which is necessary e.g. to determine the incoming solar radiation, or the Coriolis factor.

4.2 Grid partitioners

Atlas already contains a partitioner targeted at global grids: the `EqualRegions` partitioner. More information on this partitioner is given in deliverable 1.3. Relevant for this document is that this partitioner has the property that a region around a pole is entirely attributed to a single processor. This is beneficial to limit the amount of communications. However, for LAM grids, the situation is opposite: since these grids don't ‘reduce’ to a single point at the boundary of the domain, assigning the (North or South) boundary entirely to a single processor would actually increase the amount of communications.

Therefore, a new `Checkerboard` partitioner class is introduced in Atlas, which distributes a regular grid as a checkerboard pattern over a set of processors. This partitioner is quite close to the partitioner of the LAM grids of the models in the ALADIN-HIRLAM system.

Figure 3 shows the difference in behavior between the `EqualRegions` partitioner and the `Checkerboard` partitioner.

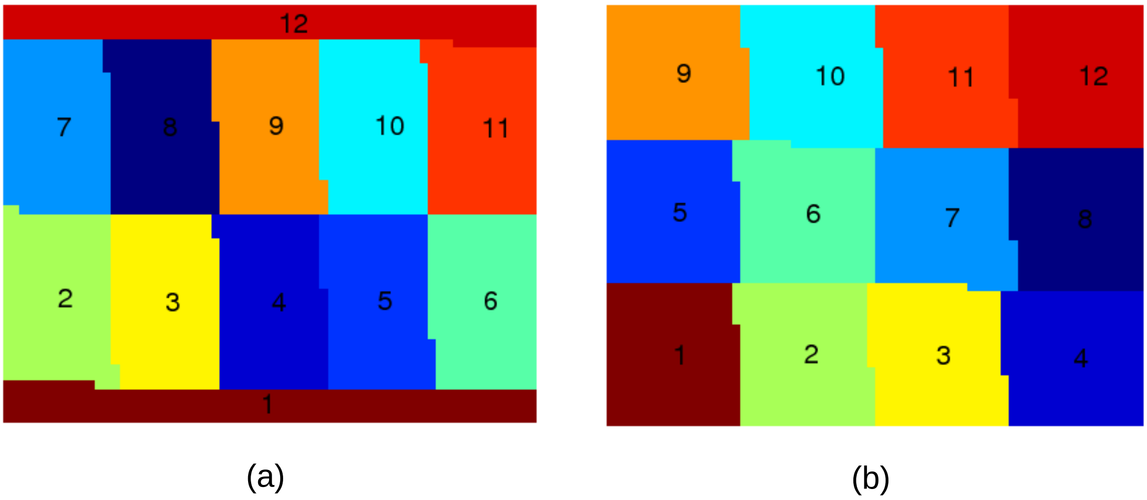


Figure 3: Partitioners in Atlas for a regular grid: (a) EqualRegions partitioner, where the poles (upper and lower boundaries) are attributed to a single processor; (b) Checkerboard partitioner, which is more suitable for LAM grids.

5 Meshes and mesh generators

A `Mesh` object describes the connectivity between nodes, and is created by a `MeshGenerator`. Two differences exist between the generation of global and LAM meshes:

- LAM grids are regular, i.e. the number of grid points per row is constant. Consequently, LAM meshes will entirely consist of quadrangular elements. This differs from (reduced) global grids, which are composed of quadrangular and triangular elements.
- The boundary conditions for LAM grids differ from those of global grids. LAM grids have no poles, but they can be bi-periodic in the case of spectral models like those of the ALADIN-HIRLAM system.

A new `RegularMeshGenerator` class has been introduced in Atlas to suit these specific needs of LAM mesh generation. It has configurable options `periodic_x` and `periodic_y`. An example of a bi-periodic mesh is given in section 7.

6 FunctionSpaces

6.1 Numerics

For most numeric operators (see next section for an exception), no formal distinction is necessary between global grids and LAM grids. These numeric operators either act on the elements of the mesh (as for finite-volume methods and finite-element methods), or on the points of the grid (as for finite-difference methods). An element (e.g. a quadrangular cell) behaves in the same way when belonging to a LAM grid as when belonging to a global grid. Similarly, there is no difference between a grid point belonging to a global grid and one belonging to a LAM grid. Therefore, no modifications are necessary on the numerical operators of the function spaces in Atlas, to apply them to LAM grids.

However, as mentioned in section 2.2.3, one (small) aspect should not be neglected here, which is the effect of a geographic projection on numeric operators. Currently, this is hardcoded in Atlas assuming a longitude-latitude projection. The more general and correct way to do this is by using the metrics of the geographic projection, as described in Zdunkowski and Bott (2003).

6.2 Spectral function space

One function space where global grids differ from LAM grids is the spectral function space. For global models, the spectral transform is based on spherical harmonics, while for LAM grids it is based on bi-Fourier functions. This issue (i.e. making the spectral function space aware of the global/regional character of the grid) will be addressed once the development of function spaces in Atlas reaches a more mature stage.

6.3 Interpolations

Support for interpolations in Atlas is useful for several reasons: for running physical parameterisations at reduced resolution (Task 1.3), or for multi-grid preconditioning of iterative solvers (Task 1.4). Also in the context of limited-area modeling, interpolations are necessary to go from a global geometry to a regional geometry. However, it can be stressed here that there is nothing special about LAM grids in this regard: all the interpolator requires is a set of grid point locations, to which the fields are interpolated. As indicated in section 6.1, there is no difference between a grid point belonging to a global grid and one belonging to a LAM grid.

7 Gallery

7.1 Bi-periodic Mesh

Topologically, a biperiodic grid is represented by a torus. A mesh action `BuildTorusXYZField` has been implemented to generate the coordinates of the gridpoints on a torus. It should be stressed that this torus has no geographic meaning; the motivation for the `BuildTorusXYZField` action is merely to verify if the biperiodism (i.e. the connection of opposite sides of the rectangular domain through elements) is correct. For this example, a regular LAM domain with a Lambert projection is used.

Configuration

```
{
  "grid" :
  {
    "type" : "regional",
    "nx" : 50,
    "ny" : 40,
    "dx" : 150000,
    "dy" : 150000,
    "lonlat(centre)" :
[4,50],
    "projection"     : {
      "type" : "lambert",
      "latitude1"  : 50,
      "longitude0" : 4
    }
  },
  "mesh" :
  {
    "periodic_x"  : true,
    "periodic_y"  : true
  }
}
```

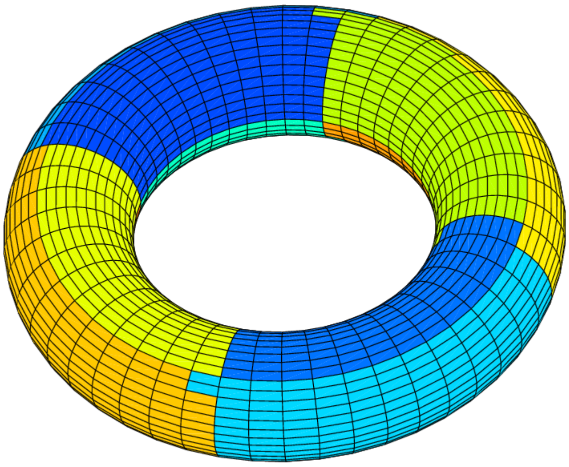


Figure 4: Bi-periodic mesh represented as a torus.

7.2 ARPEGE mesh

ARPEGE is the global model of MeteoFrance. It shares its dynamical core with the IFS model. ARPEGE uses a rotated stretched Gaussian grid, such that the high-resolution pole lies above France.

Configuration

```
{
  "grid" :
  {
    "type"       : "classic_gaussian",
    "N"          : 16,
    "projection" : {
      "type"      : "rotated_schmidt",
      "stretching_factor" : 2.0,
      "north_pole"        : [3.0,47.0]
    }
  }
}
```

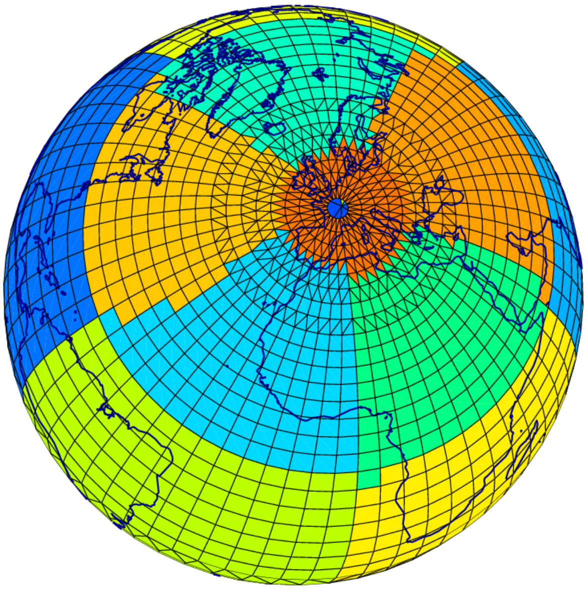


Figure 5: ARPEGE mesh.

7.3 LAM meshes

The following figures illustrate several projections with a LAM grid.

Configuration

```
{
  "grid" :
  {
    "type" : "regional",
    "nx"   : 50,
    "ny"   : 40,
    "dx"   : 50000,
    "dy"   : 50000,
    "lonlat(centre)" : [4,50],
    "projection" : {
      "type"        : "lambert",
      "latitude1"   : 50,
      "longitude0" : 4
    }
  }
}
```

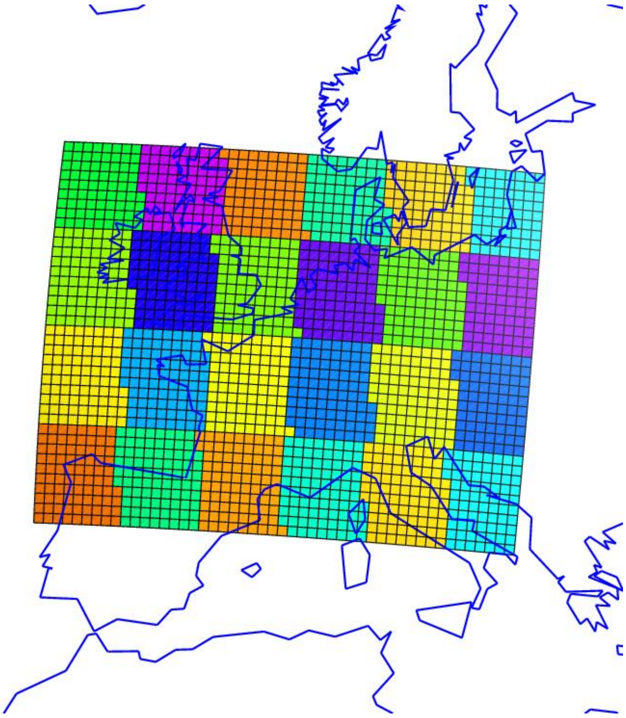


Figure 6: LAM grid with a Lambert projection.

Configuration

```
{
  "grid" :
  {
    "type"             : "regional",
    "lonlat(centre)" : [4.0,50.0],
    "nx"               : 50,
    "ny"               : 40,
    "dx"               : 0.5,
    "dy"               : 0.5,
    "projection" :
    {
      "type"        : "rotated_lonlat",
      "north_pole" : [-176.0,40.0]
    }
  }
}
```

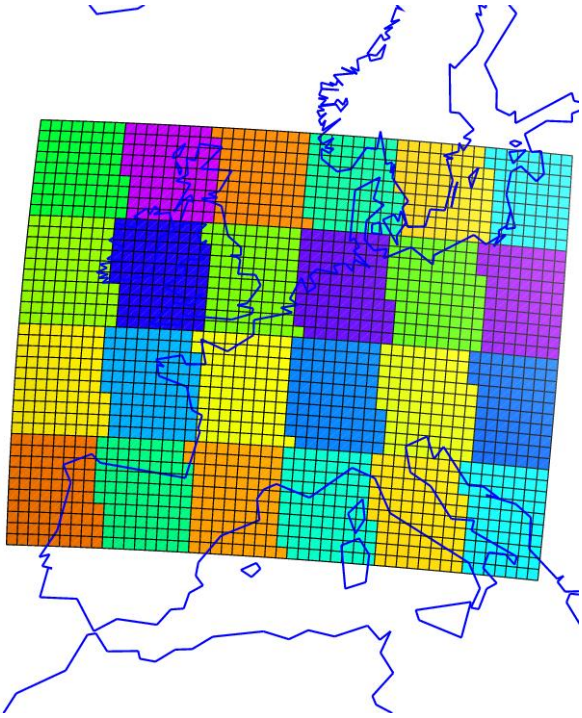
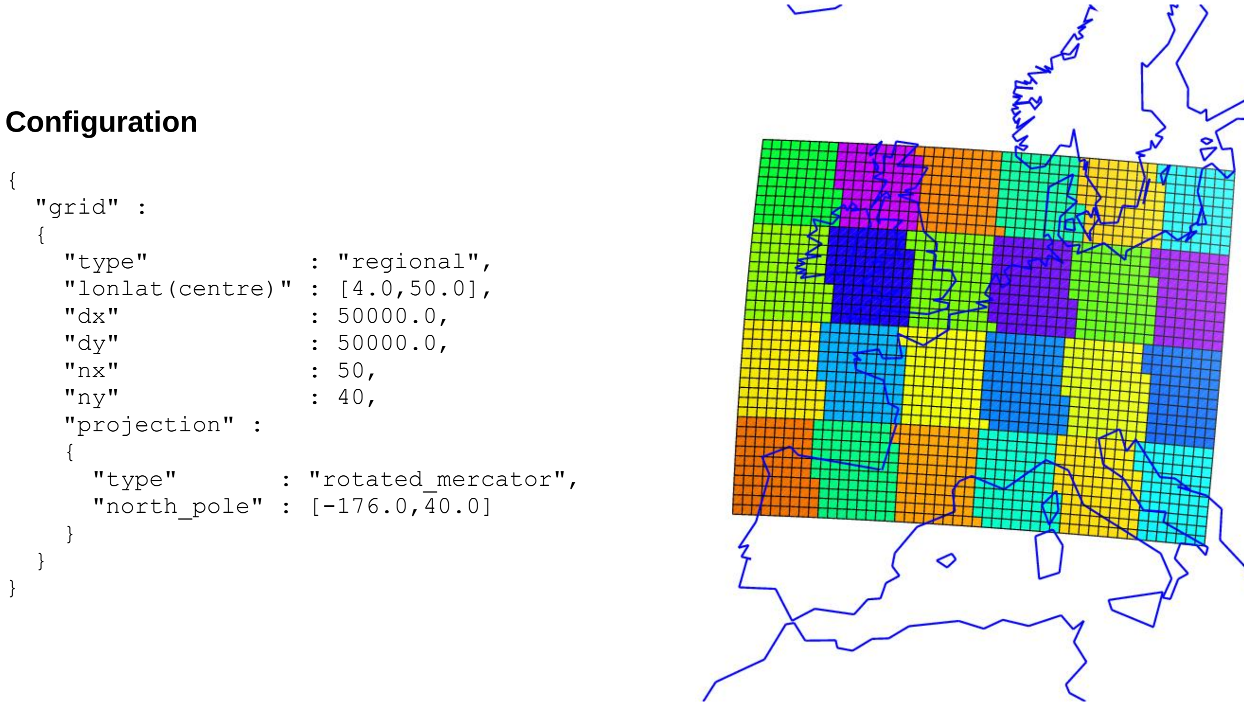


Figure 7: LAM grid with a rotated longitude-latitude projection.



Configuration

```
{
  "grid" :
  {
    "type"             : "regional",
    "lonlat(centre)" : [4.0,50.0],
    "dx"               : 50000.0,
    "dy"               : 50000.0,
    "nx"               : 50,
    "ny"               : 40,
    "projection" :
    {
      "type"       : "rotated_mercator",
      "north_pole" : [-176.0,40.0]
    }
  }
}
```

Figure 8: LAM grid with a rotated Mercator projection.

8 Conclusion

The necessary features have been developed in Atlas in order to support LAM grids and meshes.

9 References

Snyder, John P. (1987). *Map projections: A working manual*. USGS Report 1395

Zdunkowski, W. and A. Bott (2003). *Dynamics of the Atmosphere: A Course in Theoretical Meteorology*. Cambridge University Press.

Document History

Version	Author(s)	Date	Changes
0.1	Daan Degrauwe	March 10, 2017	Initial version
1.0	Daan Degrauwe	March 29, 2017	Accounted for internal reviewer comments: - Added section on FunctionsSpaces, including interpolations. - Added references for projections. Adapted according to D1.3 developments.

Internal Review History

Internal Reviewers	Date	Comments
Alastair McKinstry**,** ICHEC	2017-03-22	Good. Recommend for submission with minor changes above**.**
Alexander Macfaden, OSYS	2017-03-21	This report is appropriate for submission provided that the interpolation feature discussed above is accounted for, and under the assumption that the work identified in §2.2.3 is ultimately followed up.

Effort Contributions per Partner

Partner	Efforts
RMI	4m
ECMWF	1m
Total	5

ESCAPE

ECMWF Shinfield Park Reading RG2 9AX UK
Contact: peter.bauer@ecmwf.int

The statements in this report only express the views of the authors and the European Commission is not responsible for any use that may be made of the information it contains.