Vision, Modeling, and Visualization (2016)
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

Icosahedral Maps for a Multiresolution Representation of Earth Data

Mohammad Imrul Jubair[1], Usman Alim[1], Niklas Röber[2], John Clyne[3] and Ali Mahdavi-Amiri[1]

[1]University of Calgary, [2]German Climate Computing Centre, [3]National Center for Atmospheric Research

Abstract

The icosahedral non-hydrostatic (ICON) model is a digital Earth model based on an icosahedral representation and used for numerical weather prediction. In this paper, we introduce icosahedral maps that are designed to fit the geometry of different cell configurations in the ICON model. These maps represent the connectivity information in ICON in a highly structured two-dimensional hexagonal representation that can be adapted to fit different cell configurations. Our maps facilitate the execution of a multiresolution analysis on the ICON model. We demonstrate this by applying a hexagonal version of the discrete wavelet transform in conjunction with our icosahedral maps to decompose ICON data to different levels of detail and to compress it via a thresholding of the wavelet coefficients.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

The output generated by climate simulations is increasing in size, as well as complexity. Both aspects pose challenges for visualization and an interactive data analysis. The increase in complexity is due to maturing models that are able to better describe the intricacies of the climate system, while the gain in data size is a direct result of an increased spatial and temporal resolution used by such models.

To visualize, analyze, and integrate such large data sets, a Digital Earth framework can be used. A Digital Earth framework is a 3D representation of the Earth on which data sets from different sources can be represented in a multiresolution manner [MAAS15]. In climate science many different grids are used, all with different advantages and drawbacks. Regular grids such as lat/lon can be used and are easy to work with, but difficulties arise through mathematical instabilities at poles and a globally unequal cell area. To overcome these limitations, a Discrete Global Grid System (DGGS) can be used in which the Earth is represented by a spherical polyhedron [SWK03, MAAS15]. To provide a multiresolution representation on a DGGS, the faces of the polyhedron are initially refined (i.e. tessellated) and then projected to the sphere to form a set of cells. Different types of cells such as quads (squares or diamonds), triangles, or hexagons; different types of projected polyhedra such as the cube, icosahedron, or octahedron; and different types of projections such as equal area or conformal can be used to represent the Earth. Among these possibilities, equal area projections defined for an icosahedron are very common [PYX, Sah08, TBW*13]. This is due to the fact that equal area projections provide uniform (i.e. equal area) cells on the Earth and an icosahedron usually produces less distortion in comparison with other platonic solids [SM01, WKO92].

ICON, the ICOsahedral Non-hydrostaic model, jointly developed by the Max Planck Institute for Meteorology (MPI-M) and the German Weather Service (DWD), is a framework defined on an icosahedron with an equal area projection on which data sets are sampled via primal triangular cells, dual hexagonal cells and hybrid quadrilateral cells [ZRRB15] (Sec. 2). Although ICON data is not as easily accessible as data stored on a regular or curvilinear grid, it has several advantages. ICON has no poles, allows for an easy refinement in local areas if needed, and the coupling between the oceanic and atmospheric components is now much easier, as both models now share the same grid layout with different resolutions. Similar models to ICON – in terms of grid layout and structure – are the American MPAS, the Model for Prediction Across Scales, and the Japanese NICAM, the Nonhydrostatic ICosahedral Atmospheric Model, which makes the methods developed for ICON and discussed within this paper almost directly applicable to MPAS and NICAM as well. [RPH*13, PJR*15, SMT*07].

More recently, the ICON model was extended to permit large eddy simulations at cloud resolving resolutions as part of the HD(CP)^2 project [DSH*15]. The project aims at ultra-high resolutions to closely simulate clouds and precipitation processes. Although the simulation domain *only* includes Germany, with a spatial resolution of 100m and 150 vertical levels, one 3D variable already stores 2.5 billion cells per timestep. And this is just the beginning. In order to explore the data and to gain knowledge and detect new features and correlations, one needs to interact with these large data sets. This is only possible using in-situ compression and a level-of-detail (LoD) rendering approach. In here the data is decomposed into different LODs and compressed before it is written out to disk in a way that facilitates an interactive access later. In a second step, a visualization application that supports LoD rendering is used for a classic user-driven post visualization approach. Such an application accesses data in an out-of-core fashion; only the data that is relevant to the visible viewport is fetched.

Other approaches to visualize large data sets are in-situ visual-

DOI: 10.2312/vmv.20161355

ization, as it can be implemented using Visit [CBW*12] and ParaView [AGL05, Aya15], and distributed rendering on a large visualization cluster. Classic in-situ visualization integrates the visualization step within the simulation, in a way that the simulated data is visualized while it is available in main memory. Only the visualization results, either in the form of images or geometry, are written out to disk. While this approach saves a lot of disk space and IO bandwidth, a detailed exploration of the entire data set is not possible anymore. The distributed rendering approach requires a large and well balanced visualization cluster, that while running in this mode only serves a single user. Therefore, we are developing a visualization pipeline that allows in-situ compression of ultra-high resolution ICON/MPAS simulations, which can later be interactively explored and visualized on a single node visualization system.

This paper explores suitable wavelets and data structures that can be used to efficiently and effectively decompose and compress high resolution ICON data. The topic of out-of-core rendering of ICON data at different LODs is left for future work. ICON stores cell data in an unstructured manner; the underlying icosahedral discretization however is highly structured. Exploiting this fact in light of the above goals, this paper makes the following contributions:

- We introduce *icosahedral maps*, efficient data structures that map the connectivity information *for all cell-types* in ICON to a structured representation. These maps are an application of the atlas of connectivity maps (ACM) data structure [MAS14] and represent each face of the icosahedron using a 2D hexagonal grid. We provide the necessary grid traversal schemes that suitably map the connectivity of all cell types in ICON to a common hexagonal representation.
- We demonstrate how to decompose ICON data into different LoDs using a dyadic hexagonal wavelet scheme. Given the icosahedral maps, the wavelet filters associated with the hexagonal discrete wavelet transform are efficiently applied via 2D convolution, upsampling, and downsampling operations. Moreover, the same wavelet can be used for all cell-types.

The remainder of the paper is organized as follows. We provide an overview of the ICON grid in Section 2 followed by a review of related work in Section 3. In Section 4, we provide the essential mathematical ideas behind the proposed hexagonal wavelet scheme. Section 5 provides a more detailed account of our proposed icosahedral maps. Section 6 demonstrates the use of icosahedral maps with ICON datasets and also compares the effectiveness of the wavelet scheme for different cell types. Possible avenues of future exploration are discussed in Section 7.

2. ICON Grid Overview

The development of ICON falls into the design choice step of DGGS. An icosahedron is chosen as the base polyhedron and its faces are repetitively refined using a 1-to-4 refinement scheme to increase resolution. The cells thus created are the primal cells for ICON. In order to have similar topology on the corresponding spherical surface of the Earth, vertices are projected onto the sphere. However, the refinement leaves twelve *extraordinary vertices* on the Earth's surface that have a valence of five and correspond to the vertices of the base icosahedron; the regular ones have six. After discretizing, prognostic variables are assigned to the cells to hold data (Fig. 1a). Typically, data are computed at the centroid of the cell. However, solutions of numerical models often compute different quantities at different cell locations for improved accuracy. For example, ICON is C-Grid staggered [SNCB13], where data is computed

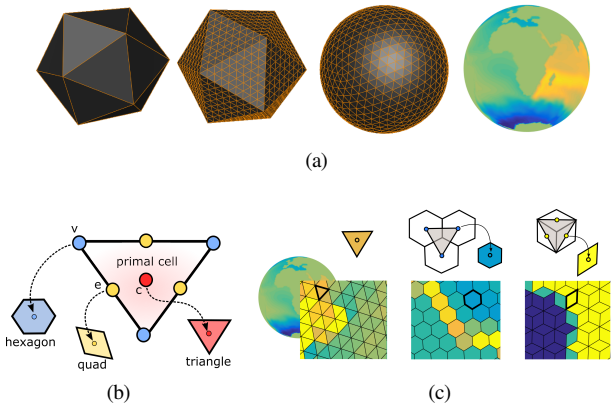


Figure 1: (a) (Left to right) Icosahedron as base polyhedron, its refinement, projection and data assignment. (b) Data at the centroid (red), vertices (blue) and edge midpoints (yellow) of the triangles are associated with (c) triangular, hexagonal and quadrilateral cells respectively.

at the (1) centroids, (2) vertices and (3) edge midpoints of the triangles (Fig. 1b). This results in additional types of cells each sampling data at their respective centroids; the centroids of the triangles are associated with the primal cells, the vertices of the triangles are the centroids of the dual hexagonal cells, and the edge midpoints of the triangles are the centroids of quad cells. Each quad cell is obtained by connecting the edge vertices with the centroids of two adjacent triangles (Fig. 1c).

Vertex information for all cell types in ICON is stored as lists of individual polygons without their adjacency information; this is often referred to as a 'polygon soup' representation. Data sampled at the cell centroids are stored according to the index of cell centroids. This property makes ICON an unstructured grid where the neighborhood of a specific vertex cannot be found explicitly (Figure 2). The soup representation of ICON makes convolution — a vital operation in multiresolution analysis — prohibitively expensive. The main reason behind this is the absence of neighborhood information which necessitates a search in the entire polygon soup for the neighbouring vertices that contribute to the convolution result at a particular vertex. In order to circumvent this problem, a structured grid-like representation can be used whereby the neighbourhood of a vertex can be efficiently determined via simple index calculations. This representation also facilitates straightforward dyadic subsampling of the grid, which needs to access alternate rows and columns of a grid. Therefore, a conversion technique is needed to transform ICON grid into a structured form whereby the connectivity information of a vertex can be easily obtained.

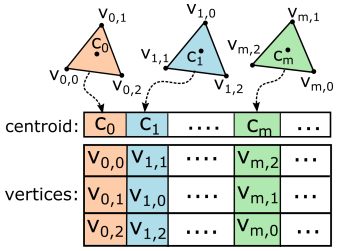


Figure 2: Example of a triangle soup with three arbitrary triangles in a mesh. Triangle with centroid c_m has vertex information $v_{m,0}$,$v_{m,1}$ and $v_{m,2}$ but the connectivity between triangles is not provided in the soup.

3. Related Work

In order to compress and visualize ICON data, the two main ingredients are an efficient data structure, and a wavelet that operates on that data structure. Each component is reviewed separately below.

3.1. Mappings and Data Structures

The problem of sampling functions defined on a sphere are frequently encountered in computer graphics. Environment mapping [Gre86] is a typical example. In its most basic form, an environment map is a 2D texture obtained from a latitude/longitude parametrization of the sphere. This representation however suffers from severe texture compression near the poles. Cube maps that use a cube as the underlying spherical polyhedron, are a popular alternative [Gre86]. This mapping stores the cube faces in six 2D textures and therefore provides a memory friendly Cartesian structure that expedites routine operations such as interpolation and filtering. Octahedron maps provide a more uniform discretization of the sphere, and use a single 2D texture to unfold and store the eight faces of the octahedron. They have been explored both in computer graphics [ED08] and Earth sciences [Dut].

The idea of cube maps and octahedron maps can be generalized to an atlas of connectivity maps (ACM) [MAS14]. ACM stores the vertex/face information of semiregular meshes in an atlas of 2D maps that can be indexed using a 2D coordinate system. Applying ACM to a spherical icosahedron results in diamond shape connectivity maps [MAS14]. In this paper, we apply the concept behind ACM to ICON data and show how to map the connectivity information of all cell types (triangles, quads and hexagons) to icosahedral maps that consist of 2D diamonds (Figure 5) where each diamond has a hexagonal grid associated with it.

3.2. Multiresolution

A visualization application used in the geosciences that supports both, LoD rendering as well as reading wavelet compressed data sets is VAPOR [CR05, CMNR07]. VAPOR operates on rectangular grids, but not irregular grids such as ICON. Separable 3D wavelets that are tensor product extensions of 1D wavelets are also commonly used to compress and render volumetric data that are sampled on 3D Cartesian grids [RGG*12].

Since ICON data has a semi-regular hexagonal structure, a non-separable multiresolution scheme is needed. We can take inspiration from the field of geometry processing where non-separable subdivision and reverse subdivision processes are typically used to represent models at different scales. The filters for these processes are usually obtained via a *geometry-based* approach to multiresolution [SDS]. In this context, loop subdivision and biorthognal loop subdivision wavelets [Ber04] are relevant as they are designed to work with triangular meshes with hexagonal connectivity. Another option is a wavelet based on the $\sqrt{3}$ (1-to-3 refinement) subdivision scheme [WQS07] which treats the primal and dual cells together. Both these approaches however do not take into consideration, the continuous nature of the data being approximated.

Consequently, we take a *function-based* approach to multiresolution [SDS]. This approach is commonly used in signal and image processing and centers on the idea of a sequence of nested function spaces that approximate a given function at different scales. The spaces are generated by a scaling function and one more more wavelets, and their orthogonality relationships are used to obtain discrete filters for the analysis (decomposition) and synthesis (reconstruction) tasks.

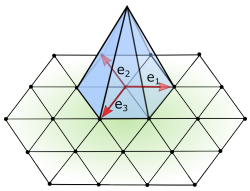


Figure 3: The linear box spline with its compact support over a hexagonal lattice. It takes a value of 1 at its center and linearly falls off to 0 at the six neighboring lattice sites.

Interpolation is a key step in many rendering and visualization tasks; when data resides at the vertices of triangles, the rendering pipeline in modern GPUs commonly performs barycentric interpolation. On the hexagonal lattice, barycentric interpolation is tantamount to applying a linear hexagonal box-spline kernel to the data values [CVDV06]. We are therefore interested in a wavelet that is associated with the hexagonal lattice and makes use of the linear hexagonal box-spline kernel as the scaling function. At the same time, we would like the decomposition and reconstruction filters to be compact so that the data can be processed efficiently. Towards this end, the hexagonal wavelet bases proposed by Cohen and Schlenker [CS93] offer a simple and viable solution. These biorthogonal wavelet bases are based on the linear box spline and are compactly supported, thus allowing perfect reconstruction using compact analysis and synthesis filters.

4. Multiresolution Scheme

We only review the essential mathematical ideas behind the proposed multiresolution scheme. The interested reader is referred to the original paper [CS93] for construction details. Another useful resource on hexagonal image processing is the book by Middleton [MS].

A 2D hexagonal lattice is generated by the matrix $\mathbf{L}=[\mathbf{e}_1\,\mathbf{e}_2]$ where $\mathbf{e}_1=[1 \quad 0]^T$ and $\mathbf{e}_2=[-1/2 \quad \sqrt{3}/2]^T$ are the basis vectors that span the lattice, i.e. any point on the lattice is given by the matrix-vector product $\mathbf{Lk}$ where $\mathbf{k}$ is a 2D integer vector that defines an integer coordinate system on the lattice. The lattice yields a regular triangle mesh whose vertices correspond to the lattice points (Figure 3).

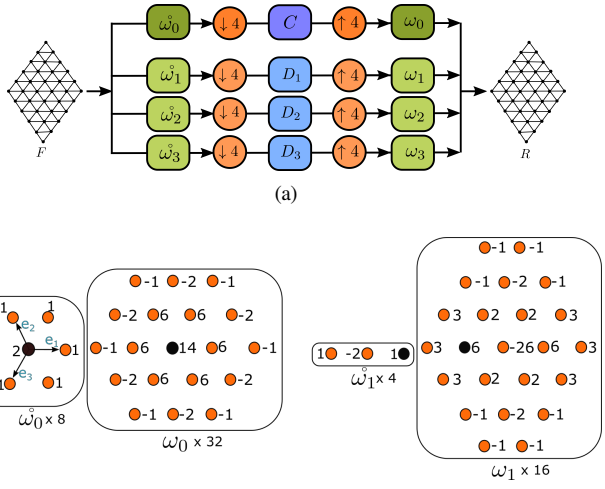
Let $F[\mathbf{k}]$ denote the sample value of a function at the lattice site indexed by $\mathbf{k}$. A linear approximation of the function is then given by

$$f(\mathbf{x})=\sum_{\mathbf{k}} F[\mathbf{k}]\mathbf{x}-\mathbf{Lk}), \tag{1}$$

where $\varphi(\mathbf{x})$ is the linear box spline (Fig. Figure 3). This is equivalent to a barycentric interpolation of the triangular mesh from the vertex values $F[\mathbf{k}]$. In the context of a dyadic multiresolution analysis, $f(\mathbf{x})$ lies in the fine-scale space V_0 that is spanned by the lattice shifts of the generating function φ. In order to approximate $f(\mathbf{x})$ at a coarser scale, one seeks the mutually orthogonal spaces V_1, W_1^1, W_1^2 and W_1^3 where $V_1 \subset V_0$ and contains the coarse scale approximation, and the spaces W_1^1, W_1^2 and W_1^3 contain the residual details so that $V_0 = V_1 \oplus W_1^1 \oplus W_1^2 \oplus W_1^3$. The coarse-to-fine reconstruction process can be written as

$$f(\mathbf{x})=\underbrace{\sum_{\mathbf{k}} C[\mathbf{k}]\varphi(\mathbf{x}/2-\mathbf{Lk})}_{\in V_0}+\sum_{i=1}^{3}\underbrace{\sum_{\mathbf{k}} D_i[\mathbf{k}]\psi_i(\mathbf{x}/2-\mathbf{Lk})}_{\in W_1^i}. \tag{2}$$

Here, ψ_i ($i \in \{1,2,3\}$) is a wavelet function that is related to φ according



(a) (b)

Figure 4: (a) Four channel subband coding scheme, ↓ represents a dyadic downsampling operation along the two lattice directions, and ↑ represents an analogous upsampling operation. (b) The low-pass analysis and synthesis filters $\mathring{\omega}_0$ and ω_0, and the high-pass analysis and synthesis filters $\mathring{\omega}_1$ and ω_1. The analysis high-pass filters $\mathring{\omega}_2$ and $\mathring{\omega}_3$ are obtained by successive $2\pi/3$ rotations of the analysis filter $\mathring{\omega}_1$. Similarly, the synthesis filters ω_2 and ω_3 are successive $2\pi/3$ rotations of ω_1.

to

$$\psi_i(\mathbf{x}/2) = \sum_{\mathbf{k}} \omega_i[\mathbf{k}]\varphi(\mathbf{x} - \mathbf{L}\mathbf{k}), \tag{3}$$

where ω_i is a synthesis wavelet filter. Similarly, φ also satisfies a two-scale relationship given by

$$\varphi(\mathbf{x}/2) = \sum_{\mathbf{k}} \omega_0[\mathbf{k}]\varphi(\mathbf{x} - \mathbf{L}\mathbf{k}), \tag{4}$$

where ω_0 is known as the synthesis scaling filter. The approximation and detail coefficients C and D_i in (2) are obtained through a decomposition process that takes the form of a discrete convolution, i.e.

$$\begin{aligned} C[\mathbf{k}] &= (F * \mathring{\omega}_0)[2\mathbf{k}], \\ D_i[\mathbf{k}] &= (F * \mathring{\omega}_i)[2\mathbf{k}], (i \in \{1,2,3\}) \end{aligned} \tag{5}$$

where $\mathring{\omega}_i$ ($i \in \{0,1,2,3\}$) are the analysis filters. The decomposition and reconstruction processes can be succinctly expressed as a subband coding scheme as shown in Fig. 4a. Compactly supported analysis and synthesis filters that achieve perfect reconstruction are shown in Fig. 4b. The scheme can be reapplied to the coarse-scale coefficients to obtain additional levels of detail. This gives rise to a discrete hexagonal wavelet transform (DHWT). Observe that the high pass analysis filters $\mathring{\omega}_1$, $\mathring{\omega}_2$ and $\mathring{\omega}_3$ compute directional derivatives in the three directions principle directions of the mesh (pictured in Fig. 4b).

The application of this multiresolution scheme to data sampled at the triangle vertices in ICON is straightforward. Other cells can also be handled provided that they can be converted into a hexagonal triangle mesh. This conversion is outlined in more detail in the following section.

5. Icosahedral Maps

Icosahedral maps convert unstructured vertex information into a structured grid so that the DHWT outlined in the previous section can be applied in an efficient manner. In order to perform this conversion

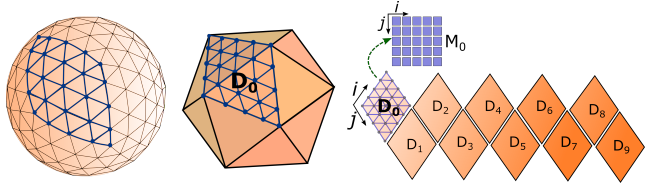
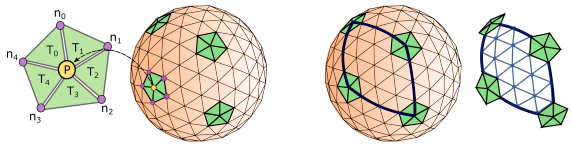
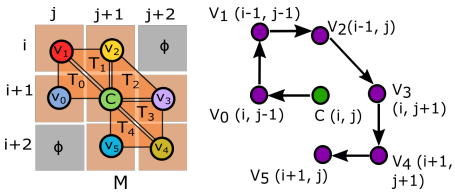


Figure 5: (Left) The Earth's surface unfolded into a net consisting of ten diamonds; each diamaond refers to a paired face of the base icosahedron. (Right) The vertex information of each diamond is stored in a rectangular 2D grid that corresponds to a hexagonal lattice associated with the diamond.



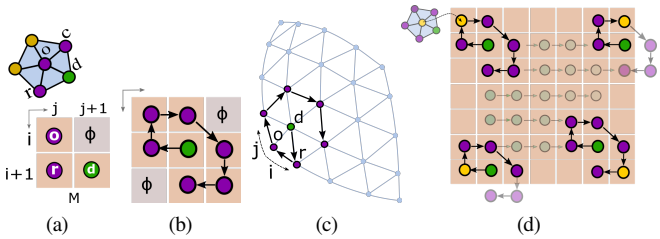
(a) (b)

Figure 6: (a) An extraordinary vertex P with its five neighboring vertices $n_0,\ldots,n_4$. (b) A diamond has four extraordinary vertices at its corners.



(a) (b)

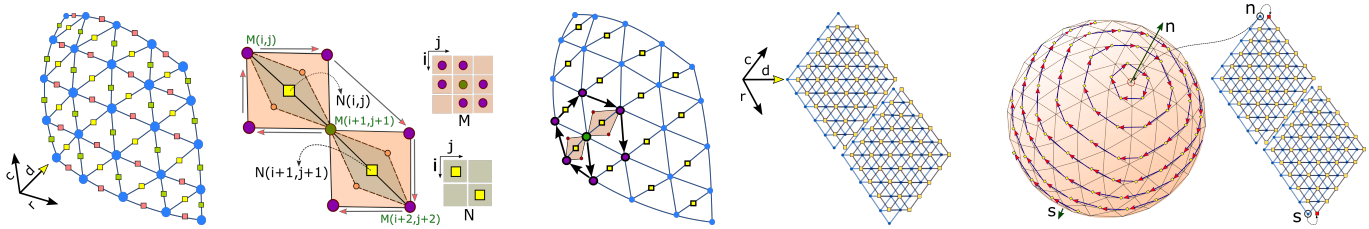
Figure 7: (a) A hexagonal fan around a vertex. (b) The order in which vertices are placed into the grid.



(a) (b) (c) (d)

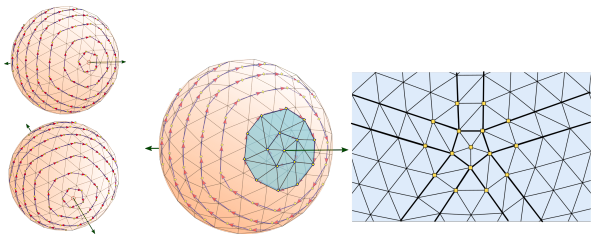
Figure 8: (a) Storing o, r and d from a pentgon to initialize array M. (b) Hexagonal fan placed at the starting vertex, and (c) the corresponding mesh vertices. (d) The fan is swept in a row-major order until the extraordinary vertices at the corners are found.

in the context of ICON, we need a technique with the following two key features: (1) it takes a polygon soup of cells as input and finds the explicit vertex connectivity information, and (2) converts the three types of cells into regular triangle meshes, where data is located at triangle vertices. Our proposed technique is an extention of ACM which unfolds the Earth's surface into ten diamonds each consisting of a pair of adjacent triangular faces of the base icosahedron (Figure 5). Vertex information on a diamond is stored in a corresponding 2D grid which is indexed using the coordinate system associated with the hexagonal lattice. This allows us to easily perform vertex neighbour lookups. The application of ACM to the triangle vertices in ICON is straightforward.



(a) (b) (c) (d) (e)

Figure 9: (a) Edge midpoints along three lattice directions (b) Modified form of hexagonal fan to map edge midpoints along the diagonal direction d, and (c) the corresponding mesh vertices. (d) Resultant shifted grids for two adjacent diamonds along the antipodal axis. (e) The edge midpoints viewed as a rotation of the primal grid, the polar vertices are obtained from the primal cells.



(a) (b)

Figure 10: (a) Earth's surface with edge midpoints along two different directions. (b) Triangulation obtained from the edge midpoints in one direction.

In our case, we need to handle all three cell types in a unified way. We therefore propose to convert the cells into a hexagonal lattice representation and store the vertex information and associated data in an unfolded net consisting of the ten diamonds of the base icosahedron. We call these representations *icosahedral maps*. Our conversion process consists of the following key steps: (1) finding extraordinary vertices, (2) retrieving vertex connectivity from the polygon soup using a *hexagonal fan* traversal scheme, and (3) storing the vertex information in icosahedral maps. Each of these steps is explained in more detail below.

Finding Extraordinary Vertices: Detecting twelve extraordinary vertices on the discretized Earth's surface is the initial step in unfolding the Earth surface into diamonds. After unfolding, each diamond has four extraordinary vertices at its four corners (Fig. 6b). Finding extraordinary vertices depends on the way the data is stored. In the context of ICON, the extraordinary vertices have five cell vertices (a pentagon) in the soup while the ordinary ones have six. Therefore, a vertex which is shared by five triangles in the triangle soup is taken to be an extraordinary vertex (Fig. 6a). Using this criteria, we can find twelve vertices on the Earth's surface along with their neighbors.

Hexagonal Fan: Given a triangle from the triangle soup, we use a hexagonal traversal scheme to find its neighbors and store the vertex information in a 2D grid. This traversal scheme allows us to map the connectivity of all cell types in ICON in a consistent manner. Figure 7 explains the details of our traversal method. Let C, V_0 and V_1 be the vertices of the provided triangle T_0 which belongs to the soup. Also, let M be a 2D array indexed by the coordinates (i,j) where i is the row index and j is the column index starting from $(0,0)$ at the upper left corner. Initially, C, V_0 and V_1 are stored in M at the locations $(i+1,j+1)$, $(i+1,j)$ and (i,j) respectively. Next, we search through the soup to find an adjacent triangle T_1 which shares the vertices C and V_1 with T_0. The remaining vertex V_2 of T_1 is stored at $M(i,j+1)$. In the next step, triangle T_2 is found from the soup that shares two vertices C and V_2 of T_2. As a result, the remaining vertex V_3 of T_2 is stored at $M(i,j+2)$. In a similar way V_4 and V_5 can be found from the triangles T_3 and T_4. Note that, $M(i+2,j)$ and $M(i,j+2)$ remain empty (indicated by ϕ in the figure). In this fashion, the connectivity information of the hexagonal neighbors around C is found by placing a hexagonal fan at C; each face of the fan is a triangle as shown in Figure 7.

Mapping Centroids of Hexagons: As explained earlier, the centroids of the hexagons reside at the vertices of the triangles. Mapping the hexagon centroids is straightforward using a hexagonal fan traversal. Figure 8 explains the main steps of the mapping procedure. An extraordinary vertex o is chosen and placed at the origin of an array M. The associated neighbors r and d that form a triangle are stored at $M(1,0)$ and $M(1,1)$ respectively (Fig. 8a). We then place a hexagonal fan at vertex d ($M(1,1)$) to determine the neighbors of d. The hexagonal fan then sweeps the grid horizontally filling up vertex information from the corresponding diamond until the extraordinary vertex at the top right corner is reached. At this point, sweeping proceeds in a row major order until two other extraordinary vertices are found. In a similar manner, the entire Earth can be unfolded into ten diamond shaped triangular grids as shown in Figure 5.

Mapping Centroids of Quads: The centroids of the quads are located at the edge midpoints of the primal cells in ICON. The edge midpoints can be found in three directions along the three edges of the triangles in the primal grid. Figure 9 shows three edge midpoints (indicated by red, green and yellow squares in Fig. 9a) in the directions r, c and d respectively. The edge midpoints can be mapped by extending our hexagonal fan traversal method used to map the hexagon centroids (triangle vertices). The four vertices of each quad are stored as a quad soup in ICON, where two of the vertices correspond to triangle edge vertices, and the other come from the centroids of two adjacent triangles. As we are sweeping the hexagonal fan to determine the connectivity of the triangle vertices, an edge can easily be found and used to find the edge's midpoint by searching in the quad soup for the corresponding quad. Fig. 9b illustrates this concept. The pictured case shows the edge midpoints along the diagonal direction d; N is the corresponding array that holds vertex information associated with the edge midpoints along d. At each iteration of the hexagonal fan, $N(i,j)$ and $N(i+1,j+1)$, the midpoints of the diagonal edges $\{M(i,j),M(i+1,j+1)\}$ and $\{M(i+1,j+1),M(i+2,j+2)\}$ respectively, are filled up. Here, M is the 2D array that stores the primal cells. This process generates a shifted version of the primal triangular grid with missing vertices at the two poles. The shifted grid can also

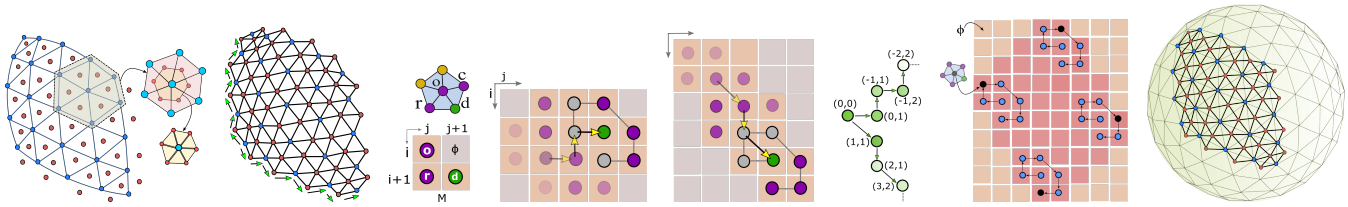


Figure 11: (a) Splitting hexagons into triangles. (b) The resulting refined hexagonal grid; the red and blue circles are the centroids and vertices of the primal cells. (c) Initializing array M with neighbors of the extraordinary vertices in the refined grid. (d and e) The first few steps of the modified sweeping scheme along the two directions. (f) The two sweeping directions shown together. (g) Empty (ϕ) entries in M and conditions to test for extraordinary vertices at the four corners of a diamond. (h) The corresponding mesh vertices exhibit a zigzag pattern on the diamond boundaries.

be seen as a rotation of the base icosahedron about an antipodal axis (Fig. 9e). The polar vertices are filled by placing the actual vertices of the triangle from the primal grid. Data at the two poles are determined by averaging neighboring data. The midpoints corresponding to the other edge directions can be mapped in a similar manner. As a result, we get three unfolded icosahedral maps corresponding to the three directional edge midpoints, each direction corresponds to a rotation about a distinct antipodal axis as shown in Fig. 10a. Fig. 10b shows the triangulation obtained from the edge midpoints in one direction. The triangles along the diamaond boundaries experience a slight distortion but the overall topology is preserved.

Mapping Centroids of Triangles: Centroids of the triangles are situated at the vertices of the voronoi cells (hexagons) of the primal grid. The voronoi cell information is stored as a hexagonal cell soup in ICON. Our mapping technique for the triangle centroids is illustrated in Figure 11. First, we split the hexagons into triangles by connecting each vertex of a hexagon to its centroid. This creates refined triangles which can also be seen as the result of applying a 1-to-3 refinement to the triangular cells. Consequently, we obtain a finer hexagonal lattice as show in Fig. 11b. This lattice is a rotated and scaled version of the primal lattice. As data is only stored at the vertices of the voronoi cells (red circles in Fig. 11b), data at the centroids of the voronoi cells (blue circles) are obtained by averaging neighboring data. Capturing the connectivity information of these vertices is similar to the procedure of mapping centroids of the hexagons, except this time the sweeping direction of the hexagonal fan needs to be adapted to the geometry of the finer hexagonal lattice. Instead of sweeping the fan horizontally and vertically, we modify the direction at each step while scanning (Fig. 11d and Fig. 11e). The horizontal direction is changed by adding $(0,1)$ for every odd step and by adding $(-1,0)$ for every even step. On the other hand, vertically the direction is changed by adding $(1,1)$ for every odd step and by adding $(1,0)$ for every even step. The resultant array M in this case has empty (ϕ) entries at the four corners (Fig. 11g) and the diamond has a zigzag pattern at its border (Fig. 11g). We also need a stopping criterion to terminate the sweep. This is provided by the extraordinary vertices as shown in Fig. 11h. It should be noted that due to the hexagonal fan traversal, the topology of the resulting 2D grid is consistent with the previous two cases. Hence, the DHWT can be applied without any modifications.

6. Results and Discussion

We mapped the connectivity information of different cell types in several ICON datasets to icosahedral maps. Owing to the simple rectangular indexing scheme associated with the 2D grids in an icosahedral map, the convolution, upsampling and downsampling operations associated with the DHWT (Figure 4) are equivalent to the corresponding operations associated with the 2D discrete wavelet transform conventionally used in image processing.

Since an icosahedral map consists of ten 2D grids, we need a way to handle boundary conditions in a consistent manner when applying the filters. In our implementation, we take advantage of our hexagonal fan to extract data at the border of each diamond from its neighboring diamonds. This data is used to pad the 2D arrays corresponding to the diamonds. The padding size is determined by the support size of the filter with the widest extent (3 in our case corresponding to ω_1). For the triangle centroids, the hexagonal fan is run along the zig-zag border of a diamond to extract data from neighbouring diamonds. This process ensures that we use actual data at the boundaries and avoids artefacts due to a sudden truncation.

To visualize scalar data on the Earth's surface, we performed a simple color mapping of normalized vertex data and used barycentric interpolation when rendering triangular faces. We experimented with ICON data as well as synthetic texture data to observe the behaviour of the proposed wavelet for different cell types.

Mapping and compression results: Figure 12 shows the results of applying icosahedral maps and the DHWT on different cell types. In order to observe how the data sets respond to compression, we applied quantile thresholding to the details from the first decomposition level. Only those details whose magnitudes fall within a user specified percentile are retained, and the remaining are set to 0. In the reported results, a threshold of n means that the bottom n percent are discarded and the top $(100-n)$ percent are retained. Peak signal-to-noise ratio (PSNR) is used to quantify the effect of thresholding. From Figure 12, we can observe that the wavelet responds very well to compression. This can be attributed to the smooth nature of the datasets which is evident in Figure 13; the PSNR values fall gradually with increasing threshold and we achieve visually acceptable results even with an aggressive threshold level.

Comparison between different cell-types: The quality of compression depends on the nature of the data being compressed. In order to observe how the wavelet behaves in the presence of non-smooth data, we performed a texture mapping experiment where the latitude/longitude coordinates of ICON vertices were used as texture coordinates to map a checkerboard texture to the sphere. This texture has sharp lines which experience some smoothing when mapped to ICON cells. Figure 14 shows the results for one diamond and for different cell types.

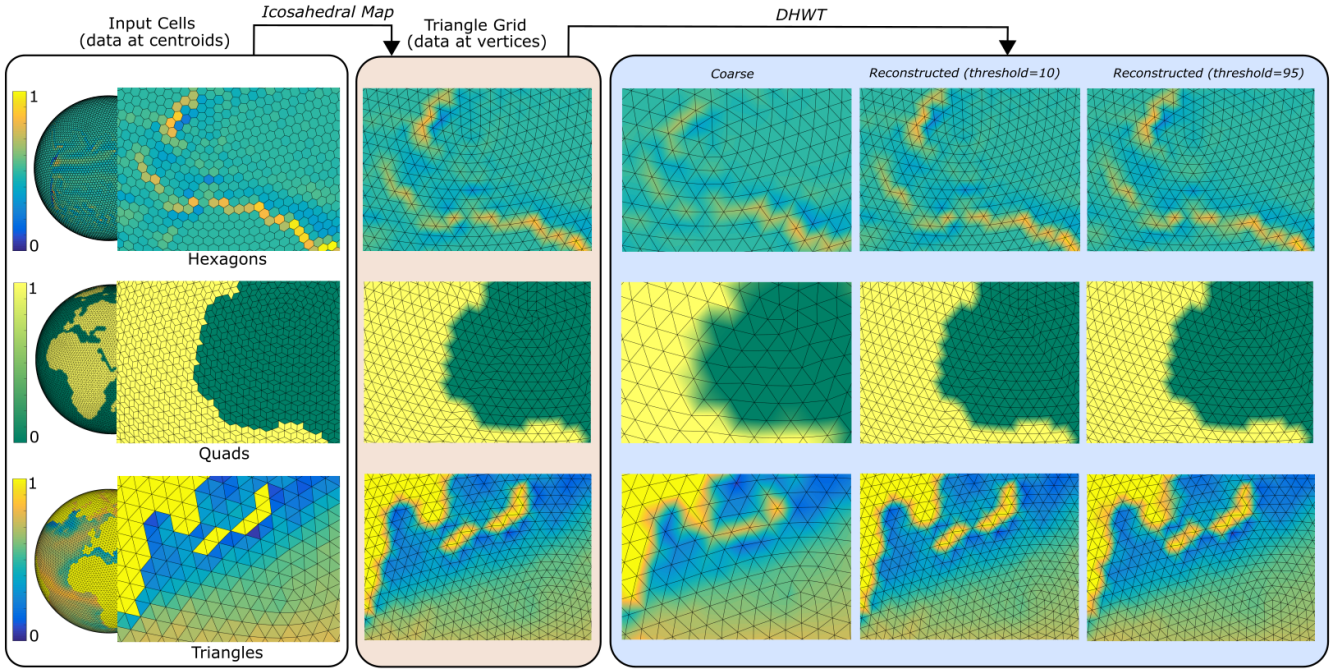
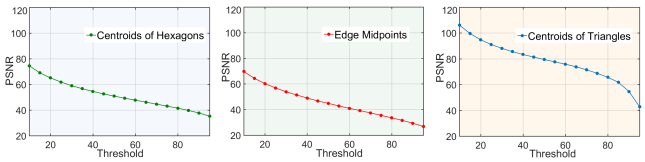


Figure 12: Results of applying icosahedral maps and the DHWT on ICON data at the centroids of hexagons (top), edge midpoints in one direction (middle), and the centroids of triangles (bottom). The grid resolutions per diamond from top to bottom are 65×65, 65×65 and 99×67 respectively. Each row (left to right) illustrates the original cell-type in ICON (data at centroids), result of converting to a triangular grid (data at vertices) via the icosahedral map, applying DHWT on the converted grid to obtain coarse data, and reconstructed data using two different thresholds: 10 and 95. The figure also focuses on the borders of the diamonds around an extraordinary vertex to show the smoothness along boundaries due to padding.



(a) (b) (c)

Figure 13: PSNR vs. threshold graphs for three different scalar ICON variables sampled at: (a) centroids of hexagons, (b) edge midpoints in one direction, and (c) centroids of triangles.

Observe that the texture is better preserved on the triangle centroids as compared to the hexagon centroids or the edge mid-points in one direction. This is to be expected since the triangle centroids are twice as dense compared to the hexagon centroids. Using our icosahedral maps, the hexagon centroids and the edge midpoints in one direction yield similar triangulations which are just shifted versions of each other. This explains why these two types of vertices behave similarly.
Additionally, when an aggressive threshold value is chosen, we can observe more pronounced compression artefacts in regions where the texture undergoes a sharp transition. The PSNR-vs-threshold curves in Fig. 14d suggest that the triangle centroids outperform the other cell types. However, all three experience a sharp deterioration for high threshold values.

7. Conclusion

We proposed icosahedral maps, data structures that are ideally suited to store the connectivity information of different cell configurations in the ICON model. Our icosahedral maps convert the cell geometry to a common hexagonal representation irrespective of the cell type. This is achieved via a hexagonal fan sweeping scheme that is adapted to the geometry of the different cell types. We further demonsrated how to apply a hexagonal wavelet scheme to the icosahedral maps in order to render scalar data at a coarser resolution or to compress it via thresholding.

Even though our tests have been limited to low resolution models, the methodology presented in this paper can be easily extended to high resolution models. Since the icosahedral maps consist of simple 2D grids, bricking schemes commonly employed in rendering high resolution volumetric data [BHP15] are directly applicable. The maps/bricks can be stored in the texture memory of modern GPUs. This paves the way for visibility-driven and out-of-core rendering strategies on the GPU.

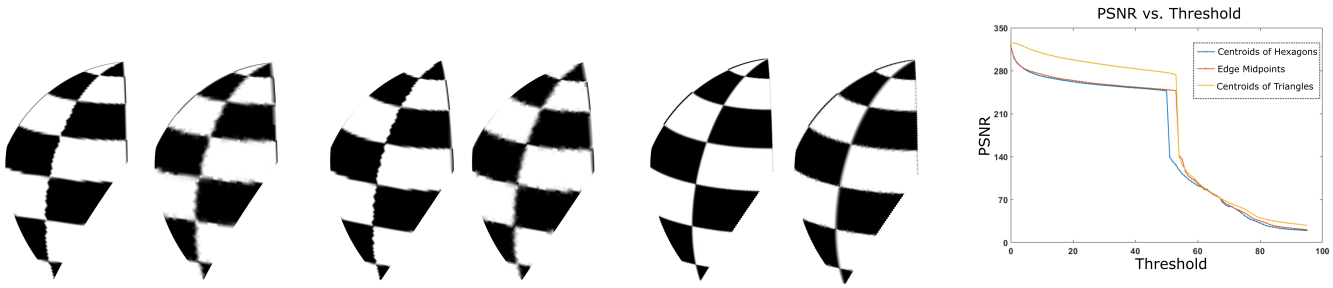
In future, besides conducting a more thorough evaluation with the domain experts, we would like to explore the scalability of our icosahedral maps. We are also interested in reducing the footprint of the maps associated with the triangle centroids. The use of non-dyadic wavelets in this context is also a topic that deserves further attention.

References

[AGL05] AHRENS J., GEVECI B., LAW C.: *Visualization Handbook*. Elsevier, 2005, ch. ParaView: An End-User Tool for Large Data Visualization. 2

[Aya15] AYACHIT U.: *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2015. 2

[Ber04] BERTRAM M.: Biorthogonal loop-subdivision wavelets. *Computing 72*, 1 (2004), 29–39. 3



(a) (b) (c) (d)

Figure 14: Result of applying the DHWT on a checkerboard texture sampled on one diamond: (a) centroids of hexagons, (b) edge midpoints in one direction, and (c) centroids of triangles. For each cell type, visual comparison between the original (left) and compressed with threshold of 95 (right) is presented. (d) PSNR vs. threshold graph for the three cell types.

[BHP15] BEYER J., HADWIGER M., PFISTER H.: State-of-the-art in gpu-based large-scale volume visualization. *Computer Graphics Forum 34*, 8 (2015), 13–37. URL: http://dx.doi.org/10.1111/cgf.12605, doi:10.1111/cgf.12605. 7

[CBW*12] CHILDS H., BRUGGER E., WHITLOCK B., MEREDITH J., AHERN S., PUGMIRE D., BIAGAS K., MILLER M., HARRISON C., WEBER G. H., KRISHNAN H., FOGAL T., SANDERSON A., GARTH C., BETHEL E. W., CAMP D., RÜBEL O., DURANT M., FAVRE J. M., NAVRÁTIL P.: VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*. Taylor and Francis, Oct 2012, pp. 357–372. 2

[CMNR07] CLYNE J., MININNI P., NORTON A., RAST M.: Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation. *New Journal of Physics 9*, 8 (2007), 301. 3

[CR05] CLYNE J., RAST M.: A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations. In *Electronic Imaging 2005* (2005), International Society for Optics and Photonics, pp. 284–294. 3

[CS93] COHEN A., SCHLENKER J. M.: Compactly supported bidimensional wavelet bases with hexagonal symmetry. *Constructive Approximation 9*, 2 (1993), 209–236. 3

[CVDV06] CONDAT L., VAN DE VILLE D.: Three-directional box-splines: Characterization and efficient evaluation. *IEEE Signal Processing Letters 13*, 7 (2006), 417–420. 3

[DSH*15] DIPANKAR A., STEVENS B., HEINZE R., MOSELEY C., ZÄNGL G., GIORGETTA M., BRDAR S.: Large eddy simulation using the general circulation model icon. *Journal of Advances in Modeling Earth Systems 7*, 3 (2015), 963–986. 1

[Dut] DUTTON G.: *A Hierarchical Coordinate System for Geoprocessing and Cartography*. Lecture notes in earth sciences. 3

[ED08] ENGELHARDT T., DACHSBACHER C.: Octahedron environment maps. In *13th International Symposium on Vision, Modeling, and Visualization 2008, VMV 2008* (2008), pp. 383–388. 3

[Gre86] GREENE N.: Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications 6*, 11 (Nov 1986), 21–29. 3

[MAAS15] MAHDAVI-AMIRI A., ALDERSON T., SAMAVATI F.: A survey of digital earth. *Computers & Graphics 53, Part B* (2015), 95 – 117. 1

[MAS14] MAHDAVI-AMIRI A., SAMAVATI F.: Atlas of connectivity maps. *Computers & Graphics 39* (2014), 1 – 11. 2, 3

[MS] MIDDLETON L., SIVASWAMY J.: *Hexagonal Image Processing: A Practical Approach*. Advances in Computer Vision and Pattern Recognition. 3

[PJR*15] PETERSEN M. R., JACOBSEN D. W., RINGLER T. D., HECHT M. W., MALTRUD M. E.: Evaluation of the arbitrary lagrangian-eulerian vertical coordinate method in the mpas-ocean model. *Ocean Modelling 86* (2015), 93–113. 1

[PYX] PYXIS INNOVATION INC.: How PYXIS works? URL: http://www.pyxisinnovation.com/pyxwiki/index.php?title=How_PYXIS_Works. 1

[RGG*12] RODRÍGUEZ M. B., GOBBETTI E., GUITIÁN J. I., MAKHINYA M., MARTON F., PAJAROLA R., SUTER S.: State-of-the-Art in Compressed GPU-Based Direct Volume Rendering. *Computer Graphics Forum* (2012). 3

[RPH*13] RINGLER T., PETERSEN M., HIGDON R., JACOBSEN D., JONES P., MALTRUD M.: A multi-resolution approach to global ocean modeling. *Ocean Modelling 69* (2013), 211–232. 1

[Sah08] SAHR K.: Location coding on icosahedral aperture 3 hexagon discrete global grids. *Computers, Environment and Urban Systems 32*, 3 (2008), 174–187. 1

[SDS] STOLLNITZ E., DEROSE T., SALESIN D.: *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann series in computer graphics and geometric modeling. 3

[SM01] SNYDER J., MITCHELL D.: *Sampling-efficient mapping of spherical images*. Tech. rep., Microsoft Research, 2001. 1

[SMT*07] SATOH M., MATSUNO T., TOMITA H., MIURA H., NASUNO T., IGA S.: Nonhydrostatic icosahedral atmospheric model (nicam) for global cloud resolving simulations. *Computational Physics, special issue on Predicting Weather, Climate and Extreme events 227* (2007), 3486–3514. 1

[SNCB13] SARAH N. COLLINS ROBERT S JAMES P. R. K. C. A. L., BROWNLEE J.: Grids in numerical weather and climate models. In *Climate Change and Regional/Local Responses*, Ray D. P., (Ed.). InTech, 2013, ch. 4. doi:10.5772/55922. 2

[SWK03] SAHR K., WHITE D., KIMERLING A. J.: Geodesic discrete global grid systems. *Cartography and Geographic Information Science 30*, 2 (2003), 121–134. 1

[TBW*13] TONG X., BEN J., WANG Y., ZHANG Y., PEI T.: Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system. *International Journal of Geographical Information Science 27*, 5 (2013), 898–921. 1

[WKO92] WHITE D., KIMERLING J. A., OVERTON S. W.: Cartographic and geometric components of a global sampling design for environmental monitoring. *Cartography and Geographic Information Science 19*, 1 (1992), 5–22. 1

[WQS07] WANG H., QIN K., SUN H.: 8730;3-subdivision-based biorthogonal wavelets. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (Sept 2007), 914–925. 3

[ZRRB15] ZÄNGL G., REINERT D., RÍPODAS P., BALDAUF M.: The icon (icosahedral non-hydrostatic) modelling framework of dwd and mpi-m: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society 141*, 687 (2015), 563–579. 1