IMPLEMENTING SCENZ-GRID CELL FAMILIES

ALEXANDER RAICHEV

ABSTRACT. These notes describe SCENZ-Grid cell families in mathematical language and specify requirements for data structures implementing cell families and sets of cell families.

The New Zealand government lab Landcare Research is currently developing a software system for geospatial analysis and worldwide scientific collaboration called SCENZ-Grid. The SCE stands for Spatial Computation Engine/Science Collaboration Environment and the NZ stands for New Zealand, of course :-) When completed the SCENZ-Grid source code will be released under an open source license.

The SCENZ-Grid development team thus far comprises Robert Gibb (Landcare Research; team leader), Michael Speth (Landcare Research), Arno Leist (Landcare Research), and me (University of Auckland).

The next step in SCENZ-Grid's development is choosing compact data structures to implement cell families and sets of cell families. The goal of these notes is to describe in mathematical language what a cell family is (Section 1) and to specify requirements for implementing cell families and sets of cell families (Section 2).

1. BACKGROUND

1.1. **Projection.** For analyzing geospatial data SCENZ-Grid uses a projection of the surface of the Earth onto a plane. More specifically SCENZ-Grid projects the WGS84 ellipsoid model of the Earth onto a plane via the rHEALPix projection, a rearrangement of the HEALPix projection. The rHEALPix projection is an injection from longitude-latitude space $D := [-\pi, \pi) \times (-\pi/2, \pi/2) \cup \{(-\pi, \pi/2), (-\pi, \pi/2)\}$ to the plane $\mathbb{R}^2$, and its image is pictured in Figure 1.1(c)*. I omit the projection's mathematical formula, because it is not relevant to these notes. All angles below are measured in radians and all lengths in meters unless indicated otherwise.

1.2. **Grids.** To index regions of the Earth, SCENZ-Grid uses a set $\mathcal{G} = \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_L\}$ of nested partitions of the rHEALPix projection image. Here L is a fixed positive integer, which will probably be set to 21 at the end of SCENZ-Grid development. Each partition $\mathcal{G}_l$ is called a **grid**, and its elements are called **cells**. Each cell c has an **ID**, i.e. a name, denoted $\mathrm{ID}(c)$ that is a string in Γ, the set of all strings beginning with a character from the alphabet $\{N, O, P, Q, R, S\}$ and followed by zero or more characters from the alphabet $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. Oftentimes I will refer to cells by their IDs, and sometimes I will blur the distinction between numbers and strings when the context is clear.

The set $\mathcal{G}$ is defined recursively as follows. Let R denote the radius of the authalic sphere of the WGS84 ellipsoid, which is 6371007.18088 m. Let $\mathcal{G}_0$ be the set of 6 squares

Date: June 11, 2012.

Key words and phrases. SCENZ-Grid, HEALPix, rHEALPix, grid, cell, cell family.

* Actually, the rHEALPix projection pictured is just one of 16 possible rHEALPix projections. In general the (n, s)-rHEALPix projection is the projection obtained by combining the HEALPix north polar triangles onto triangle $n \in \{0, 1, 2, 3\}$ and combining the HEALPix south polar triangles onto triangle $s \in \{0, 1, 2, 3\}$.

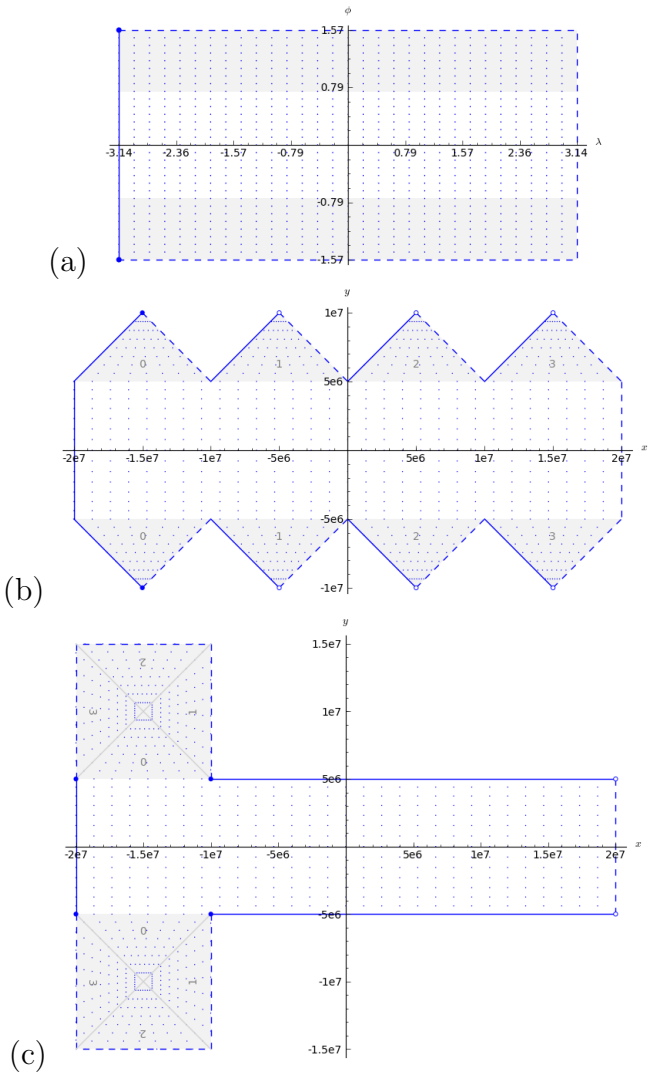

FIGURE 1.1. (a) Longitude-latitude coordinate space D with the polar region $|\phi| > \phi_0 := \arcsin(2/3)$ shaded (b) The image of D under the HEALPix projection with polar caps numbered (c) The image of D under the rHEALPix projection

of width $R\pi/2$ whose upper left vertices lie at $R(-\pi, 3\pi/4)$, $R(-\pi, \pi/4)$, $R(-\pi/2, \pi/4)$, $R(0, \pi/4)$, $R(\pi/2, \pi/4)$, and $R(-\pi, -\pi/4)$. Assign these cells the IDs N, O, P, Q, R, and S, respectively. For example, the level 0 grid $\mathcal{G}_0$ is pictured in Figure 1.2.

Because the rHEALPix projection image does not completely contain its boundary (see Figure 1.1(c)), the cells of $\mathcal{G}_0$ can not contain all their edges. Let us declare somewhat arbitrarily that cells N and S contain none of their edges and that cells O–R contain their top, left, and bottom edges. These subtleties of boundary are not relevant to these notes, though, and so I will ignore them.

Given the grid $\mathcal{G}_l$, let the grid $\mathcal{G}_{l+1}$ be the partition of the rHEALPix image that refines $\mathcal{G}_l$ as follows. Partition each cell $c \in \mathcal{G}_l$ into 9 squares of width $(R\pi/2)3^{-l-1}$, called the **children** of c, and assign these cells the IDs $s0, s1, \ldots, s8$ from left to right and top to bottom, where $s = \mathrm{ID}(c)$ and juxtaposition of strings denotes concatenation. See Figure 1.2(b). Let $\mathcal{G}_{l+1}$ be the set of all child cells described above for all cells in $\mathcal{G}_l$. For example, $\mathcal{G}_1$ is pictured in Figure 1.2(c).

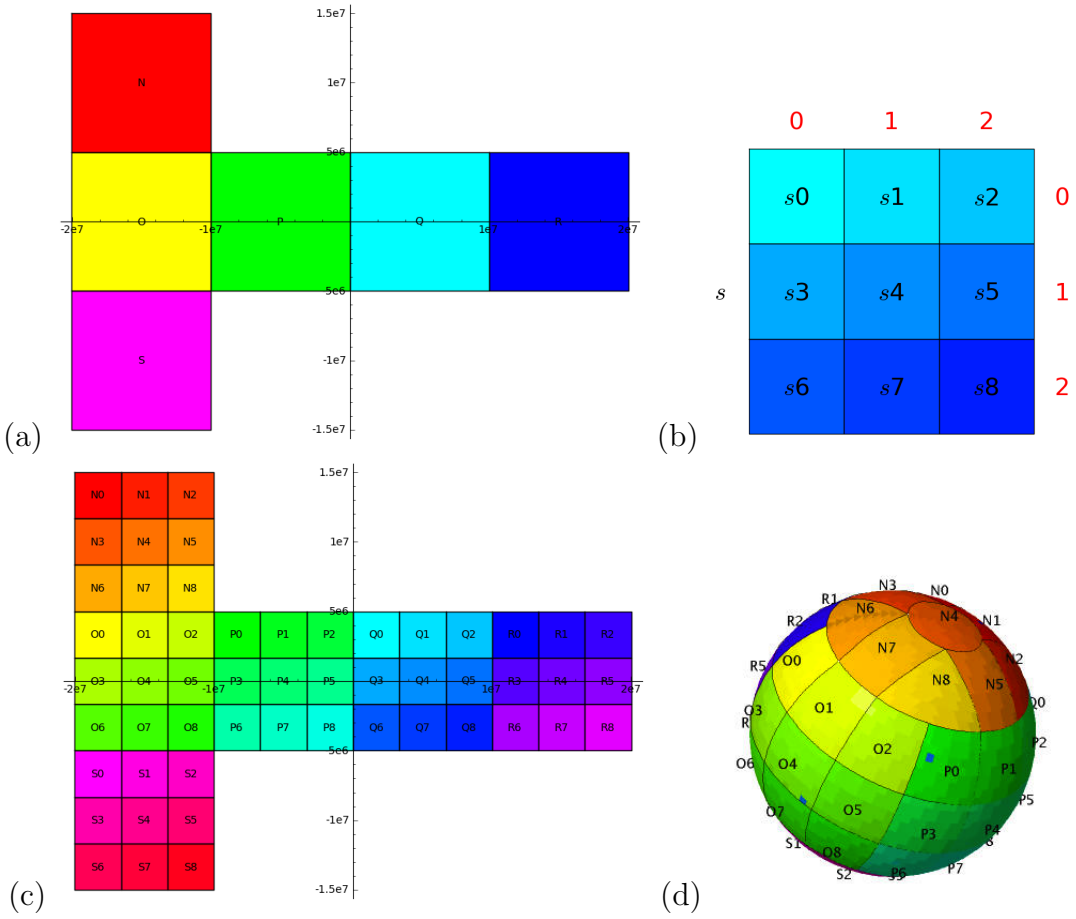

FIGURE 1.2. (a) $\mathcal{G}_0$ (b) A generic level l cell with ID s and its 9 level $l+1$ subcells; row and column numbers highlighted (c) $\mathcal{G}_1$ (d) $\mathcal{G}_1$ projected back onto the WGS84 ellipsoid

We call $\mathcal{G} := \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_L\}$ the **rHEALPix grid hierarchy**. This grid hierarchy naturally induces a grid hierarchy on the WGS84 ellipsoid via reverse projection, since these spaces are in bijective correspondence, and this induced grid hierarchy is pictured in Figure 1.2(d).

1.3. **Cell Order.** For some calculations it is convenient to linearize SCENZ-Grid's hierarchical grid structure via the following linear order on the set $\cup\mathcal{G}$ of all cells of all grids.

Define the binary relation $\trianglelefteq$ on the set of strings Γ via $s \trianglelefteq t$ iff

- t is a prefix of s or
- $s_k <_{\text{lex}} t_k$, where k is the least index at which s and t differ and where $<_{\text{lex}}$ is the lexicographic order (with $N <_{\text{lex}} < O <_{\text{lex}} \cdots <_{\text{lex}} S$ and $0 <_{\text{lex}} 1 <_{\text{lex}} \cdots <_{\text{lex}} 8$).

For example, we have $N \triangleleft P13 \triangleleft P1$, where $\triangleleft$ denotes strictly less than (and not equal to). It is straightforward to show that $\trianglelefteq$ is a linear order on Γ.

Now extend $\trianglelefteq$ to a linear order on the set of all cells $\cup\mathcal{G}$ by defining $c \trianglelefteq d$ for cells iff $\text{ID}(c) \trianglelefteq \text{ID}(d)$.

One way to understand $\trianglelefteq$ is to imagine all cell IDs organized into a rooted tree T of height $L+1$, where the root is unlabeled and has the six children $\{N, O, P, Q, R, S\}$ in lexicographic order, and each non-root node ID s has the nine children $s0, s1, \ldots, s8$ in lexicographic order. See Figure 1.3.

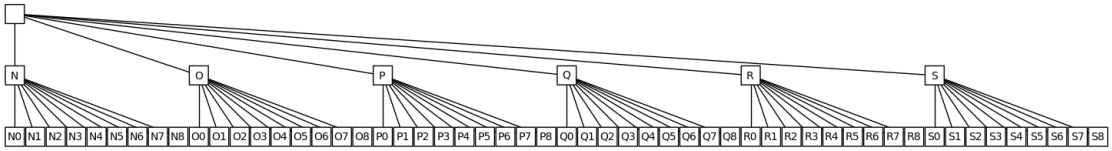

FIGURE 1.3. Part of the tree T of all cell IDs

With the tree T in mind, $c \trianglelefteq d$ for cells iff $\mathrm{ID}(c)$ lies to the left of or below $\mathrm{ID}(d)$ in T iff $\mathrm{ID}(c)$ comes before $\mathrm{ID}(d)$ in the postorder traversal of T.

For cells $c, d \in \cup\mathcal{G}$, let us say that c is a **subcell** of d if c is a subset of d. Notice that $c \subseteq d$ implies $c \trianglelefteq d$.

1.4. **Cell Families.** Given a set X of cells, let us call the union $\cup X \subset \mathbb{R}^2$ the **region** of X. Also, let us call X **subcell-free** if for every cell $c \in X$, no subcell of c lies in X. Finally, let us call a subcell-free set of cells a **cell family**. For example, $\{N, P13\}$ is a cell family but $\{N, P13, P1\}$ is not since $P13 \subset P1$ (as cells, not as strings).

Let us call a set of cells with region A **minimal** if no smaller (in cardinality) set of cells has region A. Notice that every minimal cell set is a cell family but not vice versa. Notice also that for every set X of cells there exists a unique minimal cell family with the same region as X. For example, the minimal cell family of the set comprising all level 1 cells along with the cell P is the cell family $\{\mathtt{N}, \mathtt{O}, \mathtt{P}, \mathtt{Q}, \mathtt{R}, \mathtt{S}\}$ of all the level 0 cells. Figure 1.4 displays another example.

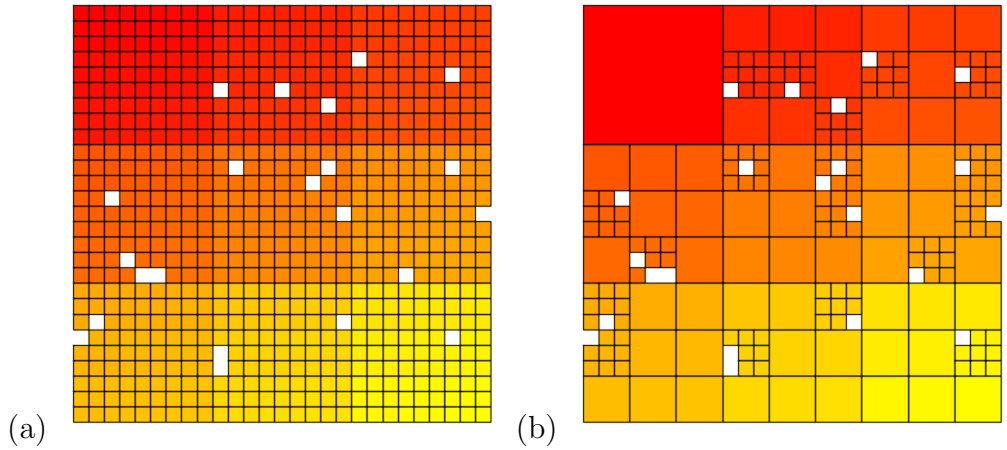

FIGURE 1.4. (a) A cell family with cell IDs omitted (b) The minimal cell family for (a)

Finding minimal cell families for the union and intersection of regions of cell sets are two common operations within SCENZ-Grid. For example, let $X = \{P0, P2, P4, P6, P8\}$ and $Y = \{P018, P1, P3, P5, P7\}$. Then the minimal cell family for the union $(\cup X) \cup (\cup Y)$ of the regions of X and Y is $\{P\}$, and the minimal cell family for the intersection $(\cup X) \cap (\cup Y)$ of the regions of X and Y is $\{P018\}$. Figure 1.4 displays another example.

Given a cell family F of n cells sorted in $\trianglelefteq$-order, the minimal cell family for F can be computed in $O(n)$ cell operations by scanning along F and replacing groups of cells with

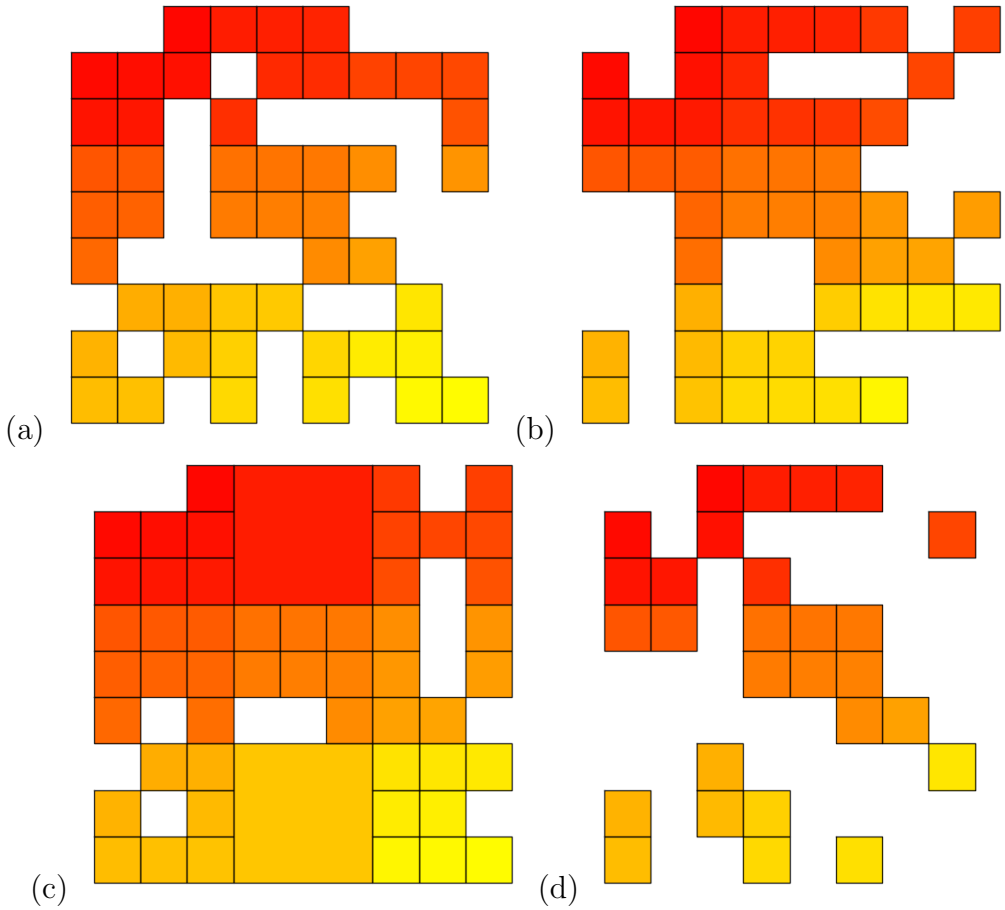

FIGURE 1.5. (a) A cell family within cell N with IDs omitted (b) Another cell family within cell N (c) The minimal cell family for the union of the regions in (a) and (b) (d) The minimal cell family for the intersection of the regions in (a) and (b)

their parents when possible. This is one reason for introducing an order such as $\trianglelefteq$ on all cells[†].

Given two cell families F and G with m and n elements already sorted in $\trianglelefteq$-order, the minimal cell family for the union of their regions can be computed in $O(\max\{m, n\})$ cell operations by scanning through both cell families in their entireties to compute the union of their regions and then minimizing the resulting cell family. The minimal cell family for the intersection of their regions can be computed in $O(\max\{m, n\})$ cell operations by scanning through both cell families to compute the intersection of their regions, stopping when the smaller cell family has been exhausted, and then minimizing the resulting cell family.

2. Implementation Requirements

2.1. **Cell Families.** SCENZ-Grid will store geographic data in cell families organized by attribute. For example, there will be a cell family containing all the cells in the system that have a soil pH attribute associated to them, a cell family containing all the cells in

[†] Alternatively, one could organize the cells of F into a tree beforehand and then compute the minimal family in $O(n)$ cell operations. Either way, the point is that taking advantage of the hierarchical grid in which the cells lie yields fast algorithms.

the system that have a soil type attribute associated to them, a cell family containing all the cells in the system that have a cation-exchange capacity attribute associated to them, and so on with possibly thousands of attributes. SCENZ-Grid cell family data will persist over the course SCENZ-Grid's lifetime and will grow slowly as each user adds new data to the system.

With all of the above in mind, the data structure used to represent an attribute cell family

- must take advantage of the tree-structure/$\trianglelefteq$-order structure of cell IDs so that searching for individual cells, computing minimal cell families, computing region unions, and computing region intersections is fast (linear or sublinear time);
- need not be dynamic, since additions to the data store will be relatively infrequent and can be done offline;
- must allow for easy parallelization since SCENZ-Grid will run on a cluster/grid/cloud architecture of commodity Intel servers.

2.2. **Sets of Cell Families.** Notice that there will be much repetition across attribute cell families. For example, cells that have had their soil pH measured probably also had their soil type and cation-exchange capacity measured. With this in mind, the data structure used to represent the set of all attribute cell families

- must take advantage of the highly repetitive nature of the data so that storage space is minimized;
- must not compromise the speed of the cell family operations listed above;
- need not be dynamic, since additions to the data store will be relatively infrequent and can be done offline;
- must allow for easy parallelization since SCENZ-Grid will run on a cluster/grid/cloud architecture of commodity Intel servers.

University of Auckland, Department of Computer Science, Private Bag 92019, Auckland, New Zealand

E-mail address: raichev@cs.auckland.ac.nz