CAMBRIDGE

Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

Part I

Grid Generation, Discretizations, and Solvers

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

1

Unstructured PEBI Grids Conforming to Lower-Dimensional Objects

RUNAR L. BERGE, ØYSTEIN S. KLEMETSDAL, AND KNUT-ANDREAS LIE

Abstract

The `upr` module in the MATLAB Reservoir Simulation Toolbox (MRST) can construct unstructured Voronoi grids that conform to polygonal boundaries and geometric constraints in arbitrary dimensions prescribed inside the reservoir volume. The resulting volumetric tessellations are usually realized as locally orthogonal, perpendicular bisector (PEBI) grids, in which cell faces can be aligned to accurately preserve objects of codimension one (curves in 2D and surfaces in 3D) and/or cell centroids can be set to follow curves in 2D or 3D. This enables you to accurately model faults, let grid cells follow horizontal and multilateral well paths, or create lower-dimensional or volumetric representations of fracture networks. The module offers methods for improving grid quality, like configurable policies for treating intersecting geometric objects and handling conflicts among constraints, methods for locating and removing conflicting generating points, as well as force-based and energy-minimization approaches for optimizing the grid cells. You can use `upr` to create a consistent hierarchy of grids that represent the reservoir volume, the constraining geometric objects (surfaces and curves), as well as their intersections. The hierarchy is built such that the cell faces of a given (sub)grid conform to the cells of all bounding subgrids of one dimension lower.

1.1 Introduction

The basic geometric description of a reservoir or aquifer will typically consist of multiple surfaces representing the top and base of the reservoir and the main bounding faults, as well as surfaces that represent internal structures such as depositional environments, lithological contrasts, minor faults, and fractures that restrict or guide the fluid flow. It is important to respect these geological boundaries in the computational grid to achieve accurate simulations, and the number of surfaces

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

to be respected tends to increase (steeply) as more details are added to the reservoir characterization. Until recently, identifying and tracing geological surfaces in processed seismic data has mainly been a manual process. If emerging automated interpretation approaches based on machine learning become more widespread, one should expect a significant growth in both the number and the complexity of the surfaces users will desire to incorporate in simulation models.

The MATLAB Reservoir Simulation Toolbox's (MRST) grid structure is very flexible and allows for completely unstructured topologies and general polyhedral cell geometries. (You can find a detailed discussion in chapter 3 of the MRST textbook [11].) The core module of MRST includes several functions for creating a wide variety of grids, from simple rectilinear meshes, via corner-point grids and unstructured simplex grids, to hybrid and multiblock grids. The grid structure puts few restrictions on the types of grids you can represent, but constructing very complex grids that adapt to outer and inner constraints on the reservoir geometry can be a tedious and complicated process using the basic functionality from the core module.

The main motivation for the `upr` module was to develop a family of relatively simple yet flexible methods for generating grids that adapt automatically to internal geometric structures that delineate structural, stratigraphic, sedimentological, or diagenetic heterogeneities. From a geological perspective, these structures are of (very) different natures. However, for grid-generation purposes, we divide the corresponding constraints into two groups: constraints for which the cell centroids should align with the geometric object and constraints for which the faces of the cells should align with the geometric object. Alignment of cell centroids is typically desirable to trace out the paths of deviated or horizontal wells. Wells are usually modeled by analytical or semi-analytical inflow performance relationships (see [11, subsection 11.7]), which in their basic form assume that the well is perforated at the center of the cell. One can also use the same functionality to trace out fractures that should be represented as volumetric objects. Boundary alignment is desirable to trace out faults and various forms of internal layering and zonation within the reservoir, as well as fractures that are to be represented as lower-dimensional objects in discrete fracture–matrix (DFM) models.

In the `upr` module, we use so-called clipped Voronoi diagrams [19] to create unstructured polyhedral and polygonal grids that allow two different types of conformity requirements: cell centroids tracing prescribed lines in 2D or 3D or cell faces tracing surfaces in 3D and lines in 2D. In the literature, the names "Voronoi mesh," "Voronoi diagram," and "perpendicular bisector (PEBI) grid" are all used to denote the same types of grids. The `upr` module uses PEBI as name convention, because this is most common in the petroleum industry. Table 1.1 lists the entry-level functions for generating grids using `upr`. The list is by no means exhaustive

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

Table 1.1 *Short overview of the entry-level functions in the* `upr` *module you can use to generate different types of PEBI grids. In the description, the word "sites" refers to generating points that control the Voronoi diagrams or the underlying Delaunay tessellations.*

Function name	Description
`pebiGrid2D`	Generate a 2D PEBI grid that conforms to internal constraints
`compositePebiGrid2D`	Similar to pebiGrid2D but with a structured background grid
`clippedPebi2D`	From a given set of sites, create a 2D clipped PEBI grid
`CPG2D`	From a given set of sites, optimize the site positions
`compositePebiGrid3D`	Generate a 3D PEBI grid that conforms to internal constraints
`mirroredPebi3D`	From a given set of sites, create a 3D PEBI grid
`CPG3D`	From a given set of sites, optimize the site positions

but offers an overview of the most important functions and an introductory point to `upr`. See also the overview in Figure 1.26 at the end of this chapter. The main purpose of this chapter is to give a basic introduction to these functions, briefly explain some of the underlying theory, and give several code-centric examples of how the functionality in the module can be used to generate complex grids that conform to geological objects and well paths. For a more comprehensive overview of alternative methods and previous research, the reader can consult [2, 6, 8, 13–15, 18].

We emphasize that the `upr` module is a research tool for constrained gridding and not a robust industry-grade geomodeling tool. You can use it to generate reservoir models with grid topologies and cell shapes that are representative of what one may encounter in complex geological descriptions of real-life problems. However, functions in the module use geometric algorithms and tolerances that primarily have been tested and adjusted for grids of unit size and close to unit aspect ratios. To create grids with more realistic dimensions and aspect ratios, we generally advise you to scale and translate inner and outer constraints to the unit domain in the first quadrant (or moderate multiples thereof) when creating the grid and then rescale and translate the grid back to the desired size and position afterward.

1.2 Basic Introduction to PEBI Grids

PEBI grids are closely related to Delaunay triangulations; in fact, the two are the dual of each other, as we will see. Both Delaunay and PEBI grids are uniquely defined (up to any degeneracy) by a set of generating points, or *sites* for short.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

For Delaunay, the sites, denoted by $\{\vec{s}_i\}_{i=1,\dots,n} = \mathcal{S}$, correspond to the vertices of the grid, whereas the sites are associated with cells for the PEBI grid.

In this section, we first introduce the Delaunay triangulation, describe the PEBI grid, and review some of its important properties. We also explain how you can construct simple unstructured grids using MRST and features from the `upr` module. This section hence gives the basic tools needed to construct conforming PEBI grids. You can find the complete source code for all examples in this section in the script `uprBookSection2.m` in the `examples/book-ii` directory of the `upr` module.

1.2.1 Delaunay Triangulation

There exist many different methods for creating the Delaunay triangulation given a set of sites; see, e.g., the textbook by Shewchuk et al. [17]. Discussing these methods is outside the scope of this chapter, but to make the discussion as self-contained as possible, we will nonetheless introduce some important properties that are useful when constructing the dual PEBI grid. Let us start by giving a precise definition of the Delaunay triangulation.

> **Theorem 1.1.** *Let $\mathcal{S}$ be a set of points. The Delaunay triangulation, denoted by $\mathcal{T}$, is a tessellation of the convex hull of $\mathcal{S}$ into simplices, such that the interior of the circumsphere of each simplex contains no sites from $\mathcal{S}$.*

This theorem does not present an obvious way to construct the Delaunay triangulation but gives a straightforward condition you can use to check whether a triangulation is Delaunay: Draw the circumsphere around each element in the triangulation and check that the spheres do not contain any sites from $\mathcal{S}$. The empty circumsphere principle is shown in Figure 1.1 for a Delaunay triangulation of six sites.

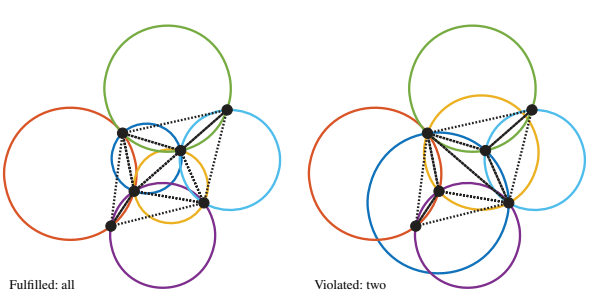


Figure 1.1 The empty circumsphere principle for Delaunay triangulations in 2D. The circumcircles of each of the six triangles shown in different colors should not contain any vertices in their interior. (Not fulfilled for the dark blue/yellow circles in the right plot.)

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

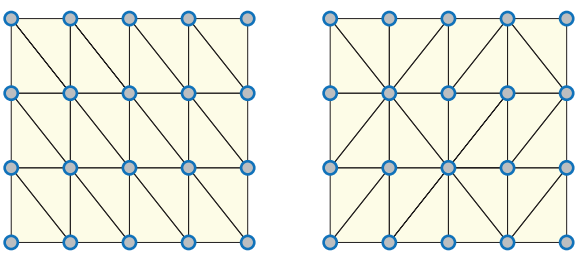
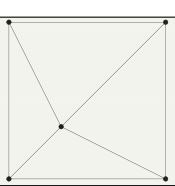


Figure 1.2 Two valid Delaunay triangulations of a Cartesian point set created by `delaunay` (left) and `delaunayn` (right). The four vertices on each quadrangle can be placed on the same circle, and both ways of splitting the quadrangle are therefore equivalent.

There is a lot of triangulation software available, and instead of implementing its own Delaunay triangulation, MRST offers an interface to transform a general triangulation into a grid object `G`. The following code uses the built-in MATLAB function `delaunay` to create a Delaunay grid from five generation points (or sites):

```
% Define sites
sites = [0, 0; 1, 0; 1, 1; 0, 1; 1/3, 1/3];
% Create Delaunay triangulation
t = delaunay(sites);
% Convert to MRST grid
G = triangleGrid(sites, t);
```



The Delaunay triangulation for a set of sites is unique up to any degenerate points. We say that $d+2$ sites from $\mathcal{S} \subset \mathbb{R}^d$ are degenerate if there exists a sphere that intersects all $d+2$ sites. In 2D, this happens, e.g., when the four sites form a quadrangle (Figure 1.2), because no matter which diagonal we pick as an edge in the triangulation, we have a valid Delaunay triangulation. Most triangulation algorithms are able to handle these degenerate cases, some algorithms will pick a diagonal at random, whereas others will always pick the same. The `delaunay` function in MATLAB produces a structured triangulation, whereas `delaunayn` and versions of `delaunay` prior to R2009b use Qhull [1] and produce the unstructured variation.

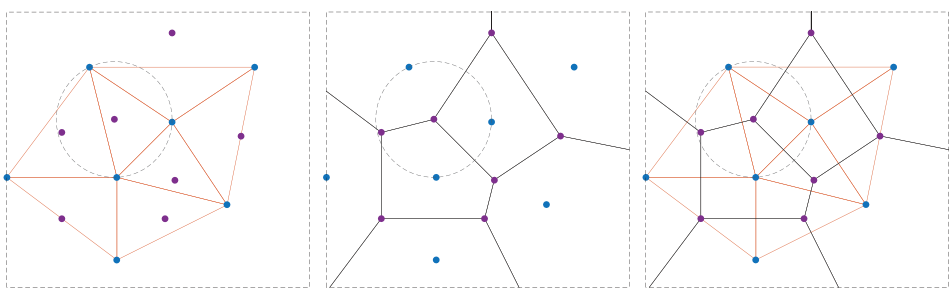
1.2.2 PEBI Grids

We denote the PEBI grid of a set of sites by $\mathcal{P}$. These sites uniquely define the PEBI grid, and for each site $\vec{s}_i \in \{\vec{s}_j\}_{j=1,\dots,n} = \mathcal{S}$, we associate a cell v_{s_i} that is defined by

$$v_{s_i} = \{\vec{x} : \vec{x} \in \mathbb{R}^d, \ \|\vec{x} - \vec{s}_i\| < \|\vec{x} - \vec{s}_k\|, \ \forall k \in \{1, \dots, n\} \setminus \{i\}\}. \tag{1.1}$$

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information



(a) (b) (c)

Figure 1.3 The duality of a Delaunay triangulation and a PEBI grid in 2D. Delaunay vertices correspond to PEBI cells (blue points). Delaunay edges are perpendicular to the corresponding PEBI edges. The circumcenter of a Delaunay triangle corresponds to a PEBI vertex (purple points).

An intuitive explanation of this is that a cell v_{s_i} is defined by all points in $\mathbb{R}^d$ that are closer to $\vec{s}_i$ than any other sites; see Figure 1.3 for an example of a PEBI grid.

A face between the two cells v_{s_i} and v_{s_j}, denoted by v_{s_i,s_j}, can now be defined as the intersection between the closure of the two cells $v_{s_i,s_j} = \bar{v}_{s_i} \cap \bar{v}_{s_j}$. For a PEBI grid, the face can alternatively be expressed as

$$v_{s_i,s_j} = \{\vec{x} : \vec{x} \in \mathbb{R}^d, \ \|\vec{x} - \vec{s}_i\| = \|\vec{x} - \vec{s}_j\| < \|\vec{x} - \vec{s}_k\|, \ \forall k \in \{1, \ldots, n\} \setminus \{i, j\}\}.$$

This means that the face v_{s_i,s_j} between the two cells v_{s_i}, v_{s_j} consists of all points that are closer to the two sites $\vec{s}_i$ and $\vec{s}_j$ than any other site.

In general, we define a k-face of the grid as the k-dimensional intersection between cells. In 3D, a 2-face is an interface between two neighboring cells, a 1-face is an edge that defines the intersection between at least two cell interfaces, and 0-faces are nodes/vertices in the grid that represent intersections among cell edges. In addition, each cell is said to be a 3-face. For a PEBI grid in $\mathbb{R}^d$, the $(d - i + 1)$-face is defined by a set of sites $\{\vec{s}_1, \ldots, \vec{s}_i\} \subset \mathcal{S}$ as

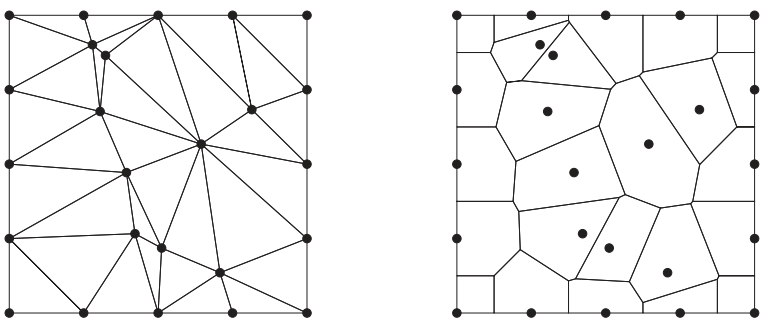
$$v_{s_1,\ldots,s_i} = \{\vec{x} : \vec{x} \in \mathbb{R}^d, \ \|\vec{x} - \vec{s}_1\| = \ldots = \|\vec{x} - \vec{s}_i\| < \|\vec{x} - \vec{s}_k\|, \\ k = i + 1, \ldots, n\}. \quad (1.2)$$

Thus, the sites $\vec{s}_1, \ldots, \vec{s}_i$ (with $i = 4$ in 3D and $i = 3$ in 2D) define a vertex $v_{s_1,\ldots,s_i}$ if and only if the interior of the ball that intersects the sites $\vec{s}_1, \ldots, \vec{s}_i$ does not contain any other sites from $\mathcal{S}$. The circle intersecting three sites in Figure 1.3 demonstrates this.

The duality between the Delaunay triangulation and the PEBI grid gives us some important properties and relations we can exploit to construct conforming

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information



(a) (b)

Figure 1.4 A set of sites that are perturbed randomly: (a) The Delaunay triangulation of the sites; (b) The PEBI grid of the sites.

PEBI grids. It will also give us a better understanding of why the conforming algorithms presented later in this chapter work. The duality of the two grids is stated as follows.

Theorem 1.2. *Let S be a generic point set (maximum $d+1$ points can be intersected by one sphere) in $\mathbb{R}^d$. Let $\mathcal{P}$ and $\mathcal{T}$ be the associated PEBI grid and Delaunay triangulation, respectively. Define a subset $P = \{\vec{s}_1, \ldots, \vec{s}_j\} \subset S$. Then, the convex hull of P is a k-face of $\mathcal{T}$ if and only if $v_{s_1,\ldots,s_j}$ is a $(d-k)$-face of $\mathcal{P}$.*

Most important, this means that each node in $\mathcal{T}$ is associated with a cell in $\mathcal{P}$ and each cell in $\mathcal{T}$ is associated with a node in $\mathcal{V}$. In fact, the center of the circumsphere around a cell from the Delaunay triangulation will be a vertex in the PEBI grid. Further, if there is an edge in $\mathcal{T}$ between two sites, then the two sites share a face in $\mathcal{P}$; see Figure 1.3 for an illustration of the duality.

The function `pebi` creates PEBI grids directly from a Delaunay triangulation in 2D by connecting the perpendicular bisectors of all edges with the circumcenters of each cell. This method is relatively fast but is only implemented in 2D. Also be aware that it will fail along the domain boundary if the circumcenter of any triangle lies outside the convex hull of the sites.

As an example, we create and perturb a set of sites and generate the Delaunay triangulation and PEBI grid shown in Figure 1.4:

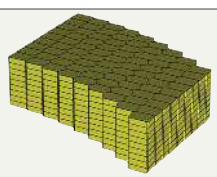
```
n = 5;
[X,Y] = meshgrid(linspace(0,1,n));
isIn  = false(n,n);  isIn(2:end-1,2:end-1) = true;
sites = [X(:),Y(:)];
sites(isIn(:),:) = sites(isIn(:), :) + 0.1*randn((n-2)^2, 2);
Gt   = triangleGrid(sites);
Gp   = pebi(Gt);
```

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE
Cambridge University Press
978-1-316-51996-7 — Advanced Modelling with the MATLAB Reservoir Simulation Toolbox
Edited by Knut-Andreas Lie , Olav Møyner
Excerpt
More Information

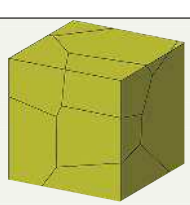
MATLAB also offers the `voronoin` function, based on Qhull [1], for computing Voronoi vertices and cells from a set of generating points. The `upr` module has a routine that transforms the resulting set of vertices and cells into a valid MRST grid, illustrated by the following code example for a set of points in 3D:

```
keep    = false(11,11,11);   % Flag used later to remove
keep(2:10,2:10,1:11)=true;   % boundary cells
[X,Y,Z] = meshgrid(linspace(0,1,11));
sites   = [X(:)+.5*Y(:).^2 Y(:) Z(:)];
[V,C]   = voronoin(sites);
G       = voronoi2mrstGrid3D(V,C(keep));
```



Here, `V` holds the vertices and `C` maps from each cell to its vertices. Note that `voronoin` creates an unbounded Voronoi diagram in which the boundary cells extend to infinity, which is not suitable in a practical grid. Such cells are disregarded by `voronoi2mrstGrid3D`, but here we clip away the outer cell layer explicitly using the Boolean array `keep`. The `upr` module also includes another function `mirroredPebi3D` that creates the PEBI grid of a convex domain by placing mirror sites outside the boundary. This function is a wrapper around the `voronoin` function and will clip the PEBI grid by the convex hull of the specified boundary. We discuss clipped 2D PEBI grids further in the next subsection.

```
pts = rand(10, 3);
% Define the unit cube as boundary
bnd = [0 0 1 1 0 0 1 1; ...
       0 1 1 0 0 1 1 0; ...
       0 0 0 0 1 1 1 1]';
% Create clipped Pebi grid
G = mirroredPebi3D(pts, bnd);
```



1.2.3 Clipping PEBI Grids

Having PEBI cells that extend to infinity is not desirable when the grid is to be used for numerical approximations of a finite domain. To resolve this, we restrict our domain to a bounded subset $\Omega \subset \mathbb{R}^d$:

$$v_{s_i} = \{\vec{x} : \vec{x} \in \Omega,\ \|\vec{x} - \vec{s}_i\| < \|\vec{x} - \vec{s}_k\|,\ \forall k \in \{1, \ldots, n\} \setminus \{i\}\}.$$

MRST's `pebi` function assumes that the domain is bounded by the *convex hull* of the sites. To generate a grid that would fit a more general domain, either we

© in this web service Cambridge University Press www.cambridge.org