Society of Petroleum Engineers

SPE 182666-MS

Unstructured Gridding and Consistent Discretizations for Reservoirs With Faults and Complex Wells

Ø.S. Klemetsdal, Norwegian University of Science and Technology; R.L. Berge, University of Bergen; K.-A. Lie, H.M. Nilsen and O. Møyner, SINTEF.

This paper was prepared for presentation at the SPE Reservoir Simulation Conference held in Montgomery, Texas, USA, 20-22 February 2017.

Abstract

This work consists of two parts. In the first part, we present new methods for generating unstructured polyhedral grids that align to prescribed geometric objects. Control-point alignment of cell centroids is introduced to accurately represent horizontal and multilateral wells, but can also be used to create volumetric representations of fracture networks. Boundary alignment of cell faces is introduced to accurately preserve geological features such as layers, fractures, faults, and/or pinchouts. Prescribed geometric objects will often intersect each other. To handle such cases, we propose a conflict-point handling scheme that creates conforming cells even at intersections. We also discuss how to generalize this method to 3D. Here, our method honors control-point alignment of cell centroids and boundary alignment of cell faces away from object intersections.

The predominant discretization method for multiphase flow in reservoir simulation is the two-point flux-approximation (TPFA) method. This finite-volume method is mass conservative, but only conditionally consistent and hence susceptible to grid-orientation effects. In the second part of the paper, we review a series of consistent methods and compare and contrast these methods both with respect to accuracy and monotonicity. Our comparisons include a multipoint flux-approximation (MPFA-O) method, the nonlinear TPFA method, mimetic methods, and the more recent virtual element methods. To limit the discussion, we focus on incompressible flow, for which we study the effects of deformed cell geometries, anisotropic permeability, and robustness with respect to various approaches to grid near wells and adapt it to lower-dimensional objects like faults and fractures.

Introduction

The basic geometric description of a petroleum reservoir consists of a collection of surfaces representing stratigraphic layering and fault surfaces and horizons that represent the structural architecture. These surfaces delineate the major compartments of the reservoir and often provide first-order control on in-place fluid volumes and fluid movement during production (Branets et al. 2009b). To correctly split the reservoir volume into sub-volumes, build flow units with similar or correlated petrophysical properties, and resolve flow pattern, it is important that a volumetric simulation grid conforms as closely as possible to these surfaces. Early simulation grids were either simple Cartesian boxes or block-centered grids used for dipping bedding, in which each rectangular cell could be compactly represented by four numbers (top depth and extent in each axial direction). These grid types are simple to construct, but cannot represent stratigraphy and structural architecture very well. To better model sloping horizons, fault planes, and erosion surfaces, corner-point or pillar grids were introduced by Ponting (1989). In these grids, each cell is defined by four coordinate lines (pillars) and eight depth values, two along each line. In its simplest form, the coordinate lines are vertical and distributed on a regular areal mesh, but more generally, the pillars are sloping lines defined over a curvilinear areal grid. The format has also been extended to include patches with local grid refinement (LGR) to improve the resolution in the near-well zone. By construction, corner-point grids have an inherent Cartesian topology, which is advantageous for simulation. Unless the reservoir is heavily faulted or has extensive erosion that introduce non-neighboring connections, most cells will have six neighbors, which in turn leads to discretization matrices with reasonably regular sparsity patterns. On the other hand, corner-point grids are time-consuming to generate, require specialized software, have many subtle geometrical challenges, and can easily give significant grid-orientation and grid-deviation effects (Wu and Parashkevov 2009) unless care is taken. The format is also inflexible and unable to accurately represent more complex features like y-shaped faults, thrust faults and other overturned structures, etc. Likewise, it is very difficult to introduce new horizontal wells without changing the grid in large parts of the reservoir. Modern wells are typically long and horizontal and may have complex geometry that consists of multiple branches. Wells are increasingly becoming the main determining factor for reservoir flow patterns because of their long reach, but also as a result of various techniques for modifying the near-well region to increase injectivity or use of (intelligent) inflow devices to control fluid production. Accurate and flexible description of well paths and the near-well region is crucial to evaluate and choose different drilling, completion, and production strategies. To meet these challenges, many have started looking into more flexible unstructured grids.

Unstructured grids were introduced in reservoir simulation in the late 1980s and early 1990s (Heinemann et al. 1991; Palagi and Aziz 1994; Guerillot and Swaby 1993; Fung et al. 1992). The earliest techniques would embed refinements in a structured background grid in areas of interest. A popular unstructured grid is the perpendicular bisector (PEBI) grid. The properties of PEBI-grids used for reservoir simulations are discussed by Verma and Aziz (1997). Courrioux et al. (2001) were among the first to create a PEBI representation of a full-scale reservoir. The main drawback of these first attempts is the inability to represent complex structures, such as pinchouts and intersections of multiple faults. Later, Branets et al. (2009a) proposed a method that handles intersection of multiple faults, and faults intersecting at sharp angles. A similar method is also presented by Toor et al. (2015). These methods create a protection layer around the features using constrained Delaunay triangulation and recover the faults exactly. Pinchouts and intersecting faults are treated by mirroring PEBI-sites (seed points for cell centers used in the grid generation) around the features. A disadvantage with these methods is that they often lead to congested PEBI-sites around the features. In an attempt to deal with these problems, Ding and Fung (2015) introduced a conflict-point removal scheme. First, a structured background grid is created. A set of PEBI-sites are placed equidistant around each fault. Each PEBI-site is given a priority, and when two sites are too close, the site with lowest priority is removed. The generated grid conforms to faults and has fairly uniform cells, but fails to treat intersections and pinchouts. A different approach is taken by Merland et al. (2011, 2014), who suggest to place the PEBI-sites by an optimization method that minimizes the volume of the cells that are cut in two by a fault. This method is promising, but one often needs to treat the grid manually after the optimization. Especially cells at fault intersections can be bad. PEBI-grids are not the only unstructured grids that have received attention. There have been several attempts at creating triangular grids that adapt to faults and fractures. Brewer et al. (2015) present a method for exactly representing fractures by a triangulation. Methods for approximating faults and fractures by triangles have also been investigated (Mustapha 2011; Holm et al. 2006). Another method that has gained popularity in the latest years is the cut-cell method (Gringarten et al. 2008, 2009; Mallison et al. 2014). This method generates a grid by creating a mapping from a Cartesian grid to the physical domain, and then creates general polyhedral cells by cutting the Cartesian cells by crossing faults.

In the first part of the paper, we present new methods for generating unstructured grids in both 2D and 3D that conform to geological structures and wells. We will look at two different conformity requirements: *(i)* structures that should be traced by faces of the grid, and *(ii)* structures that should be traced by cell centroids. Typically, wells should be traced by cell centroids, whereas internal boundaries such as fractures, faults, horizons, and erosion surfaces should be traced by faces. The new methods have been implemented as a separate module in MRST (Lie 2015).

Real reservoirs have anisotropic and spatially varying permeability. The standard two-point flux-approximation (TPFA) method uses, as the name suggests, the pressure values at the centroids of two neighboring cells to compute the flux across their common interface. The method is therefore only consistent for K-orthogonal grids in which the principal directions of the permeability tensor align with the vectors joining the pressure points used in the discretization stencil. For simple cases, K-orthogonality can be enforced by rotating the axial directions of the grid, or by transforming physical space into an isotropic space in which orthogonal grids can be generated (Gunasekera et al. 1997). The petrophysical properties of a simulation model are typically the result of an upscaling process, which will tend to introduce full-tensor permeabilities with nonzero off-diagonal elements. In principle, it is possible to diagonalize such permeabilities if the tensor is symmetric and positive, but in most cases this is not feasible, in particular when the grid needs to conform to well trajectories, faults, erosion surfaces, and other geological properties. We can therefore, as a rule, expect that adapted grids will not be K-orthogonal in some parts of the reservoir. Since the TPFA method is then not consistent, it will not be convergent, but more importantly: it will introduce grid-orientation effects that will adversely affect the solution quality. In the second part of the paper, we will therefore discuss several alternative discretization methods that are consistent on general polyhedral grids.

To ensure consistency, multipoint flux-approximation (MPFA) methods (Aavatsmark 2002) introduce auxiliary pressure points at the cell interfaces, which are coupled inside local interaction regions that together form a dual grid. The auxiliary pressures are then eliminated locally by imposing flux continuity, so that the flux across each interface is approximated using pressure values from multiple cells in the local region around the interface. MPFA methods retain the same number of unknowns as the TPFA method, but have a more dense stencil, and are somewhat cumbersome to implement for complex grids. In mimetic methods (Brezzi et al. 2005; Lie et al. 2012), the auxiliary pressure points are kept as primary unknowns. Since local interactions do not need to be constructed explicitly, mimetic methods are very flexible, both in terms of grid geometry and the type of discretization they reduce to on simple grids. However, they involve a significantly higher number of unknowns. New developments of mimetic methods have lead to virtual element methods (Beirão da Veiga et al. 2013, 2014), which constitute a uniform framework for higher-order discretizations that require significantly fewer degrees of freedom. These methods are generally not conservative, but are useful for discretizing other physical effects like diffusion, dispersion, and/or mechanical effects, and can be made conservative by postprocessing the non-conservative velocities. The last method we consider is the NTPFA method (Nikitin et al. 2014a), in which the transmissibility used for the two-point approximation for each interface varies as a function of pressure. The resulting stencil is nonlinear and the method is therefore, unlike other first-order consistent methods, guaranteed to be monotone. Using a number of test cases involving deformed cell geometries and anisotropic permeabilities, the performance of the various methods is compared and contrasted. All methods are implemented as freely available open-source modules in MRST, and the paper can therefore be seen as an update of (Lie et al. 2012).

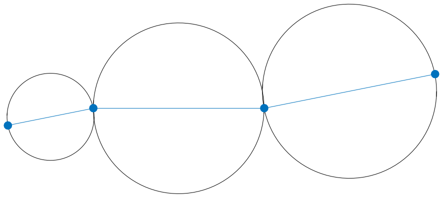


Figure 1—The well condition is satisfied if the interior of the circles does not contain any sites.

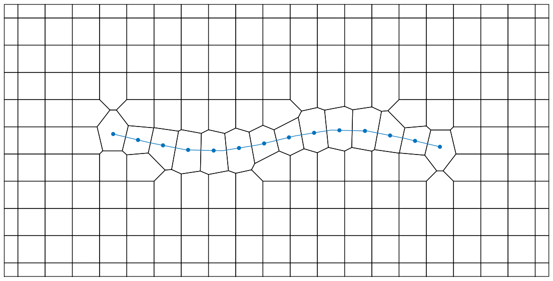


Figure 2—A grid of a single well. The blue line is the well path, and the blue points the well cell centroids.

PEBI-Grids

In this section, we will go through the process of generating PEBI-grids that conform to control points and to internal boundaries. Control-point conformity will mainly be used to represent wells, whereas internal boundaries are primarily faults. Henceforth, the two types of conformity are therefore referred to as wells and faults for simplicity, even though the same techniques can be used to adapt to other features like fracture networks, for which control-point conformity would be used to generate volumetric fractures and boundary alignment would be used for lower-dimensional representation.

To simplify the discussion, we introduce some more terminology. We let $S = \{\boldsymbol{s}_i\}_{i=1\dots n}$ be a set of generating points in $\mathbb{R}^d$, which we will refer to as *sites* to distinguish them. In all algorithms, we will work with three different sets of sites: well sites, fault sites, and reservoir sites. Well and fault sites are created to make the grid conform to control points and internal boundaries, respectively. Reservoir sites are all other sites that generate the background grid. To define the PEBI (or Voronoi) grid, we say that a point $\boldsymbol{x}$ belongs to the *PEBI-cell* Ω_i if it is at least as close to $\boldsymbol{s}_i$ as any other sites in S; that is, $\Omega_i = \{\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^d,\ |\boldsymbol{x} - \boldsymbol{s}_i| \leq |\boldsymbol{x} - \boldsymbol{s}_j|,\ j = 1 \dots n\}$. The PEBI-grid is then defined as the set of all PEBI-cells. This grid is the dual of the Delaunay triangulation of the set of well, fault, and reservoir sites. In the following we will only consider so called clipped PEBI-grids, where we restrict the PEBI-cells to a bounded set $\Omega \subset \mathbb{R}^d$; $\Omega_i^{\text{clipped}} = \Omega_i \cap \Omega$ (Berge 2016; Yan et al. 2010).

Generating well and fault sites in 2D We start by presenting a generalization of the conflict-point removal algorithms proposed by Fung et al. (2014) and Ding and Fung (2015) in 2D. In the next subsections, we first present our novel approach to robustly handle fault–fault, fault–well, and well–well intersections, then present various methods for generating reservoir sites, before we finally discuss how to generalize the generation of conforming PEBI-grids to 3D.

Well sites In reservoir simulation, the diameter of the wellbore will typically be much smaller than the size of the grid cell, so that the inflow or outflow in wells occur on a subgrid scale. To account for the significant pressure variation inside the cell, it is common to use an analytical or semi-analytical model of Peaceman type. These models typically assume that the wellbore passes through the centroid of each perforated cell. To maximize accuracy, a good grid should therefore trace wells using cell centroids. In a well-shaped PEBI-grid, the cell-centroids coincide with their respective sites (Du et al. 1999). We therefore place a set of well sites along each well trajectory with the distance between consecutive sites given by a user-defined function. A requirement we put on the well sites is that consecutive well sites that belong to the same well segment should be connected by edges in the Delaunay triangulations of the sites. In the dual PEBI-grid, consecutive well sites will then be neighbors.

Definition (Well condition)**.** If $\boldsymbol{s}_1$ is a well site and $\boldsymbol{s}_2$ is a consecutive site, the *well condition* is satisfied if the circle intersecting the two sites centered at their midpoint does not contain any other sites from S.

The circles defining the well condition are shown in Fig. 1. The line segment between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ will be an edge in the Delaunay triangulation if the well condition is satisfied (Berge 2016). Further, the neighbor edge in the dual PEBI-grid will contain this point. Fig. 2 shows a grid of a single well, where the distance between well sites are increasing along the well. Because the well condition is satisfied, the neighbor edge between two consecutive well sites always intersects the well path.

Fung et al. (2014) suggest to add a protection layer around the well trajectories to make the shapes of well cells more regular and to better capture the symmetric flow in or out of wells. Sun and Schechter (2015) showed that one can grid the radius of a well explicitly by adding protection sites around the well sites. To add a layer of protection sites, we trace the well paths and place the protection sites normal to the well path. Each well site will have two protection sites, one on each side. Fig. 3 shows one well with a protection layer. The distance d the protection sites are placed from the well paths also equals the diameter of the corresponding well cell. We allow the distance d to vary along the well path. This is practical when we for example wish to create explicit volumetric representation of fractures with varying width (Sun and Schechter 2015). In the figure, the distance function is permuted randomly for each set of protection sites. The careful reader will probably observe that this may introduce very short faces, which should preferably be eliminated, as shown in Fig. 3c, to make the grid more suitable for discretization of

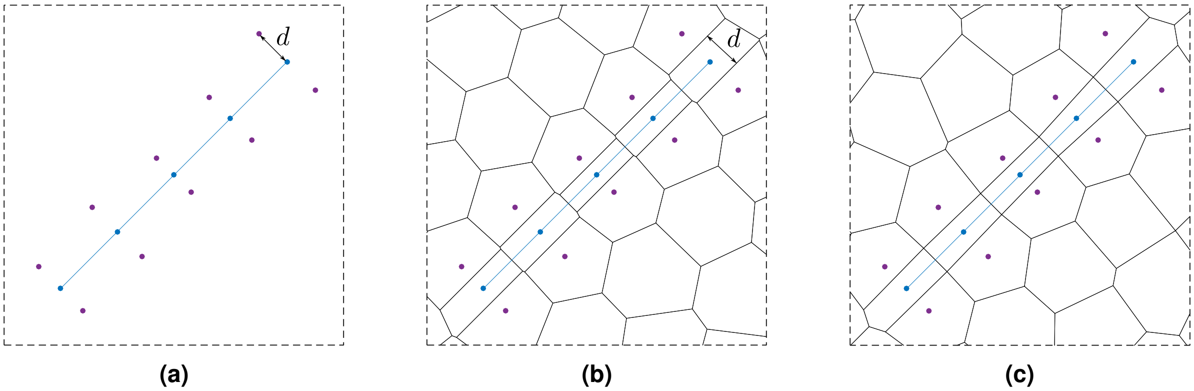
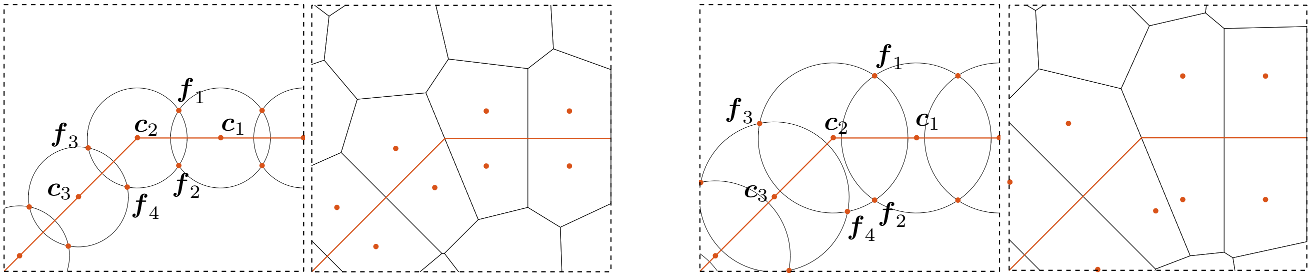


Figure 3—Illustration of a well with a protection layer. Blue points are well sites, while purple points are protection sites. The distance d from a well site and its protection sites also equals the diameter of the corresponding well cell. In (c) we have postprocessed the grid and removed any short edges. Note that this grid is not strictly PEBI, whereas the gird in (b) is.



(a) Circle factor 0.6.

(b) Circle factor 0.9.

Figure 4—The creation of fault sites $\boldsymbol{f}_1$, $\boldsymbol{f}_2$, $\boldsymbol{f}_3$, and $\boldsymbol{f}_4$, from the circle centers c_1, c_2, and c_3. The fault condition is satisfied if the interior of the three circles does not contain any sites.

flow equations.

Fault sites To create a grid where edges trace a fault, we have to place fault sites equidistant on opposite sides of the fault. To achieve this, we are inspired by the method described by Ding and Fung (2015), who propose to place fault sites equidistant along a fault by calculating the intersection of circles. They only discuss the specialized case when all circles have the same radius and distance apart, and their method also fails to represent faults exactly at intersections. We will generalize the idea of intersecting circles and give a rigorous discussion of how one can guarantee conformity also at intersections of faults.

We start by placing a set of points $\{\boldsymbol{c}_i\}$ along the fault, which we refer to as circle centers. The distance, or step length, between two consecutive circle centers $d_i = |\boldsymbol{c}_{i+1} - \boldsymbol{c}_i|$ is set by the user through a density function $d_i = \rho(\boldsymbol{c}_i, \boldsymbol{c}_{i+1})$. We then draw a circle around each circle center. Two consecutive circles should intersect, which gives us an upper and lower bound on the radii of the circles:

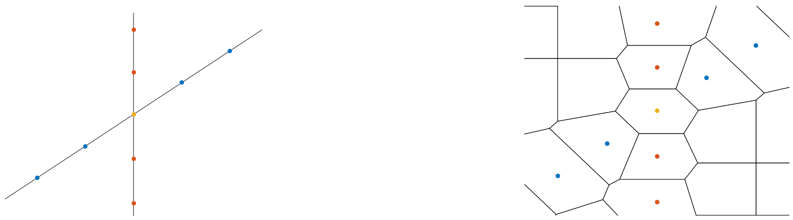
$$d_i \leq R_i + R_{i+1}, \quad |R_i - R_{i+1}| \leq d_i. \tag{1}$$

In our implementation, we have chosen to set the radius of a circle as a scaled average of the distance to the circles on either side

$$R_i = c_f \frac{d_i + d_{i-1}}{2}.$$

The constant c_f is called the circle factor, and it controls how far from the fault the fault sites should be placed. Normal values for the circle factor lie in the interval $(0.5, 1)$. The fault sites $\{\boldsymbol{f}_j\}$ are placed where two circles intersect. If c_f is small, the fault sites will be placed close to the fault, and if c_f is large, the fault sites will be placed far from the fault. In Fig. 4 we see the difference of circle factor 0.6 and circle factor 0.9. By construction, the site pairs $\boldsymbol{f}_1\boldsymbol{f}_2$ and $\boldsymbol{f}_3\boldsymbol{f}_4$ are placed equidistant on each side of the fault, which under the following condition will make the PEBI-grid conform to the fault:

Definition (Fault condition)**.** Let $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ be the two fault sites that are generated from the intersection of two circles. The *fault condition* is satisfied if the interior of the two circles contains no sites from S.



(a) The well paths and well sites.

(b) A grid created using a Cartesian background-grid.

Figure 5—Intersection of two wells (black lines). Blue points are well sites for the diagonal well, red points are well sites for the vertical well, and the yellow point is a shared well site.

The fault condition is a necessary *and* sufficient condition for the line segment between the two circles to be the neighbor edge between PEBI-cell of site $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$. If the fault condition is satisfied for all fault site pairs, the grid is guaranteed to conform exactly to the faults (Berge 2016).

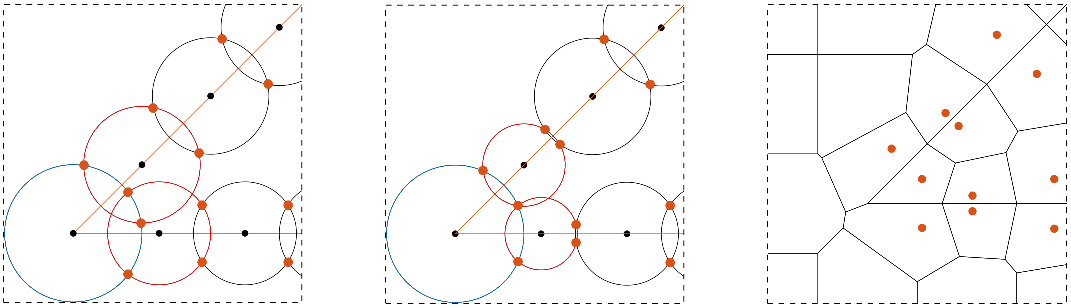
Handling intersections The algorithms introduced above for generating well and fault sites are sufficient to give good grids as long as the individual well trajectories and/or fault lines do not cross each other. If this is not the case, additional care needs to be taken to ensure that the grids conform in a feasible manner. There are three types of intersections a robust grid generator should handle; well–well, well–fault and fault–fault intersections. Our grid generator handles all of these cases automatically, also for harder cases such as the intersection of multiple faults.

Well-well intersections When two well trajectories cross, we have to be careful when placing the well sites. If we place the sites of each well independently, consecutive sites will in general not be connected by Delaunay edges over the intersection. We may also create small and badly shaped cells. To treat these cases, all well paths are divided into segments by the well intersections. A well segment does not intersect any other well segments, except possibly at the end-points. When we place the well sites, we first place a well site at each intersection. A well site at an intersection is shared by all well segments starting or ending in this intersection. The remaining well sites are placed along the well segments as normal. Fig. 5 shows the intersection of two wells. The yellow site is shared by both wells, and the other sites are in this case placed equidistant along the wells. This method ensures a consistent size of the well cells, even at intersections of multiple wells. For the wells that intersect, we do not place a set of protection sites at the intersection of the wells. Note that this will result in a cell that has an incorrect size, which has to be taken into consideration. A simple solution is to scale the porosity of the intersection cells to accurately represent the flow through the intersection.

Fault-fault intersections If we place the fault sites for each fault independently, we will in general not be able to represent the faults exactly. At the intersection of two faults, fault sites from either fault may interfere with each other and violate the fault condition. To create a conforming grid, the first step is to calculate the intersection of all faults. As for the well-well intersection, faults are split into fault segments at these intersections. The fault segments will not have any intersections, except possibly at the end-points. This way, we do not have to distinguish between different intersection cases, such as pinchouts or intersection of multiple faults.

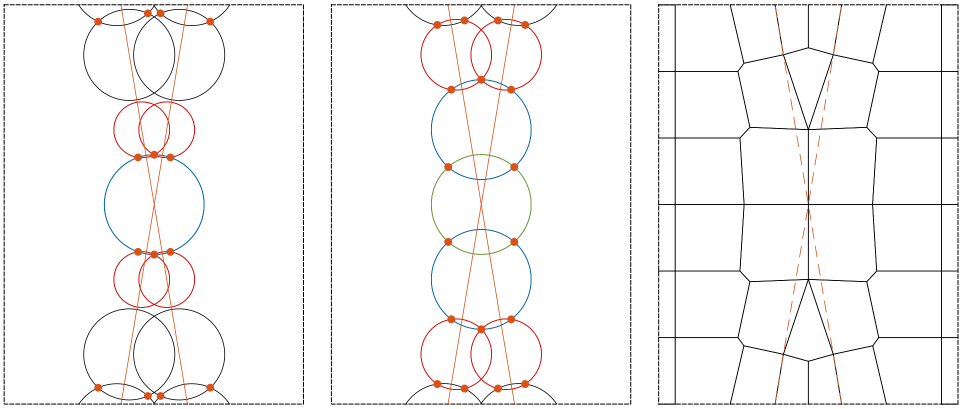
At each intersection, we place a circle that is shared by all fault segments ending in that intersection. The other circles are placed as normal along the fault segments. We color all intersection circles blue, and all neighbor circles of blue circles are colored red. On each red circle, one of three actions is performed: *(i)* Nothing is modified, *(ii)* the radius is changed, *(iii)* the circle is merged with another red circle. If the interior of a circle does not contain any sites, it is not modified. If the interior of a circle contains fault site $\boldsymbol{f}_i$, we locate the red circle that generated $\boldsymbol{f}_i$. These two circles are tagged as conflict circles. The radii of the conflict circles are shrunk as shown in Fig. 6. The new radii are chosen such that the blue circle and the two red circles intersect at the midpoint of the two faults. When multiple faults intersect, a circle might have multiple conflict pairs. We then calculate the new circle radius for each conflict pair and choose the smallest of them. If the radius of a red circle is shrunk too much, it might violate the radius condition of Eq. 1. For those cases, we locate the other red conflict circle sharing a fault site with this circle. These two circles are merged to one circle centered at the midpoint of them. The merged circle is colored blue, and we repeat the procedure above. Fig. 7 shows one iteration of the merging. In this case, two sets of conflict circles are merged, one on each side of the intersection. This is is enough to satisfy the fault condition. If the intersection had been sharper, more than two pairs of circles might have been merged successively.

A second case that triggers two red conflict circles to merge is if they are too close to each other. From Fig. 8; if the ratio δ/d is smaller than a given tolerance, we merge the circles. By changing the tolerance parameter, we can control how we grid sharp intersections. For a large tolerance, more circles will be merged than for a small tolerance.



(a) Original sites. **(b) Merged sites.** **(c) PEBI-grid**

Figure 6—Merging two conflict sites in a pinchout. The orange lines are two intersecting faults and orange points the fault sites.



(a) **(b)** **(c)**

Figure 7—The procedure of merging circles. (a) The circle at the intersection is colored blue, and its neighbors are colored red. The radii of the red circles are shrunk until they intersect the blue circle at the same point. The red circles now do not intersect their neighbors. They are therefore merged. (b) The merged circles are colored blue, and their neighbors are colored red. The green circle is already processed and is therefore not colored red. The procedure from (a) is repeated until the fault condition is satisfied. (c) An associated grid.

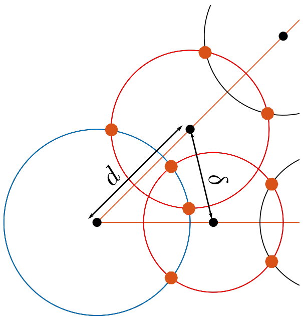
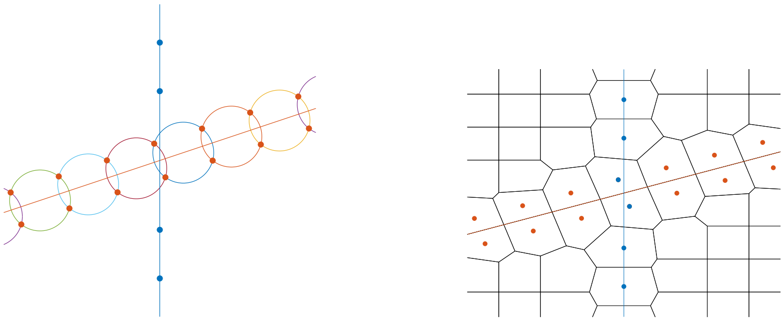


Figure 8—Merging two circles in a pinchout. If the ratio of the distance between circles on the same fault d, and the distance between the circles on opposite faults δ is too small, the circles are merged.



(a) Well and fault sites.

(b) PEBI-grid.

Figure 9—Intersection of a fault (red) and a well (blue). The fault sites are placed as close as possible to the well trajectory. The fault sites at the intersection are labeled as well sites.

Our method of splitting faults into fault segments and placing circle centers along these segments makes it easy to handle barely intersecting faults. If a fault segment is shorter than a specified length, we do not place any circles along it. In our implementation we have set this minimum length to be 80% of the desired length between circle centers.

Well-fault intersections The last type of intersection we need to consider is well-fault intersections. As for the two other cases, all faults and wells are split at the intersections. Fig. 9 shows the intersection of a well and a fault. The first circle center of a fault segment starting in a well-fault intersection is placed half a step length d from the start. Equivalently, the last circle center of a fault segment ending in a well-fault intersection is placed half a step length from the end. The two fault sites created from the circle before and after the well-fault intersection are labeled as well sites. These two sites are the first and last well site for the well segments starting and ending in the intersection, respectively.

Generating reservoir sites The reservoir sites can be placed any way the user may see fit. The most obvious choice is to create a structured grid by placing sites equidistant in each direction or on a rectilinear mesh. We can then make use of the simplicity of the Cartesian topology away from the wells and faults. When placing the reservoir sites, we ignore all faults and wells. After the reservoir sites are created, we remove any sites violating the fault or well condition. The resulting grid is then guaranteed to conform to faults and wells. Even if the fault and well conditions are satisfied, some cells might be small or badly shaped. We therefore also remove sites that are too close to each other. A grid size is defined for each well and fault site. For well sites, the grid size is the distance between two consecutive well sites. For fault sites, the grid size is set to the distance between the two sites that are generated by the same two circles. If a reservoir site is closer to a well or fault site than that site's grid size, the reservoir site is removed. This ensures that the well- and fault-cells have a consistent size.

There are many different methods to choose from to create a fully unstructured grid. Here, we have chosen to place the reservoir sites using the force-based method proposed by Persson and Strang (2004). One reason is that this method is available as a free open-source implementation in MATLAB. The method associates the edges in the Delaunay triangulation with springs, whereas vertices are associated with joints connecting the springs. An initial triangulation is given, and the algorithm then finds an equilibrium position for the vertices. When solving for equilibrium, the well and fault sites that have been created using the algorithms explained above are set as fixed points; that is, they are not allowed to move during the optimization procedure. An example of an optimum triangulation and its dual PEBI-grid is shown in Fig. 10 for a case where initial reservoir sites were placed semi-randomly in the domain.

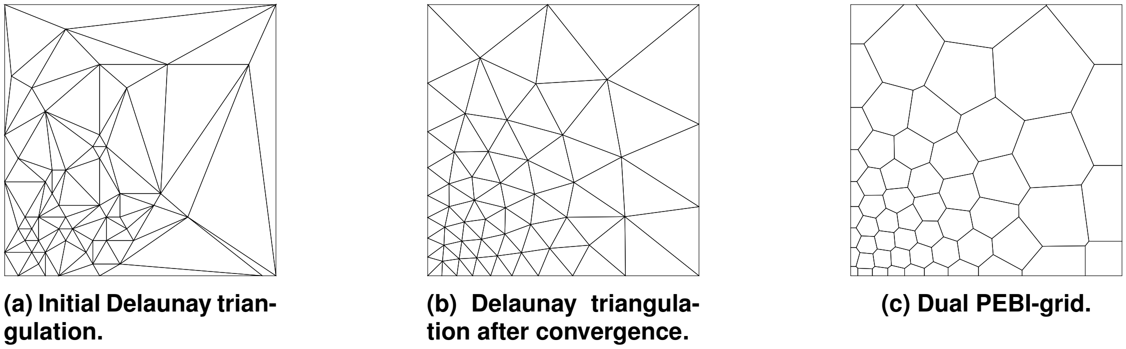
As an alternative to the force-based method, we will use a similar algorithm in which we optimize the PEBI-grid directly instead of optimizing the dual Delaunay triangulation. We define the Centroidal PEBI-grid (CPG) *energy function* as (Du et al. 1999; Iri et al. 1984)

$$F(\boldsymbol{s}) = \sum_{i=1}^{n} \int_{\Omega_i \cap \Omega} |\boldsymbol{y} - \boldsymbol{x}_i|^2 \mathrm{d}\boldsymbol{y}.$$

The variable $\boldsymbol{s} = [\boldsymbol{s}_1{}^\mathsf{T}, \ldots, \boldsymbol{s}_n{}^\mathsf{T}]^\mathsf{T}$ is a vector of the PEBI-sites. The variable $\boldsymbol{x}_i$ is the mass centroid for PEBI-cell Ω_i. A necessary condition for F to be minimized is $\boldsymbol{s}_i = \boldsymbol{x}_i$, that is, the PEBI-sites coincide with the mass centroids (Du et al. 1999). The gradient of F is

$$\frac{\partial F}{\partial \boldsymbol{s}_i} = 2A_i(\boldsymbol{s}_i - \boldsymbol{x}_i),$$

where A_i is the area of the associated PEBI-cell. To find the minimum of the CPG energy function we use the L-BFGS method, which is a quasi-Newton method (Nocedal and Wright 2006).

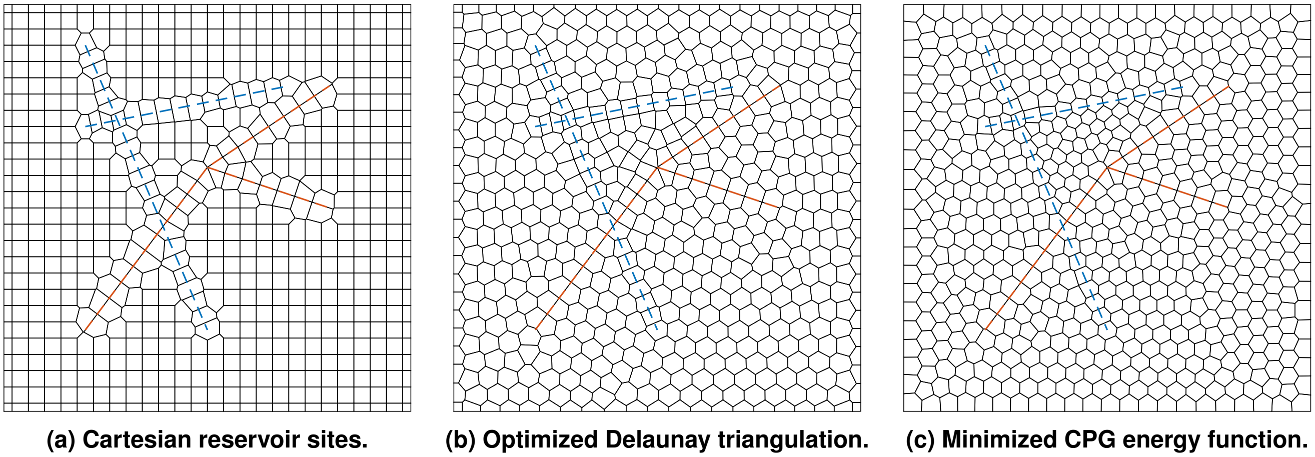


(a) Initial Delaunay triangulation.

(b) Delaunay triangulation after convergence.

(c) Dual PEBI-grid.

Figure 10—Optimization of a triangulation using the force-based algorithm. The size of the elements is proportional to the distance from the origin squared $h(x,y) \sim x^2 + y^2$.



(a) Cartesian reservoir sites.

(b) Optimized Delaunay triangulation.

(c) Minimized CPG energy function.

Figure 11—Three grids of a reservoir. The reservoir has two wells (dotted blue lines) and two faults (dotted orange lines). The well and fault sites are the same for all three grids and are created using the methods described in this section. The reservoir sites are created by three different methods; a Cartesian grid, optimizing the dual Delaunay triangulation, and minimizing the CPG energy function.

To be able to use the CPG formulation on a grid with fault and well sites, we propose a small change to the gradient. The fault and well sites are treated as fixed points, that is, we do not move them during the optimization procedure. We incorporate this into the L-BFGS algorithm by setting the derivatives $\frac{\partial F}{\partial \boldsymbol{s}_i}$ with respect to fixed points to zero. By doing so, the L-BFGS algorithm does not move the fixed points, and the resulting grid will conform to faults and wells.

A comparison of the three methods for placing reservoir sites is shown in Figure 11. The two optimization methods have more uniform cells than the Cartesian background grid. Also, the cell centroids for the optimization methods are very close to the well path. The minimization of the CPG energy function creates better reservoir cells in congested areas, e.g., look at the area between the well intersection and fault intersection.

Conforming 3D PEBI-grids Creating PEBI-grids conforming to faults is much harder in 3D than in 2D. Instead of dealing with lines, one has to consider surfaces, and the complexity increases drastically. In this section, we will present a generalization of the 2D method presented in the previous section. This is a novel method, which to the best of our knowledge is new in the literature.

Fault sites To generate the fault sites, we represent a fault by a surface triangulation. A triangle is the equivalent to a line segment in the 2D gridding method. We draw a sphere around all vertices in the triangulation and let the radius of the spheres vary according to a user-specified function. The radius of a sphere should be approximately the same as the length of the edges in the corresponding triangle. For each triangle, we find the intersection of the three spheres centered at the triangle's vertices. This intersection gives us two unique points equidistant on each side of the triangle, and we place one fault site at each of these points. The fault condition from the 2D case is trivially extended to 3D; the interior of the three spheres generating a fault site should not contain any sites from S. If the fault condition is satisfied, the resulting PEBI-grid will conform exactly to the triangulation of the fault.

To handle intersections of faults, we introduce a priority scheme. Each fault is given a priority by the user. If a fault site is

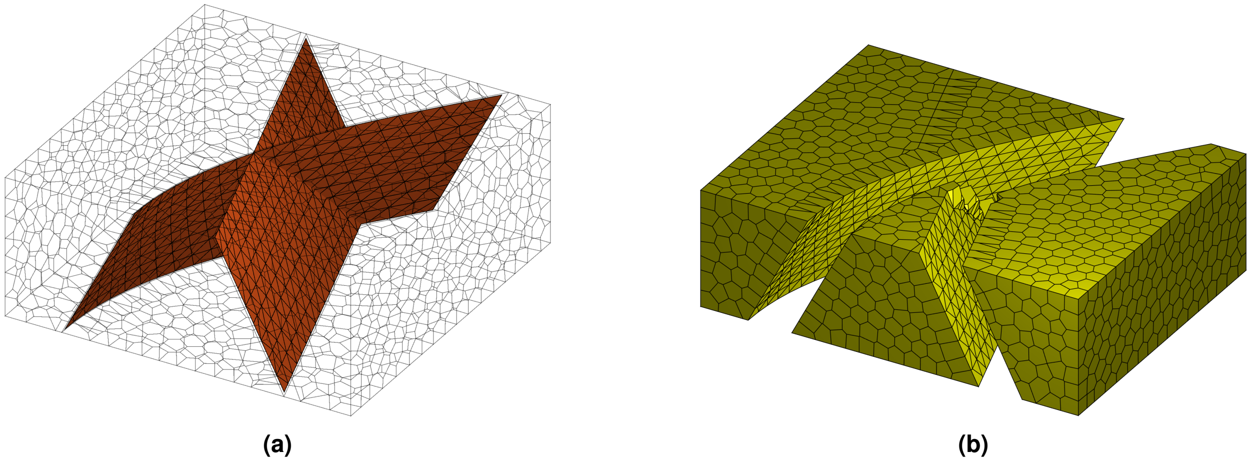


Figure 12—Two intersecting faults in 3D. At the intersection, only the curved fault is represented exactly.

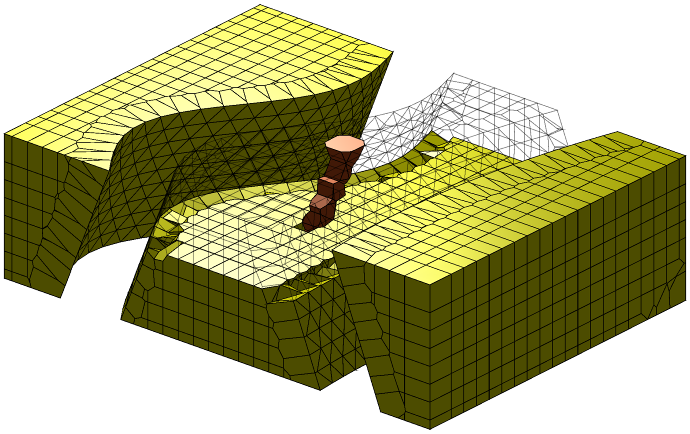


Figure 13—Reservoir consisting of three blocks separated by faults. Well cells are colored red.

in the interior of a circle of a fault with higher priority, it is removed. This priority scheme is similar to the one presented by Fung et al. (2014). We start by finding any sites that violate the fault condition for the fault with highest priority. These sites are removed. We then consider the fault with second highest priority and remove any sites inside the generating circles of this fault. This process continues until we have checked all faults. When we search for sites that violate the fault condition for a fault, we only check sites that have a lower priority than the current fault. Fig. 12 shows a grid with two intersecting faults. The fault with higher priority is represented exactly, while the fault with lower priority is only approximated over the intersection. Contrary to the 2D method, the 3D method does not conform exactly over fault intersections. However, away from any intersections, the grid conforms exactly to all faults.

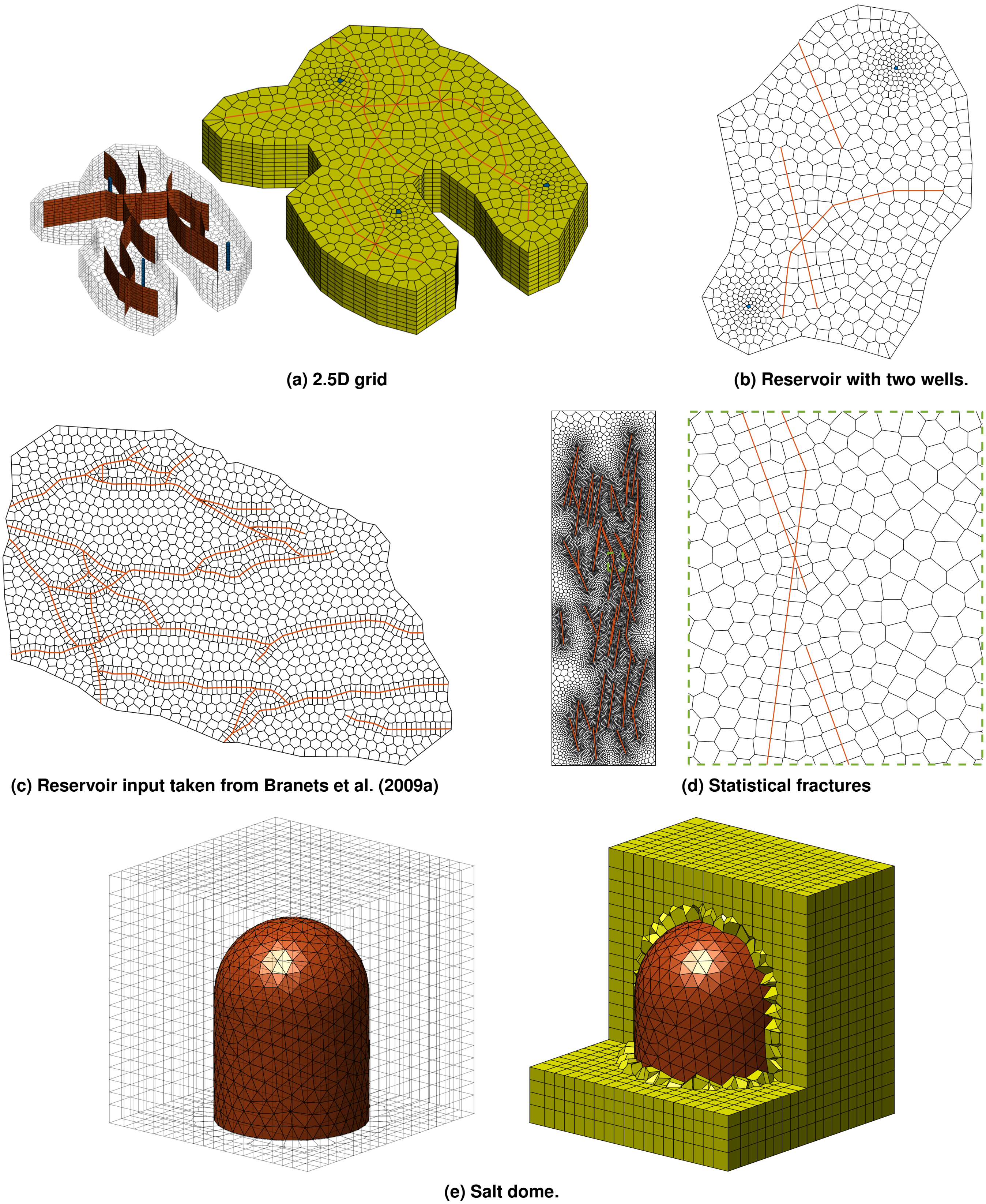
Well and reservoir sites The extension of wells sites to 3D is straightforward; instead of tracing well trajectories in 2D, we add a coordinate and place well sites along well trajectories embedded in a 3D domain. Intersection of wells are treated equivalent to the 2D case. Likewise, the reservoir sites can be created by any preferred method. The methods discussed for 2D grids easily generalize to higher dimensions. As long as the fault and well condition is satisfied for all spheres, the associated PEBI-grid will conform to the faults and wells. Fig. 13 shows a grid of a reservoir consisting of three blocks separated by two inclined and curved faults. The reservoir also contains one well going through the middle block.

For more examples of the possibilities of our gridding method, see Fig. 14.

Consistent Discretizations for Polyhedral Grids

In this section, we will review a variety of methods that can be used for consistent discretizations on general polyhedral grids. To this end, we consider incompressible flow,

$$\nabla \cdot \boldsymbol{v} = q, \quad \boldsymbol{v} = -\mathbf{K}\nabla p, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \tag{2}$$



(a) 2.5D grid

(b) Reservoir with two wells.

(c) Reservoir input taken from Branets et al. (2009a)

(d) Statistical fractures

(e) Salt dome.

Figure 14—Various grids showing the capabilities of our gridding method.

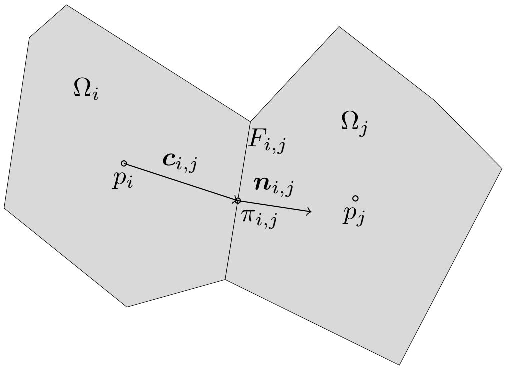


Figure 15—Flux between two neighboring cells.

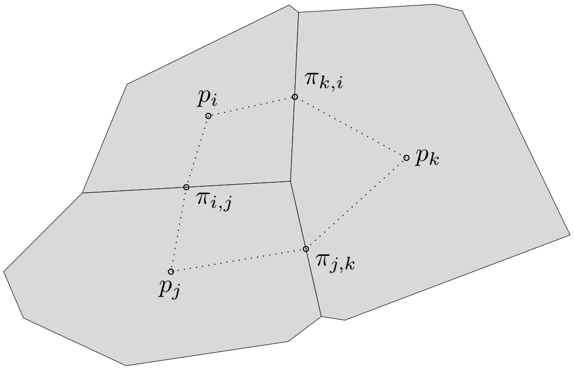


Figure 16—Illustration of the MPFA interaction region.

where $\boldsymbol{v}$ is Darcy velocity, p fluid pressure, K permeability tensor, and q is a source or sink term. For reasons that soon will be clear, the flow equation has been written on mixed first-order elliptic form that has p and $\boldsymbol{v}$ as unknowns, but the flow equation can also be written as a second-order, Poisson-type equation $-\nabla \cdot \mathsf{K}\nabla p = q$ for pressure only. The pressure equations essentially remains the same for multiphase flow, except that K is multiplied by a mobility function λ that depends on phase saturation(s). In the following, we assume that we have a grid $\mathcal{T}_h = \cup_{i=1}^{N} \Omega_i$ consisting of N polygonal or polyhedral cells Ω_i covering Ω, and that the permeability K is constant on each cell. Moreover, we will assume that suitable boundary conditions are given.

Finite-volume methods rely on the integral form of the first equation of Eq. 2,

$$\int_{\partial\Omega_i} \boldsymbol{v} \cdot \boldsymbol{n} \, \mathrm{d}s = \int_{\Omega_i} q \, \mathrm{d}\boldsymbol{x} = q_i, \tag{3}$$

and differ in the way they approximate the flux across the individual grid faces that make up $\partial\Omega_i$. Consider two cells Ω_i and Ω_j and let $F_{i,j}$ denote their common interface, as depicted in Fig. 15. This interface can be interpreted as two *half-faces*: The half-face $F_{i,j}$ is the common face of cell Ω_i and cell Ω_j, with orientation given by the normal vector $\boldsymbol{n}_{i,j}$ pointing from cell Ω_i to cell Ω_j. In the same manner, $F_{j,i}$ shares vertices and area with $F_{i,j}$, but is oriented with $\boldsymbol{n}_{j,i} = -\boldsymbol{n}_{i,j}$. We denote the flux over half-face $F_{i,j}$ by $v_{i,j}$. With this notation set, we can start presenting the different schemes.

Two-point flux-approximation The two-point flux approximation (TPFA) should be well-known to most readers, but we still introduce it in some detail to prepare the presentation of consistent schemes. Introducing an auxiliary pressure point $\pi_{i,j}$ at the centroid of the half-face $F_{i,j}$, we can use the midpoint rule to approximate the integral across $F_{i,j}$ with a two-point finite-difference to approximate Darcy's law,

$$v_{i,j} = \int_{F_{i,j}} \boldsymbol{v} \cdot \boldsymbol{n}_{i,j} \approx |F_{i,j}| \frac{\boldsymbol{c}_{i,j}^{\mathsf{T}} \boldsymbol{K}_i \boldsymbol{n}_{i,j}}{|\boldsymbol{c}_{i,j}|^2} (p_i - \pi_{i,j}) = T_{i,j}(p_i - \pi_{i,j}). \tag{4}$$

The proportionality constant $T_{i,j}$ is referred to as the half-face transmissibility. We now impose continuity of fluxes across faces, $v_{ij} = v_{i,j} = -v_{j,i}$, and continuity of face pressures, $\pi_{ij} = \pi_{i,j} = \pi_{j,i}$. This gives the system

$$\sum_{j=1}^{N} T_{ij}(p_i - p_j) = q_i \quad \forall \Omega_i \in \mathcal{T}_h, \qquad T_{ij} = \left(T_{i,j}^{-1} + T_{j,i}^{-1}\right)^{-1}, \tag{5}$$

where T_{ij} is the transmissibility associated with the connection between Ω_i and Ω_j. Although robust and monotone, the TPFA method is only consistent for K-orthogonal grids, in which $\boldsymbol{K}_i \boldsymbol{n}_{i,j}$ is parallel to $\boldsymbol{c}_{i,j}$ for all cells. The method can therefore suffer from significant grid-orientation effects (Aavatsmark 2007; Wu and Parashkevov 2009).

Multipoint flux approximation To obtain a consistent discretization of Eq. 2 for grids that are not K-orthogonal, we need accurate approximations of the pressure derivative parallel to the cell faces. To develop a multipoint stencil (here we only discuss the MFPA-O method), we start by defining an interaction region around each node in the grid, restricted by the faces that connect the cell centroids and the face centroids adjacent to the node, as shown in Fig. 16. Inside these interaction regions, we define a set of linear basis functions for the pressure, and impose pressure continuity at the face centroids and flux continuity across the face patches inside the interaction region. A globally coupled system is then obtained by using continuity requirements to express the

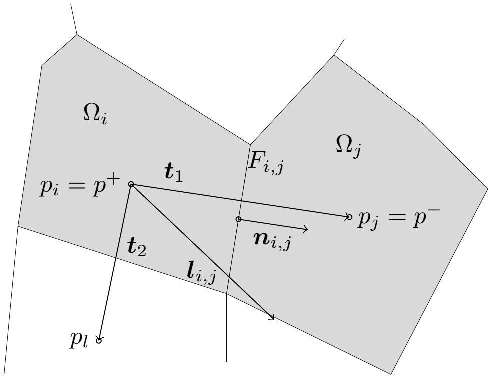


Figure 17—Decomposition of $\boldsymbol{l}_{i,j} = \boldsymbol{K}_i\boldsymbol{n}_{i,j}$ used to develop the NTPFA scheme.

gradients of the basis functions in terms of the cell pressures p_i, giving fluxes across the face patches of the interaction region. Finally, cell pressures are determined by summing the fluxes across all face patches, and imposing mass conservation. This yields a method that is consistent for all grids. The cost for this unconditional consistency is that we have to solve a much denser linear system of equations than the one obtained from the TPFA scheme. The interested reader is referred to Aavatsmark (2002) for a more general discussion of MPFA schemes.

Nonlinear two-point flux approximation The multipoint flux approximation achieves consistency by approximating inter-cell fluxes using more than two points. The nonlinear two-point flux approximation (NTPFA) method (Le Potier 2009; Lipnikov et al. 2007, 2009; Nikitin et al. 2014b) also uses additional points to estimate fluxes, but the fluxes themselves are a result of a two-point expression on the form

$$v_{i,j} = T_i(\boldsymbol{p})p_i - T_j(\boldsymbol{p})p_j,$$

where the transmissibilities depend on one or more pressure values, giving a nonlinear method. To derive a nonlinear two-point scheme, we consider the vector $\boldsymbol{l}_{i,j}$,

$$\boldsymbol{l}_{i,j} = \boldsymbol{K}_i\boldsymbol{n}_{i,j}.$$

This vector is shown in Figure 17 for some permeability that significantly changes the flow direction. Approximating the vector $\boldsymbol{l}_{i,j}$ using only the component of the vector normal to the interface, as done in the linear two-point scheme, will introduce significant errors. Instead, we define a decomposition of $\boldsymbol{l}_{i,j}$ onto a basis of d vectors with positive weights,

$$\boldsymbol{l}_{i,j} = |\boldsymbol{l}_{i,j}|\sum_{k=1}^{d} c_k \frac{\boldsymbol{t}_k}{|\boldsymbol{t}_k|}, \qquad c_k > 0\,\forall\, k,$$

where $\boldsymbol{t}_i$ is the vector from the cell center to another point where we assume to know the pressure values. We derive an expression for the flux over the interface F_{ij} as seen from cell Ω_i (i.e., flux across the half-face $F_{i,j}$,

$$v_{i,j} = \int_{F_{i,j}} \boldsymbol{v}\cdot\boldsymbol{n}_{i,j} = -\int_{F_{i,j}} \nabla p\cdot(\boldsymbol{K}\boldsymbol{n}_{i,j}) = -\int_{F_{i,j}} |\boldsymbol{l}_{i,j}|\frac{\partial p}{\partial \boldsymbol{l}_{i,j}} = -\int_{F_{i,j}} |\boldsymbol{l}_{i,j}|\sum_{k=1}^{d} c_i^+ \frac{\partial p}{\partial \boldsymbol{t}_k^+} \approx |F_{ij}||\boldsymbol{l}_{i,j}|\sum_{k=1}^{d} c_i^+ \frac{p_k - p^+}{|\boldsymbol{t}_k^+|},$$

where we have assumed that $\boldsymbol{K}$ is a constant tensor. Similarly, we derive a flux $v_{j,i}$ for the flux as seen from cell Ω_j (i.e., flux across the half-face $F_{j,i}$). The flux is then defined as convex combination of these two flux expressions,

$$v_{ij} = v_{i,j}\mu^+ - v_{j,i}\mu^- = |F_{ij}||\boldsymbol{l}_{i,j}|\left[\mu^+\sum_{k=1}^{d}\frac{c_k^+}{|\boldsymbol{t}_k^+|}p_k - \mu^-\sum_{k=1}^{d}\frac{c_k^-}{|\boldsymbol{t}_k^-|}p_k - \mu^+\sum_{k=1}^{d}\frac{c_k^+}{|\boldsymbol{t}_k^+|}p^+ + \mu^-\sum_{k=1}^{d}\frac{c_k^-}{|\boldsymbol{t}_k^-|}p^-\right],$$

where the weights are chosen such that the weighting of fluxes is not biased (i.e., $\mu^+ + \mu^- = 1$) and that the terms that do not correspond to pressures on either side of the interface cancel,

$$\mu^-\sum_{k=1}^{d}\frac{c_k^-}{|\boldsymbol{t}_k^-|}p_k = \mu^+\sum_{k=1}^{d}\frac{c_k^+}{|\boldsymbol{t}_k^+|}p_k.$$

This gives a consistent two-point flux approximation for a given pressure, where the transmissibilities themselves depend on pressure values not included in the two-point stencil. Note that if $\boldsymbol{K}$ varies, the flux expression is instead found by constructing a half-face flux with a face pressure that can be subsequently eliminated by assuming continuity.

Mimetic finite differences TPFA and MPFA are special cases of a wider family of mass-conservative schemes that for a cell Ω_i with n faces can be written on the form

$$\boldsymbol{v}_i = \boldsymbol{T}_i(\boldsymbol{e}_i p_i - \boldsymbol{\pi}_i).$$

Here, $\boldsymbol{v}_i = (v_{i1}, \dots, v_{in})^{\mathsf{T}}$ is the vector of fluxes across the faces of Ω_i, $\boldsymbol{e}_i = (1, \dots, 1)^{\mathsf{T}} \in \mathbb{R}^n$, $\boldsymbol{\pi}_i$ is the vector of face pressures, and $\boldsymbol{T}_i$ is a matrix of half-face transmissibilities. Looking at Eq. 4, we see that this is indeed true for the TPFA method, in which case $\boldsymbol{T}_i$ is the diagonal matrix with scalar entries $(T_{i,1}, \dots, T_{i,n})$. However, in the general case, $\boldsymbol{T}_i$ is a full matrix. We write this as

$$\boldsymbol{M}_i \boldsymbol{v}_i = \boldsymbol{e}_i p_i - \boldsymbol{\pi}_i, \tag{6}$$

where $\boldsymbol{M}_i = \boldsymbol{T}_i^{-1}$ can be associated to the local inner products,

$$\int_{\Omega_i} \boldsymbol{u}^{\mathsf{T}} \boldsymbol{K}_i^{-1} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}. \tag{7}$$

If we now add pressure and flux continuity as separate equations, we obtain the following discrete system,

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{D}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix}. \tag{8}$$

Here, $\boldsymbol{v}$ are the outward fluxes for each half-face of the cell. This gives that both v_{ij} and $-v_{ij}$ appears in $\boldsymbol{v}$ for each face F_{ij}. In the same manner, $\boldsymbol{p}$ and $\boldsymbol{\pi}$ contain the cell and face pressures. The matrix $\boldsymbol{B}$ is block diagonal, with diagonal entries $\boldsymbol{M}_1, \dots, \boldsymbol{M}_N$, and $\boldsymbol{C}$ is a block diagonal matrix with diagonal $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_N)$. Finally, a column of $\boldsymbol{D}$ corresponds to a unique face in the grid, and has two unit entries for each face. We refer to Eq. 8 as the *hybrid formulation* of Eq. 2, see e.g., (Lie et al. 2012) for details.

The idea of a k-th order mimetic finite difference (MFD) method (Brezzi et al. 2005; da Veiga et al. 2014) is to construct approximations of the local inner products $\boldsymbol{M}_i$ so that these are exact whenever one of the velocities $\boldsymbol{u}$ or $\boldsymbol{v}$ in Eq. 7 is a vector of polynomials of degree k or less. When none of the velocities are such a polynomial, we only approximate the local inner product to the right order. We refer to these properties as the *consistency* and *stability* properties of the method.

In a first-order mimetic method, the pressure is approximated by a piecewise linear function on each cell, such that $p_i = \boldsymbol{x} \cdot \boldsymbol{a} + b$ for some vector $\boldsymbol{a}$ and constant b. The flux across $F_{i,j}$ and pressure drop from the cell centroid to the face centroid are then

$$v_{ij} = -|F_{i,j}| \boldsymbol{n}_{i,j}^{\mathsf{T}} \boldsymbol{K}_i \boldsymbol{a}, \quad p_i - \pi_{ij} = \boldsymbol{c}_{i,j}^{\mathsf{T}} \boldsymbol{a}.$$

We insert this into Eq. 6 to obtain the following consistency conditions for the local inner products:

$$\boldsymbol{M}_i \boldsymbol{N}_i \boldsymbol{K}_i = \boldsymbol{C}_i, \quad \boldsymbol{N}_i \boldsymbol{K}_i = \boldsymbol{T}_i \boldsymbol{C}_i, \tag{9}$$

where

$$\boldsymbol{N}_i = (|F_{i,1}| \boldsymbol{n}_{i,1}, \dots, |F_{i,n}| \boldsymbol{n}_{i,n})^{\mathsf{T}}, \quad \boldsymbol{C}_i = (\boldsymbol{c}_{i,1}, \dots, \boldsymbol{c}_{i,n})^{\mathsf{T}}.$$

For simplicity, we henceforth omit cell index i. From this, we can derive that any valid first-order mimetic approximation of the inner products can be written as

$$\boldsymbol{M} = \frac{1}{|\Omega_i|} \boldsymbol{C} \boldsymbol{K}^{-1} \boldsymbol{C}^{\mathsf{T}} + \boldsymbol{M}_s,$$

where $\boldsymbol{M}_s$ is a matrix defined such that $\boldsymbol{M}$ is consistent and symmetric positive definite. In a sense, it is trough the term $\boldsymbol{M}_s$ that we *mimic* the physical and mathematical properties of the problem. Since we are free to construct this term, we can obtain certain other methods such as TPFA and MPFA as special cases of the MFD formulation. A thorough description of first-order mimetic formulations of Eq. 8 is given by Lie et al. (2012), who also discuss possible choices for the stability term $\boldsymbol{M}_s$. Some of these are also given in the appendix.

Note that we initially use the cell pressures, face pressure and face fluxes as degrees of freedom. However, as explained by Lie et al. (2012), we can use the Schur complement of Eq. 8 to reduce it to a system with only the face pressures as unknowns. The cell pressures and face fluxes can then be reconstructed by back substitution.

The Virtual Element Method Higher-order mimetic methods can be obtained by adding degrees of freedom associated to quantities such as moments over cells and faces. However, the construction of such methods has proved cumbersome. Recently, it has become clear that if we disregard the requirement of explicit representation of basis functions in the cell interior, it is possible to obtain much simpler discretization methods, and a more unified framework for higher-order methods. This has led to a finite-element–like approach called the virtual element method (VEM) (Beirão da Veiga et al. 2013, 2014).

We combine Eq. 2 into an elliptic equation:

$$-\nabla \cdot \mathsf{K} \nabla p = q, \tag{10}$$

and assume, for simplicity, that we have homogeneous Dirichlet boundary conditions. Multiplying Eq. 10 by a function $v \in H_0^1(\Omega)$ and integrating over Ω yields the weak formulation: Find $p \in H_0^1(\Omega)$ such that

$$a(p, v) = \int_\Omega \nabla p \cdot \mathbf{K} \nabla v \, \mathrm{d}\boldsymbol{x} = \int_\Omega q v \, \mathrm{d}\boldsymbol{x}, \qquad \forall v \in H_0^1(\Omega). \tag{11}$$

Note that a can be split into a sum of local bilinear forms as

$$a(u, v) = \sum_{\Omega_i \in \mathcal{T}_h} \int_{\Omega_i} \nabla u \cdot \mathbf{K} \nabla v \, \mathrm{d}\boldsymbol{x} = \sum_{\Omega_i \in \mathcal{T}_h} a^i(u, v).$$

We will not go into the details of constructing the virtual element function spaces for this problem, and the interested reader is referred to Klemetsdal (2016); Beirão da Veiga et al. (2013); Ahmad et al. (2013); Beirão da Veiga et al. (2014).

As for MFD methods, the idea of a kth order virtual element method is to approximate the bilinear form such that it satisfies the same kind of consistency and stability properties. However, unlike in MFD, we do not know the exact expression of our basis functions. Instead, we define our virtual element function space, in which the trial and test functions are contained, such that it consists of all polynomials of degree k or less, along with some possibly non-polynomial functions. Moreover, we choose the degrees of freedom such that when one of the entries is a polynomial of degree k or less, the approximate bilinear form can be computed exactly *using only the degrees of freedom*. As we will see, we never actually compute the non-polynomial basis functions, but rather, their polynomial projection, which we can calculate from the degrees of freedom. Hence, the term *virtual*.

To construct an approximate bilinear form satisfying these properties, we apply the following Pythagoras identity:

$$a^i(u, v) = a^i\left(\mathit{\Pi} u, \mathit{\Pi} v\right) + a^i\left(u - \mathit{\Pi} u, v - \mathit{\Pi} v\right),$$

where $\mathit{\Pi} v$ is an a^i-orthogonal projection of v onto the space of polynomials of degree k or less. We then choose our approximate bilinear form to be

$$a_h^i(u, v) = a^i\left(\mathit{\Pi} u, \mathit{\Pi} v\right) + s^i\left(u - \mathit{\Pi} u, v - \mathit{\Pi} v\right),$$

where s^i a stability term. To obtain a discrete version of Eq. 11, we cleverly define our local virtual element function space and its degrees of freedom such that we are able to calculate the projection $\mathit{\Pi} v_h$ for all functions v_h in this function space. We then proceed to choose the basis for this function space to be the functions $\{\phi^i\}$ such that the jth degree of freedom of ϕ^i is 1 if $i = j$, and zero otherwise. We now arrive at the following linear system:

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{q}, \quad \boldsymbol{A}_{i,j} = a_h(\phi^i, \phi^j) = \sum_{k=1}^{N} a^k(\mathit{\Pi}\phi^i, \mathit{\Pi}\phi^j) + s^k(\phi^i - \mathit{\Pi}\phi^i, \phi^j - \mathit{\Pi}\phi^j), \quad \boldsymbol{q}_i = \sum_{k=1}^{N} \int_{\Omega_k} q\phi^i \, \mathrm{d}\boldsymbol{x}.$$

We see that by construction, any choice of the stability term yields a consistent method. However, in order to ensure that the stability property is satisfied, we must choose s^i such that a_h^i is positive definite, and scales as a^i. As for MFD, since we are free to choose this stability term, we can obtain certain other numerical methods from VEM. This is for instance true for FEM and some finite difference methods for certain cell geometries. An investigation of the stability term was done by Klemetsdal (2016), and some of these results are summarized in the appendix.

Flux reconstruction From the construction of our approximate bilinear form a_h^i, we know that it is exact whenever one of the entries is a polynomial of degree k or less. By choosing the polynomials to be x, y and z in turn, we thus obtain

$$\int_{\Omega_i} \boldsymbol{K}_i \nabla p \, \mathrm{d}\boldsymbol{x} = \left(a^i(p, x), a^i(p, y), a^i(p, z)\right)^\mathsf{T} = \left(a_h^i(p, x), a_h^i(p, y), a_h^i(p, z)\right)^\mathsf{T}.$$

By the construction of the virtual element function space, we know that we can calculate this exactly when p is the approximated pressure. We use this to approximate the cell velocities

$$-(\mathbf{K}\nabla p)|_{\Omega_i} \approx -\frac{1}{|\Omega_i|} \int_{\Omega_i} \boldsymbol{K}_i \nabla p \, \mathrm{d}\boldsymbol{x} = \boldsymbol{v}_i.$$

The velocity over half-face $F_{i,j}$ is then taken to be the weighted average

$$\boldsymbol{v}_{ij} = \theta_{i,j}\boldsymbol{v}_i + (1 - \theta_{i,j})\boldsymbol{v}_j,$$

where $\theta_{i,j}$ are coefficients such that $0 < \theta_{i,j} < 1$. Herein, we make the simple choice $\theta_{i,j} = 1/2$. An alternative is to follow Burman and Zunino (2006) and Ern et al. (2008) and choose

$$\theta_{i,j} = \frac{\delta_{i,j}}{\delta_{i,j} + \delta_{j,i}}, \quad \delta_{i,j} = \boldsymbol{n}_{i,j}^\mathsf{T} \boldsymbol{K}_i \boldsymbol{n}_{i,j}. \tag{12}$$

We can now calculate the flux over a half-face $F_{i,j}$ as

$$v_{i,j} = \begin{cases} |F_{i,j}| \boldsymbol{v}_{ij} \cdot \boldsymbol{n}_{i,j}, & F_{i,j} \text{ is an internal face,} \\ |F_{i,j}| \boldsymbol{v}_{i} \cdot \boldsymbol{n}_{i,j}, & F_{i,j} \text{ is a boundary face.} \end{cases}$$

Here, we use the convention that $F_{i,j}$ is a boundary face if either i or j is zero. Naturally, this calculation is not needed for boundary faces with a prescribed flux.

Postprocessing Whereas TPFA, MPFA, NTPFA, and MFD are locally conservative, VEM is not, and may potentially lead to unphysical states when applied as part of a multiphase simulation. To obtain conservative fluxes, we will postprocess the non-conservative velocity field using a method proposed by Odsæter et al. (2016). In accordance with Eq. 3, we define the residual in cell Ω_i to be

$$r_i = q_i - \sum_{j=1}^{N} v_{i,j}.$$

We now want to find a correction function so that the fluxes are locally conservative. In particular, we want to determine a function H, constant on each face, so that $q_i = \sum_j (v_{i,j} + |F_{i,j}| H_{ij} \boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i,j})$, where H_{ij} denotes the value of H on the common face of Ω_i and Ω_j. The vector $\boldsymbol{n}_{ij}$ is a predetermined direction for the flux over this face, and is either $\boldsymbol{n}_{i,j}$ or $\boldsymbol{n}_{j,i}$, so that $\boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i,j} = \pm 1$. To uniquely determine H, we pick the one that minimizes a weighted L^2-norm over the faces. It can be shown that H can be expressed through a function y, constant on each cell, such that $H_{ij} \boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i,j} = \omega_{ij}^{-1}(y_i - y_j)$, where y_i is the value of y on cell Ω_i. The coefficients ω_{ij} are the weights of the weighted L^2-norm. This yields the system

$$q_i - \sum_{j=1}^{N} (v_{i,j} + |F_{i,j}| H_{ij} \boldsymbol{n}_{ij} \cdot \boldsymbol{n}_{i,j}) = r_i - \sum_{j=1}^{N} |F_{i,j}| \omega_{ij}^{-1} (y_i - y_j) = 0 \quad \forall \Omega_i \in \mathcal{T}_h. \tag{13}$$

Taking a closer look at Eq. 13, we see that it is similar to the TPFA scheme in Eq. 5. In fact, if we choose the weight function ω so that $|F_{i,j}|\omega_{ij}^{-1}$ equals the transmissibility T_{ij}, we recover the TPFA scheme. This is not as surprising as it might seem. Indeed, let us denote the exact velocity $\boldsymbol{v} = \boldsymbol{v}_h + \boldsymbol{h}$, where $\boldsymbol{v}_h$ is the approximated velocity and $\boldsymbol{h}$ is a correction term. Similarly, we denote the exact pressure by $p = p_h + r^p$. Inserting this into Eq. 2 yields

$$\nabla \cdot \boldsymbol{h} = q - \nabla \cdot \boldsymbol{v}_h, \quad \boldsymbol{h} = -\mathbf{K} \nabla r^p.$$

It is now easy to check that the TPFA formulation of this problem is equivalent to Eq. 13, with $|F_{i,j}|\omega_{ij}^{-1}$ equal to the transmissibilities T_{ij}. For our simulations, we follow Odsæter et al. (2016) and use the weights

$$\omega_{ij} = \frac{\delta_{i,j} + \delta_{j,i}}{2\delta_{i,j}\delta_{j,i}},$$

where the coefficients $\delta_{i,j}$ are given in Eq. 12.

We conclude this section with Fig. 18, showing a polygon with the degrees of freedom of the different methods indicated.

Numerical Simulations

We now consider a number of test cases to compare and contrast the various discretization methods discussed above. In the discussion, we will use different inner productions for the mimetic and the virtual element methods; for detailed definitions of these inner product see the appendix and (Lie et al. 2012).

Example 1 (Maximum principle)**.** A fundamental property of Eq. 2 is that it obeys an elliptic maximum principle, which implies that if there is a single source within the domain, the pressure will decrease monotonically towards the boundary. Discretization schemes should obey a similar maximum or monotonicity principle (Nordbotten et al. 2007). It is well-known that whereas the TPFA scheme is monotone, many consistent schemes like the MPFA-O and mimetic methods are not. To compare and contrast the various schemes, we consider the problem of a point source placed at the center of a quadratic domain, with zero pressure on the boundary. The permeability ratio is 500:1, rotated by an angle $\pi/12$. We compute the pressure fields on a Cartesian grid of 51×51 cells using the methods described above. For MFD and VEM, we have used all inner products described in the appendix. Contour plots of some of the methods are reported in Fig. 19. The MFD method shown uses the inner product M1 that resembles the mixed Raviart–Thomas inner product. The VEM methods shown uses the inner products V1 and V4, where the latter resembles a finite difference stencil. The weight w for the finite difference stencil is chosen trough a minimization of the absolute value of the negative cell pressures, resulting in $w = 0.0499$, see the appendix for details.

We observe that the TPFA solution completely fails to capture the effect of the rotated permeability field. The MFD-M1 and VEM1-V4 solutions exhibit similar oscillations with regions of negative pressure that are symmetric with respect to the point

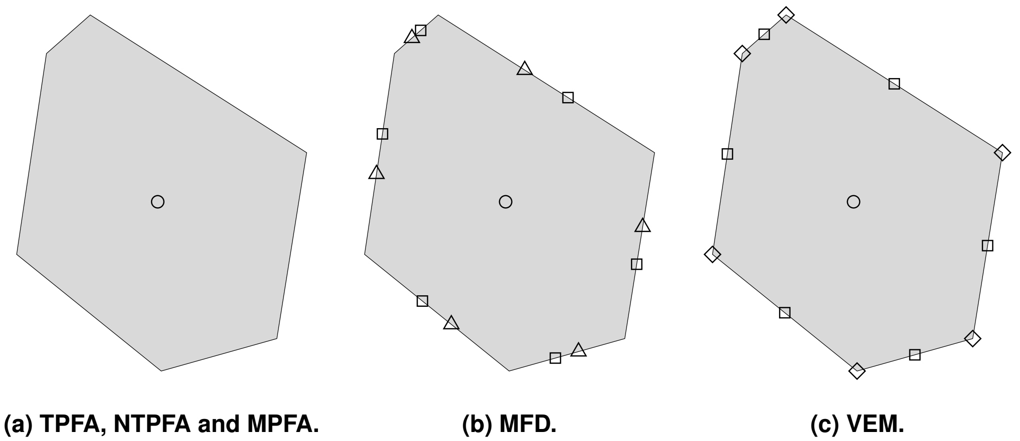


Figure 18—Visualization of the degrees of freedom of the different methods: Cell pressure ○, face pressure □, node pressure ◇, and face flux △. Note that all initial degrees of freedom for MFD are indicated, even though we only solve for the face pressures. For VEM, the degrees of freedom of a second-order method are shown. In a first-order method, only the nodal degrees of freedom are used.

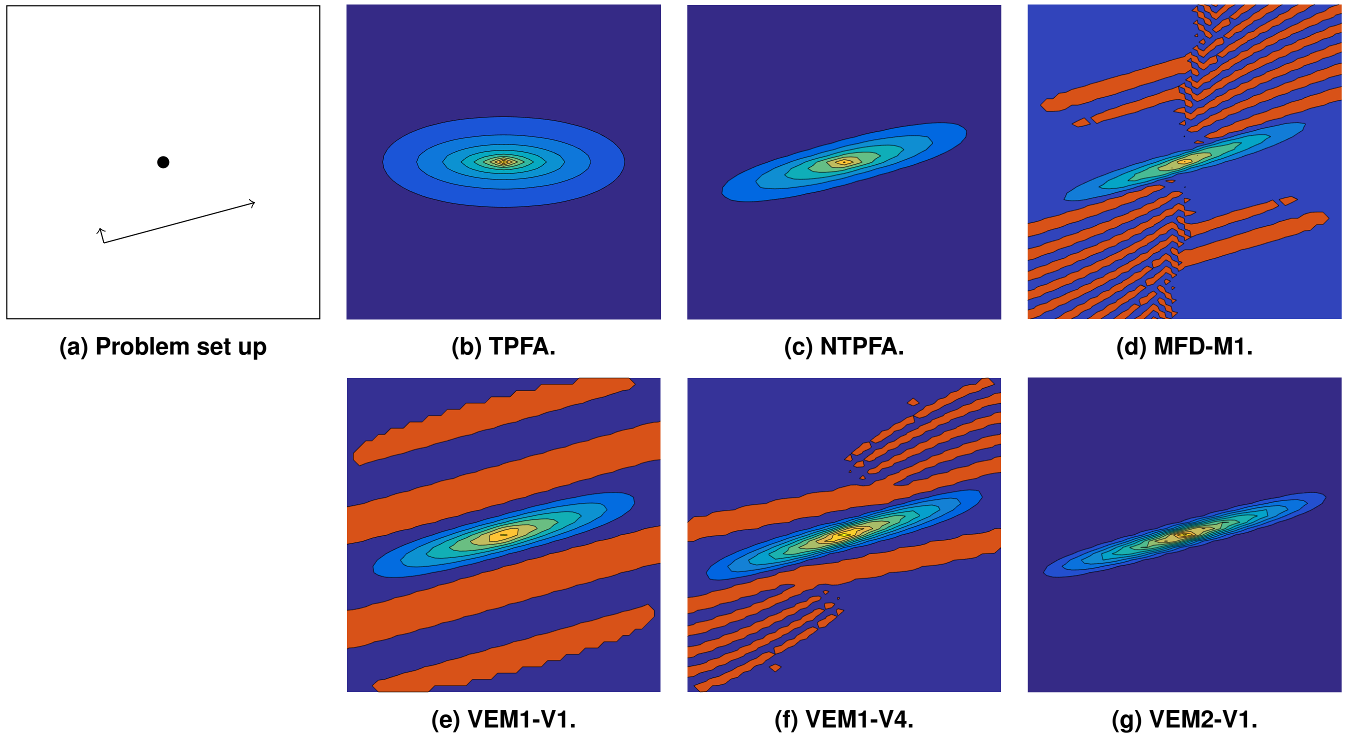
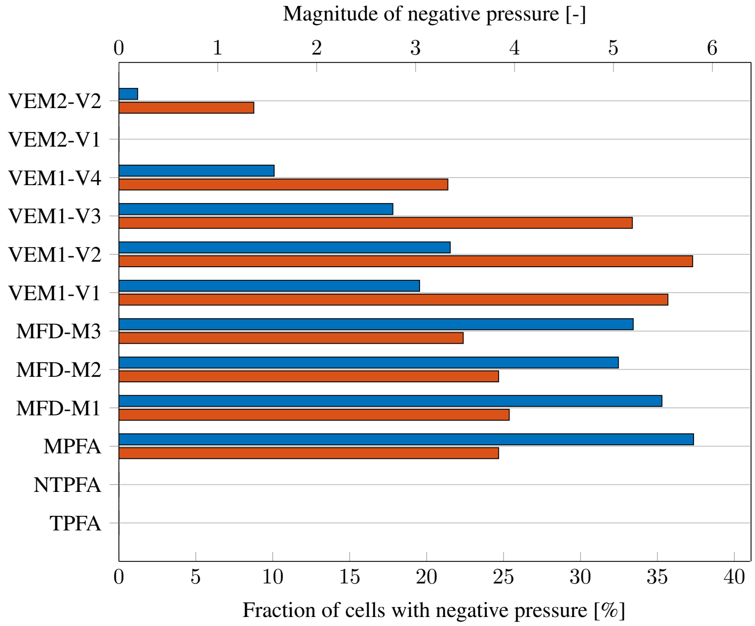


Figure 19—Contour plots of the pressure fields for the point source problem using different solution methods. Regions of negative pressure are indicated in red.

source. Oscillations are also evident in the first-order VEM solution using V1, but these appear in larger contiguous patches. The NTPFA solution is monotone by construction and thus free for oscillations. We also observe that the second-order VEM solution is completely free of regions with negative pressure. Moreover, we notice a sharper and more localized peak in the pressure towards the source in the MFD-M1, VEM1-V4, and VEM2-V1 solutions.

Fig. 20 reports a more comprehensive comparison of all the methods with different inner products used for MFD and VEM. In particular, we have indicated the fraction of cells with negative pressure for each method, along with a measure of the magnitude of the negative pressures, defined as

$$\mu = 100 \times \frac{\sum_{p_i<0} |p_i|}{\sum_i |p_i|},$$



dof	nnz	ratio	cond
10609	162817	15.35	1.31e+06
10609	162817	15.35	1.06e+06
2704	22704	8.40	5.57e+04
2704	22704	8.40	5.14e+04
2704	22704	8.40	6.32e+04
2704	22704	8.40	5.16e+04
5100	35096	6.88	4.73e+03
5100	35096	6.88	4.90e+03
5100	35096	6.88	6.66e+03
2805	23005	8.20	1.60e+03
2805	24221	8.63	1.19e+06
2601	12801	4.92	1.39e+03

Figure 20—Comparison of the different methods and inner products. Red bars show the fraction of cells with negative pressure. Blue bars show a measure of the magnitude of the negative pressures. More details about the various inner products (M1 to M3, V1 to V4) can be found in the appendix. In the tabular, `dof` denotes the number of primary unknowns, `nnz` the number of nonzero entries in the discretization matrix, `ratio` the ratio between `nnz` and `dof`, and `cond` the condition number.

where we recall that p_i is the pressure at the centroid of cell Ω_i. Note that strictly speaking, the cell pressure is not a degree of freedom for the second-order VEM method, but rather, the average of the pressure over the cell. This average cell pressure is not a degree of freedom for the first-order VEM method, but can be calculated using the nodal degrees of freedom. Hence, for the VEM methods, we use these average cell pressures as approximations of the cell pressures. On the right-hand side of the bar chart we have reported key figures about the resulting linear systems such as the number of unknowns (dof), the number of non-zero elements (nnz), the ratio between these two numbers (ratio), and the lower bound of the condition number (cond), defined by $\|\boldsymbol{A}\|_1/\|\boldsymbol{A}^{-1}\|_1$.

Since both the NTPFA and TPFA solutions are monotone by construction, these methods have no cells with negative pressure. However, as Fig. 19 clearly shows, the TPFA method completely fails to resolve the effect of the rotated anisotropy in the permeability field. Comparing the remaining solutions, the second-order VEM methods outperform the first-order methods both in terms of fraction and magnitude of negative cell pressures, with the inner product V1 being the best. Except for the V4 inner product, the first-order VEM solutions have the highest fraction of negative cell pressures, but the negative cell pressures have lower magnitude than for MFD and MPFA. Notice that the method using the V3 inner product, which resembles a standard finite-element method, is slightly better than V1 and V2 in both measures, while the method using V4, which resembles a finite-difference method, is the best among all the linear first-order methods. All MFD solutions yield similar results, with M2 and M3 being slightly better than M1. The M3 inner product corresponds to a Raviart–Thomas method on cuboids, while M2 corresponds to the TPFA method on K-orthogonal grids. The MPFA-O solution is comparable to the MFD solutions. We conclude that for this particular test case, the second-order method yields far better results in terms of physically meaningful pressure fields.

Finally, let us look in some more detail at the linear(ized) systems arising from the different schemes. Referring back to Fig. 18, we see that the cell-centered TPFA method will have $51^2 = 2601$ degrees of freedom. The MPFA and NTPFA methods are cell-centered like TPFA, but both methods introduce extra face pressures along the model perimeter to handle boundary conditions and therefore get a higher number of degrees of freedom ($51^2 + 4 \cdot 51 = 2805$). Since the MPFA method uses a multipoint approximation for each intercell flux, the corresponding stencil is denser (almost twice as many nnzs per degree of freedom as TPFA), but the condition number is only slightly worse. The NTPFA method has the same flux expression as TPFA, but since the transmissibilities depend on pressure, the resulting sparsity pattern is similar to the MPFA matrix. The condition number reported is averaged over all iterations since the coefficients of the stencil vary with each nonlinear iteration. For the mimetic method, we use a Schur complement technique to reduce Eq. 8 to a positive-definite system for face pressures. Hence, the scheme has $2 \cdot 51 \cdot 52 - 4 \cdot 51 = 5100$ degrees of freedom. The stencil is less dense than for MFPA, but gives somewhat worse condition numbers. First-order VEM is vertex centered and hence has $52^2 = 2704$ degrees of freedom, which is only slightly more than TPFA and less than MPFA/NTPFA. The stencil is slightly denser than MPFA and gives worse condition

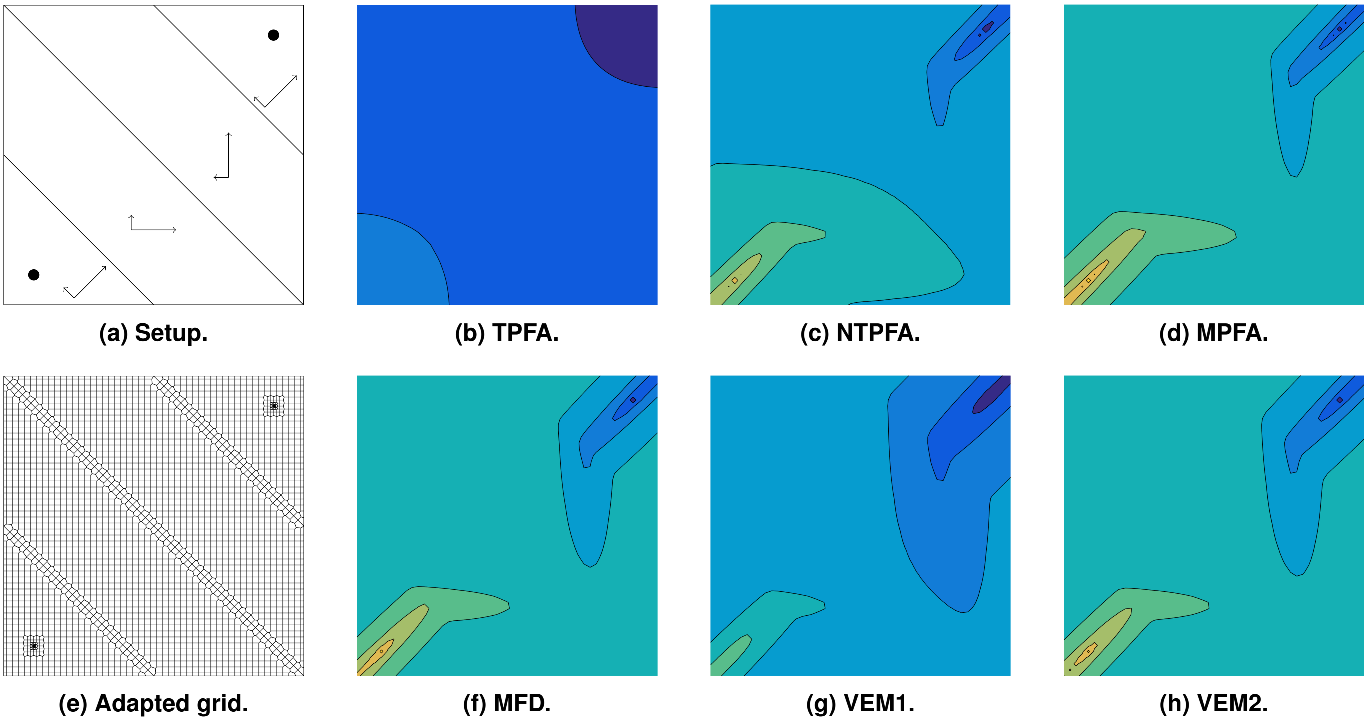

(a) Setup.
(b) TPFA.
(c) NTPFA.
(d) MPFA.
(e) Adapted grid.
(f) MFD.
(g) VEM1.
(h) VEM2.

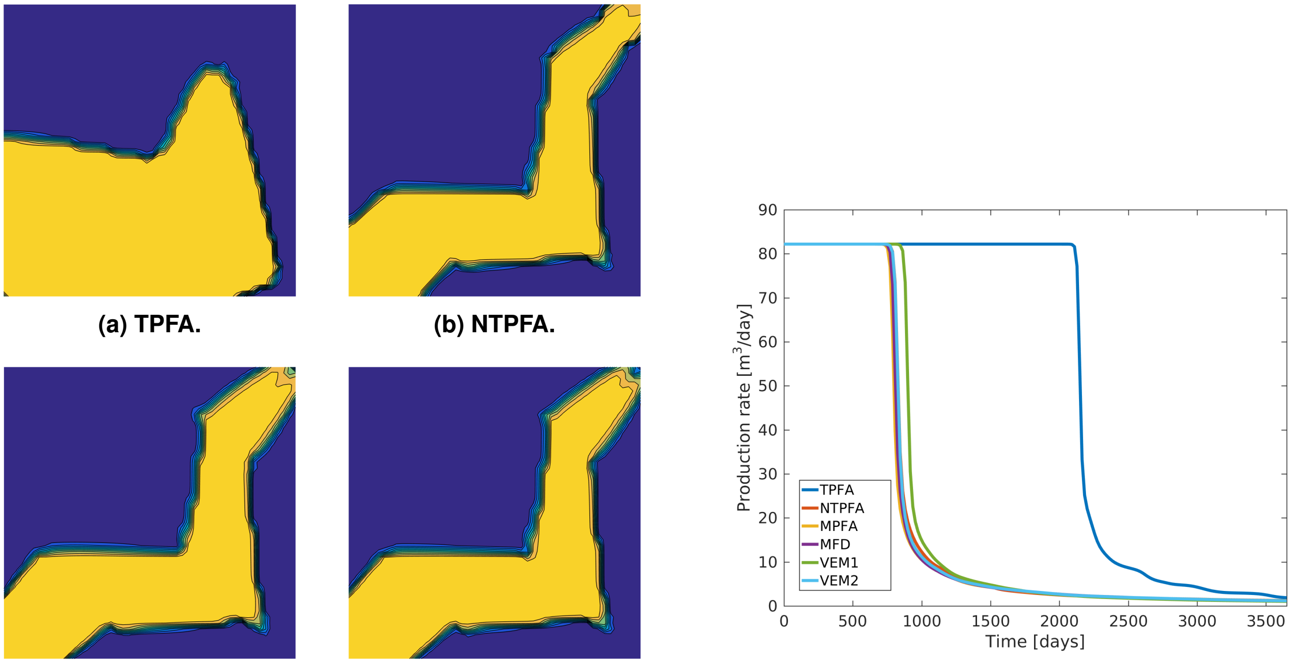
Figure 21—Setup and pressure profiles for the injector/producer problem.

numbers than the other schemes. Second-order VEM has degrees of freedom at vertices plus cell and face centers, giving a total of $52^2 + 51^2 + 2 \cdot 51 \cdot 52 = 10609$ degrees of freedom. The resulting stencil is three times denser than the TPFA stencil and has three orders of magnitude worse condition number. In summary, we see that the condition numbers for VEM are significantly higher than for the other methods, in particular for the second-order VEM methods.

Example 2 (Injector/producer)**.** Next, we consider an example examined earlier by Nikitin et al. (2014a) for nonlinear TPFA methods. The problem is posed on a unit square of 1000 m $\times$1000 m, with fluid flowing from an injector placed in the lower left corner, to a producer placed in the upper right corner. The domain is divided into four regions of different permeability. In all regions, the permeability tensor is $\mathbf{K} = \mathrm{diag}(1000, 10)\times$ mD, so that we have an anisotropy ratio of 100:1. The tensor is rotated by an angle $\pi/4$ in the fist and last regions, not rotated in the second region, and rotated by $\pi/2$ in the third region. We impose no-flow boundary conditions. The setup and the corresponding grid are shown in Fig. 21. Notice how the grid aligns to the boundaries defining the different permeability regions, with refinement around the wells. Fig. 21 also shows the resulting pressure profiles computed with TPFA, NTPFA, MPFA, MFD, VEM1, and VEM2. We have used inner products M1 and V1 for MFD and VEM, respectively. The pressure profiles are almost identical for NTPFA, MPFA, MFD, and VEM2. The first-order VEM method gives the same pressure contours, but with a less sharp pressure gradient. Again, the TPFA method suffers from grid-orientation effects.

To investigate the transport properties of the resulting flux fields, we consider a simple two-phase flow problem in which the flow field is only computed once and not updated to account for changes in total mobility. Initially, the reservoir is filled with oil, and we inject water at a constant rate. Fig. 22 reports the resulting saturation profiles for some of the methods. The saturation field predicted by TPFA is completely different from the other methods. Since TPFA is not able to resolve the effects of anisotropy, it grossly exaggerates the areal sweep and hence predicts that the displacement profile will break through much later than for the other methods. The remaining saturation plots are qualitatively similar, but we see that VEM1 gives a displacement profile that is slightly more smeared out orthogonal to the flow directions than the other methods, and hence VEM1 will give later breakthrough and better sweep. This is confirmed in Fig. 22e, which reports the predicted production rates for the different methods. We see that all production curves show decay in production after water breaks through, and that the breakthrough occurs much later for the TPFA method. **Table 1** reports various characteristics of the resulting discretization matrices for the five solution methods. The relative differences in the number of unknowns and in the sparsity pattern are the same as in the previous example. In terms of condition numbers, MPFA is now worse and first-order VEM better, seen relative to the other methods.

Example 3 (Different gridding techniques)**.** As we have seen, there are a number of gridding techniques that can be used to represent a reservoir. In this example, we consider a reservoir with two internal boundaries, one injector, and two producers. The permeability tensor is isotropic, and we impose no-flow boundary conditions on all boundaries. We represent this reservoir using four different gridding approaches: *(i)* a classical Cartesian grid, *(ii)* a triangle grid, *(iii)* a composite PEBI-grid, and *(iv)* a fully



(a) TPFA.

(b) NTPFA.

(c) VEM1.

(d) VEM2.

(e) Oil production rates.

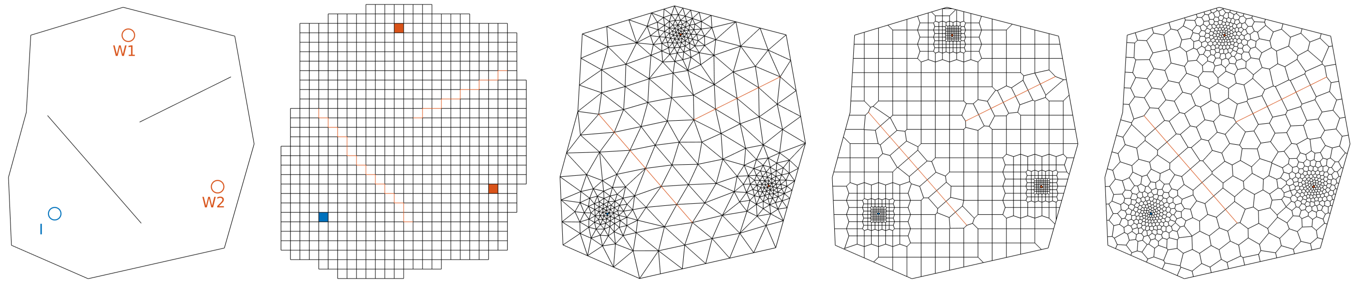
Figure 22—Saturation and oil production profiles for the injector/producer problem. Saturation profiles for MPFA and MFD are almost identical to VEM 2 and hence not reported.

Table 1—Key characteristics of the discrete system for the producer/injector case. Here, `dof` denotes the number of primary unknowns, `nnz` the number of nonzero entries in the discretization matrix, `ratio` the ratio between `nnz` and `dof`, and `cond` the condition number.

	dof	nnz	ratio	cond
TPFA	2698	14110	5.23	8.12e+04
NTPFA	2898	23531	8.12	1.19e+06
MPFA	2898	23535	8.12	1.06e+07
MFD	5906	45598	7.72	1.11e+06
VEM1	3209	31489	9.81	4.41e+05
VEM2	11813	206017	17.44	1.73e+07

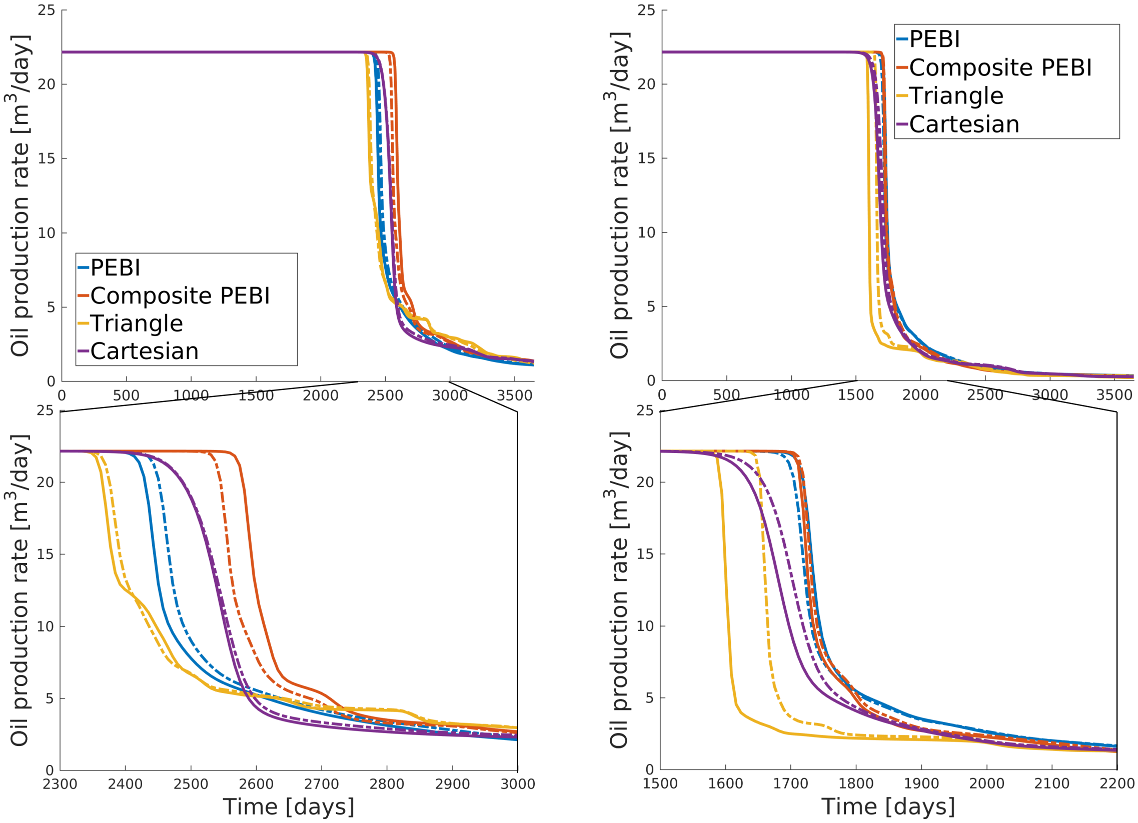
unstructured PEBI-grid. The setup and the different grids are shown in Fig. 23. Notice that the Cartesian grid does not align with the external or internal boundaries.

A consistent method should produce similar results for all grids. While pressure profiles can be quite similar for different grids, differences in flux fields are usually easier to detect. As in the previous example, we therefore consider a simple two-phase flow problem, in which the flow field is only computed once. The reservoir is initially filled with oil and we inject water at a constant rate. The resulting production curves predicted by TPFA and MFD for the two producers are shown in Fig. 24, where we have used the M1 inner product for MFD. To better compare the production rates in the two wells, the figure presents zooms of the production curves for a period of 700 days around the time when the different methods predict water breakthrough. For W1, oil rates computed on the PEBI-grids follow a somewhat similar pattern, but with a significant difference in time to water breakthrough. The slowest decay in the production is predicted by both methods on the triangle grid. The lowest production rate is predicted by TPFA on the triangle grid, while the highest production rate is predicted by the same method on the composite PEBI-grid. Altogether, there are larger differences between different grids than between TPFA and MFD on the same grid. However, the deviation is smaller for MFD. The best correspondence between the two methods is observed for the Cartesian grid, as expected. For W2, there is a close correspondence between the TPFA and MFD solutions for the PEBI and composite PEBI-grids. The rates predicted on the triangular grid deviate significantly from those predicted on the two PEBI curves, but are similar for both method. The same is evident for the Cartesian grid curves. The lowest production is predicted by TPFA on the triangle grid, while the highest production is predicted by TPFA on the PEBI-grid. In both wells, the transition period is more smeared out on the Cartesian grid. Looking at the production curves for well W1, there are large differences in both the TPFA and MFD solutions between the various grids. This indicates that error caused by a (too) low number of cells in the region between



(a) Setup. **(b) Cartesian grid.** **(c) Triangle grid.** **(d) Composite PEBI-grid.** **(e) PEBI-grid.**

Figure 23—Setup and grids for Example 3. The reservoir consists of an injection well I and two producers W1 and W2



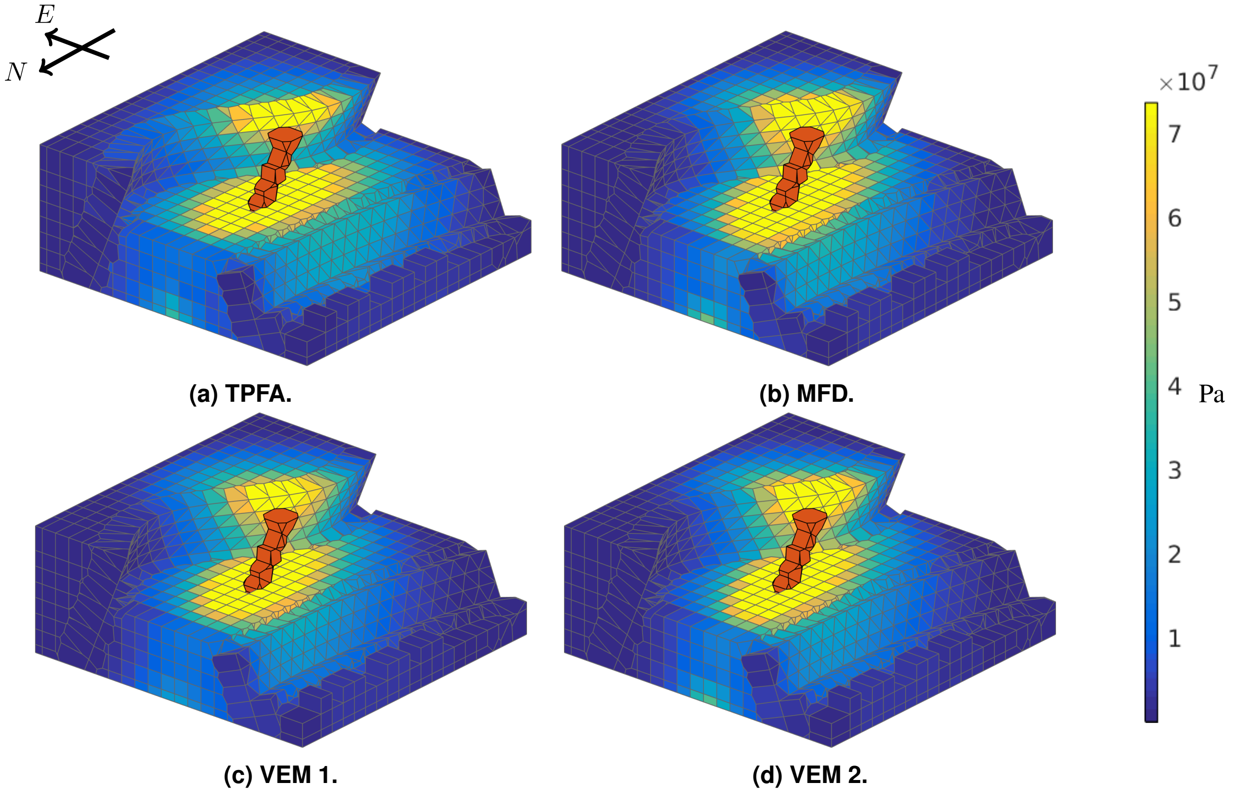
(a) Production in W1. **(b) Production in W2.**

Figure 24—Production curves for the two wells in Example 3. Solid lines are the TPFA solutions, while dotted lines are MFD solutions.

the end of the leftmost internal boundary and the model perimeter dominates the error in the flow path connecting wells I and W1. Moreover, we observe that there is a close correspondence between all production curves of the PEBI-grids in W2. This indicates that the PEBI-grids are favorable for the TPFA method, since these grids are designed to preserve orthogonality of interfaces for isotropic permeability fields. Overall, however, we conclude that both methods are subject to grid effects, with TPFA differing more than MFD between the various grids. Moreover, the PEBI-grid solution for both methods are very similar.

Example 4 (Reservoir with faults)**.** We consider a case in which two curved and sloping faults divide a box-shaped reservoir into three regions with very different permeability. The grid is shown in Fig. 13, with well cells indicated in red. The permeability field is constructed starting from a lognormal, isotropic realization. This gives a scalar value k_i for the permeability in cell Ω_i. Regions are numbered from left to right, and we set permeability tensor in each cell to be

$$\boldsymbol{K}_i = k_i \boldsymbol{R}(\theta_j) \boldsymbol{D}_j \boldsymbol{R}(\theta_j)^{\mathsf{T}},$$



(a) TPFA.

(b) MFD.

(c) VEM 1.

(d) VEM 2.

Figure 25—Pressure fields for Example 4, obtained from four different methods.

where $\boldsymbol{D}_j$ is a diagonal matrix scaling the permeability in each of the axial directions in region j, with

$$\boldsymbol{D}_1 = \mathrm{diag}(10, 1, 1), \quad \boldsymbol{D}_2 = \mathrm{diag}(1, 10, 1), \quad \boldsymbol{D}_3 = \mathrm{diag}(0.1, 0.1, 0.1),$$

and $\boldsymbol{R}(\theta)$ is a rotation about the z-axis;

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

with $\theta_1 = \theta_2 = \pi/12$, and $\theta_3 = 0$. The permeabilities k_i are in the magnitude of 100 mD. We define the well cells as sources with a constant injection rate, and impose Dirichlet boundary conditions $p = 0$ on the sides of the domain, and no-flow boundary conditions on the top and bottom. Approximate solutions computed by TPFA, the mimetic method, and the two virtual element methods are shown in Fig. 25. For MFD and VEM, we have used the V1 and M1 inner products. For both the MPFA and the NTPFA methods, our solvers failed to compute transmissibilities, likely due to nearly colinear faces.

Again, the TPFA method fails to capture the effect of the anisotropic permeability field, while the remaining methods all give similar solutions. **Table 2** reports the relative difference between the methods defined as $100 \times |\boldsymbol{p}^i - \boldsymbol{p}^j|/|\boldsymbol{p}^i|$, where $\boldsymbol{p}^i$ is the vector of cell pressure values of method i. Likewise, **Table 3** reports the fraction of the injected fluid flowing out of each of the vertical sides. The sides are labeled by their compass direction, with the northern boundary being the leftmost visible vertical boundary. Altogether, the tables confirm that TPFA deviates most from the other methods. Of the four methods, VEM2 and MFD are closest to each other, followed by VEM2 and VEM1. Notice however, that the relative difference between VEM1 and TPFA is smaller than between VEM1 and MFD.

Table 4 reports key characteristics of the resulting discretization matrix for the four solution methods. The consistent methods are obviously much more expensive to compute than the TPFA method: the mimetic method has 4.5 times as many unknowns and 8.6 times as many nonzero elements, first-order VEM has 2.6 times as many unknowns but 11.7 times as many nonzero elements, whereas the second-order VEM has 14.6 times as many unknowns and 167.5 (!) as many nonzero elements. Notice also that unlike in 2D, VEM1 gives a much denser systems than the mimetic method and hence can be expected to be more computationally expensive. Finally, we note that the condition number for TPFA, which is cell-centered, is much better that for the methods using node and face pressures as degrees of freedom. This is due to the fact that the distances between the cell centroids are, by construction, much more uniform than distances between nodes and face centroids. One could redefine the gridding algorithms to yield grids with favorable properties for node- and face-centered methods. However, in multiphase flow problems, which have a hyperbolic character and are typically solved by finite-volume methods, we are forced to use grids with relatively uniform cell volumes.

Example 5 (Multilateral well)**.** Finally, we consider a reservoir containing a single multilateral well to which we have fitted a 3D PEBI-grid as shown in Fig. 26a. The reservoir volume is limited by the convex hull of a set of points. This hull is represented as a polyhedral, whose straight edges can be seen along the top of the reservoir in all plots of the reservoir in Fig. 26. All PEBI-cells close to the boundary are clipped against the limiting polyhedral, which creates additional face patches as seen in Fig. 26c. The permeability tensor varies in the z-direction in the following way

$$\mathsf{K}(z) = \boldsymbol{R}(\pi z) \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{R}(\pi z)^{\mathsf{T}},$$

where $\boldsymbol{R}(\theta)$ gives the rotation about the z axis, as defined in Eq. 14. Here, z is the normalized reservoir height, and runs from 0 at the bottom of the reservoir, to 1 at the top. In other words, the permeability at height z has a ratio of 10 : 1 in the xy-plane, and is rotated by an angle πz around the z-axis. The x-component of the permeability tensor is shown in Fig. 26b. As in the previous example, the permeability is given in units of 100 mD. We define the well cells as sources with constant injection rate, and impose no-flow boundary conditions on the top and bottom boundaries, and homogeneous Dirichlet conditions on the remaining vertical sides. The resulting solutions using TPFA, NTPFA, MFD, VEM1 and VEM2 are shown in Fig. 27. Again, we have used the M1 and V1 inner products for MFD and VEM.

As in the previous cases with grids that are not K-orthogonal, we observe that the TPFA method fails to capture the effect of the rotated permeability field. The NTPFA and MFD method seems to accurately capture this effect. The same is also evident in the first-order VEM solution, but the effect is not as pronounced, while the second-order VEM solution is comparable to the NTPFA and MFD solutions.

Table 5 reports the relative difference between the methods, confirming the observations above. In particular, we note that the NTPFA and VEM2 solutions are in close agreement. **Table 6** reports key characteristics of the resulting discrete systems for the methods. The VEM2 condition number is not reported, as the MATLAB function `condest` was not able to calculate this for the resulting linear system, possibly due to the size and density of the matrix. Finally, **Table 7** reports the fraction of the injected fluid flowing out of each vertical side. Boundary B_1 is the large side at the front of the grid, from which the numbering runs counter-clockwise around the reservoir. We observe that the fluxes out of the sides are comparable for all methods except TPFA, and that the closest agreement is between MFD and second-order VEM.

Closing Remarks

In the first part of this work, we have presented a method for generating PEBI-grids that conform to different geometric structures in subsurface reservoirs. Our method successfully creates both control-point aligned grids and boundary aligned grids. The key advantages of this method are; *(i)* user specified cell density allows for grid refinements in areas of interest; *(ii)* robust and automatic handling of intersections; and *(iii)* high-quality cells even in constricted areas. We have generalized the method for creating control-point and boundary aligned grids to 3D, and the method honors non-intersecting boundary aligned features exactly. In future work, the method for handling intersections in 2D should also be generalized to 3D.

Table 2—Relative difference in percent between pairs of solutions for the case with two curved faults.

	TPFA	MFD	VEM1	VEM2
TPFA	0.00	31.98	23.35	24.39
MFD	26.91	0.00	27.37	13.67
VEM1	25.37	35.34	0.00	20.54
VEM2	23.26	15.48	18.03	0.00

Table 3—Fraction of total flux trough each vertical side of the case with two curved faults.

	N	E	S	W
TPFA	55.49	3.68	23.84	16.99
MFD	59.58	2.88	17.53	20.01
VEM1	56.71	2.83	17.63	22.84
VEM2	57.60	2.96	17.70	21.74

Table 4—Key characteristics for the discrete system for the case with two curved faults. `dof` denotes the number of primary unknowns, `nnz` the number of nonzero entries in the discretization matrix, `ratio` the ratio between `nnz` and `dof`, and `cond` the condition number.

	dof	nnz	ratio	cond
TPFA	3959	37939	9.58	3.13e+04
MFD	17853	328015	18.37	1.62e+17
VEM1	10472	442801	42.28	9.59e+11
VEM2	57977	6356043	109.63	8.67e+12

Table 5—Relative difference in percent between pairs of solutions for the multilateral well case.

	TPFA	NTPFA	MFD	VEM1	VEM2
TPFA	0.00	65.50	56.82	61.25	66.18
NTPFA	81.20	0.00	26.06	21.88	5.51
MFD	60.80	22.49	0.00	34.78	21.68
VEM1	88.48	25.49	46.95	0.00	26.47
VEM2	82.22	5.52	25.18	22.76	0.00

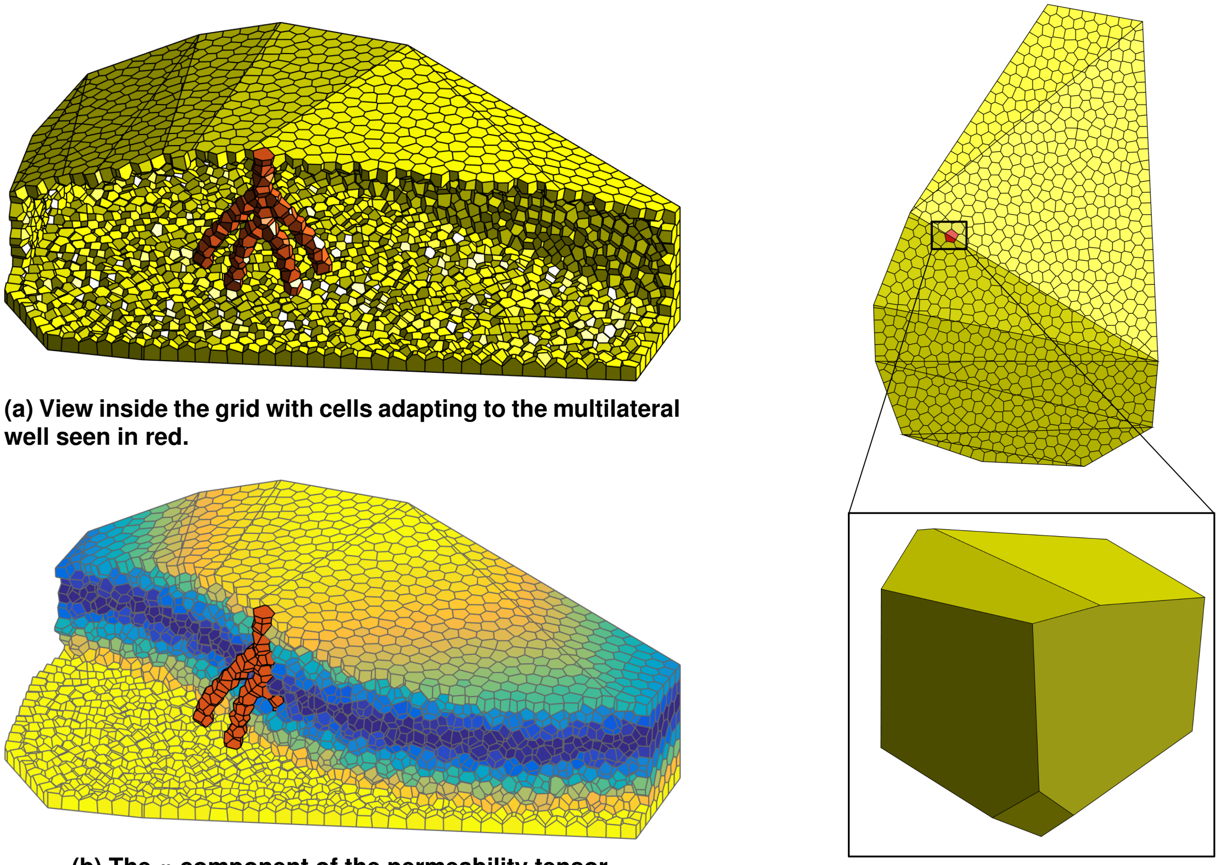
Table 6—Key characteristics of the discrete system for the multilateral well case.

	dof	nnz	ratio	cond
TPFA	9026	126002	13.96	9.64e+02
NTPFA	11920	280703	23.55	2.92e+07
MFD	60321	1538305	25.50	9.37e+15
VEM1	52350	3404977	65.04	4.91e+11
VEM2	227459	38770593	170.45	—

Table 7—Fraction of total flux through each vertical side of the multilateral well case.

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
TPFA	41.91	1.05	15.97	9.56	3.42	3.25	3.69	7.89	6.53	6.72
NTPFA	33.02	3.38	19.09	5.77	2.04	2.76	4.02	11.55	10.72	7.65
MFD	31.12	3.99	19.50	5.05	1.93	2.70	3.96	12.23	11.51	8.01
VEM1	33.85	3.19	18.54	5.85	1.98	2.84	4.49	11.63	10.02	7.61
VEM2	31.25	3.87	19.31	5.17	2.01	2.74	4.05	12.34	11.31	7.95

Through five test cases, we have demonstrated the importance of using a consistent discretization when modeling subsurface flow. In particular, we conclude that if the grid does not align with the permeability tensor, TPFA will give significant consistency errors. We have also seen that higher-order methods can be beneficial in such cases. However, this drastically increases the computational cost. A possible approach can be order refinement, in which we apply a higher-order method only in certain parts

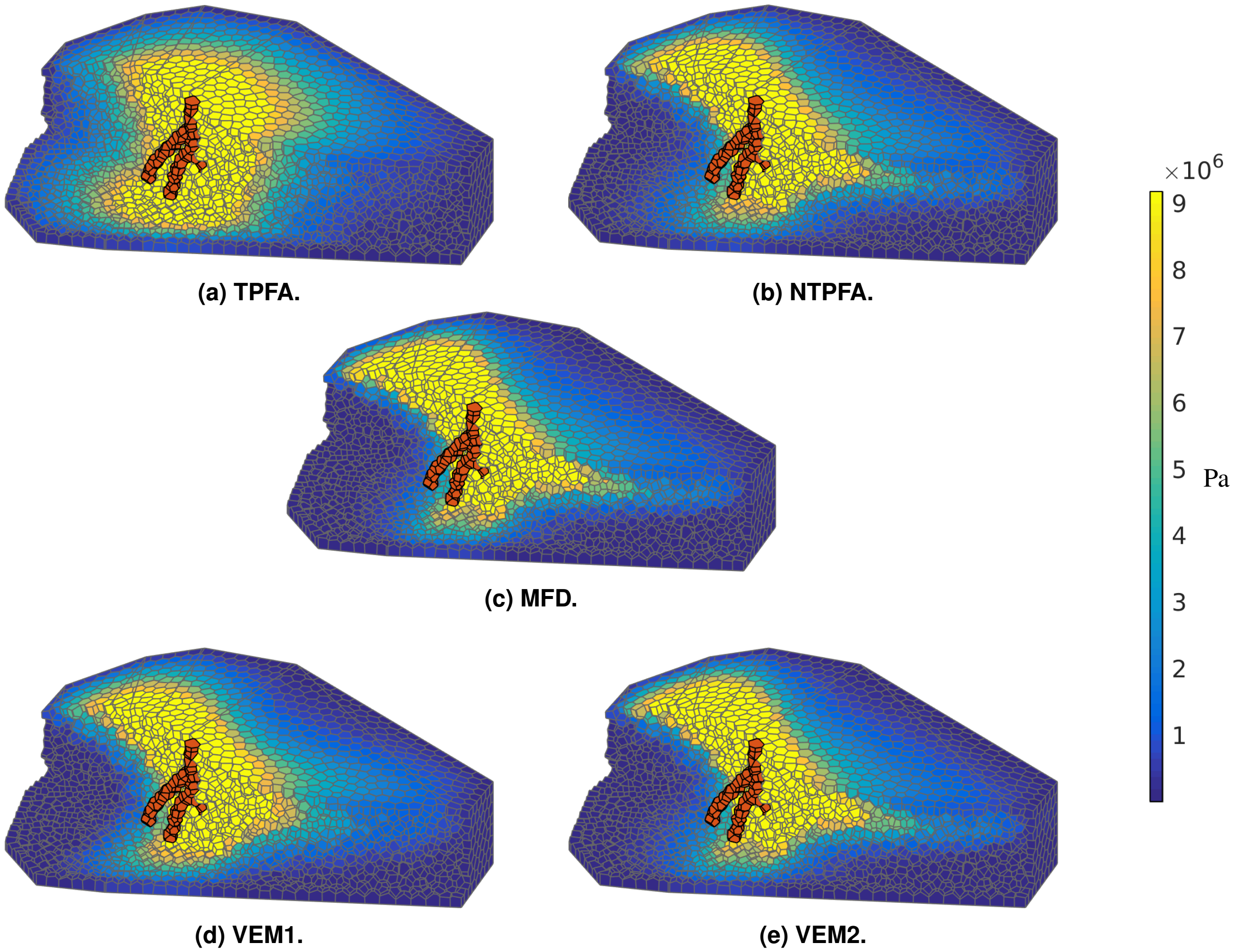


(a) View inside the grid with cells adapting to the multilateral well seen in red.

(b) The x-component of the permeability tensor.

(c) Grid cells clipped by reservoir geometry

Figure 26—Fully unstructured 3D PEBI-grid with a multilateral well inside. The grid is constructed so that individual cells are set to adapt to the well trajectory, whereas cells along the model perimeter are clipped by the polyhedral that describes the extent of the reservoir.



(a) TPFA.

(b) NTPFA.

(c) MFD.

(d) VEM1.

(e) VEM2.

Figure 27—Pressure fields around a multilateral well computed with four different methods.

of the grid, for example in the near-well regions.

The computational cost for the consistent methods discussed in this paper is obviously higher than for the classical two-point method. For multiphase flow simulations, the difference between TPFA and NTPFA may not be that big in practice when both schemes are used as part of an iterative procedure for nonlinear equations. Likewise, the cost of computing a single-phase flow solution with a consistent scheme is small compared with a full multiphase flow solution, and can thus be used to evaluate errors due to anisotropy and skew and irregular cell geometries. Furthermore, since all methods except TPFA are consistent for general grids, they can be used as part of an upscaling algorithm, and/or to modify the transmissibilities for the TPFA method to ensure that this method better honors the global flow properties. Overall, we believe the combination of these methods can be valuable in supporting simulations on more general grids.

Acknowledgments

The research by Klemetsdal is funded in part by the Research Council of Norway through the PETROMAKS2 program, grant number 244361. The research of Berge is funded in part by the TheMSES project, grant no. 250223 and by and Statoil, through the Akademia agreement.

Appendix: Mimetic and Virtual Inner Products

Mimetic inner products Lie et al. (2012) presents various inner products that can be used to specify various first-order mimetic methods, some of which coincide with other known finite-element or finite-volume methods for special grid types. To keep our paper as self-contained as possible, we present the inner products that have been used in our numerical experiments in some detail.

(M1) The following inner product was used by Aarnes et al. (2008), and chosen to resemble the Raviart–Thomas (RT0) inner product:

$$\boldsymbol{M} = \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^T + \frac{d|\Omega_i|}{6\,\mathrm{tr}(\boldsymbol{K})}\boldsymbol{A}^{-1}\left(\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathsf{T}}\right)\boldsymbol{A}^{-1},$$

where $\boldsymbol{A}$ is a diagonal matrix with $\boldsymbol{A}_{j,j} = |F_{i,j}|$, and $\boldsymbol{Q} = \left(\boldsymbol{A}^{-1}\boldsymbol{N}\right)^{\perp}$. This inner product is referred to as `ip_simple`, both by Lie et al. (2012) and in MRST.

(M2) As we have seen, the TPFA method requires a diagonal transmissibility tensor $\boldsymbol{T}$. However, we see from Eq. 9 that this is only possible when $\boldsymbol{K}\boldsymbol{n}$ is parallel to $\boldsymbol{c}$. The following inner product coincides with TPFA in its region of validity and gives a valid mimetic inner product for any combination of grid and permeability tensor:

$$\boldsymbol{T} = \frac{1}{|\Omega_i|}\left(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + 2\boldsymbol{P}_C^{\perp}\,\mathrm{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})\boldsymbol{P}_C^{\perp}\right), \quad \boldsymbol{M} = \boldsymbol{T}^{-1}.$$

Here, $\boldsymbol{P}_C^{\perp} = \boldsymbol{I} - \boldsymbol{Q}_C\boldsymbol{Q}_C^{\mathsf{T}}$ with $\boldsymbol{Q}_C^{\perp}$ being an orthonormal basis for the null space of $\boldsymbol{C}^{\mathsf{T}}$. Lie et al. (2012) refer to this inner product as `ip_qtpfa`, while its MRST name is `ip_quasitpf`.

(M3) Finally, we consider an inner product that is equivalent to the mixed Raviart–Thomas inner product for orthogonal grids with the same principal axes as the permeability tensor:

$$\boldsymbol{M} = \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^T + \frac{|\Omega_i|}{6}\boldsymbol{P}_N^{\perp}\left(\mathrm{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})\right)^{-1}\boldsymbol{P}_N^{\perp}.$$

Here, $\boldsymbol{P}_N^{\perp} = \boldsymbol{I} - \boldsymbol{Q}_N\boldsymbol{Q}_N^{\mathsf{T}}$ with $\boldsymbol{Q}_N^{\perp}$ begin an orthonormal basis for the null space of $\boldsymbol{N}^{\mathsf{T}}$. Lie et al. (2012) refer to this inner product as `ip_qrt`, while its MRST name is `ip_quasirt`.

Virtual inner products In the following, we briefly discuss the construction of some possible choices for the VEM stability term. For a more detailed discussion, see Klemetsdal (2016).

(V1) As explained above, for a_h^i to satisfy the stability property, we must choose s^i such that a_h^i is symmetric and positive definite. Moreover, it can be shown that the bilinear form a^i scales as $h_i^{d-2}\,\mathrm{tr}(\boldsymbol{K}_i)$, where h_i is the diameter of cell Ω_i. Hence, the simplest possible choice of stability term is

$$s^i(u,v) = h_i^{d-2}\,\mathrm{tr}(\boldsymbol{K}_i)\hat{\boldsymbol{u}}^{\mathsf{T}}\hat{\boldsymbol{v}},$$

where $\hat{\boldsymbol{v}}$ is the vector with the jth degree of freedom of v in position j. The matrix form of this stability term is thus the identity matrix multiplied by $h_i^{d-2}\,\mathrm{tr}(\boldsymbol{K}_i)$. The MRST name of this inner product is `ip_simple`.

(V2) Since we only use the stability term for functions on the form $v - \Pi v$, we only need it to be symmetric positive definite for functions in the null space of Π. To this end, we let $\{\psi^i\}$ be a basis for this null space, i.e., $\Pi\psi^i = 0$. It can be shown that any positive definite matrix on the null space of Π can be written on the form

$$\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^{\mathsf{T}},$$

where column j of $\boldsymbol{Q}$ is the vector $\hat{\boldsymbol{\psi}}^j$, and $\boldsymbol{\Sigma}$ is a positive definite matrix of dimension $(N_i - \dim\mathbb{P}_k(\Omega_i)) \times (N_i - \dim\mathbb{P}_k(\Omega_i))$. Here, N_i is the number of degrees of freedom of the local virtual element space on cell Ω_i, and $\mathbb{P}_k(\Omega_i)$ is the space of polynomials of degree k or less on Ω_i. We use this to formulate a family of stability terms:

$$s^i(u,v) = \hat{\boldsymbol{u}}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^T\hat{\boldsymbol{v}},$$

where $\boldsymbol{\Sigma}$ is a positive definite matrix, scaling with $h_i^{d-2}\,\mathrm{tr}(\boldsymbol{K}_i)$. This inner product can be found in MRST under the name `ip_qfamily`. In this paper, we have used $\boldsymbol{\Sigma} = h_i^{d-2}\,\mathrm{tr}(\boldsymbol{K}_i)\boldsymbol{I}$.

(V3) As shown in Klemetsdal (2016), if we choose $\{\psi^j\}$ to be an L^2-orthonormal basis for the null space of Π, we can recover the exact bilinear form so that $a_h^i = a^i$. This is obtained by choosing the stability term

$$s^i(u,v) = \hat{\boldsymbol{u}}^{\mathsf{T}}\boldsymbol{Q}(\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Q})^{-1}\boldsymbol{\Lambda}(\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Q})^{-1}\boldsymbol{Q}^{\mathsf{T}}\hat{\boldsymbol{v}}, \tag{15}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with the generalized eigenvalues

$$\lambda_j = a^i\left(\psi^j, \psi^j\right)$$

on the diagonal. Notice that this involves determining the basis $\{\psi^j\}$ explicitly, which is not possible in general since we do not know the exact expression of our basis functions. However, it can be shown that the local function spaces for the first-order finite element method and the first-order VEM coincide for certain cell geometries. This is for instance true for Cartesian grids with parallelogram cells, and we can use this fact to compute $\{\psi^j\}$ and obtain the exact stability term. In MRST, this particular inner product is called `ip_fem`.

(V4) It is possible to construct a virtual element inner product which, on Cartesian grids with equal, rectangular or cuboid cells, resembles a combination of two finite difference stencils; one using a regular Cartesian coordinate system, and one using a diagonal coordinate system, see Klemetsdal (2016) for details. In 2D, this is obtained by using a stability term on the same form as Eq. 15, with $\mathbf{\Lambda}$ equal to the scaled eigenvalue matrix

$$\mathbf{\Lambda}_{j,j} = 3w\lambda_j, \quad w \in \mathbb{R}\,.$$

If we denote the discretization of $\nabla \cdot \mathsf{K}\nabla$ using the Cartesian stencil by $\boldsymbol{F}_c$, and the discretization using the diagonal stencil by $\boldsymbol{F}_d$, the parameter w is the weight we use to combine the stencils:

$$\boldsymbol{F} = w\boldsymbol{F}_c + (1-w)\boldsymbol{F}_d, \quad 0 \le w \le 1. \tag{16}$$

In MRST, we refer to this inner product as `ip_fd`.

References

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2008. Multiscale mixed/mimetic methods on corner-point grids. *Computational Geosciences*, 12(3):297–315. doi: 10.1007/s10596-007-9072-8.

Aavatsmark, I. 2002. An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences*, 6(3-4):405–432. doi: 10.1023/A:1021291114475.

Aavatsmark, I. 2007. Interpretation of a two-point flux stencil for skew parallelogram grids. *Computational Geosciences*, 11(3):199–206. doi: 10.1007/s10596-007-9042-1.

Ahmad, B., Alsaedi, A., Brezzi, F., Marini, L. D., and Russo, A. 2013. Equivalent projectors for virtual element methods. *Computers & Mathematics with Applications*, 66(3):376–391. doi: 10.1016/j.camwa.2013.05.015.

Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. 2013. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(01):199–214. doi: 10.1142/S0218202512500492.

Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. 2014. The hitchhiker's guide to the virtual element method. *Mathematical models and methods in applied sciences*, 24(08):1541–1573. doi: 10.1142/S021820251440003X.

Berge, R. L. 2016. Unstructured pebi grids adapting to geological feautres in subsurface reservoirs. Master's thesis, Norwegian University of Science and Technology.

Branets, L., Ghai, S. S., Lyons, S. L., and Wu, X.-H. 2009a. Efficient and Accurate Reservoir Modeling Using Adaptive Gridding with Global Scale Up. In *Proceedings of the SPE Reservoir Simulation Symposium*, The Woodlands, Texas. doi: 10.2118/118946-MS.

Branets, L. V., Ghai, S. S., Lyons, S. L., and Wu, X.-H. 2009b. Challenges and technologies in reservoir modeling. *Communications in Computational Physics*, 6(1):1–23.

Brewer, M., Camilleri, D., Ward, S., and Wong, T. 2015. Generation of hybrid grids for simulation of complex, unstructured reservoirs by a simulator with MPFA. *SPE Reservoir Simulation Symposium, 23-25 February, Houston, Texas, USA*. doi: 10.2118/173191-MS.

Brezzi, F., Lipnikov, K., and Simoncini, V. 2005. A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences*, 15(10):1533–1551. doi: 10.1142/S0218202505000832.

Burman, E. and Zunino, P. 2006. A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis*, 44(4):1612–1638. doi: 10.1137/050634736.

Courrioux, G., Nullans, S., Guillen, A., Boissonnat, J. D., Repusseau, P., Renaud, X., and Thibaut, M. 2001. 3D volumetric modelling of Cadomian terranes Northern Brittany, France): an automatic method using Voronoi diagrams. *Tectonophysics*, 331:181–196. doi: 10.1016/S0040-1951(00)00242-0.

da Veiga, L. B., Lipnikov, L., and Manzini, G. 2014. *Mimetic Finite Difference Method for Elliptic Problems*, volume 11. Springer.

Ding, X. Y. and Fung, L. S. K. 2015. An unstructured gridding method for simulating faulted reservoirs populated with complex wells. In *Proceedings of the SPE Reservoir Simulation Symposium*, Houston, Texas, USA. doi: 10.2118/173243-MS.

Du, Q., Faber, V., and Gunzburger, M. 1999. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676. doi: 10.1137/S0036144599352836.

Ern, A., Stephansen, A. F., and Zunino, P. 2008. A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity. *IMA Journal of Numerical Analysis*. doi: 10.1093/imanum/drm050.

Fung, L.-K., Hiebert, A., and Nghiem, L. 1992. Reservoir simulation with a control-volume finite-element method. *SPE Reservoir Engineering*, 7(3):349–356. doi: 10.2118/21224-PA.

Fung, L. S. K., Ding, X. Y., and Dogru, A. H. 2014. Unconstrained Voronoi grids for densely spaced complex wells in full-field reservoir simulation. *SPE Journal*, 19(5):803–815. doi: 10.2118/163648-PA.

Gringarten, E. J., Arpat, G. B., Haouesse, M. A., Dutranois, A., Deny, L., Jayr, S., Tertois, A.-L., Mallet, J.-L., Bernal, A., and Nghiem, L. X. 2008. New grids for robust reservoir modeling. In *SPE Annual Technical Conference and Exhibition, 21-24 September, Denver, Colorado, USA*. Society of Petroleum Engineers. doi: 10.2118/116649-MS.

Gringarten, E. J., Haouesse, M. A., Arpat, G. B., and Nghiem, L. X. 2009. Advantages of using vertical stair step faults in reservoir grids for flow simulation. In *SPE Reservoir Simulation Symposium, 2-4 February, The Woodlands, Texas*. Society of Petroleum Engineers. doi: 10.2118/119188-MS.

Guerillot, D. and Swaby, P. 1993. An interactive 3D mesh builder for fluid flow reservoir simulation. *SPE Computer Applications*, 5(6):5–10. doi: 10.2118/26227-PA.

Gunasekera, D., Cox, J., and Lindsey, P. 1997. The generation and application of K-orthogonal grid systems. In *SPE Reservoir Simulation Symposium, 8-11 June, Dallas, Texas*. Society of Petroleum Engineers. doi: 10.2118/37998-MS.

Heinemann, Z. E., Brand, C. W., Munka, M., and Chen, Y. M. 1991. Modeling reservoir geometry with irregular grids. *SPE Reservoir Engineering*, 6(2):225–232. doi: 10.2118/18412-PA.

Holm, R., Kaufmann, R., Heimsund, B.-O., ian, E., and Espedal, M. S. 2006. Meshing of domains with complex internal geometries. *Numerical Linear Algebra with Applications*, 13(9):717–731. doi: 10.1002/nla.505.

Iri, M., Murota, K., and Ohya, T. 1984. *System Modelling and Optimization: Proceedings of the 11th IFIP Conference Copenhagen, Denmark, July 25–29, 1983*, chapter A fast Voronoi-diagram algorithm with applications to geographical optimization problems, pages 273–288. Springer Berlin Heidelberg, Berlin, Heidelberg. doi: 10.1007/BFb0008901.

Klemetsdal, Ø. S. 2016. The virtual element method as a common framework for finite element and finite difference methods – numerical and theoretical analysis. Master's thesis, Norwegian university of science and technology.

Le Potier, C. 2009. A nonlinear finite volume scheme satisfying maximum and minimum principles for diffusion operators. *International Journal on Finite Volumes*, pages 1–20.

Lie, K.-A. 2015. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST)*. SINTEF ICT.

Lie, K.-A., Krogstad, S., Ligaarden, I. S., Natvig, J. R., Nilsen, H. M., and Skaflestad, B. 2012. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322. doi: 10.1007/s10596-011-9244-4.

Lipnikov, K., Shashkov, M., Svyatskiy, D., and Vassilevski, Y. 2007. Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes. *Journal of Computational Physics*, 227(1):492–512.

Lipnikov, K., Svyatskiy, D., and Vassilevski, Y. 2009. Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes. *Journal of Computational Physics*, 228(3):703–716.

Mallison, B., Sword, C., Viard, T., Milliken, W., and Cheng, A. 2014. Unstructured Cut-Cell Grids for Modeling Complex Reservoirs. *SPE Journal*, 19(02):340–352. doi: 10.2118/163642-PA.

Merland, R., Caumon, G., Lévy, B., and Collon-Drouaillet, P. 2014. Voronoi grids conforming to 3D structural features. *Computational Geosciences*, 18(3-4):373–383. doi: 10.1007/s10596-014-9408-0.

Merland, R., Lévy, B., and Caumon, G. 2011. Building PEBI grids conforming to 3D geological features using centroidal voronoi tessellation. In Marschallinger, R. and Zolb, R., editors, *Proceedings of IAMG*, page 12, Salzburg.

Mustapha, H. 2011. G23fm: a tool for meshing complex geological media. *Computational Geosciences*, 15(3):385–397. doi: 10.1007/s10596-010-9210-6.

Nikitin, K., Terekhov, K., and Vassilevski, Y. 2014a. A monotone nonlinear finite volume method for diffusion equations and multiphase flows. *Computational Geosciences*, 18(3-4):311–324. doi: 10.1007/s10596-013-9387-6.

Nikitin, K., Terekhov, K., and Vassilevski, Y. 2014b. A monotone nonlinear finite volume method for diffusion equations and multiphase flows. *Computational Geosciences*, 18(3-4):311–324.

Nocedal, J. and Wright, S. J. 2006. *Numerical Optimization*. Springer-Verlag New York, 2 edition. doi: 10.1007/978-0-387-40065-5.

Nordbotten, J. M., Aavatsmark, I., and Eigestad, G. T. 2007. Monotonicity of control volume methods. *Numerische Mathematik*, 106(2):255–288. doi: 10.1007/s00211-006-0060-z.

Odsæter, L. H., Wheeler, M. F., Kvamsdal, T., and Larson, M. G. 2016. Postprocessing of non-conservative flux for compatibility with transport in heterogeneous media. *arXiv preprint arXiv:1605.04076*.

Palagi, C. and Aziz, K. 1994. Use of Voronoi Grid in Reservoir Simulation. *Society of Petroleum Engineers*, 2(2):69–77. doi: 10.2118/22889-PA.

Persson, P.-O. and Strang, G. 2004. A Simple Mesh Generator in MATLAB. *SIAM Review*, 46(2):329–345. doi: 10.1137/S0036144503429121.

Ponting, D. K. 1989. Corner point geometry in reservoir simulation. In King, P., editor, *Proceedings of the 1st European Conference on Mathematics of Oil Recovery, Cambridge, 1989*, pages 45–65, Oxford. Clarendon Press. doi: 10.3997/2214-4609.201411305.

Sun, J. and Schechter, D. 2015. Optimization-Based Unstructured Meshing Algorithms for Simulation of Hydraulically and Naturally Fractured Reservoirs With Variable Distribution of Fracture Aperture, Spacing, Length, and Strike. *SPE Reservoir Evaluation & Engineering*, 18(04):463 – 480. doi: 10.2118/170703-PA.

Toor, S. M., Edwards, M. G., Dogru, A. H., and Shaalan, T. M. 2015. Boundary aligned grid generation in three dimensions and CVD-MPFA discretization. In *Proceedings of the SPE Reservoir Simulation Symposium*, Houston, Texas, USA. doi: 10.2118/173313-MS.

Verma, S. and Aziz, K. 1997. A control volume scheme for flexible grids in reservoir simulation. In *SPE Reservoir Simulation Symposium, 8-11 June, Dallas, Texas*. Society of Petroleum Engineers. doi: 10.2118/37999-MS.

Wu, X.-H. and Parashkevov, R. 2009. Effect of grid deviation on flow solutions. *SPE Journal*, 14(01):67–77. doi: 10.2118/92868-PA.

Yan, D.-M., Wang, W., Lévy, B., and Liu, Y. 2010. *Advances in Geometric Modeling and Processing: 6th International Conference, GMP 201 0, Castro Urdiales, Spain, June 16-18, 2010. Proceedings*, chapter Efficient Computation of 3D Clipped Voronoi Diagram, pages 269–282. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-13411-1_18.