6

Consistent Discretizations on Polyhedral Grids

The two-point flux-approximation (TPFA) scheme is robust in the sense that it generally gives a linear system that has a solution regardless of the variations in $\mathbf{K}$ and the geometrical and topological complexity of the grid. The resulting solutions will also be monotone, but the scheme is only consistent for certain combinations of grids and permeability tensors $\mathbf{K}$. This implies that a TPFA solution will not necessarily approach the true solution when we increase the grid resolution. It also means that the scheme may produce different solutions depending upon how the grid is oriented relative to the main flow directions; we will discuss this in more detail in Chapter 10. In this chapter, we first explain the lack of consistency for TPFA, before we introduce a few consistent schemes implemented in MRST. If you have only limited interest in discretizations, my advice is that you read Sections 6.1 and 6.3 before continuing to Chapter 7.

6.1 The TPFA Method Is Not Consistent

To provide some background for the following discussion, we start by giving a quick and informal recap of some important concepts from basic numerical analysis. When we approximate the Laplace operator $\mathcal{L} = \nabla \cdot \mathbf{K}\nabla$ by a numerical scheme, the approximation can be written as follows:

$$\mathcal{L}p = \mathcal{L}_h p + \mathcal{T}(h),$$

where h is the characteristic size of the cells in our grid, $\mathcal{L}_h$ is the approximation to the Laplace operator, and $\mathcal{T}$ is the *truncation error*, i.e., the difference between the true operator and our approximation. A numerical scheme is said to be *consistent* if the truncation error vanishes as h tends to zero, or in other words, if the difference between the true operator and our approximation diminishes as the grid is refined. A fundamental result in numerical analysis (Lax's equivalence theorem) states that *a consistent numerical method is convergent if and only if the method is stable*. By *convergent*, we mean that our approximations will approach the true solution as the size of the grid cells approaches zero. By *stable*, we mean that errors we make in approximating parameters determining the solution (for Poisson's equation: the source term q and the permeability $\mathbf{K}$) do not give unbounded amplification of the error in the computed solution. Here, a word of caution is

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

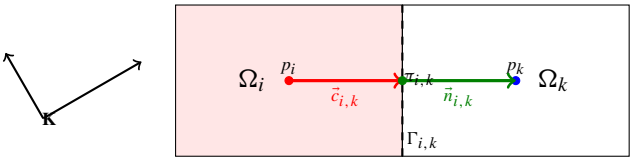

Figure 6.1 Two cells in a Cartesian grid and a full permeability tensor whose principal axes are not aligned with the coordinate system.

in order: *a scheme does not need to be consistent to be convergent*; showing consistency and stability is only a convenient approach to prove convergence.

Example 6.1.1 *To show that the TPFA scheme is generally not consistent, we start by a simple example. Recall from Section 4.4.1 that if p_i and p_k denote the average pressures in two neighboring cells Ω_i and Ω_k, then the flux across the interface Γ_{ik} between them is given as*

$$v_{ik} = T_{ik}(p_i - p_k), \tag{6.1}$$

where the transmissibility T_{ik} depends on the geometry of the two cells and the associated permeability tensors $\mathbf{K}_i$ and $\mathbf{K}_k$. To see that the method is not consistent, we assume that $\mathbf{K}$ is a homogeneous symmetric tensor

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{xy} & \mathbf{K}_{yy} \end{bmatrix}.$$

Consider now the flux across an interface Γ_{ik} between two cells Ω_i and Ω_k in a 2D Cartesian grid, whose normal vector $\vec{n}_{i,k}$ points in the x-direction so that $\vec{n}_{i,k} = \vec{c}_{i,k} = (1,0)$; see Figure 6.1. If follows from Darcy's law that $\vec{v} \cdot \vec{n} = -(\mathbf{K}_{xx}\partial_x p + \mathbf{K}_{xy}\partial_y p)$, from which it is easy to see that the flux across the interface will have two components, one orthogonal and one transverse,

$$v_{ik} = -\int_{\Gamma_{ik}} \mathbf{K}\nabla p \cdot \vec{n}\, ds = -\int_{\Gamma_{ik}} \left(\mathbf{K}_{xx}\,\partial_x p + \mathbf{K}_{xy}\,\partial_y p\right) ds.$$

The two flux components correspond to the x-derivative and the y-derivative of the pressure, respectively. In the two-point approximation (6.1), the only points we use to approximate the flux are the pressures p_i and p_k. Because these two pressures are associated with the same y-value, their difference can only be used to estimate $\partial_x p$ and not $\partial_y p$. This means that the TPFA method cannot account for the transverse flux contribution $\mathbf{K}_{xy}\partial_y p$ and will hence generally not be consistent. The only exception is if $\mathbf{K}_{xy} \equiv 0$ in all cells in the grid, in which case there are no fluxes in the transverse direction.

To link the derivation in Example 6.1.1 to the geometry of the cells, we can use the one-sided definition of fluxes (4.51) of the two-point scheme to write

$$v_{i,k} \approx T_{i,k}(p_i - \pi_{i,k}) = \frac{A_{i,k}}{|\vec{c}_{i,k}|^2}\,(\mathbf{K}\vec{c}_{i,k}) \cdot \vec{n}_{i,k}(p_i - \pi_{i,k}).$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

More generally, we have that the TPFA scheme is only convergent for *K-orthogonal grids*. An example of a K-orthogonal grid is a grid in which all cells are parallelepipeds in 3D or parallelograms in 2D and satisfy the condition

$$\vec{n}_{i,j} \cdot \mathbf{K}\vec{n}_{i,k} = 0, \qquad \forall j \neq k, \tag{6.2}$$

where $\vec{n}_{i,j}$ and $\vec{n}_{i,k}$ denote normal vectors to faces of cell number i.

What about grids that are not parallelepipeds or parallelograms? For simplicity, we only consider the 2D case. Let $\vec{c} = (c_1, c_2)$ denote the vector from the center of cell Ω_i to the centroid of face Γ_{ik}; see Figure 4.10. If we let $\vec{c}_\perp = (c_2, -c_1)$ denote a vector that is orthogonal to $\vec{c}$, then any constant pressure gradient inside Ω_i can be written as $\nabla p = p_1\vec{c} + p_2\vec{c}_\perp$, where p_1 can be determined from p_i and $\pi_{i,k}$, and p_2 is some unknown constant. Inserted into the definition of the face flux, this gives

$$v_{i,k} = -\int_{\Gamma_{ik}} \left[p_1 (\mathbf{K}\vec{c}) \cdot \vec{n} + p_2 (\mathbf{K}\vec{c}_\perp) \cdot \vec{n}\right] ds.$$

For the two-point scheme to be correct, the second term in the integrand must be zero. Setting $\vec{n} = (n_1, n_2)$, we find that

$$0 = (\mathbf{K}\vec{c}_\perp) \cdot \vec{n} = (K_1 c_2, -K_2 c_1) \cdot (n_1, n_2) = (K_1 n_1 c_2 - K_2 n_2 c_1) = (\mathbf{K}\vec{n}) \times \vec{c}.$$

In other words, a sufficient condition for a polygonal grid to be K-orthogonal is that $(\mathbf{K}\vec{n}_{i,k}) \parallel \vec{c}_{i,k}$ for all cells in the grid, in which case the transverse flux contributions are all zero and hence need not be estimated. Consequently, the TPFA method will be consistent.

The lack of consistency in the TPFA scheme for grids that are not K-orthogonal may lead to significant *grid orientation effects* [8, 319], i.e., artifacts or errors in the solution that will appear in varying degree depending upon the angles the cell faces make with the principal directions of the permeability tensor. We have already seen an example of grid effects in Figure 5.7 in Section 5.4.3, where an irregular triangular grid caused large deviations from what should have been an almost symmetric pressure drawdown. To demonstrate grid orientation effects and lack of convergence more clearly, we look at a simple example that has been widely used to teach reservoir engineers the importance of aligning the grid with the principal permeability directions.

Example 6.1.2 *Consider a homogeneous reservoir in the form of a 2D rectangle* $[0,4] \times [0,1]$. *Flow is driven from the north to the south side of the reservoir by a fluid source with unit rate located at* $(2, 0.975)$ *along the north side, and two fluid sinks located at* $(0.5, 0.025)$ *and* $(3.5, 0.025)$ *along the south side. The two source terms each have rate equal one half and are symmetric to the line* $x = 2$ *and will therefore result in a symmetric pressure distribution.*

To challenge TPFA, we discretize the domain by a skewed grid that is uniform along the north side and graded towards east along the south side:

```
G = cartGrid([41,20],[2,1]);
makeSkew = @(c) c(:,1) + .4*(1-(c(:,1)-1).^2).*(1-c(:,2));
G.nodes.coords(:,1) = 2*makeSkew(G.nodes.coords);
```

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

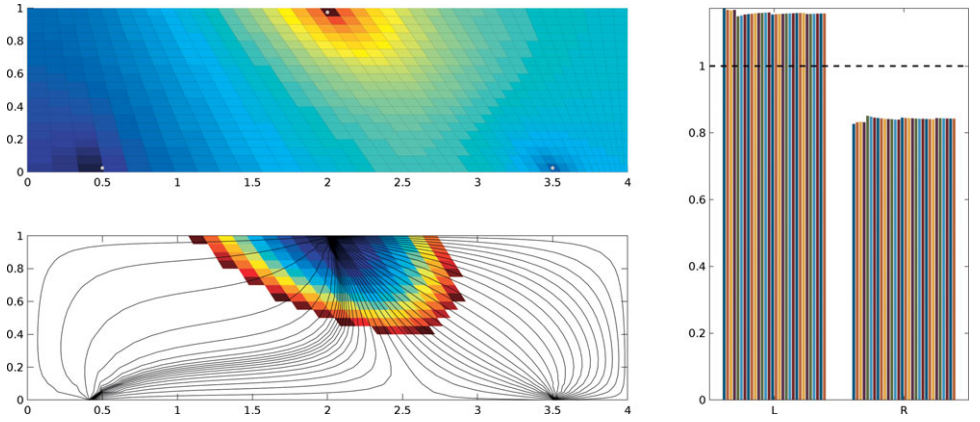

Figure 6.2 Solution of a symmetric flow problem in a homogeneous domain using the TPFA method on a skew grid that is not K-orthogonal. The upper-left plot shows the pressure distribution, whereas the lower-left plot shows streamlines and values of time-of-flight less than 0.25 PVI. The plot to the right shows time-of-flight in the left (L) and right (R) sink for a series of refined grids of dimension $(20n + 1) \times 10n$ for $n = 1, \ldots, 30$. The solutions do not converge toward the analytical solution shown as the dashed red line.

If we assume an isotropic and homogeneous permeability field, the grid is not K-orthogonal, and we therefore cannot expect that the TPFA method will converge to the correct physical solution. You can find a complete setup of the problem in the script `1phase/showInconsistentTPFA.m`.

Figure 6.2 reports the result of a convergence study. All the computed solutions exhibit the same lack of symmetry you see in the solution on the 41 × 20 grid. Moreover, the large difference in travel times from the injector to the two producers does not decay with increasing grid resolution, which confirms the expected lack of convergence for an inconsistent scheme.

Grid orientation errors are discussed in more detail in Section 10.4.2. It is a good advice to try to make grids so that cell faces align with the principle directions of the permeability tensor. A main motivation for using PEBI grids is to maintain flexibility in areal representation while keeping grid orientation errors at a minimum; see [128, 121, 210].

6.2 The Mixed Finite-Element Method

There are several ways to formulate consistent discretization methods. The mixed finite-element method [53] is based on a quite different formulation than the two-point method. The first new idea is that instead of discretizing the second-order Poisson equation, we form a system that consists of two discrete equations representing mass conservation and Darcy's law. Hence, instead of solving for pressure and computing discrete fluxes by postprocessing

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

the pressure solution, we solve for pressure and fluxes simultaneously. The second new idea is to look for solutions that satisfy the flow equations in a weak sense; that is, look for solutions that fulfill the equation when it is multiplied with a suitable *test function* and integrated in space. The third new idea is to express the unknown solution as a linear combination of a set of *basis functions*, i.e., piecewise polynomials with localized support.

MRST does not implement the mixed finite-element method directly in the classical sense, but the key ideas of this method are instrumental when we develop a general class of finite-volume discretizations in Section 6.3 and discussed in more details in the remaining parts of the chapter. As a precursor to this discussion, we review the mixed finite-element method in some detail.

6.2.1 Continuous Formulation

We start by restating the continuous flow equation written as a first-order system:

$$\nabla \cdot \vec{v} = q, \qquad \vec{v} = -\mathbf{K}\nabla p, \qquad \vec{x} \in \Omega \subset \mathbb{R}^d \tag{6.3}$$

with boundary conditions $\vec{v} \cdot \vec{n} =$ for $\vec{x} \in \partial\Omega$. Gravity has been omitted for brevity, but can easily be included. For compatibility, we require that $\int_\Omega q \, d\vec{x} = 0$. Because this is a pure Neumann boundary-value problem, pressure p is only defined up to an arbitrary constant, and as an extra constraint, we require that $\int_\Omega p \, d\vec{x} = 0$.

In the mixed method, we look for solutions in an abstract function space. To this end, we need two spaces: $L^2(\Omega)$ is the space of square integrable functions, and H_0^{div} is a so-called Sobolev space defined as

$$H_0^{\text{div}}(\Omega) = \{\vec{v} \in L^2(\Omega)^d : \nabla \cdot \vec{v} \in L^2(\Omega) \text{ and } \vec{v} \cdot \vec{n} \text{ on } \partial\Omega\}, \tag{6.4}$$

i.e., as the set of square-integrable, vector-valued functions with compact support in Ω, whose divergence is also square integrable. The *mixed formulation* of (6.3) now reads: find a pair $(p, \vec{v})$ that lies in $L^2(\Omega) \times H_0^{\text{div}}(\Omega)$ and satisfies

$$\begin{aligned} \int_\Omega \vec{u} \cdot \mathbf{K}^{-1}\vec{v} \, d\vec{x} - \int_\Omega p \, \nabla \cdot \vec{u} \, d\vec{x} = 0, \qquad & \forall \vec{u} \in H_0^{\text{div}}(\Omega), \\ \int_\Omega w \, \nabla \cdot \vec{v} \, d\vec{x} = \int_\Omega q w \, d\vec{x}, \qquad & \forall w \in L^2(\Omega). \end{aligned} \tag{6.5}$$

The first equation follows by multiplying Darcy's law by $\mathbf{K}^{-1}$ and applying the divergence theorem to the term involving the pressure gradient. (Recall that $\vec{u}$ is zero on $\partial\Omega$ by definition so that the term $\int_{\partial\Omega} p\vec{u} \cdot \vec{n} \, ds = 0$.)

We can write equation (6.5) on a more compact form if we introduce the following three inner products

$$b(\cdot,\cdot) : H_0^{\text{div}} \times H_0^{\text{div}} \to \mathbb{R}, \qquad b(\vec{u},\vec{v}) = \int_\Omega \vec{u} \cdot \mathbf{K}^{-1}\vec{v} \, d\vec{x}$$

$$c(\cdot,\cdot) : H_0^{\text{div}} \times L^2 \to \mathbb{R}, \qquad c(\vec{u},p) = \int_\Omega p \, \nabla \cdot \vec{u} \, d\vec{x}$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

$$(\cdot,\cdot) : L^2 \times L^2 \to \mathbb{R}, \qquad (q,w) = \int_\Omega q\, w\, d\vec{x}, \tag{6.6}$$

and write

$$b(\vec{u},\vec{v}) - c(\vec{u},p) = 0, \qquad c(\vec{v},w) = (q,w). \tag{6.7}$$

Alternatively, we can derive (6.7) by minimizing the energy functional

$$I(\vec{v}) = \frac{1}{2}\int_\Omega \vec{v}\cdot\mathbf{K}^{-1}\vec{v}\, d\vec{x},$$

over all mass-conservative functions $\vec{v} \in H_0^{\mathrm{div}}(\Omega)$, i.e., subject to the constraint

$$\nabla\cdot\vec{v} = q.$$

The common strategy for solving such minimization problems is to introduce a Lagrangian functional, which in our case reads

$$L(\vec{v},p) = \int_\Omega \Big(\tfrac{1}{2}\vec{v}\cdot\mathbf{K}^{-1}\vec{v} - p\big(\nabla\cdot\vec{v} - q\big)\Big)\, d\vec{x} = \frac{1}{2}b(\vec{v},\vec{v}) - c(\vec{v},p) + (p,q),$$

where p is a so-called Lagrangian multiplier. At a minimum of L, we must have $\partial L/\partial\vec{v} = 0$. Looking at an increment $\vec{u}$ of $\vec{v}$, we have the requirement that

$$0 = \frac{\partial L}{\partial\vec{v}}\vec{u} = \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\big[L(\vec{v}+\varepsilon\vec{u},p) - L(\vec{v},p)\big] = b(\vec{u},\vec{v}) - c(\vec{u},p)$$

for all $u \in H_0^{\mathrm{div}}(\Omega)$. Similarly, we can show that

$$0 = \frac{\partial L}{\partial p}w = \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\big[L(\vec{v},p+\varepsilon w) - L(\vec{v},p)\big] = -c(\vec{v},w) + (q,w)$$

for all $w \in L^2(\Omega)$. To show that the solution $\vec{v}$ is a minimal point, we consider a perturbation $\vec{v}+\vec{u}$ that satisfies the constraints so that $\vec{u} \in H_0^{\mathrm{div}}$ and $\nabla\cdot\vec{u} = 0$. For the energy functional we have that

$$\begin{aligned} I(\vec{v}+\vec{u}) &= \frac{1}{2}\int_\Omega (\vec{v}+\vec{u})\cdot\mathbf{K}^{-1}(\vec{v}+\vec{u})\, d\vec{x} = I(\vec{v}) + I(\vec{u}) + b(\vec{u},\vec{v}) \\ &= I(\vec{v}) + I(\vec{u}) + c(\vec{u},p) = I(\vec{v}) + I(\vec{u}) > I(\vec{v}), \end{aligned}$$

which proves that $\vec{v}$ is indeed a minimum of I. We can also prove that the solution $(p,\vec{v})$ is a saddle-point of the Lagrange functional, i.e., that $L(\vec{v},w) \le L(\vec{v},p) \le L(\vec{u},p)$ for all $\vec{u} \ne \vec{v}$ and $w \ne p$. The right inequality can be shown as follows

$$\begin{aligned} L(\vec{v}+\vec{u},p) &= \tfrac{1}{2}b(\vec{v}+\vec{u},\vec{v}+\vec{u}) - c(\vec{v}+\vec{u},p) + (q,p) \\ &= L(\vec{v},p) + I(\vec{u}) + b(\vec{u},\vec{v}) - c(\vec{u},p) = L(\vec{v},p) + I(\vec{u}) > L(\vec{v},p). \end{aligned}$$

The left inequality follows in a similar manner.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

6.2.2 Discrete Formulation

To discretize the mixed formulation, (6.5), we introduce a grid $\Omega_h = \cup\Omega_i$, replace $L^2(\Omega)$ and $H_0^{\text{div}}(\Omega)$ by finite-dimensional subspaces U and V defined over Ω_h, and rewrite the inner products (6.6) as sums of integrals localized to cells

$$\begin{aligned} b(\cdot,\cdot)_h &: V \times V \to \mathbb{R}, \qquad b(\vec{u},\vec{v})_h = \sum_i \int_{\Omega_i} \vec{u} \cdot \mathbf{K}^{-1}\vec{v}\, d\vec{x}, \\ c(\cdot,\cdot)_h &: V \times U \to \mathbb{R}, \qquad c(\vec{u},p)_h = \sum_i \int_{\Omega_i} p\, \nabla \cdot \vec{u}\, d\vec{x}, \\ (\cdot,\cdot)_h &: U \times U \to \mathbb{R}, \qquad (q,w)_h = \sum_i \int_{\Omega_i} q\, w\, d\vec{x}. \end{aligned} \tag{6.8}$$

To obtain a practical numerical method, the spaces U and V are typically defined as piecewise polynomial functions that are nonzero on a small collection of grid cells. For instance, in the Raviart–Thomas method [262, 53] of lowest order for triangular, tetrahedral, or regular parallelepiped grids, $L^2(\Omega)$ is replaced by

$$U = \{p \in L^2(\cup\Omega_i) : p|_{\Omega_i} \text{ is constant } \forall \Omega_i \subset \Omega\} = \text{span}\{\chi_i\},$$

where χ_i is the characteristic function of grid cell Ω_i. Likewise, $H_0^{\text{div}}(\Omega)$ is replaced by a space V consisting of functions $\vec{v} \in H_0^{\text{div}}(\cup\Omega_i)$ that have linear components on each grid cell $\Omega_i \in \Omega$, have normal components $\vec{v}\cdot\vec{n}_{ik}$ that are constant on each cell interface Γ_{ik}, and are continuous across these interfaces. These requirements are satisfied by functions $\vec{v} \in P_1(\Omega_i)^d$, i.e., first-order polynomials that on each grid cell take the form $\vec{v}(\vec{x}) = \vec{a} + B\vec{x}$, where $\vec{a}$ is a constant vector and B is a matrix. These functions can be parametrized in terms of the faces between two neighboring cells. This means that we can write $V = \text{span}\{\vec{\psi}_{ik}\}$, where each function $\vec{\psi}_{ik}$ is defined as

$$\vec{\psi}_{ik} \in \mathcal{P}_1(\Omega_i)^d \cup \mathcal{P}_1(\Omega_k)^d \quad \text{and} \quad (\vec{\psi}_{ik}\cdot\vec{n}_{jl})|_{\Gamma_{jl}} = \begin{cases} 1, & \text{if} \quad \Gamma_{jl} = \Gamma_{ik}, \\ 0, & \text{otherwise.} \end{cases}$$

Figure 6.3 illustrates the four nonzero basis functions for the special case of a Cartesian grid in 2D.

To derive a fully discrete method, we use χ_i and $\vec{\psi}_{ik}$ as our *trial functions* and express the unknown $p(\vec{x})$ and $\vec{v}(\vec{x})$ as sums over these trial functions, $p = \sum_i p_i \chi_i$ and $\vec{v} = \sum_{ik} v_{ik}\vec{\psi}_{ik}$. We now have two different degrees of freedom, p_i associated with each cell and v_{ik} associated with each interface Γ_{ik}, which we collect in two vectors $\boldsymbol{p} = \{p_i\}$ and $\boldsymbol{v} = \{v_{ik}\}$. Using the same functions as test functions, we derive a linear system of the form

$$\begin{bmatrix} \boldsymbol{B} & -\boldsymbol{C}^T \\ \boldsymbol{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{q} \end{bmatrix}. \tag{6.9}$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

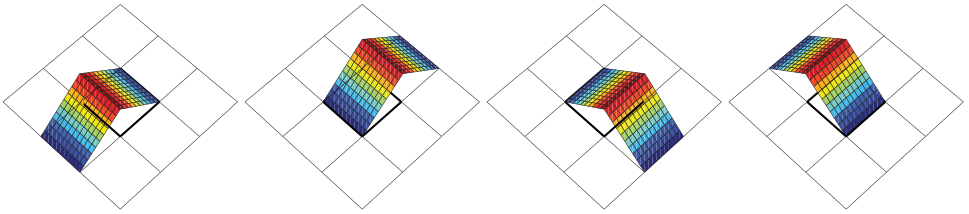

Figure 6.3 Illustration of the velocity basis for the lowest-order Raviart–Thomas method for a 2D Cartesian grid. There are four basis functions that have nonzero support in the interior of the center cell. The basis functions correspond to degrees of freedom associated with the west, east, south, and north cell faces.

Here, $\boldsymbol{B} = [b_{ik,\,jl}]$, $\boldsymbol{C} = [c_{i,kl}]$, and $\boldsymbol{q} = [q_i]$, where:

$$\begin{aligned} b_{ik,\,jl} &= b(\vec{\psi}_{ik}, \vec{\psi}_{jl})_h = \sum_{\ell} \int_{\Omega_\ell} \vec{\psi}_{ik} \cdot \mathbf{K}^{-1} \vec{\psi}_{jl}\, d\vec{x}, \\ c_{i,kl} &= c(\vec{\psi}_{kl}, \chi_i)_h = \int_{\Omega_i} \nabla \cdot \vec{\psi}_{kl}\, d\vec{x}, \\ q_i &= (q, \chi_i)_h = \int_{\Omega_i} q\, d\vec{x}. \end{aligned} \tag{6.10}$$

Note that for the first-order Raviart–Thomas finite elements, we have

$$c_{i,kl} = \begin{cases} 1, & \text{if } i = k, \\ -1, & \text{if } i = l, \\ 0, & \text{otherwise.} \end{cases}$$

The matrix entries $b_{ik,\,jl}$ depend on the geometry of the grid cells and whether $\mathbf{K}$ is isotropic or anisotropic. Unlike the two-point method, mixed methods are consistent and will therefore be convergent also on grids that are not K-orthogonal.

In [2], we presented a simple MATLAB code that in approximately seventy-five code lines implements the lowest-order Raviart–Thomas mixed finite-element method on regular hexahedral grids for flow problems with diagonal tensor permeability in two or three spatial dimensions. The code is divided into three parts: assembly of the $\boldsymbol{B}$ block, assembly of the $\boldsymbol{C}$ block, and a main routine that loads data (permeability and grid), assembles the whole matrix, and solves the system. Figure 6.4 illustrates the sparsity pattern of the mixed system for a $n_x \times n_y \times n_z$ Cartesian grid. The system matrix clearly has the block structure given in (6.9). The matrix blocks $\boldsymbol{B}$ and $\boldsymbol{C}$ each have three nonzero blocks that correspond to velocity basis functions oriented along the three axial directions. That is, degrees of freedom at interfaces (fluxes) have been numbered in the same way as the grid cells; i.e., first faces orthogonal to the x-direction, then faces orthogonal to the y-direction, and finally faces orthogonal to the z-direction, resulting in a heptadiagonal structure.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

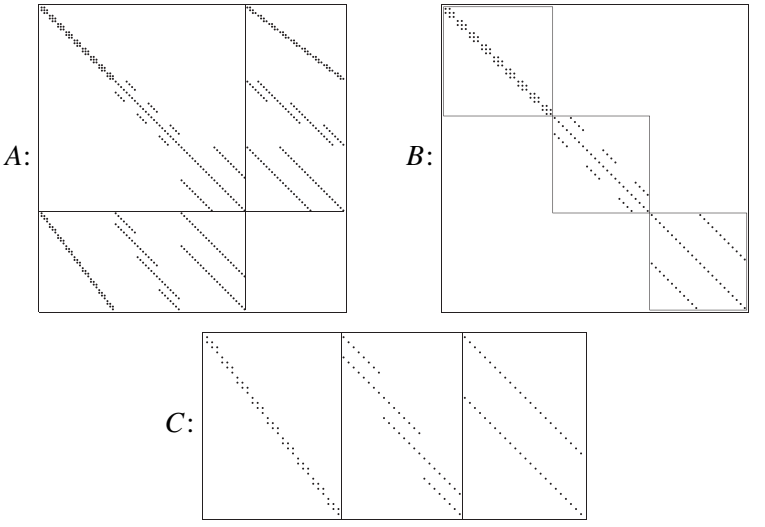

Figure 6.4 Sparsity patterns for the full mixed finite-element matrix $\boldsymbol{A}$, and the matrix blocks $\boldsymbol{B}$ and $\boldsymbol{C}$ for a $4 \times 3 \times 3$ Cartesian grid.

The code presented in [2] exploits the simple geometry of the Cartesian grid to integrate the Raviart–Thomas basis functions exactly and perform a direct assembly of the matrix blocks, giving a MATLAB code that is both compact and efficient. For more general grids, you would typically perform an element-wise assembly by looping over all cells in the grid, map each cell them back to a reference element, and then integrate the individual inner production by use of a suitable numerical quadrature rule. In our experience, this procedure is cumbersome to implement for general stratigraphic and polyhedral grids and would typically require the use of many different reference elements or subdivision of cells.

MRST is designed to work on general polyhedral grids in 3D and does not supply any direct implementation of mixed finite-element methods. Instead, the closest we get to a mixed method is a finite-volume method in which the half-transmissibilities are equivalent to the discrete inner products for the lowest-order Raviart–Thomas (RT0) method on rectangular cuboids; Section 6.4 on page 188 explains more details. In the rest of the current section, we introduce an alternative formulation of the mixed method that gives a smaller linear system that is better conditioned.

6.2.3 Hybrid Formulation

In the same way as we proved that the solution of the mixed problem is a saddle point, we can show that the linear system (6.9) is indefinite. Indefinite systems are harder to solve and generally require special linear solvers. We therefore introduce an alternative formulation that, when discretized, gives a positive-definite discrete system and thereby simplifies the computation of a discrete solution. Later in the chapter, this so-called *hybrid formulation*

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

will form the basis for a general family of finite-volume discretizations on polygonal and polyhedral grids.

A hybrid formulation avoids solving a saddle-point problem by lifting the constraint that the normal velocity must be continuous across cell faces and instead integrate (6.3) to get a weak form that contains jump terms at the cell interfaces. Continuity of the normal velocity component is then reintroduced by adding an extra set of equations, in which the pressure π at the cell interfaces plays the role of Lagrange multipliers. (Recall how Lagrange multipliers were used to impose mass conservation as a constraint in the minimization procedure used to derive the mixed formulation in Section 6.2.1.) Introducing Lagrange multipliers does not change $\vec{v}$ or p, but enables the recovery of pressure values at cell faces, in addition to introducing a change in the structure of the weak equations. Mathematically, the mixed hybrid formulation reads: find $(\vec{v}, p, \pi) \in H_0^{\text{div}}(\Omega_h) \times L^2(\Omega_h) \times H^{\frac{1}{2}}(\Gamma_h)$ such that

$$\begin{aligned} \sum_i \int_{\Omega_i} \left(\vec{u} \cdot \mathbf{K}^{-1} \vec{v} - p \, \nabla \cdot \vec{u} \right) d\vec{x} + \sum_{ik} \int_{\Gamma_{ik}} \pi \, \vec{u} \cdot \vec{n} \, ds = 0, \\ \sum_i \int_{\Omega_i} w \, \nabla \cdot \vec{v} \, d\vec{x} = \int_{\Omega_h} q \, w \, d\vec{x}, \\ \sum_{ik} \int_{\Gamma_{ik}} \mu \, \vec{v} \cdot \vec{n} \, ds = 0 \end{aligned} \tag{6.11}$$

for all test functions $\vec{u} \in H_0^{\text{div}}(\Omega_h)$, $w \in L^2(\Omega_h)$, and $\mu \in H^{\frac{1}{2}}(\Gamma_h)$. Here, Γ_h denotes all the interior faces of the grid and $H^{\frac{1}{2}}(\Gamma_h)$ is the space spanned by the traces[1] of functions in $H^1(\Omega_h)$, i.e., the space of square integrable functions whose derivatives are also square integrable. As for the mixed formulation, we can introduce inner products to write the weak equations (6.11) in a more compact form,

$$\begin{aligned} b(\vec{u}, \vec{v}) - c(\vec{u}, p) + d(\vec{u}, \pi) = 0 \\ c(\vec{v}, w) = (q, w) \\ d(\vec{v}, \mu) = 0, \end{aligned} \tag{6.12}$$

where the inner products $b(\cdot,\cdot)$, $c(\cdot,\cdot)$, and $(\cdot,\cdot)$ are defined as in (6.6), and $d(\cdot,\cdot)$ is a new inner product defined over the interior faces,

$$d(\cdot,\cdot)_h : H_0^{\text{div}}(\Omega_h) \times H^{\frac{1}{2}}(\Gamma_h) \to \mathbb{R}, \qquad d(\vec{v}, \pi) = \sum_{ik} \int_{\Gamma_{ik}} \pi \, \vec{v} \cdot \vec{n} \, ds. \tag{6.13}$$

To derive a fully discrete problem, we proceed in the exact same way as for the mixed problem by first replacing the function spaces L^2, H_0^{div}, and $H^{\frac{1}{2}}$ by finite-dimensional subspaces V and U that are spanned by piecewise polynomial basis functions with local

[1] If you are not familiar with the notion of a trace operator, think of it as the values of a function along the boundary of the domain this function is defined on.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

support, as discussed earlier. In the lowest-order approximation, the finite-dimensional space Π consists of functions that are constant on each face,

$$\Pi = \text{span}\{\mu_{ik}\}, \qquad \mu_{ik}(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \in \Gamma_{ik}, \\ 0, & \text{otherwise.} \end{cases} \tag{6.14}$$

Using these basis functions as test and trial functions, one can derive a discrete linear system of the form,

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{D}^{\mathsf{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix}, \tag{6.15}$$

where the vectors $\boldsymbol{v}$, $\boldsymbol{p}$, and $\boldsymbol{\pi}$ collect the degrees of freedom associated with fluxes across the cell interfaces, face pressures, and cell pressures, respectively, the matrix blocks $\boldsymbol{B}$ and $\boldsymbol{C}$ are defined as in (6.10), and $\boldsymbol{D}$ has two nonzero entries per column (one entry for each side of the cell interfaces).

The linear system (6.15) is an example of a sparse, symmetric, indefinite system, i.e., a system $\boldsymbol{A}$ whose quadratic form $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$ takes both positive and negative values. Several methods for solving such systems can be found in the literature, but these are generally not as efficient as solving a symmetric, positive-definite system. We will therefore use a so-called Schur-complement method to reduce the mixed hybrid system to a positive-definite system. The Schur-complement method basically consists of using a block-wise Gaussian elimination of (6.15) to form a positive-definite system (the Schur complement) for the face pressures,

$$\left(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{D} - \boldsymbol{F}^{\mathsf{T}}\boldsymbol{L}^{-1}\boldsymbol{F}\right)\boldsymbol{\pi} = \boldsymbol{F}^{\mathsf{T}}\boldsymbol{L}^{-1}\boldsymbol{q}. \tag{6.16}$$

Here, $\boldsymbol{F} = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{D}$ and $\boldsymbol{L} = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{C}$. Given the face pressures, the cell pressures and fluxes can be reconstructed by back-substitution, i.e., by solving

$$\boldsymbol{L}\boldsymbol{p} = \boldsymbol{q} + \boldsymbol{F}\boldsymbol{\pi}, \qquad \boldsymbol{B}\boldsymbol{u} = \boldsymbol{C}\boldsymbol{p} - \boldsymbol{D}\boldsymbol{\pi}. \tag{6.17}$$

Unlike the mixed method, the hybrid method has a so-called explicit flux representation, which means that the intercell fluxes can be expressed as a linear combination of neighboring values for the pressure. This property is particularly useful in fully implicit discretizations of time-dependent problems, as discussed in Chapter 7.

This is all we will discuss about mixed and mixed-hybrid methods herein. If you want to learn more about these methods, and how to solve the corresponding linear systems, you should consult some of the excellent books on the subject, for instance [50, 45, 53]. In the rest of the chapter, we discuss how ideas from mixed-hybrid methods can be used to formulate finite-volume methods that are consistent and hence convergent on non-K-orthogonal grids.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

6.3 Finite-Volume Methods on Mixed Hybrid Form

We will now present a large class of consistent, finite-volume methods. Our formulation borrows ideas from the mixed methods, but is much simpler to formulate and implement for general polygonal and polyhedral grids. For simplicity, we assume that all methods can be written on the following local form

$$\boldsymbol{v}_i = \boldsymbol{T}_i(\boldsymbol{e}_i p_i - \boldsymbol{\pi}_i). \tag{6.18}$$

This is essentially the same formulation we used for the one-sided fluxes to develop the TPFA method in (4.51), except that now $\boldsymbol{v}_i$ is a vector of all one-sided fluxes associated with cell Ω_i, $\boldsymbol{\pi}_i$ is a vector of face pressures, $\boldsymbol{T}_i$ is a matrix of one-sided transmissibilities, and $\boldsymbol{e}_i = (1, \ldots, 1)^{\mathsf{T}}$ has one unit value per face of the cell. If we define $\boldsymbol{M}_i = \boldsymbol{T}_i^{-1}$, the local discretization (6.18) can alternatively be written as

$$\boldsymbol{M}_i \boldsymbol{v}_i = \boldsymbol{e}_i p_i - \boldsymbol{\pi}_i. \tag{6.19}$$

Consistent with the discussion in Section 6.2, we refer to the matrix $\boldsymbol{M}_i$ as the *local inner product*. From the local discretizations (6.18) or (6.18) on each cell, we derive a linear system of discrete global equations written on mixed or hybridized mixed form. Both forms are supported in MRST, but herein we only discuss the hybrid form for brevity. In the two-point method, we assembled the linear system by combining mass conservation and Darcy's law into one second-order discrete equation for the pressure. In the mixed formulation, mass conservation and Darcy's law were kept as separate first-order equations that together formed a coupled system for pressure and face fluxes. In the hybrid formulation, the continuity of pressures across cell faces was introduced as a third equation that together with mass conservation and Darcy's law constitute a coupled system for pressure, face pressure, and fluxes. Not all consistent methods need to, or can be, formulated in this form. Nevertheless, using the mixed hybrid formulation will enable us to give a uniform presentation of a large class of schemes.

Going back to the TPFA method formulated in Section 4.4.1, it follows immediately from (4.52) that we can write the method as in (6.18) and that the resulting matrix $\boldsymbol{T}_i$ is diagonal with entries

$$(\boldsymbol{T}_i)_{kk} = \vec{n}_{ik} \cdot \mathbf{K}\vec{c}_{ik}/|\vec{c}_{ik}|^2, \tag{6.20}$$

where the length of the normal vector $\vec{n}_{ik}$ is assumed to be equal the area of the corresponding face. The equivalent form (6.19) follows trivially by setting $\boldsymbol{M}_i = \boldsymbol{T}_i^{-1}$. Examples of consistent methods that can be written in the form (6.18) or (6.19) include the lowest-order RT0 method seen in the previous section, multipoint flux approximation (MPFA) schemes [9, 94, 7], and recently developed mimetic finite-difference methods [52], as well as the coarse-scale formulation of several multiscale methods [95]. For all these methods, the corresponding $\boldsymbol{T}_i$ and $\boldsymbol{M}_i$ will be full matrices. We will come back to more details about specific schemes later in the chapter.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

To derive the global linear system on mixed hybrid form, we augment (6.19) with flux and pressure continuity across cell faces. Assembling contributions from all cells in the grid, gives the following linear system

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{C}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{D}^{\mathsf{T}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ -\boldsymbol{p} \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{q} \\ \mathbf{0} \end{bmatrix}, \tag{6.21}$$

where the first row in the block-matrix equation corresponds to (6.19) for all grid cells. Here, vector $\boldsymbol{v}$ contains the outward fluxes associated with half-faces ordered cell-wise so that fluxes over interior interfaces in the grid appear twice, once for each half-face, with opposite signs. Likewise, the vector $\boldsymbol{p}$ contains the cell pressures, and $\boldsymbol{\pi}$ the face pressures. The matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are block diagonal with each block corresponding to a cell,

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{M}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{M}_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \boldsymbol{M}_n \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{e}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{e}_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \boldsymbol{e}_n \end{bmatrix}, \tag{6.22}$$

where $\boldsymbol{M}_i = \boldsymbol{T}_i^{-1}$. Similarly, each column of $\boldsymbol{D}$ corresponds to a unique interface in the grid and has two unit entries for interfaces between cells in the interior of the grid. The two nonzero entries appear at the indexes of the corresponding internal half-faces in the cell-wise ordering. Similarly, $\boldsymbol{D}$ has a single nonzero entry in each column that corresponds to an interface between a cell and the exterior.

The hybrid system (6.21) is obviously much larger than the linear system for the standard TPFA method, but can be reduced to a positive-definite system for the face pressures, as discussed in Section 6.2.3, and then solved using either MATLAB's standard linear solvers or a highly efficient, third-party solver like AGMG [238, 15]. From (6.16) and (6.17), we see that to compute the Schur complement to form the reduced linear system for face pressures, and to reconstruct cell pressures and face fluxes, we only need $\boldsymbol{B}^{-1}$. Moreover, the matrix $\boldsymbol{L}$ is by construction diagonal, and computing fluxes is therefore an inexpensive operation. Many schemes – including the mimetic method, the MPFA-O method, and the standard TPFA scheme – yield algebraic approximations for the $\boldsymbol{B}^{-1}$ matrix. Thus, (6.21) encompasses a family of discretization schemes whose properties are determined by the choice of $\boldsymbol{B}$, which we will discuss in more detail in Section 6.4.

Before digging into details about specific, consistent methods, we revisit the example discussed on page 177 to demonstrate how use of a consistent method significantly reduces the grid orientation effects seen in Figure 6.2. In MRST, the mimetic methods are implemented in a separate module. Replacing TPFA by a mimetic solver is easy; we first load the `mimetic` module by calling

```
mrstModule add mimetic
```

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

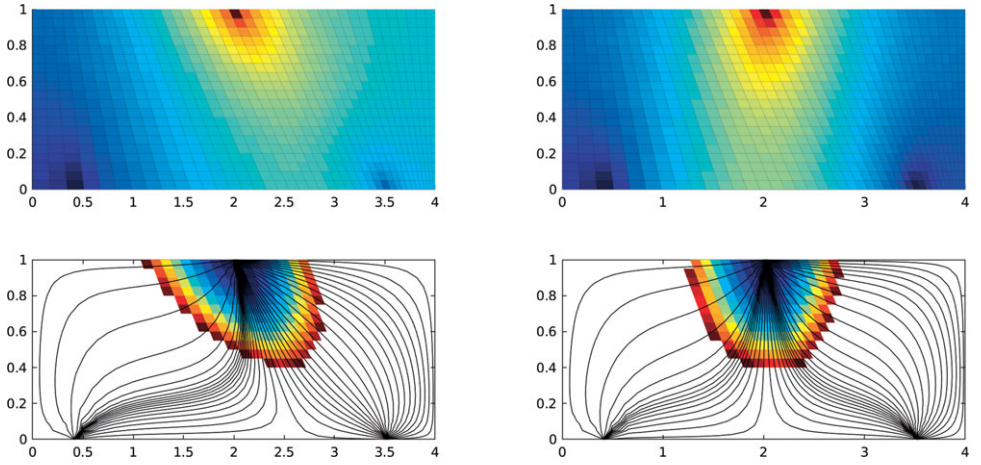

Figure 6.5 Solution of a symmetric flow problem in a homogeneous domain using the TPFA method (left) and the mimetic method (right) on a skew grid that is not K-orthogonal.

Then, we can replace computation of half-face transmissibilities by a call that constructs the local mimetic half-transmissibility $\boldsymbol{T}_i$ or its equivalent inner product $\boldsymbol{M}_i = \boldsymbol{T}_i^{-1}$ for each cell

```
% hT = computeTrans(G, rock);
  S  = computeMimeticIP(G, rock);
```

Likewise, we replace each call to the two-point solver by a call to a function that assembles and solves the global mixed hybrid system

```
% state = incompTPFA(state, G, hT, fluid);
  state = incompMimetic(state, G, S, fluid);
```

We can also use the same routine to assemble a mixed system, and in certain cases also a TPFA-type system. In Figure 6.5 we have applied the mimetic solver to the exact same setup as in Figure 6.2. The approximate solution computed by the mimetic method is almost symmetric and represents a significant improvement compared with the TPFA method. In particular, the difference in travel times between the injector and each producer is reduced from 17% for TPFA to less than 2% for the mimetic method. Moreover, repeating a similar grid refinement study as reported in Figure 6.2 verifies that the consistent mimetic method converges toward the correct solution.

The two calls just outlined constitute all you need to obtain a consistent discretization on general polygonal and polyhedral grids, and if you are not interested in getting to know the inner details of various consistent methods, you can safely jump to the next chapter.

Computer exercises

6.3.1 Run `mimeticExample1` to familiarize yourself with the mimetic solver, the mixed-hybrid formulation, the Schur-component reduction, etc.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

6.3.2 Go back to Exercise 5.4.1a and verify that using the mimetic method cures the grid orientation effects you can observe on the non-K-orthogonal grid.

6.3.3 Can you cure the grid orientation problems observed in Figure 5.7 by using a consistent discretization?

6.3.4 Compare the solutions computed by TPFA and the mimetic method for the SAIGUP model as set up in Section 5.4.4. In particular, you should compare discrepancies in pressure and time-of-flight values at the well perforations. Which solution do you trust most?

Hints: (i) Remember to use the correct inner product when setting up well models. (ii) The mimetic linear system is much larger than the TPFA system and you may have to use an iterative solver like AGMG to reduce runtime and memory consumption.

6.4 The Mimetic Method

Mimetic finite-difference methods are examples of so-called *compatible spatial discretizations* that are constructed so that they not only provide accurate approximation of the mathematical models but also inherit or mimic fundamental properties of the differential operators and mathematical solutions they approximate. Examples of properties include conservation, symmetries, vector calculus identities, etc. Such methods have become very popular in recent years and are currently used in wide range of applications [40].

Mimetic methods can be seen as a finite-volume generalization of finite-differences or (low-order) mixed-finite element methods to general polyhedral grids. The methods are defined in a way that introduces a certain freedom of construction that naturally leads to a family of methods. By carefully picking the parameters needed to fully specify a method, one can construct mimetic methods that coincide with other known methods, or reduce to these methods (e.g., the two-point method, the RT0 mixed finite-element method, or the MPFA-O multipoint method) on certain types of grids.

The mimetic methods discussed herein can all be written on the equivalent forms (6.18) or (6.19) and are constructed so that they are exact for linear pressure fields and give a symmetric positive-definite matrix $\boldsymbol{M}$. In addition, the methods use discrete pressures and fluxes associated with cell and face centroids, respectively, and consequently resemble finite-difference methods.

If we write a linear pressure field in the form $p = \vec{x} \cdot \vec{a} + b$ for a constant vector $\vec{a}$ and scalar b, the corresponding Darcy velocity is $\vec{v} = -\mathbf{K}\vec{a}$. Let $\vec{n}_{i,k}$ denote the area-weighted normal vector to Γ_{ik} and $\vec{c}_{i,k}$ be the vector pointing from the centroid of cell Ω_i to the centroid of face Γ_{ik}, as seen in Figure 6.6. Using Darcy's law and the notion of a one-sided transmissibility $\boldsymbol{T}_{i,k}$ (which will generally not be the same as the two-point transmissibility), the flux and pressure drop can be related as follows,

$$v_{i,k} = -\vec{n}_{i,k}\mathbf{K}\vec{a} = T_{i,k}(p_i - \pi_{i,k}) = -T_{i,k}\vec{c}_{i,k} \cdot \vec{a}. \tag{6.23}$$

To get these relations in one of the local forms (6.18) and (6.19), we collect all vectors $\vec{c}_{i,k}$ and $\vec{n}_{i,k}$ defined for cell Ω_i as rows in two matrices $\boldsymbol{C}_i$ and $\boldsymbol{N}_i$. Because relation (6.23) is

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

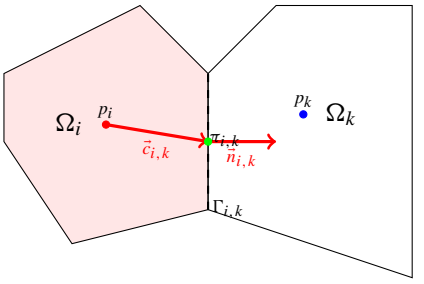

Figure 6.6 Two neighboring cells with quantities used to define mimetic methods.

required to hold for all linear pressure drops, i.e., for an arbitrary vector $\vec{a}$, we see that the matrices $\boldsymbol{M}_i$ and $\boldsymbol{T}_i$ must satisfy the following consistency conditions

$$\boldsymbol{MNK} = \boldsymbol{C}, \qquad \boldsymbol{NK} = \boldsymbol{TC}, \tag{6.24}$$

where we have dropped the subscript i that identifies cell number. These general equations give us (quite) large freedom in how to specify a specific discretization method: any method in which the local inner product $\boldsymbol{M}_i$, or equivalently the local transmissibility matrix $\boldsymbol{T}_i$, is positive definite and satisfies (6.24) will give a consistent, first-order accurate discretization. In the rest of the section, we discuss various choices of valid inner products $\boldsymbol{M}_i$ or reverse inner products $\boldsymbol{T}_i$.

General Family of Inner Products

The original article [52] that first introduced the mimetic methods considered herein, discussed inner products for discrete velocities. However, as we have seen in previous chapters, it is more common in the simulation of flow in porous media to consider intercell fluxes as the primary unknowns. We therefore consider inner products of fluxes rather than velocities in the following. The relation between the two is trivial: an inner product of velocities becomes an inner product for fluxes by pre- and post-multiplying by the inverse of the area of the corresponding cell faces. In other words, if $\boldsymbol{A}$ is a diagonal matrix with element A_{jj} equal the area of the j-th face, the flux inner product $\boldsymbol{M}_{\text{flux}}$ is related to the velocity inner product $\boldsymbol{M}_{\text{vel}}$ through

$$\boldsymbol{M}_{\text{flux}} = \boldsymbol{A}^{-1}\boldsymbol{M}_{\text{vel}}\boldsymbol{A}^{-1}. \tag{6.25}$$

To simplify the derivation of valid solutions, we start by stating a geometrical property that relates $\boldsymbol{C}$ and $\boldsymbol{N}$ as follows (for a proof, see [52]):

$$\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N} = \boldsymbol{V} = \text{diag}(|\Omega_i|). \tag{6.26}$$

Next, we multiply the left matrix equation in (6.24) by $\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N}$

$$\boldsymbol{C}\left(\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N}\right) = \left(\boldsymbol{MNK}\right)\left(\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N}\right) = \boldsymbol{MNC}^{\mathsf{T}}\boldsymbol{N} = \boldsymbol{MNV},$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

from which it follows that there exists a family of valid solution of the form

$$\boldsymbol{M} = \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{M}_2, \tag{6.27}$$

where $\boldsymbol{M}_2$ is a matrix defined such that $\boldsymbol{M}_2\boldsymbol{N} = \boldsymbol{0}$, i.e., any matrix whose rows lie in the left nullspace of $\boldsymbol{N}^{\mathsf{T}}$. Moreover, to make a sensible method, we must require that $\boldsymbol{M}$ is symmetric positive definite. In other words, any symmetric and positive-definite inner product that fulfills these requirements can be represented in two alternative compact forms,

$$\begin{aligned}\boldsymbol{M} &= \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{Q}_N^{\perp}\boldsymbol{S}_M\boldsymbol{Q}_N^{\perp\mathsf{T}} \\ &= \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{P}_N^{\perp}\boldsymbol{S}_M\boldsymbol{P}_N^{\perp}.\end{aligned} \tag{6.28}$$

Here, $\boldsymbol{S}_M$ denotes any symmetric positive-definite matrix, $\boldsymbol{Q}_N^{\perp}$ is an orthonormal basis for the left nullspace of $\boldsymbol{N}^{\mathsf{T}}$, and $\boldsymbol{P}_N^{\perp}$ is the nullspace projection $\boldsymbol{I} - \boldsymbol{Q}_N\boldsymbol{Q}_N^{\mathsf{T}}$, in which $\boldsymbol{Q}_N$ is a basis for spaces spanned by the columns of $\boldsymbol{N}$. Similarly, we can derive a closed expression for the inverse inner product $\boldsymbol{T}$ by multiplying the right matrix equation in (6.24) by $\boldsymbol{V}^{-1}$ from the left and by $\boldsymbol{V}$ from the right and using the identity (6.26),

$$\boldsymbol{T}\boldsymbol{C} = \boldsymbol{V}^{-1}(\boldsymbol{N}\boldsymbol{K})\boldsymbol{V} = \boldsymbol{V}^{-1}(\boldsymbol{N}\boldsymbol{K})(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N})^{\mathsf{T}} = \boldsymbol{V}^{-1}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})\boldsymbol{C}.$$

By the same argument as for $\boldsymbol{M}$, mutatis mutandis, we obtain the following family of inverse inner products,

$$\begin{aligned}\boldsymbol{T} &= \frac{1}{|\Omega_i|}\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + \boldsymbol{Q}_C^{\perp}\boldsymbol{S}_T\boldsymbol{Q}_C^{\perp\mathsf{T}} \\ &= \frac{1}{|\Omega_i|}\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + \boldsymbol{P}_C^{\perp}\boldsymbol{S}_T\boldsymbol{P}_C^{\perp},\end{aligned} \tag{6.29}$$

where $\boldsymbol{Q}_C^{\perp}$ is an orthonormal basis for the left nullspace of $\boldsymbol{C}^{\mathsf{T}}$ and $\boldsymbol{P}_C^{\perp} = \boldsymbol{I} - \boldsymbol{Q}_C\boldsymbol{Q}_C^{\mathsf{T}}$ is the corresponding nullspace projection.

The matrices $\boldsymbol{M}$ and $\boldsymbol{T}$ in (6.28) and (6.29) are evidently symmetric, so we only need to prove that they are positive definite. We start by writing the inner product as $\boldsymbol{M} = \boldsymbol{M}_1 + \boldsymbol{M}_2$, and observe that each of these matrices are positive semi-definite. Hence, our result follows if we can prove that this implies that $\boldsymbol{M}$ is positive definite. Let $\boldsymbol{z}$ be an arbitrary nonzero vector, which we split uniquely as $\boldsymbol{z} = \boldsymbol{N}\boldsymbol{x} + \boldsymbol{y}$, where $\boldsymbol{x}$ lies in the column space of $\boldsymbol{N}$ and $\boldsymbol{y}$ lies in the left nullspace of $\boldsymbol{N}^{\mathsf{T}}$. If $\boldsymbol{y}$ is zero, we have

$$\begin{aligned}\boldsymbol{z}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{z} &= \boldsymbol{x}^{\mathsf{T}}\boldsymbol{N}^{\mathsf{T}}\boldsymbol{M}_1\boldsymbol{N}\boldsymbol{x} \\ &= |\Omega_i|^{-1}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{N}^{\mathsf{T}}\boldsymbol{C})\boldsymbol{K}^{-1}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{N})\boldsymbol{x} = |\Omega_i|\,\boldsymbol{x}^{\mathsf{T}}\boldsymbol{K}^{-1}\boldsymbol{x} > 0\end{aligned}$$

because $\boldsymbol{K}^{-1}$ is a positive definite matrix. If $\boldsymbol{y}$ is nonzero, we have

$$\boldsymbol{z}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{z} = \boldsymbol{z}^{\mathsf{T}}\boldsymbol{M}_1\boldsymbol{z} + \boldsymbol{y}^{\mathsf{T}}\boldsymbol{M}_2\boldsymbol{y} > 0$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

because $z^{\mathsf{T}} M_1 z \geq 0$ and $y^{\mathsf{T}} M_2 y > 0$. An analogous argument holds for the matrix T, and hence we have proved that (6.28) and (6.29) give a well-defined family of consistent discretizations.

So far, we have not put any restrictions on the matrices S_M and S_T that will determine the specific methods, except for requiring that they should be positive definite. In addition, these matrices should mimic the scaling properties of the continuous equation, which is invariant under affine transformations of space and permeability,

$$\vec{x} \mapsto \sigma \vec{x} \quad \text{and} \quad \mathbf{K} \mapsto \sigma^{\mathsf{T}} \mathbf{K} \sigma, \tag{6.30}$$

To motivate how we should choose the matrices S_M and S_T to mimic these scaling properties, we look at a simple 1D example:

Example 6.4.1 *Consider the grid cell $x \in [-1, 1]$, for which $N = C = [1, -1]^{\mathsf{T}}$ and $Q_N^{\perp} = Q_C^{\perp} = \frac{1}{\sqrt{2}}[1, 1]^{\mathsf{T}}$. Hence*

$$\begin{aligned} M &= \frac{1}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \frac{1}{K} [1, -1] + \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} S_M [1, 1], \\ T &= \frac{1}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix} K [1, -1] + \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} S [1, 1]. \end{aligned} \tag{6.31}$$

The structure of the inner product should be invariant under scaling of K and thus we can write the inner product as a one-parameter family of the form

$$\begin{aligned} M &= \frac{1}{2K}\left(\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \frac{2}{t}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right), \\ T &= \frac{K}{2}\left(\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \frac{t}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right). \end{aligned} \tag{6.32}$$

Having established a plausible way of scaling the inner products, we are now in a position to go through various specific choices that are implemented in the `mimetic` module of MRST and look at correspondence between these methods and the standard two-point method, the lowest-order RT0 mixed method, and the MPFA-O method. Our discussion follows [192].

General Parametric Family

Motivated by Example 6.4.1, we propose to choose the matrix S_T as the diagonal of the first matrix term in (6.29) so that these two terms scale similarly under transformations of the type (6.30). Using this definition of S_T and allowing that M should be equal to T^{-1} suggests the following general family of inner products that only differ in the constant in front of the second (regularization) matrix term:

$$\begin{aligned} M &= \frac{1}{|\Omega_i|} C K^{-1} C^{\mathsf{T}} + \frac{|\Omega_i|}{t} P_N^{\perp} \operatorname{diag}(N K N^{\mathsf{T}})^{-1} P_N^{\perp}, \\ T &= \frac{1}{|\Omega_i|}\Big[N K N^{\mathsf{T}} + t\, P_C^{\perp} \operatorname{diag}(N K N^{\mathsf{T}}) P_C^{\perp}\Big]. \end{aligned} \tag{6.33}$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

In MRST, this family of inner products is called `'ip_qfamily'`, and the parameter t is supplied in a separate option:

```
S = computeMimeticIP(G, rock, 'InnerProduct', 'ip_qfamily', 'qparam', t);
```

As we will see shortly, mimetic inner products that reduce to the standard TPFA method or the RT0 mixed finite-element method on simple grids are members of this family.

Two-Point Type Methods

A requirement of the two-point method is that the transmissibility matrix $\boldsymbol{T}$ (and hence also the matrix $\boldsymbol{M}$ of the inner product) should be diagonal. Looking at (6.24), we see that this is only possible if the vectors $\mathbf{K}\vec{N}_{ik}$ and $\vec{c}_{ik}$ are parallel, which is the exact same condition for K-orthogonality that we argued was sufficient to guarantee consistency of the two-point method on page 176. An explicit expression for the diagonal entries of the two-point method on K-orthogonal grids has already been given in (6.20). Strictly speaking, this relation does not always yield a positive value for the two-point transmissibility on any grid. For instance, for corner-point grids it is normal to define the face centroids as the arithmetic mean of the associated corner-point nodes and the cell centroids as the arithmetic mean of the centroids of the top and bottom cell faces, which for most grids will guarantee a positive transmissibility.

The extension of the two-point to non-orthogonal grids is not unique. Here, we present a mimetic method that coincides with the two-point methods in this method's region of validity and at the same time gives a valid mimetic inner product and hence a consistent discretization for all types of grids and permeability tensors. One advantage of this method, compared with multipoint flux-approximation methods, is that the implementation is simpler for general unstructured grids. The most instructive way to introduce the general method is to look at a simple example that will motivate the general construction.

Example 6.4.2 *We consider the grid cell* $[-1,1]\times[-1,1]$ *in 2D and calculate the components that make up the inverse inner product* $\boldsymbol{T}$ *for a diagonal and a full permeability tensor (*$\mathbf{K}_1$ *and* $\mathbf{K}_2$*, respectively) and compare with the corresponding two-point discretization. We start by setting up the permeability and the geometric properties of the cell*

```
K1 = eye(2);  K2 = [1 .5; .5 1];
C = [-1 0; 1 0; 0 -1; 0 1]; N = 2*C; vol = 4;
```

Using the definition (6.20), we see that the transmissibility matrix resulting from the standard two-point discretization can be computed as:

```
T = diag(diag(N*K*C')./sum(C.*C,2))
```

The result is `T=diag([2 2 2 2])` *for both permeability tensors, which clearly demonstrates that the scheme is not consistent for* $\mathbf{K}_2$*. To construct the inverse inner product, we start by computing the nullspace projection*

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

```
Q = orth(C);
P = eye(size(C,1)) - Q*Q';
```

We saw earlier that as a simple means of providing a positive definite matrix $\boldsymbol{S}_T$ that scales similarly to the first term in the definition of $\boldsymbol{T}$ in (6.29), we could choose $\boldsymbol{S}_T$ as a multiple of the diagonal of this matrix:

```
W = (N * K * N') ./ vol;
St = diag(diag(W));
```

Collecting our terms, we see that the inverse inner product will be made up of the following two terms for the case with the diagonal permeability tensor:

```
W =  1   -1    0    0      P*St*P =  0.5   0.5    0    0
    -1    1    0    0                0.5   0.5    0    0
     0    0    1   -1                  0     0   0.5  0.5
     0    0   -1    1                  0     0   0.5  0.5
```

If we now define the inverse inner product as

```
T = W + 2*P*St*P
```

the resulting method will coincide with the diagonal transmissibility matrix for diagonal permeability and give a full matrix for the tensor permeability,

```
T1 = 2.0    0    0    0     T2 =  2.0    0   0.5  -0.5
     0    2.0    0    0           0    2.0  -0.5   0.5
     0     0   2.0    0         0.5  -0.5   2.0    0
     0     0    0   2.0        -0.5   0.5    0   2.0
```

Altogether, we have derived a discrete inner product that generalizes the two-point method to full tensor permeabilities on 2D Cartesian grids and gives a consistent discretization also when the grid is not K-orthogonal.

Motivated by this example, we define a quasi-two-point inner product that simplifies to the standard TPFA method on Cartesian grids with diagonal permeability tensor:

$$\begin{aligned}\boldsymbol{M} &= \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}} + \frac{|\Omega_i|}{2}\boldsymbol{P}_N^{\perp}\operatorname{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})^{-1}\boldsymbol{P}_N^{\perp},\\ \boldsymbol{T} &= \frac{1}{|\Omega_i|}\Big[\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + 2\,\boldsymbol{P}_C^{\perp}\operatorname{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})\boldsymbol{P}_C^{\perp}\Big].\end{aligned} \tag{6.34}$$

The observant reader will notice that this is a special case for $t = 2$ of the general family (6.33) of inner products. In MRST, this inner product is constructed by

```
S = computeMimeticIP(G, rock, 'InnerProduct', 'ip_quasitpf');
```

For completeness, the `mimetic` module also supplies a standard, diagonal two-point inner product that is not generally consistent. This is invoked by:

```
S = computeMimeticIP(G, rock, 'InnerProduct', 'ip_tpf');
```

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

Raviart–Thomas-Type Inner Product

To compute the mixed finite-element inner product on cells that are not simplexes or hexahedrons aligned with the coordinate axes, the standard approach is to map the cell back to a unit reference cell on which the mixed basis functions are defined and perform the integration there. For a general hexahedral cell in 3D, the resulting integral would conceptually look something like

$$\iiint_{\Omega_i} f(\vec{x})\, d\vec{x} = \iiint_{[0,1]^3} f(\vec{x}(\xi,\eta,\zeta))\, |J(\xi,\eta,\zeta)|\, d\xi d\eta d\zeta,$$

where $J = \partial(x,y,z)/\partial(\xi,\eta,\zeta)$ denotes the Jacobian matrix of the coordinate transformation. The determinant of J is generally nonlinear and it is therefore not possible to develop a mimetic inner product that equals the lowest-order, RT0 inner product on general polygonal or polyhedral cells, and at the same time is simpler to compute.

Instead, we develop an inner product that is equivalent to RT0 on grids that are orthogonal and aligned with the principal axes of the permeability tensor. To motivate the definition of this inner product, we first look at a simple example.

Example 6.4.3 *The RT0 basis functions defined on the reference element in two spatial dimensions read*

$$\vec{\psi}_1 = \begin{pmatrix} 1-x \\ 0 \end{pmatrix}, \quad \vec{\psi}_2 = \begin{pmatrix} x \\ 0 \end{pmatrix}, \quad \vec{\psi}_3 = \begin{pmatrix} 0 \\ 1-y \end{pmatrix}, \quad \vec{\psi}_4 = \begin{pmatrix} 0 \\ y \end{pmatrix}.$$

We compute the elements of the corresponding inner product matrix for a diagonal permeability with unit entries, $\boldsymbol{K} = \boldsymbol{I}$. This entails careful treatment of a sticky point: whereas the mixed inner product involves velocities, the mimetic inner product involves fluxes that are considered positive when going out of a cell and negative when going into the cell. To get comparative results, we thus reverse the sign of $\vec{\psi}_1$ and $\vec{\psi}_3$. Using symmetry and the fact that $\vec{\psi}_i \cdot \vec{\psi}_k = 0$ if $i = 1,2$ and $k = 3,4$, we only have to compute two integrals:

$$\int_0^1 \int_0^1 \vec{\psi}_1 \cdot \vec{\psi}_1 \, dxdy = \int_0^1 (x-1)^2 \, dx = \tfrac{1}{3}$$
$$\int_0^1 \int_0^1 \vec{\psi}_1 \cdot \vec{\psi}_2 \, dxdy = \int_0^1 (x-1)x \, dx = -\tfrac{1}{6}\,.$$

This means that the RT0 inner product in 2D reads

$$\boldsymbol{M} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & 0 & 0 \\ -\frac{1}{6} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & -\frac{1}{6} \\ 0 & 0 & -\frac{1}{6} & \frac{1}{3} \end{bmatrix}.$$

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

Similarly as in Example 6.4.2, we can compute the two matrices used to form the mimetic inner product

```
C  = .5*[-1 0; 1 0; 0 -1; 0 1]; N = 2*C; vol = 1; K = eye(2);
Q  = orth(N);
P  = eye(size(C,1)) - Q*Q';
Sm = diag(1 ./ diag(N*K*N'))*vol;
M1 = C*(K\C')./vol
M2 = P*Sm*P
```

The result of this computation is

```
M1 = 0.25  -0.25     0     0      M2 = 0.50   0.50     0     0
    -0.25   0.25     0     0           0.50   0.50     0     0
        0      0  0.25 -0.25              0      0  0.50  0.50
        0      0 -0.25  0.25              0      0  0.50  0.50
```

from which it follows that $\boldsymbol{M}_2$ should be scaled by $\frac{1}{6}$ if the mimetic inner product is to coincide with the RT0 mixed inner product.

For a general grid, we define a quasi-RT0 inner product that simplifies to the standard RT0 inner product on orthogonal grids with diagonal permeability tensors, as well as on all other cases that can be transformed to such a grid by an affine transformation of the form (6.30). The quasi-RT0 inner product reads

$$\begin{aligned}\boldsymbol{M} &= \frac{1}{|\Omega_i|}\boldsymbol{C}\boldsymbol{K}^{-1}\boldsymbol{C}^{\mathsf{T}} + \frac{|\Omega_i|}{6}\boldsymbol{P}_N^{\perp}\operatorname{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})^{-1}\boldsymbol{P}_N^{\perp},\\ \boldsymbol{T} &= \frac{1}{|\Omega_i|}\Big[\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + 6\,\boldsymbol{P}_C^{\perp}\operatorname{diag}(\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}})\boldsymbol{P}_C^{\perp}\Big].\end{aligned} \tag{6.35}$$

and is a special case of the general family (6.33) of inner products for $t = 6$. In MRST, this inner product is constructed as follows

```
S = computeMimeticIP(G, rock, 'InnerProduct', 'ip_quasitpf');
```

For completeness, the `mimetic` module also supplies a standard RT0 inner product, `'ip_rt'`, that is only valid on Cartesian grids.

Default Inner Product in MRST

For historical reasons, the default discretization in the `mimetic` module of MRST corresponds to a mimetic method with half-transmissibilities defined by the following expression,

$$\boldsymbol{T} = \frac{1}{|\Omega_i|}\Big[\boldsymbol{N}\boldsymbol{K}\boldsymbol{N}^{\mathsf{T}} + \frac{6}{d}\operatorname{tr}(\boldsymbol{K})\boldsymbol{A}\,(\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^{\mathsf{T}})\,\boldsymbol{A}\Big]. \tag{6.36}$$

Here, $\boldsymbol{A}$ is the diagonal matrix containing face areas and $\boldsymbol{Q}$ is an orthogonal basis for the range of $\boldsymbol{A}\boldsymbol{C}$. The inner product, referred to as `'ip_simple'`, was inspired by [52] and introduced in [6] to resemble the RT0 inner product; they are equal for scalar permeability

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

on orthogonal grids, which can be verified by inspection. Because the inner product is based on velocities, it involves pre- and post-multiplication of (inverse) face areas and is in this sense different from the general class of inner products discussed in (6.33). It is nevertheless possible to show that the eigenspace corresponding to the nonzero eigenvalues of the second matrix term is equal to the nullspace for $\boldsymbol{C}$.

Local-Flux Mimetic Method

Another large class of consistent discretization methods that have received a lot of attention in recent years is the multipoint flux-approximation (MPFA) schemes [9, 94, 7]. Discussing different variants of this method is beyond the scope of the current presentation, mainly because efficient implementation of a general class of MPFA schemes requires some additional mappings that are not yet part of the standard grid structure outlined in Section 3.4. These mappings are available and have been used in some of the add-on modules for geomechanics, and it should therefore not be too difficult for the competent and interested reader to also implement other MPFA type schemes.

Having said this, there *is* a module `mpfa` in MRST that gives a simple implementation of the MPFA-O method. In this implementation, we have utilized the fact that some variants of the method can be formulated as a mimetic method. This was first done by [281, 199] and is called the local-flux mimetic formulation of the MPFA method. In this approach, each face in the grid is subdivided into a set of subfaces, one subface for each of the nodes that make up the face. The inner product of the local-flux mimetic method gives exact results for linear flow and is block diagonal with respect to the faces corresponding to each node of the cell, but it is not symmetric. The block-diagonal property makes it possible to reduce the system into a cell-centered discretization for the cell pressures. This naturally leads to a method for calculating the MPFA transmissibilities. The crucial point is to have the corner geometry in the grid structure and handle the problems with corners that do not have three unique half-faces associated.

The local-flux mimetic formulation of the MPFA-O method can be used with a calling sequence that is similar to that of the TPFA and the mimetic methods:

```
mrstModule add mpfa;
hT = computeMultiPointTrans(G, rock);
state = incompMPFA(state, G, hT, fluid)
```

By default, `computeMultiPointTrans` uses MATLAB to invert a sequence of small matrices that arise in the discretization. This operation is relatively slow and the routine therefore takes an optional parameter `'invertBlocks'` that can be set equal to `'MEX'` to invoke C-accelerated routines for the inversion of the small matrix blocks. This option may not work if MRST has not been properly set up to automatically compile C-code on your computer.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

6.5 Monotonicity

The main motivation for developing consistent discretizations is to get schemes that are convergent for rough grids and full-tensor permeabilities and less influenced by grid orientation errors. All consistent methods discussed in this chapter fulfill this requirement. On the other hand, these methods are all linear and not guaranteed to satisfy discrete counterparts of the maximum principle fulfilled by the continuous solution. For the Poisson equation $\nabla \cdot \mathbf{K}\nabla p = q$, this principle implies that if there is a single source within the domain, the pressure will decrease monotonically from the source toward the boundary. The discrete solution should also follow a similar principle, in which case we say that the underlying scheme is *monotone*.

If a scheme is not monotone, the discrete solutions may contain oscillations that are not present in the underlying physical system, particularly for cases with full-tensor permeabilities and large anisotropy ratios and/or high aspect ratio grids. While such oscillations are not very serious for single-phase, incompressible flow, they may have a significant impact on multiphase flows, which can be highly sensitive to small pressure variations; think of an oil–gas system operating slightly above the bubble point, for which artificial gas is liberated if the pressure falls below the bubble point; see Chapter 11. A classical result says that a discretization scheme is monotone if it generates an *M-matrix*, i.e., a matrix whose off-diagonal elements are less than or equal zero and whose eigenvalues have positive real parts. Two-point schemes are strictly monotone if all transmissibilities are nonnegative. Multipoint schemes, on the other hand, do not necessarily give an M-matrix, and extensive research has gone into investigate their monotonicity properties [236, 234, 235, 12, 157] and proving convergence [308, 26, 103, 104, 96, 314, 10, 167, 11]. In 2D, this has lead to the somewhat negative result [157]: *no linear nine-point control volume method can be constructed for quadrilateral grids in 2D that is exact for linear solutions while remaining monotone for general problems.*.

For brevity, we will only present an example that illustrates lack of monotonicity for linear, consistent schemes. We refer the interested reader to [235] and references therein for a thorough discussion of the underlying mathematical properties that cause these unphysical oscillations.

Example 6.5.1 *(*Example 2 from [192]) *To illustrate the different behavior of TPFA and linear, consistent discretizations, we consider a 2D domain of size $n \times n$, where n is also the number of grid cells in each spatial direction. To generate a non-orthogonal grid, we use the* `twister` *routine, which you have encountered multiple times already. This routine normalizes all coordinates in a rectilinear grid to the interval* $[0, 1]$*, then perturbs all interior points by adding* $0.03 \sin(\pi x) \sin(3\pi(y - 1/2))$ *to the x-coordinates and subtracting the same value from the y-coordinate before transforming back to the original domain. This creates a non-orthogonal, but smoothly varying, logically Cartesian grid. The domain has Dirichlet boundary conditions on the left and right-hand sides, with values one and zero, respectively, and zero Neumann conditions at the top and bottom.*

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

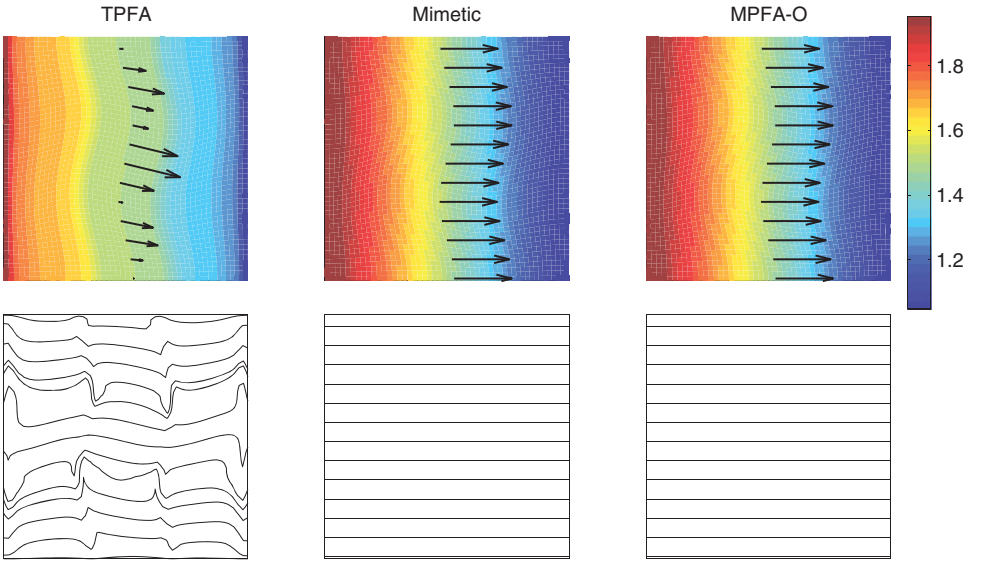

Figure 6.7 Grid orientation effects for a flow described by a linear pressure drop from the left to the right boundary. Approximate solutions computed by three different incompressible solvers from the `incomp`, `mimetic`, and `mpfa` modules, respectively. The permeability field is homogeneous with anisotropy ratio 1:1,000 aligned with the grid.

An important design principle for consistent discretizations is that they, unlike the TPFA method, should reproduce flows that result from a linear pressure drop exactly on grids that are not K-orthogonal. We therefore start by considering a diagonal permeability tensor with anisotropy ratio 1:1,000 aligned with the x-axis. Figure 6.7 compares the discrete solutions computed TPFA, the local-flux mimetic (MPFA-O) method, and the mimetic method with inner product `'ip_simple'` *on a* 101×101 *grid. Whereas the mimetic and MPFA-O schemes produce the expected result, the pressure and velocity solutions computed by TPFA show significant grid-orientation effects and are obviously wrong.*

To illustrate non-monotonic behavior, we repeat the experiment with the tensor rotated by $\pi/6$ *on a grid with* 21×21 *cells; see Figure 6.8. The solution computed by TPFA is monotone, but fails to capture the correct flow pattern in which streamlines starting near the upper-left corner should exit near the lower-right corner. Judging from the color plots, the pressure values computed by the mimetic and MPFA-O methods appear to be correct and have no discernible oscillations. However, both the streamline distribution and the velocity vectors plotted for cells along the midsection of the grid clearly show that the resulting velocity fields are highly oscillatory, in particular for the mimetic method. If we change the inner product from its default value (* `'ip_simple'`*) to, e.g., the quasi-two-point inner product (* `'ip_quasitpf'`*), the solution is qualitatively the same as for the MFPA-O method.*

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

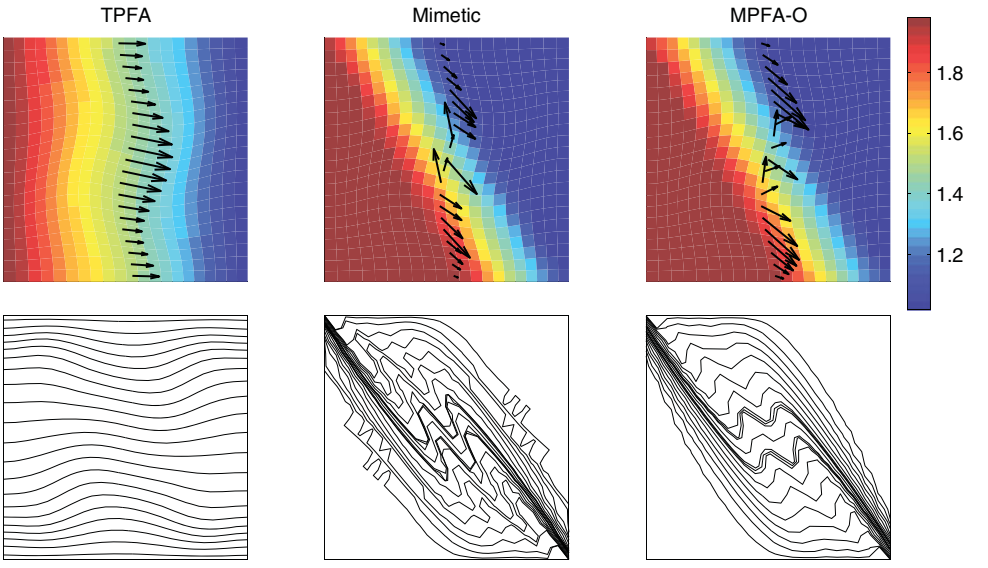

Figure 6.8 Monotonicity illustrated with the same setup as in Figure 6.8, but with the anisotropy ratio 1:1,000, making an angle $\pi/6$ with the x-direction.

6.6 Discussion

In this chapter, we have shown that TPFA is only consistent on K-orthogonal grids and hence may give significant grid orientation errors for tensor permeabilities and general cell geometries. On the other hand, the method is monotone and coercive (sum of fluxes is always positive), has a rather sparse stencil, and is hence computationally efficient. It is also possible to reduce grid orientation errors in the TPFA method by modifying the stencil. For tetrahedral grids satisfying the Delaunay criterion, one can design consistent two-point schemes by computing transmissibilities using the circumcenter rather than the cell centroid. You can also reduce numerical artifacts for general cell geometries by either optimizing the point used to compute intercell transmissibilities and/or use fluxes from a consistent method to adjust the transmissibilities. As a general precaution, I would recommend that you compute solutions using both a two-point and a consistent method to get an idea of the amount of grid orientation when working with stratigraphic or other unstructured representations of complex geology. This applies particularly to multiphase flow, for which you can get a good idea of the grid orientation errors by solving a representative single-phase problem.

Which consistent method should you choose amongst those implemented in MRST? The consistent MPFA method is cell-centered like the TPFA method and thus has the same number of unknowns, but the resulting stencil is denser and the method is therefore significantly less computationally efficient. Compared with mimetic methods, MPFA is quite involved to implement on general polyhedral meshes, and the method is not very

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

robust on very skewed and deformed grids and for high aspect ratios or strong anisotropy [267]. In my experience, the mimetic method seems more robust and would therefore be my favorite choice, despite a higher number of unknowns.

There are also many other consistent methods than those discussed herein. Virtual element methods [38, 39] can be seen as a variational analog of mimetic methods and have proved to be versatile and robust for discretizing diffusion, dispersion, and mechanical effects. Unfortunately, the methods are not locally conservative when applied to Poisson-type flow equations, and thus require postprocessing to compute useful fluxes. MRST offers an implementation in the recent `vem` module [168], but this is currently primarily of academic interest when it comes to flow equations.

To develop schemes that are both consistent *and* monotone, several authors have started looking into *nonlinear two-point methods* [179, 198, 226, 272]. In these methods, the flux approximation is written

$$v_{i,k} = T_i(\vec{p})p_i - T_k(\vec{p})p_k,$$

where the transmissibilities depend on one or more pressure values so that the method becomes nonlinear. The disadvantage is that you have to solve a nonlinear system, but this additional cost may not present a significant drawback when the method is used for compressible and multiphase flow, where you need to use a nonlinear solver to account for other nonlinear couplings in the model equations, as we will see later in the book. At the time of writing, MRST only contains a preliminar implementation of nonlinear TPFA schemes. You can find more detailed comparisons of all the methods discussed previously in [169].

Another example of a recent and promising method is the vertex approximation gradient (VAG) method [105, 106, 267]. Unlike other nodal methods, this method has unknowns associated with both vertices and cell centers and relies on a local triangulation of the grid cells to enable simple extension of finite-elements to general cell geometries. The scheme can also be written on finite-volume form, introducing explicit cell-vertex fluxes that connect each cell to its vertex. This method is not yet implemented in MRST.

Computer exercises

6.6.1 The two tutorials `mimeticExample2` and `mimeticExample3` compare the TPFA method implemented in the `incomp` module with its mimetic counterpart. Do the methods produce the same results if you perturb the grid cells using `twister` or specify an anisotropic permeability field? What happens if you use a different inner product?

6.6.2 Redo the comparison of structured and unstructured grids in Section 5.4.3 using a mimetic solver and possibly the MPFA-O scheme. Try to also compare computational efficiency with the TPFA scheme in terms of number of unknowns, number of nonzero elements in the matrix, and condition numbers.

6.6.3 A standard method to construct analytical solutions to the Laplace equation in 2D is to write the unknown as the real or imaginary part of an complex analytic function

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press

$f(z)$. We choose the function $f(z) = (z + \frac{1}{2}z^2)$ and set our analytical solution to be $p_a(x, y) = \mathrm{Im}(f(x + iy)) = y + xy$. By prescribing $p_a(x, y)$ along the boundary, we obtain a Dirichlet problem with known solution. Use this technique to study the convergence of the TPFA method and at least two different consistent methods on a sequence of rough non-Cartesian grids of increasing resolution. Try to compare grids with cells that are always non-Cartesian at all resolutions with grids in which the cells are non-Cartesian at coarse resolutions but gradually approach a Cartesian shape as the grid is refined.

https://doi.org/10.1017/9781108591416.009 Published online by Cambridge University Press