2

Nonlinear Finite-Volume Methods for the Flow Equation in Porous Media

MOHAMMED AL KOBAISI AND WENJUAN ZHANG

Abstract

This chapter explains how one can formulate nonlinear finite-volume (NFV) methods, as advanced discretization schemes, to solve the flow equation in porous media. These schemes are of particular interest because apart from being consistent, they are monotone by design. We explain the basic ideas of the NFV methods: how to construct one-sided fluxes, interpolate using harmonic averaging points, and obtain unique discrete fluxes through grid faces with convex combinations of one-sided fluxes. We outline key functions in the accompanied `nfvm` module in the MATLAB Reservoir Simulation Toolbox (MRST) and show some examples of how the method is applied.

2.1 Introduction

Two-point flux approximation (TPFA) is the default discretization scheme in almost all simulators, and this has been the case since the early days of reservoir simulation. Its popularity stems from its simplicity and properties of being locally conservative and honoring the discrete maximum principle (solution must be between the minimum and maximum boundary conditions [6]). In the context of reservoir simulation, the pressure field (solution) must be positive and bounded between the minimum and maximum boundary conditions (e.g., wells). For tensor permeability fields and non-K-orthogonal grids, TPFA fails to address the tensorial flow behavior and the errors incurred in the flow field can be quite significant. An excellent discussion on this can be found in Chapter 6 and subsection 10.4.2 in the MATLAB Reservoir Simulation Toolbox (MRST) textbook [14]. Linear multipoint flux approximation (MPFA) schemes [1, 2, 5, 7, 8, 18, 24, 25] can alleviate this problem to a certain degree, but they are "conditionally monotone" depending on the anisotropy contrast and severity of grid distortion [17]. Moreover, these linear schemes tend to increase cell stencils. To improve the quality of the discrete flux

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

approximations with respect to monotonicity and the discrete maximum principle, nonlinear discretization schemes have been proposed recently [11, 13, 15, 16, 26]. These schemes aim to preserve monotonicity and positivity of the discrete solutions using a number of methods, most notably positive coefficient interpolation strategies, MPFA-like or flux continuity interpolation, and inverse-distance interpolation.

In the following sections, we succinctly describe the mathematical problem and delve right into the discretization scheme. We also present several examples, which can be readily replicated using the `nfvm` module implemented in MRST. The examples demonstrate the efficacy of nonlinear finite volume (NFV) methods against spurious oscillations that can potentially arise in the solutions of other consistent discretization schemes.

2.2 Model Equations

To illustrate the ideas of NFV methods, we consider the following diffusion equation on an open bounded polygonal domain $\Omega \subseteq \mathcal{R}^d, d = 2$ or 3:

$$\begin{aligned} -\nabla \cdot (\mathbf{K}(\mathbf{x})\nabla p(\mathbf{x})) &= q(\mathbf{x}), \quad && \mathbf{x} \in \Omega, \\ p(\mathbf{x}) &= g_{\mathrm{D}}(\mathbf{x}), && \mathbf{x} \in \Gamma_{\mathrm{D}}. \\ -\mathbf{K}(\mathbf{x})\nabla p(\mathbf{x}) \cdot \widehat{\mathbf{n}} &= g_{\mathrm{N}}(\mathbf{x}), && \mathbf{x} \in \Gamma_{\mathrm{N}}. \end{aligned} \tag{2.1}$$

Here, p is the unknown dependent variable, called fluid pressure; the absolute permeability $\mathbf{K}$ of the porous medium is assumed to be symmetric and positive definite; and q is the source (if positive) or sink term (if negative). Dirichlet boundary conditions g_{D} and Neumann boundary conditions g_{N} are applied on the boundaries Γ_{D} and Γ_{N}, respectively. The boundary of the domain Ω is $\partial\Omega = \Gamma_{\mathrm{N}} \cup \Gamma_{\mathrm{D}}$. The unit normal vector pointing outward to the boundary is denoted by $\widehat{\mathbf{n}}$.

2.3 Nonlinear Finite-Volume Methods

Like many control volume–based discretization methods, the NFV methods begin by partitioning the computational domain Ω into a conformal mesh consisting of strongly connected nonoverlapping cells. The following notations are first defined for the mesh:

Ω_h : the set of all cells $\{\Omega_i\}_{i=1}^{n_c}$ in the mesh, where n_c is the total number of cells;
Ω_i : the ith cell in the mesh;
$\mathbf{x}_i$: the centroid of cell Ω_i;
$\mathcal{F}$: the set of all faces in the mesh;
σ : a generic face from $\mathcal{F}$;
$\mathbf{x}_\sigma$: auxiliary point associated with face σ;
$\mathcal{F}_i$: the subset of faces associated with cell Ω_i; i.e., $\partial\Omega_i = \cup_{\sigma\in\mathcal{F}_i}\sigma, \forall\Omega_i \in \Omega_h$.

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

Integrating (2.1) over cell Ω_i and applying the divergence theorem leads to

$$\int_{\Omega_i} -\nabla \cdot (\mathbf{K}\nabla p)\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega_i} -\mathbf{K}\nabla p \cdot \widehat{\mathbf{n}}\, dS = \sum_{\sigma\in\mathcal{F}_i} \int_{\sigma} -\mathbf{K}\nabla p \cdot \widehat{\mathbf{n}}\, dS = \int_{\Omega_i} q\,\mathrm{d}\mathbf{x}. \tag{2.2}$$

This equation can be seen as a mass-balance equation for cell Ω_i. The key element of finite-volume methods is to construct the discrete numerical flux v_σ through the face σ using pressure at the centroids of some cells:

$$\int_{\sigma} -\mathbf{K}\nabla p \cdot \widehat{\mathbf{n}}\,\mathrm{d}S \approx v_\sigma = \sum_{k\in\varphi_\sigma} t_k p_k. \tag{2.3}$$

Here, φ_σ is the index set of cells used for the approximation of flux v_σ, the transmissibility term t_k is associated with cell Ω_k, and p_k is the numerical approximation of pressure at the centroid of cell Ω_k. For linear finite-volume methods, the transmissibility terms t_k are constants for a given mesh and absolute permeability distribution.

The discrete flux expression given by (2.3) can be approximated linearly using the pressures of two adjacent grid blocks sharing an interface or by using additional pressures from neighboring grid blocks. The former leads to the standard TPFA method, whereas the latter leads to the so-called MPFA formalism. Variants of MPFA primarily depend on the location of the continuity points and the number of neighboring cells involved in the flux approximation. For additional details on flux linearization schemes in MPFA formulations, we refer the reader to [1].

Substituting (2.3) into (2.2) and assembling the resulting equation for all cells in the mesh leads to a system of linear equations that can be solved for pressure at cell centroids. For NFV methods, however, the transmissibility terms t_k depend on the primary unknowns, leading to a system of nonlinear equations. To derive the discrete numerical flux v_σ using NFV methods, two one-sided fluxes will first be constructed, and then a unique flux is obtained by taking a convex combination of the two one-sided fluxes. The details follow in the next subsections.

2.3.1 Construction of One-Sided Fluxes

Consider an internal face σ shared by two cells Ω_i and Ω_j for a 2D grid as depicted in Figure 2.1. Flux through the face seen from cell Ω_i (called one-sided flux) can be approximated as

$$\int_{\sigma} -\mathbf{K}_i\nabla p \cdot \widehat{\mathbf{n}}\,\mathrm{d}S \approx v_\sigma^i = -(\nabla p)_i \cdot \mathbf{K}_i\mathbf{n}_{ij}, \tag{2.4}$$

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

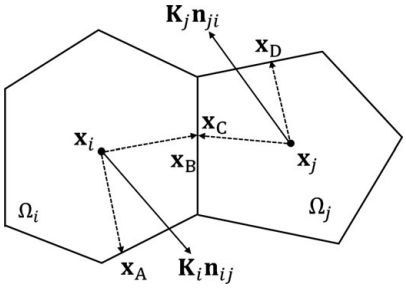

Figure 2.1 Decomposition of conormal vector $\mathbf{K}_i\mathbf{n}_{ij}$ and $\mathbf{K}_j\mathbf{n}_{ji}$ using face interpolating points as auxiliary points.

where $(\nabla p)_i$ denotes the average constant pressure gradient inside Ω_i, $\mathbf{K}_i$ is the piecewise constant permeability tensor of cell Ω_i, and $\mathbf{n}_{ij}$ is the area-weighted normal vector to the face σ pointing from cell Ω_i to cell Ω_j.

The conormal vector $\mathbf{K}_i\mathbf{n}_{ij}$ is then decomposed as

$$\mathbf{K}_i\mathbf{n}_{ij} = \alpha_{i\mathrm{A}}\left(\mathbf{x}_\mathrm{A} - \mathbf{x}_i\right) + \alpha_{i\mathrm{B}}\left(\mathbf{x}_\mathrm{B} - \mathbf{x}_i\right), \tag{2.5}$$

where $\mathbf{x}_\mathrm{A}$ and $\mathbf{x}_\mathrm{B}$ are the position vectors of two auxiliary points, each associated with one face of cell Ω_i. Determining the exact locations of the auxiliary points will be explained in the next subsection. The terms $\alpha_{i\mathrm{A}}$ and $\alpha_{i\mathrm{B}}$ are the corresponding decomposing coefficients and they are required to be nonnegative. Substituting (2.5) into (2.4) and assuming linearly varying pressure inside Ω_i leads to the following expression:

$$v_\sigma^i = \alpha_{i\mathrm{A}}\left(p_i - p_\mathrm{A}\right) + \alpha_{i\mathrm{B}}\left(p_i - p_\mathrm{B}\right), \tag{2.6}$$

where p_A and p_B are pressures at $\mathbf{x}_\mathrm{A}$ and $\mathbf{x}_\mathrm{B}$, respectively. To obtain a pure cell-centered scheme, p_A and p_B are further interpolated using the primary unknowns, which are pressures at some cell centroids:

$$p_\mathrm{A} = \sum_{k\in\mathcal{S}_\mathrm{A}} \omega_{\mathrm{A}k}\, p_k, \qquad p_\mathrm{B} = \sum_{k\in\mathcal{S}_\mathrm{B}} \omega_{\mathrm{B}k}\, p_k, \tag{2.7}$$

where $\mathcal{S}_\mathrm{A}$ and $\mathcal{S}_\mathrm{B}$ are the index sets of control volumes that are involved in the interpolation of p_A and p_B, respectively. The term $\omega_{\mathrm{A}k}$ is the weighting coefficient of pressure at the centroid of cell Ω_k in the interpolation of p_A and $\omega_{\mathrm{B}k}$ has similar meanings. Here, the weighting coefficients are again required to be nonnegative and need to sum up to unity for each auxiliary point. Finally, substituting (2.7) into (2.6) leads to

$$v_\sigma^i = \alpha_{i\mathrm{A}}\left(p_i - \sum_{k\in\mathcal{S}_\mathrm{A}} \omega_{\mathrm{A}k}\, p_k\right) + \alpha_{i\mathrm{B}}\left(p_i - \sum_{k\in\mathcal{S}_\mathrm{B}} \omega_{\mathrm{B}k}\, p_k\right). \tag{2.8}$$

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

Collecting similar terms gives the final form of the one-sided flux:

$$v_\sigma^i = t_i^i p_i - t_j^i p_j - \sum_{k \in \mathcal{S}_i} t_k^i p_k = t_i^i p_i - t_j^i p_j - r_i, \tag{2.9}$$

where t_i^i, t_j^i, and t_k^i are coefficients of p_i, p_j, and p_k, respectively, and $\mathcal{S}_i = \mathcal{S}_\mathrm{A} \cup \mathcal{S}_\mathrm{B} \backslash \{i, j\}$. Similarly, the one-sided flux flowing from Ω_j to Ω_i seen from Ω_j can be expressed as (with $\mathcal{S}_j = \mathcal{S}_\mathrm{C} \cup \mathcal{S}_\mathrm{D} \backslash \{i, j\}$)

$$v_\sigma^j = t_j^j p_j - t_i^j p_i - \sum_{k \in \mathcal{S}_j} t_k^j p_k = t_j^j p_j - t_i^j p_i - r_j. \tag{2.10}$$

One-sided fluxes for 3D grids can be derived analogously. Using the `nfvm` module in MRST, the one-sided transmissibilities are computed as:

```
T = findOStrans(G, rock, hap, 'bc', bc);
```

Here, `G`, `rock`, and `bc` are data structures for grid, rock properties, and boundary conditions, respectively. Note that the boundary condition structure `bc` is slightly different from that used in the core of MRST. Instead of providing a scalar value for each boundary face in `bc.value`, we specify a function handle for each boundary face to accommodate the boundary data g_D and g_N in (2.1). The input argument `hap` stores information of the auxiliary points used in (2.5)–(2.7), which we will discuss in the next subsection.

2.3.2 Harmonic Averaging Point

Construction of one-sided fluxes requires introducing auxiliary points (2.6) and interpolating pressure at these points by pressures at some cell centroids (2.7). The grid vertices are an obvious choice for auxiliary points, and various interpolation methods have been proposed in the literature to interpolate pressure at grid vertices using pressure at cell centroids, such as linear interpolation and inverse distance weighting (IDW) and others; see, e.g., [15, 19, 23]. Different interpolations can have a big impact on the performance of the NFV methods, and it is difficult to design an interpolation method that is both robust and accurate. As noted in [22], it can be more challenging to construct a second-order positivity-preserving interpolating algorithm than the construction of the NFV method itself. Another disadvantage of choosing grid vertices as auxiliary points is that the number of cells involved in the interpolation of pressure at a grid vertex can be quite large. Usually, all of the cells that share the vertex will be involved.

A more attractive option is the so-called harmonic averaging point that was first proposed in [4]. There is one harmonic-averaging point $\mathbf{x}_\sigma$ associated with

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

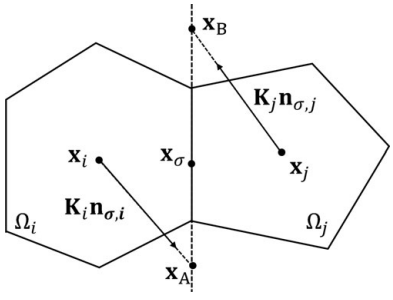

Figure 2.2 Harmonic-averaging point $\mathbf{x}_\sigma$ lies between the two points $\mathbf{x}_A$ and $\mathbf{x}_B$.

each face σ in the mesh. Interpolation of pressure at the harmonic-averaging point involves only the two neighboring cells that share the face and the interpolating coefficients are always nonnegative. Still, consider face σ shared by two cells Ω_i and Ω_j shown in Figure 2.2. First, we find the point $\mathbf{x}_\mathrm{A}$ that lies on the plane containing face σ such that the vector $\mathbf{x}_\mathrm{A} - \mathbf{x}_i$ is parallel to $\mathbf{K}_i\mathbf{n}_{ij}$. Similarly, we can find the point $\mathbf{x}_\mathrm{B}$ such that vector $\mathbf{x}_\mathrm{B} - \mathbf{x}_j$ is parallel to $\mathbf{K}_j\mathbf{n}_{ji}$. Assume that the pressure unknown p is piecewise affine, then flux out of cell Ω_i and Ω_j can be expressed individually as

$$\begin{aligned} v_\sigma^i &= -(\nabla p)_i \cdot \mathbf{K}_i\mathbf{n}_{ij} = w_i\,(p_i - p_\mathrm{A})\,, \\ v_\sigma^j &= -(\nabla p)_j \cdot \mathbf{K}_j\mathbf{n}_{ji} = w_j\,\big(p_j - p_\mathrm{B}\big)\,, \end{aligned} \tag{2.11}$$

where $w_i = \left\|\mathbf{K}_i\mathbf{n}_{ij}\right\| / \left\|\mathbf{x}_\mathrm{A} - \mathbf{x}_i\right\|$ and $w_j = \left\|\mathbf{K}_j\mathbf{n}_{ji}\right\| / \left\|\mathbf{x}_\mathrm{B} - \mathbf{x}_j\right\|$. Furthermore, assume that pressure and the tangential part of the pressure gradient $\mathbf{g}_\sigma$ are continuous on the plane containing face σ. Now, if we take an arbitrary point $\mathbf{x}$ on the plane containing face σ, the pressure at points $\mathbf{x}_\mathrm{A}$ and $\mathbf{x}_\mathrm{B}$ can be written in terms of $\mathbf{g}_\sigma$ and the pressure at $\mathbf{x}$ as

$$p_\mathrm{A} = p(\mathbf{x}) + \mathbf{g}_\sigma \cdot (\mathbf{x}_\mathrm{A} - \mathbf{x})\,, \qquad p_\mathrm{B} = p(\mathbf{x}) + \mathbf{g}_\sigma \cdot (\mathbf{x}_\mathrm{B} - \mathbf{x})\,. \tag{2.12}$$

Substituting (2.12) into (2.11) and imposing flux continuity condition $v_\sigma^i + v_\sigma^j = 0$, we can solve for pressure $p(\mathbf{x})$ as

$$p(\mathbf{x}) = \frac{w_i p_i + w_j p_j - \mathbf{g}_\sigma \cdot \big[w_i\,(\mathbf{x}_\mathrm{A} - \mathbf{x}) + w_j\,(\mathbf{x}_\mathrm{B} - \mathbf{x})\big]}{w_i + w_j}. \tag{2.13}$$

This shows that pressure at any point on the plane containing face σ is a linear convex combination of pressures at centroids of two neighboring cells plus a term accounting for pressure variation along the tangent direction. If we choose a certain point, such that the last term in the numerator vanishes regardless of the tangent gradient $\mathbf{g}_\sigma$, pressure at this point will depend on pressure at the centroids of two

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

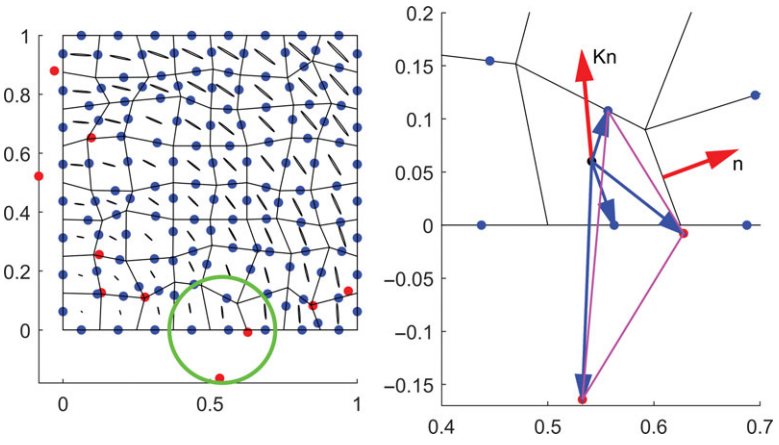

Figure 2.3 Left: Harmonic-averaging points (blue dots are harmonic-averaging points lying inside the face and red dots are points lying outside the face). The elongated ellipses in each cell represent the cell-wise constant permeability tensor. The semi-axes of the ellipses are scaled by the square root of the maximum and minimum principal permeability, respectively. Right: Zooming in on the circled part of the left plot. The red arrows represent the normal vector to the rightmost face and its associated conormal vector. The blue arrows start at the centroid of the cell and end at the harmonic averaging points associated with the faces of the cell. The triangle in magenta represent the convex hull of the involved harmonic averaging points. (After Zhang and Al Kobaisi [26])

neighboring cells only. By equating $w_i\,(\mathbf{x}_\mathrm{A}-\mathbf{x})+w_j\,(\mathbf{x}_\mathrm{B}-\mathbf{x})$ to zero, we can solve for this particular point $\mathbf{x}$ denoted as $\mathbf{x}_\sigma$:

$$\mathbf{x}_\sigma = \frac{w_i\mathbf{x}_\mathrm{A} + w_j\mathbf{x}_\mathrm{B}}{w_i + w_j}, \tag{2.14}$$

which is the location of the harmonic-averaging point associated with face σ. Pressure at this point can then be interpolated by

$$p\,(\mathbf{x}_\sigma) = \frac{w_i p_i + w_j p_j}{w_i + w_j}. \tag{2.15}$$

Unfortunately, the harmonic-averaging point is not without its drawback. For heterogeneous and anisotropic permeability tensors on nonorthogonal grids, some harmonic-averaging points may lie far outside their associated grid faces. As a result, decomposition of conormal vectors with nonnegative coefficients can easily run into difficulty. Figure 2.3 shows an example taken from [26]. The left plot of the figure shows a distorted quadrilateral mesh populated by a heterogeneous rotating permeability tensor field. The permeability tensor of each cell in the mesh is represented by an ellipse, whose semi-axes are scaled by the square root of

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

the maximum and minimum principal permeability, respectively. The permeability anisotropy ratio is a constant of 200. Because of the strong anisotropy, the ellipses look very elongated in the plot. We can see that whereas most harmonic-averaging points lie inside the associated faces (indicated by the blue dots), a small fraction of the harmonic-averaging points (indicated by the red dots in the figure) lie far away from their respective associated faces.

The right plot of Figure 2.3 shows the zoomed-in area delineated by the green circle on the left plot of the figure. The four blue arrows denote the vectors starting at the centroid of the cell and ending at the four harmonic-averaging points associated with the four faces of the quadrilateral cell. Conormal vector **Kn** associated with the rightmost face is denoted by the red arrow. It can be readily seen that decomposing **Kn** with positive coefficients using harmonic-averaging points only is impossible, because the centroid lies outside of the convex hull (represented by the triangle in magenta) of the four harmonic averaging points associated to this cell.

To circumvent this difficulty in decomposing the conormal vectors, a robust and efficient correction algorithm is given in [26] to modify the location of those "ill-placed" harmonic averaging points so that the conormal vectors can be successfully decomposed with nonnegative coefficients. The main idea of the correction algorithm is to "pull back" the ill-placed harmonic averaging points toward the face centroids, while minimizing the errors incurred in the process. Specifically, for any cell whose centroid lies outside the convex hull of the harmonic averaging points associated with its faces, the harmonic averaging point that deviates the most from the face centroid is identified and a different interpolating point is chosen to replace the original harmonic-averaging point. The location of the new interpolating point is restricted to lie within a prescribed distance from the face centroid. Interpolation of pressure at this point is still given by (2.15). Because for any $\mathbf{x} \neq \mathbf{x}_\sigma$ lying on the hyperplane containing face σ pressure at point $\mathbf{x}$ is given by (2.13), the new location is chosen such that the term $w_i(\mathbf{x}_\mathrm{A} - \mathbf{x}) + w_j(\mathbf{x}_\mathrm{B} - \mathbf{x})$ is minimized.

Without digression from the main framework of NFV methods, in what follows we shall assume that the ill-placed harmonic averaging points have been corrected and interested readers are referred to [26] for details on the correction algorithm. Figures 2.4 and 2.5 show two pictorial examples in which `correctHAP(G,hap)`, found in the `nfvm` module, has been used to correct the harmonic averaging points for a triangular cell in 2D and a tetrahedral cell in 3D, respectively. The code for constructing and correcting the harmonic averaging points is

```
hap=findHAP(G, rock, 'bc', bc); % compute hap's for all faces
hap=correctHAP(G, hap); % apply correction algorithm to the hap's
```

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

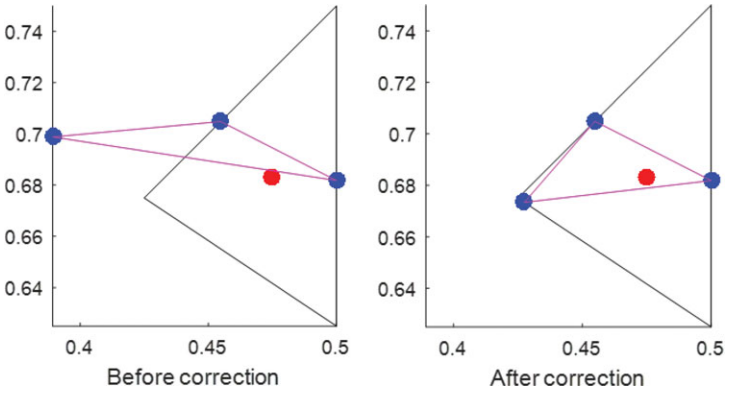

Figure 2.4 Harmonic-averaging points for a triangular cell before and after correction. (After Zhang and Al Kobaisi [26])

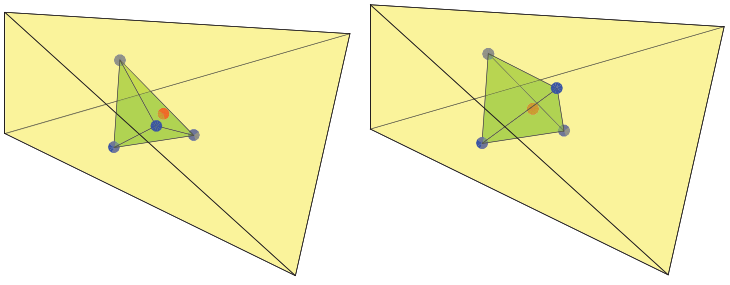

Figure 2.5 Harmonic-averaging points for a tetrahedral cell before and after correction. (After Zhang and Al Kobaisi [26])

The `hap` structure stores the coordinates (`hap.coords`) of each harmonic averaging point, the index of cells (`hap.cells`) involved in the interpolation, and the corresponding weighting coefficients (`hap.weights`). The `correctHAP` function will then modify `hap.coords` whenever necessary.

2.3.3 Nonlinear TPFA

Using the harmonic-averaging points just introduced as auxiliary points, we can construct two one-sided fluxes v_σ^i, v_σ^j for each internal face σ as detailed in Subsection 2.3.1. The unique flux v_σ through face σ flowing from cell Ω_i to Ω_j is then obtained as a convex combination of the two one-sided fluxes:

$$v_\sigma = \mu_i v_\sigma^i + \mu_j\left(-v_\sigma^j\right), \tag{2.16}$$

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

where μ_i, μ_j are the two nonnegative coefficients for the convex combination that satisfy

$$\mu_i + \mu_j = 1. \tag{2.17}$$

Substituting the two one-sided fluxes (2.9) and (2.10) into (2.16) leads to

$$v_\sigma = \left(\mu_i t_i^i + \mu_j t_i^j\right) p_i - \left(\mu_j t_j^j + \mu_i t_j^i\right) p_j - \mu_i r_i + \mu_j r_j. \tag{2.18}$$

To obtain a nonlinear TPFA scheme, the last two terms of (2.18) are forced to be zero:

$$-\mu_i r_i + \mu_j r_j = 0. \tag{2.19}$$

Combining (2.17), (2.19) and solving for μ_i, μ_j gives

$$\mu_i = \frac{r_j}{r_i + r_j}, \quad \mu_j = \frac{r_i}{r_i + r_j}, \tag{2.20}$$

when $r_i + r_j \neq 0$. If $r_i + r_j = 0$, then μ_i and μ_j are taken as 0.5. The final expression for v_σ is then given by the following:

$$v_\sigma = \left(\mu_i t_i^i + \mu_j t_i^j\right) p_i - \left(\mu_j t_j^j + \mu_i t_j^i\right) p_j = T_i p_i - T_j p_j, \tag{2.21}$$

where $T_i = \mu_i t_i^i + \mu_j t_i^j$ and $T_j = \mu_j t_j^j + \mu_i t_j^i$. Note that because r_i and r_j are dependent on pressure values at some cell centroids, T_i and T_j are also dependent on the primary unknowns. Therefore, the flux v_σ is nonlinear. In the case of K-orthogonal grids, the harmonic averaging point for a face coincides with the face centroid. Decomposition of conormal vectors becomes trivial because the conormal vector is parallel to the vector starting from the cell centroid and ending at the face centroid. The terms r_i and r_j will be identically zero and the two parameters μ_i and μ_j become constants. As a result, (2.21) simplifies to the regular linear TPFA method with the transmissibility between two cells approximated by the harmonic average of the cell transmissibilities; i.e., $v_\sigma = T_{Ha}\left(p_i - p_j\right)$.

2.3.4 Nonlinear MPFA

Though the nonlinear TPFA (NTPFA) method is monotone and preserves the nonnegativity of the pressure solution, it does not respect the discrete maximum/minimum principle. To obtain an NFV method that is also extremum preserving, a different convex combination of the one-sided fluxes can be used; see [10, 16, 20, 21]. Following the ideas presented in those works, we can write the one-sided flux expression (2.8) as

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

$$v_\sigma^i = \alpha_{i\mathrm{A}}\Big[\Big(\sum_{k\in\mathcal{S}_\mathrm{A}}\omega_{\mathrm{A}k}\Big)p_i - \sum_{k\in\mathcal{S}_\mathrm{A}}\omega_{\mathrm{A}k}p_k\Big] + \alpha_{i\mathrm{B}}\Big[\Big(\sum_{k\in\mathcal{S}_\mathrm{A}}\omega_{\mathrm{A}k}\Big)p_i - \sum_{k\in\mathcal{S}_\mathrm{B}}\omega_{\mathrm{B}k}p_k\Big]$$
$$= \alpha_{i\mathrm{A}}\sum_{k\in\mathcal{S}_\mathrm{A}}\omega_{\mathrm{A}k}\left(p_i - p_k\right) + \alpha_{i\mathrm{B}}\sum_{k\in\mathcal{S}_\mathrm{B}}\omega_{\mathrm{B}k}\left(p_i - p_k\right). \tag{2.22}$$

Noting that $\sum_{k\in\mathcal{S}_\mathrm{A}}\omega_{\mathrm{A}k} = 1$ and $\sum_{k\in\mathcal{S}_\mathrm{B}}\omega_{\mathrm{B}k} = 1$ and collecting similar terms gives the final form of the one-sided flux as

$$v_\sigma^i = t_j^i\left(p_i - p_j\right) + \sum_{k\in\mathcal{S}_i} t_k^i\left(p_i - p_k\right) = t_j^i\left(p_i - p_j\right) + R_i, \tag{2.23}$$

where t_j^i and t_k^i are coefficients of $\left(p_i - p_j\right)$ and $(p_i - p_k)$, respectively, and $\mathcal{S}_i = \mathcal{S}_\mathrm{A}\cup\mathcal{S}_\mathrm{B}\backslash\{i,j\}$. Note that the coefficients t_j^i, t_k^i here are different from those in (2.9). Similarly, we have

$$v_\sigma^j = t_i^j\left(p_j - p_i\right) + \sum_{k\in\mathcal{S}_j} t_k^j\left(p_j - p_k\right) = t_i^j\left(p_j - p_i\right) + R_j, \tag{2.24}$$

where $\mathcal{S}_j = \mathcal{S}_\mathrm{C}\cup\mathcal{S}_\mathrm{D}\backslash\{i,j\}$. A unique flux v_σ is then obtained by taking a convex combination of the two one-sides fluxes:

$$v_\sigma = \mu_i f_\sigma^i + \mu_j\left(-v_\sigma^j\right) = \left(\mu_i t_j^i + \mu_j t_i^j\right)\left(p_i - p_j\right) + \mu_i R_i - \mu_j R_j. \tag{2.25}$$

Unlike NTPFA, in the nonlinear MPFA (NMPFA) method we choose the two convex combination parameters as

$$\mu_i = \frac{\left|R_j\right|}{|R_i| + \left|R_j\right|}, \quad \mu_j = \frac{|R_i|}{|R_i| + \left|R_j\right|}, \tag{2.26}$$

when $|R_i| + \left|R_j\right| \neq 0$ and $\mu_i = \mu_j = 0.5$ otherwise. With this choice of convex combination parameters, it can be verified that when $R_i R_j \leq 0$, two algebraically equivalent fluxes are obtained:

$$\begin{aligned} v_\sigma &= \left(\mu_i t_j^i + \mu_j t_i^j\right)\left(p_i - p_j\right) + 2\mu_i\sum_{k\in\mathcal{S}_i} t_k^i\left(p_i - p_k\right) \\ -v_\sigma &= \left(\mu_i t_j^i + \mu_j t_i^j\right)\left(p_j - p_i\right) + 2\mu_j\sum_{k\in\mathcal{S}_j} t_k^j\left(p_j - p_k\right). \end{aligned} \tag{2.27}$$

Equation (2.27) can be seen as an MPFA-like formulation because of the one-sided flux expressions given by (2.23) and (2.24); hence the name NMPFA. When $R_i R_j > 0$, the last two terms of (2.25) cancel out and the flux expression becomes

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

$$v_\sigma = \left(\mu_i t_j^i + \mu_j t_i^j\right)\left(p_i - p_j\right). \tag{2.28}$$

Again, as in NTPFA, the NMPFA formulation reduces to the linear TPFA in the case of K-orthogonal grids.

2.3.5 Nonlinear Solver

The system of nonlinear equations resulting from the NTPFA and NMPFA methods can be solved by any nonlinear solver such as the widely used Newton–Raphson method. However, to guarantee the positivity of the pressure solutions during nonlinear iterations, the Picard nonlinear solver is often the method of choice and it is implemented in this module.

Choose a small number $\epsilon_{\text{non}} > 0$ and initial solution vector $\mathbf{p}^{(0)} > 0$
Repeat for $k = 1, 2, \ldots,$
  solve $\mathbf{A}(\mathbf{p}^{(k-1)})\mathbf{p}^{(k)} = \mathbf{b}(\mathbf{p}^{(k-1)})$
until $\left\|\mathbf{A}(\mathbf{p}^{(k)})\mathbf{p}^{(k)} - \mathbf{b}(\mathbf{p}^{(k)})\right\| \leq \epsilon_{\text{non}}\left\|\mathbf{A}(\mathbf{p}^{(0)})\mathbf{p}^{(0)} - \mathbf{b}(\mathbf{p}^{(0)})\right\|$.

In the following numerical examples, we take ϵ_{non} to be 10^{-7}. We also set the maximum number of Picard iterations to be 300 and exit if k exceeds 300.

2.4 Numerical Examples

In this section, we present a few examples to show how one can use the aforementioned NFV formalism, as implemented in the `nfvm` module of MRST, to solve the flow equation in porous media. We note here that the examples are not, by any means, exhaustive but rather are intended to shed some light on the potential for NFV methods to handle highly unstructured grids with strong anisotropy ratios and heterogeneity contrasts. Furthermore, in the following we only show the 2D examples for ease of displaying the results pictorially. 3D cases can be run with the exact same calls of the `nfvm` module in MRST with the arguments being in 3D. Applications in 3D with additional flow physics can be found in [26, 27].

2.4.1 Example 1: Homogeneous Permeability

In this first example, we solve the diffusion equation on a unit square with a hole in the middle, $\Omega = (0,1)^2 \backslash [4/9, 5/9]^2$. The boundary of the computational domain is composed of two disjoint sets: an outer boundary Γ_0 and an inner boundary Γ_1. Pressure on Γ_0 is set to 0 and pressure on Γ_1 is 1. Source term q is zero throughout the domain. The domain is meshed using perturbed quadrilateral grids and

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

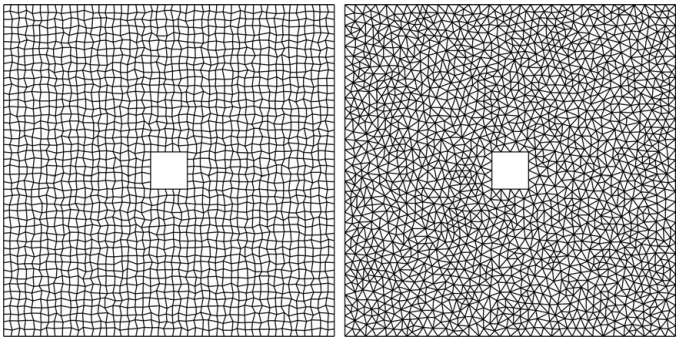

Figure 2.6 Grids used for monotonicity test on a perturbed quadrilateral mesh (left) and an unstructured triangular mesh (right).

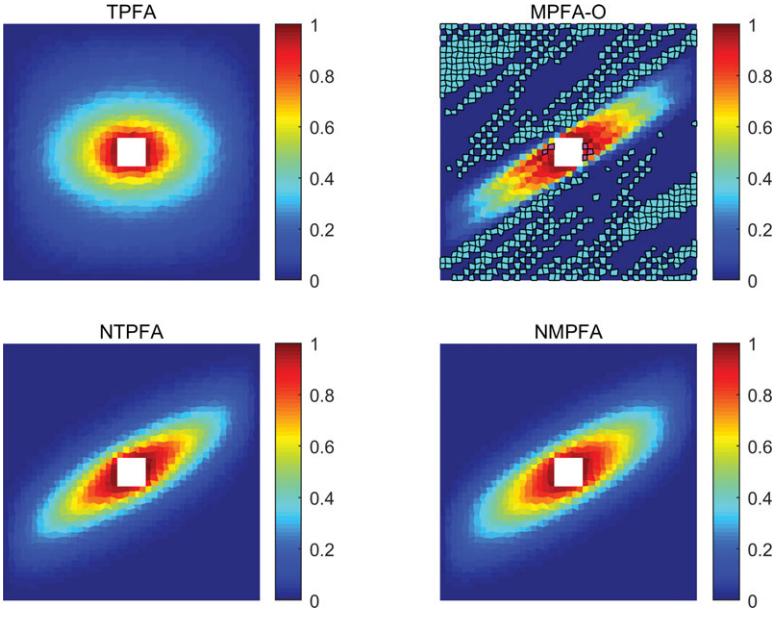

Figure 2.7 Pressure solutions using TPFA, MPFA-O, NTPFA, and NMPFA on a perturbed quadrilateral mesh. Cells with negative pressure values are colored in cyan and cells with pressure values greater than 1 are colored in magenta.

unstructured triangular grids shown in Figure 2.6. Permeability is homogeneous but anisotropic and takes the following form (where θ is 30°):

$$\mathbf{K} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

The discrete maximum principle (DMP) states that the discrete pressure solution should be bounded within the interval [0, 1]. Figure 2.7 depicts pressure

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

Table 2.1 *Computational results of Example 1.*

Method	Quadrilateral grid			Triangular grid		
	$p_{\min}$	$p_{\max}$	n_{iter}	$p_{\min}$	$p_{\max}$	n_{iter}
TPFA	0.0001	0.9686	–	0	0.9877	–
MPFA-O	−0.1443	1.1586	–	−0.9852	1.0253	–
NTPFA	0	0.9851	73	0	0.9864	82
NMPFA	0	0.9775	153	0	0.9830	300

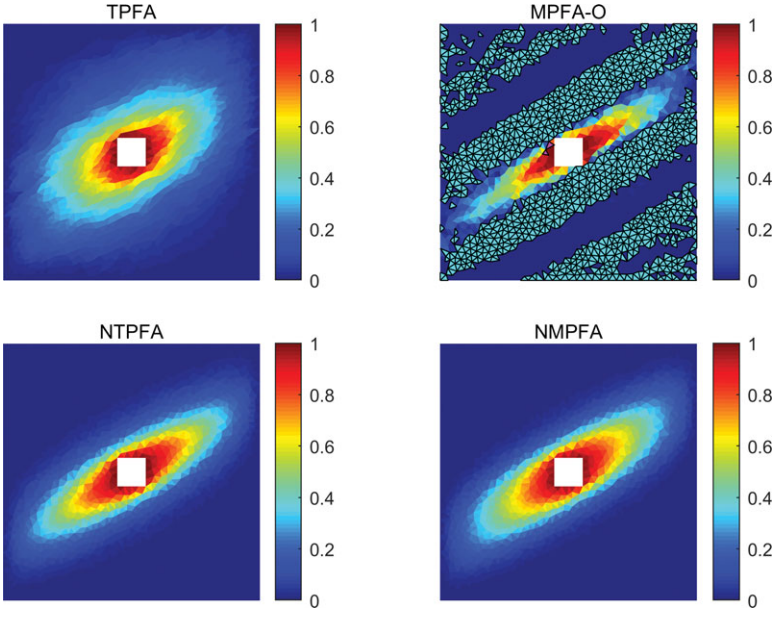

Figure 2.8 Paressure solutions using TPFA, MPFA-O, NTPFA, and NMPFA on an unstructured triangular mesh. Cells with negative pressure values are colored in cyan and cells with pressure values greater than 1 are colored in magenta.

solutions computed using linear TPFA, linear MPFA-O, NTPFA, and NMPFA on the quadrilateral mesh, and the corresponding results on the triangular mesh are shown in Figure 2.8. Table 2.1 lists the minimum and maximum pressure values $p_{\min}, p_{\max}$ for each method and the number of Picard iterations, n_{iter}, for the two nonlinear methods. As expected, the results show that only the two NFV methods respect the discrete minimum and maximum principle and perform consistently on the two different meshes. The MPFA-O method suffers from strong spurious oscillations, especially on the triangular mesh. Note here that the default MPFA-O($\eta = 0$) method is used. Monotonicity of linear MPFA methods on triangular grids is discussed in [12] and it is shown that the MPFA-O(0) method

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

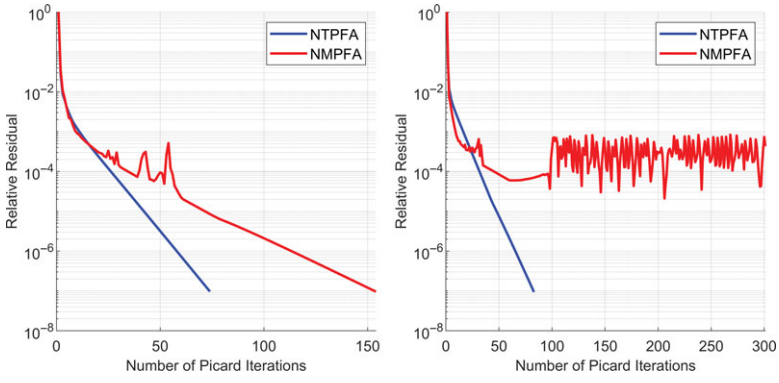

Figure 2.9 Convergence history of the Picard nonlinear solver for the two nonlinear methods on a quadrilateral mesh (left) and a triangular mesh (right).

suffers from strong unphysical oscillations when the triangular grid is not uniform, but its performance can be improved by using the alternative MPFA-O$\left(\eta = \frac{1}{3}\right)$ method [9]. The pressure solution of TPFA is also bounded between [0,1] but the method is obviously inconsistent on the two meshes.

Figure 2.9 reports the convergence history of the Picard nonlinear solver for NTPFA and NMPFA on the two meshes. Whereas NTPFA converges quickly on both meshes, NMPFA converges at a slower rate on the quadrilateral mesh and suffers from convergence issues on the triangular mesh. This may be caused by the strong anisotropy of the permeability tensor. If we take a mild anisotropy ratio of 5 and solve the diffusion equation on the same meshes again, we now see that both methods converge, albeit slower for NMPFA. The pressure solutions using NTPFA and NMPFA for this case are shown in Figure 2.10 and the convergence histories of the Picard nonlinear solver are shown in Figure 2.11. Depending on the particular case, NMPFA can run into local stiff nonlinear issues impeding its convergence. For more details on convergence analyses, see [26].

You can find details of the code in the script `nfvmExample_1.m`. Some key steps are given here. After the grid and rock properties are created in the standard MRST construct format (`G`, `rock`), the boundary condition is then specified:

```
bf = boundaryFaces(G);
xf = G.faces.centroids(bf,1); yf = G.faces.centroids(bf,2);
d  = min([abs(xf) abs(yf) abs(xf-Lx) abs(yf-Ly)],[],2);
index=d<1e-3;
bc.face  = bf;
bc.type  = repmat({'pressure'},[numel(bc.face),1]);
bc.value = cell(length(bf),1);
bc.value(index)  = repmat({@(x)0},[sum(index) 1]);
bc.value(~index) = repmat({@(x)1},[sum(~index),1]);
```

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

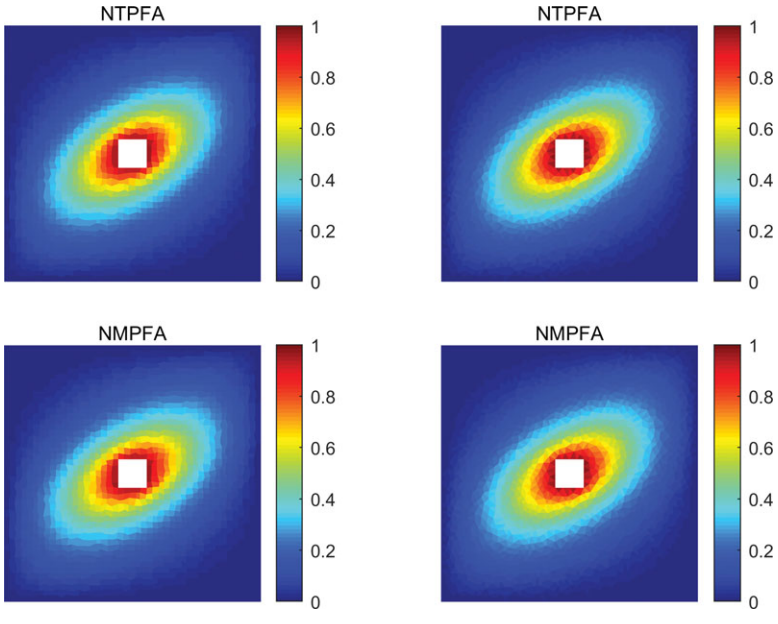

Figure 2.10 Pressure solutions using NTPFA and NMPFA on a quadrilateral mesh (left) and a triangular mesh (right) when the permeability anisotropy ratio is mild.

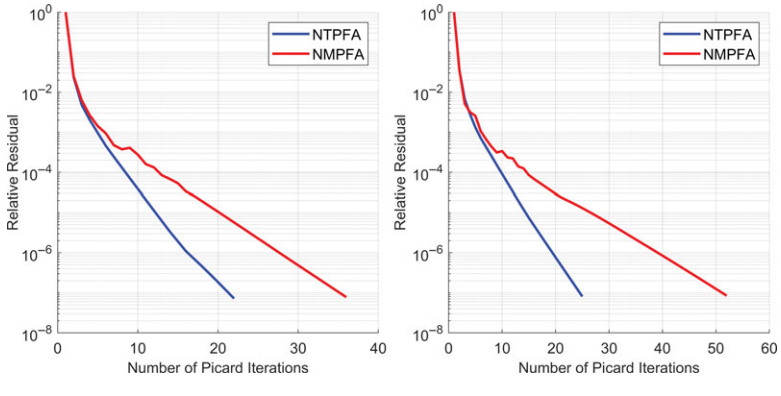

Figure 2.11 Convergence history of the Picard nonlinear solver for the two nonlinear methods on a quadrilateral mesh (left) and a triangular mesh (right) when the permeability anisotropy ratio is mild.

We then compute the one-sided transmissibility arrays and store the results in a cell array T:

```
T = transNFVM(G, rock, 'bc', bc);
```

which is nothing but a simple function that groups the three functions `findHAP`, `correctHAP`, and `findOStrans` together. The system of nonlinear equations is

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

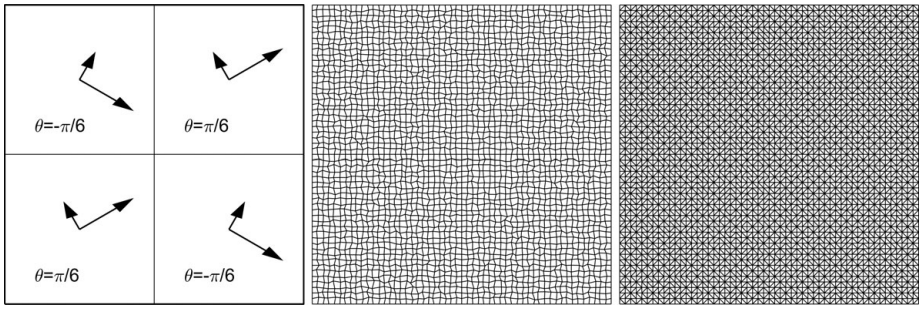

Figure 2.12 Principal directions of the **K** tensor are rotated by different angles in each quadrant (left). Middle/left: Quadrilateral/triangular grid used to mesh the domain.

then solved using the Picard solver. Parameters needed by the Picard solver are collected in a structure called `picard`:

```
picard.u0      = ones(G.cells.num, 1);
picard.tol     = 1e-7;
picard.maxIter = 300;
```

Finally the pressure solution is obtained using any of the nonlinear methods:

```
sntp = incompNTPFA(G, T, picard, 'bc', bc);
snmp = incompNMPFA(G, T, picard, 'bc', bc);
```

The current implementation of the nonlinear flow solvers does not consider the fluid properties because we are mainly concerned with solving (2.1) numerically. Therefore, no `fluid` object is passed to the nonlinear solvers. To solve the same problem using the linear TPFA and MPFA methods implemented in MRST, we can simply set the fluid viscosity to unity.

2.4.2 Example 2: Discontinuous Permeability

The next example considers a domain with significant jumps in the permeabilities. The computational domain is $\Omega = (0,1)^2$ and is composed of four quadrants. The permeability tensor is homogeneous in each quadrant but jumps across mesh edges between different quadrants. The principal directions of the permeability tensors are rotated by different angles θ with respect to the coordinate system as indicated in the left plot of Figure 2.12. The maximum and minimum principal permeabilities are $k_1 = 1\,000$ and $k_2 = 1$ for all quadrants. Distorted quadrilateral and triangular grids that honor the internal permeability discontinuity are used

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

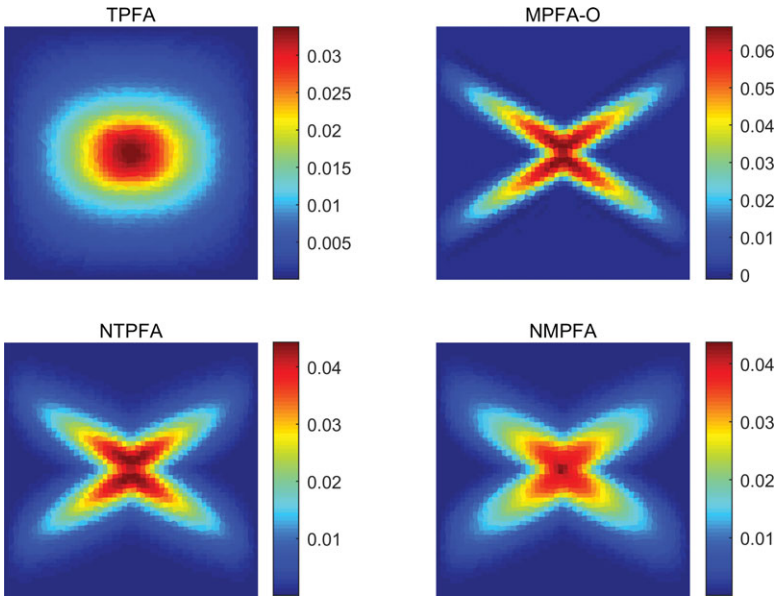

Figure 2.13 Pressure solutions using TPFA, MPFA-O, NTPFA, and NMPFA for Example 2 with a perturbed quadrilateral mesh.

to mesh the domain (see middle and right plots of Figure 2.12). Homogeneous Dirichlet boundary conditions are applied to the boundaries and the source term is given by

$$q(x,y) = \begin{cases} 1000, & (x,y) \in [7/18, 11/18]^2, \\ 0, & \text{otherwise.} \end{cases}$$

Using similar code as in the previous example, we now simulate the problem using TPFA, MPFA-O, NTPFA, and NMPFA. Figure 2.13 shows the pressure solutions computed with these four schemes on quadrilateral meshes, and Figure 2.14 gives the pressure solutions on triangular meshes. Figure 2.15 shows the convergence history of the two nonlinear methods, and we can see that NMPFA again suffers from convergence issues, although the final solution looks physically correct.

If we rerun the simulation with a permeability anisotropy ratio of 10, while keeping all other parameters the same, the four schemes produce results that are virtually indistinguishable. The convergence histories of the Picard nonlinear solver depicted in Figure 2.16 also confirm that NMPFA converges to the specified tolerance with no problem when the permeability anisotropy ratio is not too large, as we also observed in the previous example. Table 2.2 lists the computational results of the monotonicity test for this example. Note that the MPFA-O method performs acceptably on the quadrilateral mesh but performs very poorly on the triangular mesh.

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

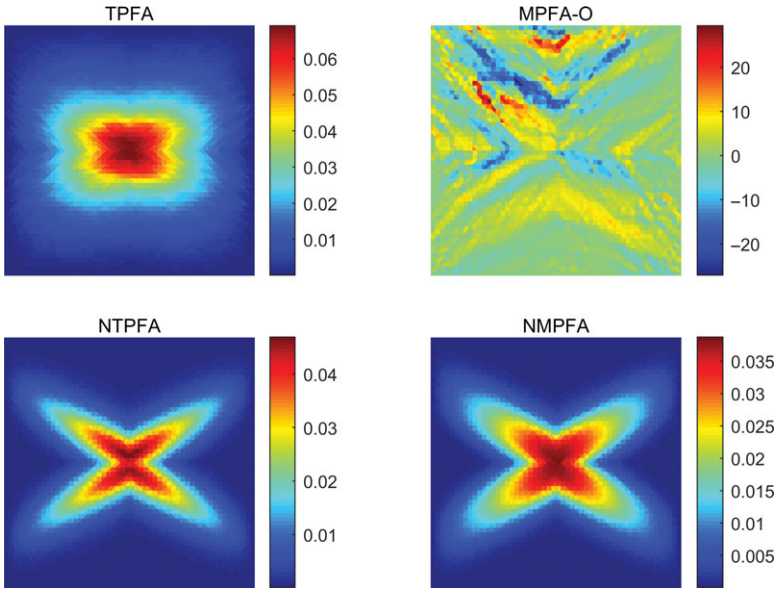

Figure 2.14 Pressure solutions using TPFA, MPFA-O, NTPFA, and NMPFA for Example 2 with a triangular mesh.

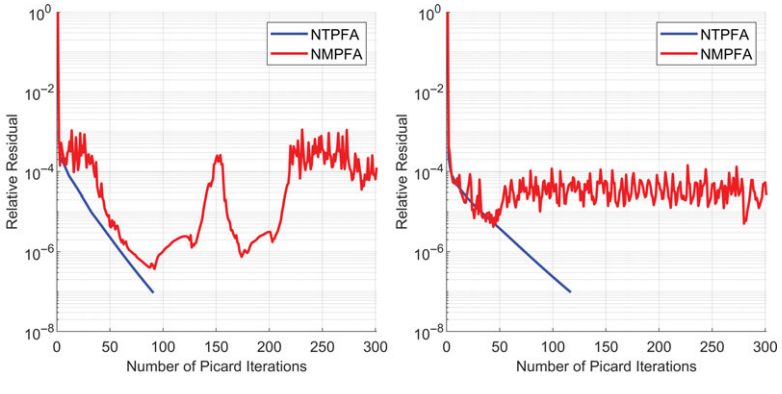

Figure 2.15 Convergence history of the Picard nonlinear solver for the two nonlinear methods on a quadrilateral mesh (left) and a triangular mesh (right) for Example 2.

The two nonlinear methods, on the other hand, resolve the principal directions of the discontinuous permeability tensors quite nicely and the pressure solutions stay nonnegative. Details of the code for this example can be found in the script `nfvmExample_2.m`.

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

Table 2.2 *Computational results of Example 2.*

Method	Quadrilateral grid			Triangular grid		
	$p_{\min}$	$p_{\max}$	n_{iter}	$p_{\min}$	$p_{\max}$	n_{iter}
TPFA	0	0.0338	–	0	0.0690	–
MPFA-O	−0.0013	0.0662	–	−27.1878	29.3887	–
NTPFA	0	0.0443	90	0	0.0469	116
NMPFA	0	0.0437	300	0	0.0387	300

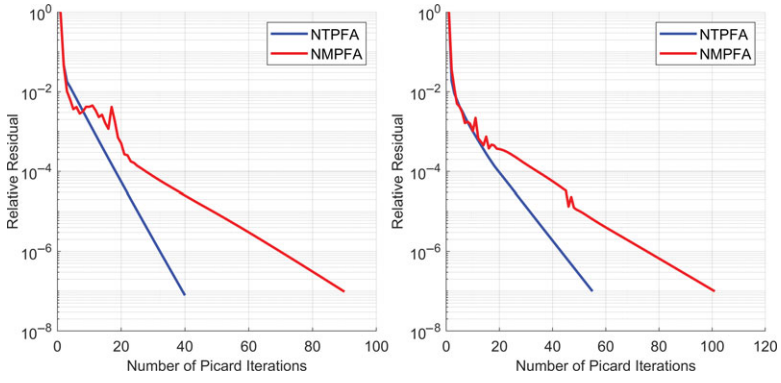

Figure 2.16 Convergence history of the Picard nonlinear solver for the two nonlinear methods on a quadrilateral mesh (left) and a triangular mesh (right) for Example 2: mild anisotropy.

2.4.3 Example 3: No-Flow Boundary Conditions

In this last example, a test problem taken from [3] is used here to further illustrate the monotonicity properties of the NFV methods. The computational domain is the unit square domain $(0,1)^2$ and it is meshed by an 11×11 Cartesian grid. Permeability is given by the following formula:

$$\mathbf{K} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

where $\theta = 67.5^\circ$. No-flow boundary conditions are applied on the exterior boundaries. Pressure of cell (4, 6) is fixed at 0 and pressure in cell (8, 6) is 1, giving rise to Dirichlet boundary conditions for faces bordering the two cells. The pressure solutions using TPFA, MPFA-O, NTPFA, and NMPFA are shown in Figure 2.17 and the convergence history of the two nonlinear methods is shown in Figure 2.18. Astute observers will reach to the following remarks: The MPFA-O result is neither monotone nor DMP adherent; the NTPFA result is monotone but

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

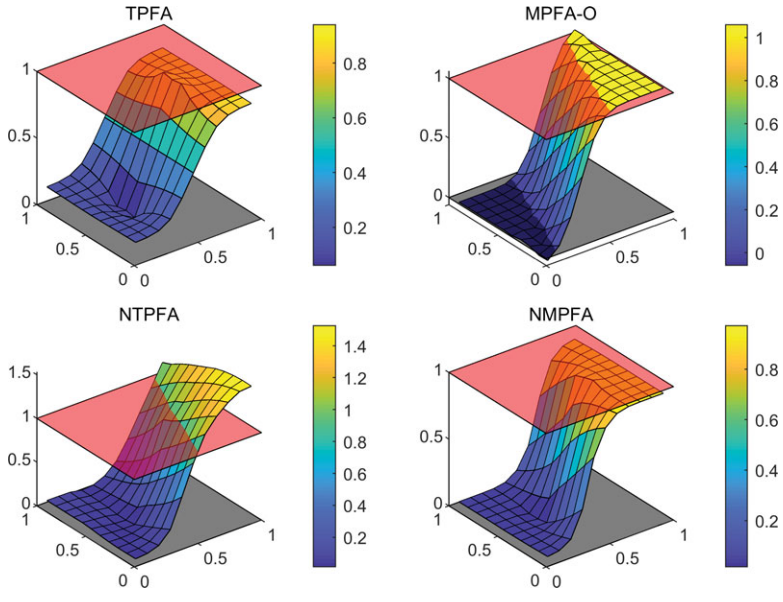

Figure 2.17 Pressure profiles using TPFA, MPFA-O, NTPFA, and NMPFA. The semi-transparent red and black planes mark the physical upper and lower bounds of the pressure solution, respectively.

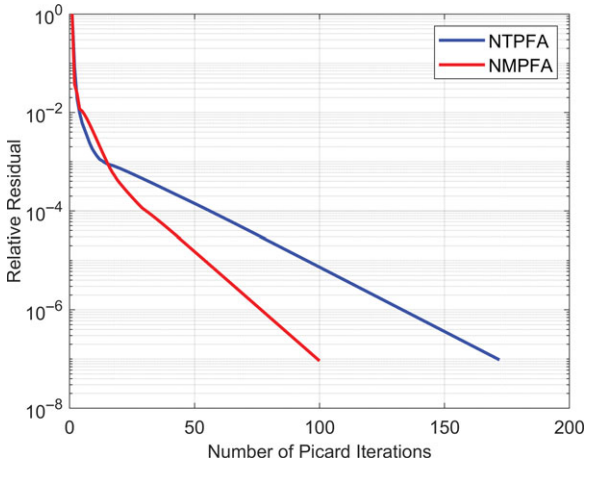

Figure 2.18 Convergence history of the Picard nonlinear solver for the two nonlinear methods on a quadrilateral mesh (left) and a triangular mesh (right) for Example 3.

violates the DMP; the NMPFA result, on the other hand, is both monotone and adheres to the DMP. Moreover, for this particular example, the NMPFA method converges faster than NTPFA. Details of the code for this example can be found in `nfvmExample_3.m`.

https://doi.org/10.1017/9781009019781.006 Published online by Cambridge University Press

2.5 Concluding Remarks

This chapter introduced you to a particular class of advanced discretizations, namely, NFV methods. Unlike their linear finite-volume counterparts, these methods enjoy full monotonicity and/or extremum preservation by design. These traits become of particular interest when dealing with highly unstructured grids and strong anisotropy ratios and heterogeneity contrasts. Because of the inherent nonlinear nature of the schemes, they do require efficient nonlinear solvers that preserve positivity of the discrete solution. Moreover, the size of the resultant discrete system remains equivalent to that of linear finite-volume approaches whenever the comparison is made on a one-to-one basis. However, computational gains are quickly realized when one compares the monotone nonlinear TPFA to a linear MPFA scheme for 3D problems. In the former, the stencil is essentially that of a linear TPFA in 3D, whereas for the latter we would be looking at a 27-point stencil for MPFA-O. Variants of the NFV formulations described herein are certainly possible, but a judicious interpolation method ensuring nonnegative flux decomposition is quite difficult to attain on challenging non-K-orthogonal grids with heterogeneous permeability fields and high anisotropy ratios. In our experience, however, the harmonic averaging point scheme combined with the correction algorithm seems to be the most robust of all.

Acknowledgement. The authors thank Knut-Andreas Lie, August Johansson, and Olav Møyner of the SINTEF Computational Geosciences group for the fruitful discussions.

References

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

[2] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media. *Journal of Computational Physics*, 127(1):2–14, 1996. doi: 10.1006/jcph.1996.0154.

[3] I. Aavatsmark, G. T. Eigestad, B. T. Mallison, and J. M. Nordbotten. A compact multipoint flux approximation method with improved robustness. *Numerical Methods for Partial Differential Equations: An International Journal*, 24(5):1329–1360, 2008. doi: 10.1002/num.20320.

[4] L. Agélas, R. Eymard, and R. Herbin. A nine-point finite volume scheme for the simulation of diffusion in heterogeneous media. *Comptes Rendus Mathematique*, 347(11–12):673–676, 2009. doi: 10.1016/j.crma.2009.03.013.

[5] Q.-Y. Chen, J. Wan, Y. Yang, and R. T. Mifflin. Enriched multi-point flux approximation for general grids. *Journal of Computational Physics*, 227(3):1701–1721, 2008. doi: 10.1016/j.jcp.2007.09.021.

[6] J. Droniou. Finite volume schemes for diffusion equations: introduction to and review of modern methods. *Mathematical Models and Methods in Applied Sciences*, 24(8):1575–1619, 2014. doi: 10.1142/S0218202514400041.

[7] M. G. Edwards and C. F. Rogers. Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Computational Geosciences*, 2(4):259–290, 1998. doi: 10.1023/A:1011510505406.

[8] M. G. Edwards and H. Zheng. A quasi-positive family of continuous Darcy-flux finite-volume schemes with full pressure support. *Journal of Computational Physics*, 227(22):9333–9364, 2008. doi: 10.1016/j.jcp.2008.05.028.

[9] H. A. Friis, M. G. Edwards, and J. Mykkeltveit. Symmetric positive definite flux-continuous full-tensor finite-volume schemes on unstructured cell-centered triangular grids. *SIAM Journal on Scientific Computing*, 31(2):1192–1220, 2009. doi: 10.1137/070692182.

[10] Z. Gao and J. Wu. A small stencil and extremum-preserving scheme for anisotropic diffusion problems on arbitrary 2D and 3D meshes. *Journal of Computational Physics*, 250:308–331, 2013. doi: 10.1016/j.jcp.2013.05.013.

[11] Z. Gao and J. Wu. A second-order positivity-preserving finite volume scheme for diffusion equations on general meshes. *SIAM Journal on Scientific Computing*, 37(1):A420–A438, 2015. doi: 10.1137/140972470.

[12] E. Keilegavlen and I. Aavatsmark. Monotonicity for MPFA methods on triangular grids. *Computational Geosciences*, 15(1):3–16, 2011. doi: 10.1007/s10596-010-9191-5.

[13] C. Le Potier. Schéma volumes finis monotone pour des opérateurs de diffusion fortement anisotropes sur des maillages de triangles non structurés [Finite volume monotone scheme for highly anisotropic diffusion operators on unstructured triangular meshes]. *Comptes Rendus Mathématique*, 341(12):787–792, 2005. doi: 10.1016/j.crma.2005.10.010.

[14] K.-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge University Press, Cambridge, UK, 2019. doi: 10.1017/9781108591416.

[15] K. Lipnikov, M. Shashkov, D. Svyatskiy, and Y. Vassilevski. Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes. *Journal of Computational Physics*, 227(1):492–512, 2007. doi: 10.1016/j.jcp.2007.08.008.

[16] K. Lipnikov, D. Svyatskiy, and Y. Vassilevski. Minimal stencil finite volume scheme with the discrete maximum principle. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 27(4):369–386, 2013. doi: 10.1515/rnam-2012-0020.

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

[18] J. M. Nordbotten and G. T. Eigestad. Discretization on quadrilateral grids with improved monotonicity properties. *Journal of Computational Physics*, 203(2):744–760, 2005. doi: 10.1016/j.jcp.2004.10.002.

[19] L. E. S. Queiroz, M. R. A. Souza, F. R. L. Contreras, P. R. M. Lyra, and D. K. E. de Carvalho. On the accuracy of a nonlinear finite volume method for the solution of diffusion problems using different interpolations strategies. *International Journal for Numerical Methods in Fluids*, 74(4):270–291, 2014. doi: 10.1002/fld.3850.

[20] D. Svyatskiy and K. Lipnikov. Second-order accurate finite volume schemes with the discrete maximum principle for solving Richards' equation on unstructured meshes. *Advances in Water Resources*, 104:114–126, 2017. doi: 10.1016/j.advwatres.2017.03.015.

[21] K. M. Terekhov, B. T. Mallison, and H. A. Tchelepi. Cell-centered nonlinear finite-volume methods for the heterogeneous anisotropic diffusion problem. *Journal of Computational Physics*, 330:245–267, 2017. doi: 10.1016/j.jcp.2016.11.010.

[22] J. Wu and Z. Gao. Interpolation-based second-order monotone finite volume schemes for anisotropic diffusion equations on general grids. *Journal of Computational Physics*, 275:569–588, 2014. doi: 10.1016/j.jcp.2014.07.011.

[23] G. Yuan and Z. Sheng. Monotone finite volume schemes for diffusion equations on polygonal meshes. *Journal of Computational Physics*, 227(12):6288–6312, 2008. doi: 10.1016/j.jcp.2008.03.007.

[24] W. Zhang and M. Al Kobaisi. A globally coupled pressure method for the discretization of the tensor-pressure equation on non-K-orthogonal grids. *SPE Journal*, 22(2):679–698, 2017. doi: 10.2118/184405-PA.

[25] W. Zhang and M. Al Kobaisi. A simplified enhanced MPFA formulation for the elliptic equation on general grids. *Computational Geosciences*, 21(4):621–643, 2017. doi: 10.1007/s10596-017-9638-z.

[26] W. Zhang and M. Al Kobaisi. Cell-centered nonlinear finite-volume methods with improved robustness. *SPE Journal*, 25(1):288–309, 2020. doi: 10.2118/195694-PA.

[27] W. Zhang and M. Al Kobaisi. Nonlinear finite volume method for 3D discrete fracture–matrix simulations. *SPE Journal*, 25(4):2079–2097, 2020. doi: 10.2118/201098-PA.