# Benchmark on Anisotropic Problems

## Numerical investigation of a mimetic finite difference method

**Bernd Flemisch, Rainer Helmig**

*Universität Stuttgart*
*Institute of Hydraulic Engineering*
*Department of Hydromechanics and Modeling of Hydrosystems*
*Pfaffenwaldring 61, 70191 Stuttgart, Germany*
*bernd@iws.uni-stuttgart.de*

ABSTRACT. *This benchmark study investigates the behavior of a mimetic finite difference method. It solves the majority of the proposed problems with convincing accuracy and robustness. It appears to be most promising for tackling real applications, which is also due to the ease of implementation.*

KEYWORDS: *anisotropy benchmark, mimetic finite difference method*

### 1. Presentation of the scheme

Mimetic finite difference methods are discretization methods for partial differential equations which promise to be robust for general unstructured polygonal and polyhedral meshes, including adaptive local refinement, non-matching interfaces, and degenerate or non-convex elements. Moreover, in view of the considered diffusion problem, they qualify for dealing with strongly heterogeneous full tensorial permeabilities. Evolving from standard finite differences, the development of the methodology started originally in [FAV 81], where the name "support operator method" has been used. A good introduction is provided by [SHA 96], a framework for the mathematical analysis in terms of mixed formulations is developed in [BRE 05a], and a quite rich annotated bibliography can be found at [ROB ]. For this benchmark study, we aim to investigate the behavior of the mimetic finite difference method developed and presented in [BRE 05b]. Since we did not develop the method ourselves, we only provide a brief and rough overview in the following. For more details, we would like to refer to the original work [BRE 05b] and the references therein.

The starting point is the continuous problem formulation of seeking a scalar pressure function $u$ and a velocity vector field $\boldsymbol{v}$ such that

$$\operatorname{div} \boldsymbol{v} = f, \quad \boldsymbol{v} = -\mathbb{K} \operatorname{grad} u$$

inside a domain $\Omega$, completed by appropriate boundary conditions. After obtaining a triangulation of $\Omega$, the space $Q_h$ of element-wise constants is selected for the discretization of the pressures. The velocities are approximated in a space $X_h$ by associating for every element $E$ a normal velocity $v_E^f$ with each face $f$ of $E$, subject to the continuity condition $v_{E_1}^f = -v_{E_2}^f$ for two neighboring elements $E_1$ and $E_2$. On $Q_h$, the usual $L^2$ scalar product is given and denoted by $(\cdot,\cdot)_{Q_h}$. The velocity space $X_h$ is also assumed to be equipped with a scalar product $(\cdot,\cdot)_{X_h}$, defined element-wise by $(\boldsymbol{v},\boldsymbol{w})_E = \boldsymbol{v}_E^{\mathrm{T}}\mathbb{M}_E\boldsymbol{w}_E$, where $\boldsymbol{v}_E$ denotes the vector with entries $v_E^{f_1},\ldots,v_E^{f_{k_E}}$, and $\mathbb{M}_E$ is a symmetric positive definite $k_E\times k_E$ matrix. The fundamental idea is now to discretize the first order differential operators div and $-\mathbb{K}\,\mathrm{grad}$ to $\mathrm{div}_h : X_h \to Q_h$ and $\mathrm{G}_h : Q_h \to X_h$, respectively, in such a way that the discrete operators are adjoint to each other, i.e.,

$$(\boldsymbol{v},\mathrm{G}_h u)_{X_h} = (\mathrm{div}_h\boldsymbol{v},u)_{Q_h}, \quad \boldsymbol{v}\in X_h,\ u\in Q_h. \qquad [1]$$

Choosing $\mathrm{div}_h$ in the canonical way, i.e., $(\mathrm{div}_h\boldsymbol{v})_E = |E|^{-1}\sum v_E^f|f|$, the crucial step is to select a suitable scalar product $(\cdot,\cdot)_{X_h}$. The missing operator $\mathrm{G}_h$ is then uniquely determined by [1]. Finally, the discrete problem is to find $(u_h,\boldsymbol{v}_h)\in Q_h\times X_h$ such that

$$\mathrm{div}_h\boldsymbol{v}_h = f_h, \quad \boldsymbol{v}_h = \mathrm{G}_h u_h. \qquad [2]$$

## 2. Numerical results

The main contribution of [BRE 05b] is to present a whole family of suitable scalar products guaranteeing the stability and convergence of the mimetic discretization scheme. Further narrowing down this choice by considering computational efficiency, we are still able and responsible to choose an element-wise constant $\tilde{u}_E$. A very robust choice turned out to be $\tilde{u}_E = c|E|^{-1}\mathrm{trace}\,\mathbb{K}_E$ with $c\in[2,80]$, see page 1547 in [BRE 05b]. For all tests except test 8, we simply chose $c=2$. The exact algorithm for calculating the inverse $\mathbb{W}_E = \mathbb{M}_E^{-1}$ is given by Algorithm 1 in [BRE 05b].

By usual hybridization known from mixed finite elements, system [2] can be transformed to a sparse system for unknown pressure traces at the element faces with a symmetric positive definite system matrix. After solving this system, the original variables $u_h$ and $\boldsymbol{v}_h$ can be efficiently calculated by a local postprocess. Thus, the number of degrees of freedom `nunkw` is given by the total number of element faces. For the evaluation points $x_E$ in the formulas for `sumf` and `erl2`, the barycenter of $E$ is selected. Moreover, the discrete numerical flux needed for `flux0`, ..., `erflm`, is simply given by $v_E^f|f|$ on the face $f$. The integrals involving the exact solution appearing in `erflx0`, ..., `erflm`, have been computed by using lowest order Gauß quadrature.

Although originally not available, discrete element-wise gradients can be obtained by a postprocess from the fluxes $v_E^f|f|$. For quadrilaterals, we first define a reference velocity $\widehat{\boldsymbol{v}}$ on the unit square $\widehat{E}$ by setting

$$\widehat{v}_x = 0.5(v_E^{f_\mathrm{r}}|f_\mathrm{r}| - v_E^{f_\mathrm{l}}|f_\mathrm{l}|), \quad \widehat{v}_y = 0.5(v_E^{f_\mathrm{t}}|f_\mathrm{t}| - v_E^{f_\mathrm{b}}|f_\mathrm{b}|),$$

where $f_\mathrm{r}$, $f_\mathrm{l}$, $f_\mathrm{t}$, and $f_\mathrm{b}$ indicate the right, left, top, and bottom face of $\widehat{E}$. Using the Piola transformation yields an element velocity $\boldsymbol{v}_E = (\det \mathbb{J})^{-1} \mathbb{J} \widehat{\boldsymbol{v}}$, with the Jacobian $\mathbb{J}$ of the usual element mapping $\widehat{E} \to E$, [HÆG 07]. For triangles, the element velocity $\boldsymbol{v}_E$ is defined as being the interpolant in terms of Raviart–Thomas finite elements. Having calculated $\boldsymbol{v}_E$, the discrete gradient $\mathrm{grad}_h u_h$ is given by $(\mathrm{grad}_h u_h)_E = -\mathbb{K}_E^{-1} \boldsymbol{v}_E$. The relative $L^2$ norm of the error in the gradient `ergrad` is chosen analogously to `erl2`.

The implementation has been performed within the multi-scale multi-physics toolbox DuMu$^\mathrm{x}$, [FLE 07], which is based on the recently released DUNE framework, [BAS 07]. The arising systems of linear equations have been solved by employing the sparse direct solver PARDISO, [SCH 04]. We note that mimetic finite difference methods are very well suited for non-conforming grids. We can simply consider each hanging node of an element as being a vertex of this element, admitting an angle of 180° which does not pose any theoretical or numerical difficulties for the approach. Nevertheless, we did not exploit this fact until now in our implementation, and therefore had to skip the tests on the corresponding meshes.

• **Test 1.1 Mild anisotropy,** $u(x,y) = 16x(1-x)y(1-y)$**,** $\min = 0$, $\max = 1$**, regular triangular mesh,** `mesh1`

| i | nunkw | nnmat | sumflux | erl2 | ergrad | ratiol2 | ratiograd |
|---|---|---|---|---|---|---|---|
| 1 | 9.20E+01 | 4.28E+02 | 1.78E-14 | 2.03E-02 | 1.72E-01 | – | – |
| 2 | 3.52E+02 | 1.70E+03 | -3.91E-14 | 4.61E-03 | 8.55E-02 | 2.21E+00 | 1.04E+00 |
| 3 | 1.38E+03 | 6.75E+03 | 3.91E-14 | 1.11E-03 | 4.26E-02 | 2.08E+00 | 1.02E+00 |
| 4 | 5.44E+03 | 2.69E+04 | 3.84E-13 | 2.75E-04 | 2.13E-02 | 2.03E+00 | 1.01E+00 |
| 5 | 2.16E+04 | 1.08E+05 | 2.31E-12 | 6.85E-05 | 1.06E-02 | 2.01E+00 | 1.01E+00 |
| 6 | 8.63E+04 | 4.30E+05 | -9.96E-12 | 1.71E-05 | 5.32E-03 | 2.00E+00 | 1.00E+00 |
| 7 | 3.45E+05 | 1.72E+06 | -3.18E-11 | 4.28E-06 | 2.66E-03 | 2.00E+00 | 1.00E+00 |

**ocvl2** = 2.01E+00, **ocvgradl2** = 1.00E+00.

| i | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| 1 | 2.00E-02 | 2.00E-02 | 2.00E-02 | 2.00E-02 | 2.87E-01 | 7.33E-02 | 9.21E-01 |
| 2 | 5.12E-03 | 5.12E-03 | 5.12E-03 | 5.12E-03 | 1.69E-01 | 1.92E-02 | 9.81E-01 |
| 3 | 1.29E-03 | 1.29E-03 | 1.29E-03 | 1.29E-03 | 9.17E-02 | 4.90E-03 | 9.95E-01 |
| 4 | 3.22E-04 | 3.22E-04 | 3.22E-04 | 3.22E-04 | 4.77E-02 | 1.23E-03 | 9.99E-01 |
| 5 | 8.06E-05 | 8.06E-05 | 8.06E-05 | 8.06E-05 | 2.43E-02 | 3.10E-04 | 1.00E-00 |
| 6 | 2.02E-05 | 2.02E-05 | 2.02E-05 | 2.02E-05 | 1.23E-02 | 7.75E-05 | 1.00E-00 |
| 7 | 1.03E-05 | 1.03E-05 | 1.03E-05 | 1.03E-05 | 6.17E-03 | 1.94E-05 | 1.00E-00 |

• **Test 1.1 Mild anisotropy,** $u(x,y) = 16x(1-x)y(1-y)$**,** $\min = 0$, $\max = 1$**, coarse (C) and fine (F) distorted quadrangular meshes,** `mesh4_1` **and** `mesh4_2`

| grid | nunkw | nnmat | sumflux | erl2 | ergrad |
|---|---|---|---|---|---|
| C | 612 | 4080 | 9.24E-14 | 3.84E-02 | 4.02E-02 |
| F | 2244 | 15312 | 1.10E-13 | 1.08E-02 | 1.11E-02 |

| grid | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| C | 1.39E-03 | 2.75E-04 | 2.24E-04 | 1.23E-03 | 1.85E-01 | 9.73E-03 | 9.45E-01 |
| F | 1.40E-04 | 1.85E-04 | 8.71E-05 | 6.66E-05 | 5.11E-02 | 2.90E-03 | 9.83E-01 |

• **Test 1.2 Mild anisotropy,** $u(x,y) = \sin((1-x)(1-y)) + (1-x)^3(1-y)^2$**,** $\min = 0,\ \max = 1 + \sin 1$**, regular triangular mesh,** `mesh1`

| i | nunkw | nnmat | sumflux | erl2 | ergrad | ratiol2 | ratiograd |
|---|---|---|---|---|---|---|---|
| 1 | 9.20E+01 | 4.28E+02 | -2.22E-15 | 6.63E-03 | 1.26E-01 | – | – |
| 2 | 3.52E+02 | 1.70E+03 | -9.77E-15 | 1.64E-03 | 6.17E-02 | 2.08E+00 | 1.06E+00 |
| 3 | 1.38E+03 | 6.75E+03 | 2.49E-14 | 4.10E-04 | 3.07E-02 | 2.03E+00 | 1.02E+00 |
| 4 | 5.44E+03 | 2.69E+04 | 2.04E-13 | 1.03E-04 | 1.53E-02 | 2.01E+00 | 1.02E+00 |
| 5 | 2.16E+04 | 1.08E+05 | 1.43E-12 | 2.56E-05 | 7.66E-03 | 2.02E+00 | 1.00E+00 |
| 6 | 8.63E+04 | 4.30E+05 | 5.04E-13 | 6.41E-06 | 3.83E-03 | 2.00E+00 | 1.00E+00 |
| 7 | 3.45E+05 | 1.72E+06 | 2.18E-12 | 1.61E-06 | 1.91E-03 | 2.00E+00 | 1.00E+00 |

**ocvl2** = 2.01E+00, **ocvgradl2** = 1.01E+00.

| i | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| 1 | 1.96E-03 | 4.28E-04 | 1.20E-02 | 8.42E-03 | 1.23E-01 | 4.92E-03 | 1.37E+00 |
| 2 | 4.36E-04 | 2.73E-04 | 3.17E-03 | 2.64E-03 | 8.26E-02 | 1.24E-03 | 1.59E+00 |
| 3 | 9.51E-05 | 8.62E-05 | 8.60E-04 | 8.26E-04 | 4.75E-02 | 3.10E-04 | 1.72E+00 |
| 4 | 2.02E-05 | 2.35E-05 | 2.34E-04 | 2.53E-04 | 2.54E-02 | 7.76E-05 | 1.78E+00 |
| 5 | 4.15E-06 | 6.09E-06 | 6.38E-05 | 7.53E-05 | 1.31E-02 | 1.94E-05 | 1.81E+00 |
| 6 | 8.09E-07 | 1.55E-06 | 1.73E-05 | 2.19E-05 | 6.67E-03 | 4.85E-06 | 1.83E+00 |
| 7 | 1.64E-07 | 3.87E-07 | 4.34E-06 | 6.30E-06 | 3.34E-03 | 1.21E-06 | 1.83E+00 |

• **Test 1.2 Mild anisotropy,** $u(x,y) = \sin((1-x)(1-y)) + (1-x)^3(1-y)^2$**,** $\min = 0,\ \max = 1 + \sin 1$**, locally refined non-conforming rectangular mesh,** `mesh3`

This test could not be made, since we have not implemented the method for non-conforming grids; see the more detailed comments above.

• **Test 2 Numerical locking,** $u(x,y) = \sin(2\pi x)e^{-2\pi\sqrt{\frac{1}{\delta}}y},\ \delta = 10^5$**,** $\min = -1,\ \max = 1$**, regular triangular mesh,** `mesh1`

| i | nunkw | nnmat | sumflux | erl2 | ergrad | ratiol2 | ratiograd |
|---|---|---|---|---|---|---|---|
| 1 | 9.20E+01 | 4.28E+02 | 4.49E-10 | 2.61E-01 | 3.54E-01 | – | – |
| 2 | 3.52E+02 | 1.70E+03 | 1.86E-09 | 3.57E+00 | 1.61E+01 | -3.90E+00 | -5.69E+00 |
| 3 | 1.38E+03 | 6.75E+03 | -1.72E-09 | 6.51E-01 | 5.75E+00 | 2.49E+00 | 1.51E+00 |
| 4 | 5.44E+03 | 2.69E+04 | 1.33E-09 | 1.00E-01 | 1.97E+00 | 2.73E+00 | 1.56E+00 |
| 5 | 2.16E+04 | 1.08E+05 | 1.31E-09 | 1.62E-02 | 6.83E-01 | 2.64E+00 | 1.54E+00 |
| 6 | 8.62E+04 | 4.30E+05 | -2.25E-09 | 2.75E-03 | 2.40E-01 | 2.56E+00 | 1.51E+00 |
| 7 | 3.45E+05 | 1.72E+06 | 3.02E-09 | 4.76E-04 | 8.44E-02 | 2.53E+00 | 1.51E+00 |

**ocvl2** = 2.54E+00, **ocvgradl2** = 1.51E+00.

| i | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| 1 | 1.81E-11 | 7.66E-12 | 1.29E-10 | 1.60E-10 | 5.69E+02 | -1.06E+00 | 1.07E+00 |
| 2 | 2.55E-11 | 2.27E-10 | 5.66E-10 | 4.00E-11 | 5.06E+02 | -6.50E+00 | 5.75E+00 |
| 3 | 5.68E-11 | 1.72E-11 | -4.95E-10 | -7.63E-10 | 4.27E+02 | -2.46E+00 | 2.29E+00 |
| 4 | 1.02E-10 | 7.27E-12 | 3.03E-11 | 7.09E-10 | 2.73E+02 | -1.10E+00 | 1.09E+00 |
| 5 | 4.57E-11 | 5.18E-10 | 1.14E-09 | -2.76E-09 | 4.78E+01 | -1.01E+00 | 1.01E+00 |
| 6 | 7.97E-11 | 5.30E-11 | 6.18E-10 | -2.70E-09 | 7.16E+01 | -1.00E+00 | 1.00E+00 |
| 7 | 2.05E-10 | 3.46E-09 | -6.62E-09 | -1.06E-08 | 3.59E+01 | -1.00E+00 | 1.00E+00 |

• **Test 2 Numerical locking,** $u(x,y) = \sin(2\pi x)e^{-2\pi\sqrt{\frac{1}{\delta}}y}$, $\delta = 10^6$**,**
$\min = -1$, $\max = 1$**, regular triangular mesh,** `mesh1`

| i | nunkw | nnmat | sumflux | erl2 | ergrad | ratiol2 | ratiograd |
|---|---|---|---|---|---|---|---|
| 1 | 9.20E+01 | 4.28E+02 | 2.01E-09 | 2.61E-01 | 3.54E-01 | – | – |
| 2 | 3.52E+02 | 1.70E+03 | 1.31E-09 | 1.13E+01 | 5.10E+01 | -5.62E+00 | -7.41E+00 |
| 3 | 1.38E+03 | 6.75E+03 | 1.73E-07 | 2.06E+00 | 1.82E+01 | 2.49E+00 | 1.51E+00 |
| 4 | 5.44E+03 | 2.69E+04 | 1.69E-08 | 3.16E-01 | 6.21E+00 | 2.73E+00 | 1.57E+00 |
| 5 | 2.16E+04 | 1.08E+05 | 2.91E-08 | 5.12E-02 | 2.16E+00 | 2.64E+00 | 1.53E+00 |
| 6 | 8.62E+04 | 4.30E+05 | 1.07E-08 | 8.66E-03 | 7.57E-01 | 2.57E+00 | 1.52E+00 |
| 7 | 3.45E+05 | 1.72E+06 | 2.68E-08 | 1.50E-03 | 2.67E-01 | 2.53E+00 | 1.50E+00 |

**ocvl2** = 2.53E+00, **ocvgradl2** = 1.51E+00.

| i | erflx0 | erflx1 | fluy0 | fluy1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| 1 | 2.94E-11 | 2.99E-10 | 1.25E-09 | -9.31E-10 | 1.80E+03 | -1.07E+00 | 1.07E+00 |
| 2 | 7.60E-10 | 9.69E-10 | 2.85E-08 | -2.85E-08 | 1.61E+03 | -1.86E+01 | 1.63E+01 |
| 3 | 3.84E-09 | 1.09E-08 | 1.09E-07 | -2.85E-08 | 1.36E+03 | -6.59E+00 | 6.08E+00 |
| 4 | 7.37E-10 | 1.89E-11 | 1.15E-08 | 8.86E-10 | 8.70E+02 | -1.84E+00 | 1.75E+00 |
| 5 | 2.03E-09 | 2.78E-10 | 1.87E-08 | -5.83E-10 | 4.51E+02 | -1.06E+00 | 1.06E+00 |
| 6 | 6.89E-10 | 4.56E-10 | 8.24E-09 | -4.75E-09 | 2.27E+02 | -1.00E+00 | 1.00E+00 |
| 7 | 4.63E-09 | 3.94E-09 | -1.87E-08 | 4.13E-08 | 1.14E+02 | -1.00E+00 | 1.00E+00 |

• **Test 3 Oblique flow,** $\min = 0$, $\max = 1$**, uniform rectangular mesh,** `mesh2`

As reference mesh, mesh2_7 with meshsize 5.52E-03 has been chosen.

| i | nunkw | nnmat | sumflux | umin | umax |
|---|---|---|---|---|---|
| 1 | 4.00E+01 | 2.32E+02 | -1.33E-15 | 7.06E-02 | 7.06E-02 |
| 2 | 1.44E+02 | 9.12E+02 | -6.63E-15 | 3.14E-02 | 9.69E-01 |
| 3 | 5.44E+02 | 3.62E+03 | -2.98E-14 | 1.63E-02 | 9.84E-01 |
| 4 | 2.11E+03 | 1.44E+04 | -1.55E-13 | 8.25E-03 | 9.92E-01 |
| 5 | 8.32E+03 | 5.75E+04 | -5.93E-13 | 4.10E-03 | 9.96E-01 |
| ref | 1.32E+05 | 9.18E+05 | -8.66E-12 | 1.00E-03 | 9.99E-01 |

**Figure 1.** *Solutions for the oblique flow on mesh2_i for i=2 (left), i=3 (center), i=4 (right)*

| i | flux0 | flux1 | fluy0 | fluy1 | ener1 | ener2 | eren |
|---|---|---|---|---|---|---|---|
| 1 | -1.93E-01 | 1.93E-01 | -9.57E-02 | 9.57E-02 | 2.19E-01 | 2.77E-01 | 2.09E-01 |
| 2 | -1.90E-01 | 1.90E-01 | -1.00E-01 | 1.00E-01 | 2.41E-01 | 2.72E-01 | 1.15E-01 |
| 3 | -1.92E-01 | 1.92E-01 | -9.99E-02 | 9.99E-02 | 2.38E-01 | 2.43E-01 | 1.88E-02 |
| 4 | -1.93E-01 | 1.93E-01 | -9.89E-02 | 9.89E-02 | 2.42E-01 | 2.43E-01 | 4.49E-03 |
| 5 | -1.93E-01 | 1.93E-01 | -9.88E-02 | 9.88E-02 | 2.42E-01 | 2.42E-01 | 1.28E-03 |
| ref | -1.93E-01 | 1.93E-01 | -9.87E-02 | 9.87E-02 | 2.42E-01 | 2.42E-01 | 1.05E-04 |

• **Test 4 Vertical fault,** $\min = 0$, $\max = 1$**, non-conforming rectangular mesh,** `mesh5`

As reference mesh, mesh2_7 with meshsize 5.52E-03 has been chosen. The test for $i = 1$ could not be made, since we have not implemented the method for non-conforming grids; see the more detailed comments above.

| i | nunkw | nnmat | sumflux | umin | umax |
|---|---|---|---|---|---|
| 1 | – | – | – | – | – |
| reg | 8.40E+02 | 5.64E+03 | -4.14E-12 | 2.12E-02 | 9.81E-01 |
| ref | 1.32E+05 | 9.18E+05 | -3.74E-11 | 1.63E-03 | 9.98E-01 |

| i | flux0 | flux1 | fluy0 | fluy1 | ener1 | ener2 | eren |
|---|---|---|---|---|---|---|---|
| 1 | – | – | – | – | – | – | – |
| reg | -4.19E+01 | 4.43E+01 | -2.34E+00 | 8.14E-04 | 4.26E+01 | 4.31E+01 | 1.17E-02 |
| ref | -4.21E+01 | 4.45E+01 | -2.47E+00 | 7.98E-04 | 4.32E+01 | 4.32E+01 | 3.53E-04 |

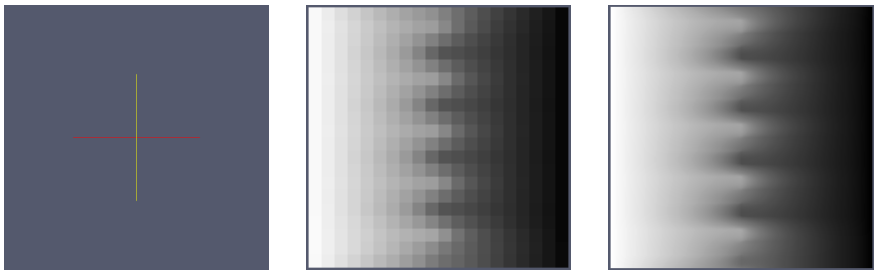

**Figure 2.** *Solution for the vertical fault on a non-conforming rectangular mesh (not available), a regular 20x20 mesh (middle), and a reference 256x256 mesh (right)*

• **Test 5 Heterogeneous rotating anisotropy,** $\min = 0,\ \max = 1$ **, uniform rectangular mesh,** `mesh2`

| i | nunkw | nnmat | sumflux | erl2 | ergrad | ratiol2 | ratiograd |
|---|---|---|---|---|---|---|---|
| 1 | 4.00E+01 | 2.32E+02 | 0.00E+00 | 1.94E-01 | 5.69E-01 | – | – |
| 2 | 1.44E+02 | 9.12E+02 | -1.78E-15 | 3.16E-02 | 2.25E-01 | 2.91E+00 | 1.45E+00 |
| 3 | 5.44E+02 | 3.62E+03 | -2.66E-15 | 5.92E-03 | 7.41E-02 | 2.59E+00 | 1.67E+00 |
| 4 | 2.11E+03 | 1.44E+04 | -7.11E-15 | 1.31E-03 | 2.02E-02 | 2.26E+00 | 1.92E+00 |
| 5 | 8.32E+03 | 5.75E+04 | 6.22E-15 | 3.17E-04 | 5.09E-03 | 2.08E+00 | 2.01E+00 |

**ocvl2** = 2.06E+00, **ocvgradl2** = 1.99E+00.

| i | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| 1 | 2.70E-01 | 2.14E-01 | 2.70E-01 | 2.14E-01 | 1.20E+00 | 1.74E-01 | 9.04E-01 |
| 2 | 8.33E-02 | 8.91E-02 | 8.33E-02 | 8.91E-02 | 6.53E-01 | 4.10E-02 | 9.71E-01 |
| 3 | 2.55E-02 | 4.29E-02 | 2.55E-02 | 4.29E-02 | 3.34E-01 | 9.78E-03 | 9.92E-01 |
| 4 | 7.68E-03 | 2.17E-02 | 7.68E-03 | 2.17E-02 | 1.68E-01 | 2.39E-03 | 9.98E-01 |
| 5 | 2.26E-03 | 1.10E-02 | 2.26E-03 | 1.10E-02 | 8.45E-02 | 5.99E-04 | 9.99E-01 |

• **Test 6 Oblique drain,** $\min = -1.2,\ \max = 0$**, coarse (C) and fine (F) oblique meshes,** `mesh6` **and** `mesh7`

The test could not be made for the fine mesh, since we have not implemented the method for non-conforming grids; see the more detailed comments above.

| grid | nunkw | nnmat | sumflux | erl2 | ergrad |
|---|---|---|---|---|---|
| C | 4.51E+02 | 2.97E+03 | -2.17E-13 | 4.99E-15 | 4.38E-14 |
| F | – | – | – | – | – |

| grid | erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|---|
| C | 1.22E-14 | 2.60E-14 | 1.53E-15 | 5.05E-14 | 3.06E-12 | -1.15E+00 | -5.38E-02 |
| F | – | – | – | – | – | – | – |

• **Test 7 Oblique barrier,** $\min = -5.575,\ \max = 0.575$**, coarse oblique mesh** `mesh6`

| nunkw | nnmat | sumflux | erl2 | ergrad |
|---|---|---|---|---|
| 4.51E+02 | 2.97E+03 | -4.45E-13 | 3.25E-15 | 4.73E-15 |

| erflx0 | erflx1 | erfly0 | erfly1 | erflm | umin | umax |
|---|---|---|---|---|---|---|
| 8.42E-13 | 1.39E-13 | 9.99E-16 | 2.50E-13 | 4.24E-13 | -5.54E+00 | 5.37E-01 |

• **Test 8 Perturbed parallelograms,** $\min = 0$**, perturbed parallelograms mesh** `mesh8`

As mentioned above, there is one parameter $\tilde{u}_E$ for tuning the scheme. In all tests before, this parameter was set element-wise to $\tilde{u}_E = c|E|^{-1}\,\mathrm{trace}(K_E)$ with $c = 2$. For this test, it seems necessary to choose the constant $c$ larger in order to stabilize the scheme and reduce the unfortunately appearing oscillations. For the following table, $c$ was set to 32. The definition of flux0, flux1, and fluy1 was changed in the obvious way to meet the requirements of the domain shape.

| nunkw | nnmat | sumflux | umin | umax |
|---|---|---|---|---|
| 2.64E+02 | 1.72E+03 | 5.51E-14 | -2.04E-02 | 8.44E-02 |

| flux0 | flux1 | fluy0 | fluy1 |
|---|---|---|---|
| -2.29E-02 | 3.43E-03 | 5.10E-01 | 5.09E-01 |

| i\j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1.57E-04 | -4.02E-04 | 1.10E-03 | -3.74E-03 | 3.99E-03 | 1.19E-02 | -2.68E-03 | 9.09E-04 | 1.76E-03 | -9.25E-04 | -2.10E-04 |
| 2 | -6.02E-04 | -7.02E-04 | 1.94E-03 | -6.77E-03 | 1.06E-02 | 2.86E-02 | -4.25E-03 | 2.55E-03 | 7.72E-03 | -3.29E-03 | 1.69E-04 |
| 3 | -1.04E-03 | -1.42E-03 | 2.06E-03 | -7.15E-03 | 1.58E-02 | 4.46E-02 | 2.41E-03 | -9.51E-03 | 1.12E-02 | -3.29E-03 | 1.22E-03 |
| 4 | -1.34E-03 | -1.91E-03 | 5.53E-03 | -1.51E-02 | 1.28E-02 | 6.51E-02 | 1.47E-03 | -1.23E-02 | 1.07E-02 | -2.21E-03 | 1.64E-03 |
| 5 | -1.26E-03 | -2.00E-03 | 8.65E-03 | -1.89E-02 | 1.08E-02 | 8.01E-02 | 2.40E-03 | -1.29E-02 | 1.46E-02 | -3.09E-03 | 1.70E-03 |
| 6 | -8.74E-04 | -2.75E-03 | 8.31E-03 | -1.63E-02 | 6.35E-03 | **8.46E-02** | -2.57E-03 | -1.68E-02 | 1.41E-02 | -2.02E-03 | 1.76E-03 |
| 7 | -1.61E-04 | -3.33E-03 | 9.55E-03 | -1.07E-02 | -8.65E-03 | 7.25E-02 | 8.50E-03 | -1.65E-02 | 9.92E-03 | -7.34E-04 | 1.45E-03 |
| 8 | 1.59E-03 | -4.47E-03 | 8.11E-03 | -2.39E-03 | -1.35E-02 | 5.86E-02 | 2.59E-02 | **-2.04E-02** | 9.54E-03 | -1.29E-03 | 8.18E-04 |
| 9 | 2.46E-03 | -5.07E-03 | 5.09E-03 | -2.39E-03 | -9.27E-03 | 5.22E-02 | 1.68E-02 | -1.73E-02 | 1.15E-02 | -1.76E-03 | 3.41E-04 |
| 10 | 1.50E-03 | -2.47E-03 | 2.65E-03 | -2.69E-03 | -3.53E-03 | 3.09E-02 | 1.07E-02 | -1.12E-02 | 9.74E-03 | -2.08E-03 | 4.84E-04 |
| 11 | 3.97E-04 | -6.82E-05 | 5.68E-05 | -3.12E-04 | 1.28E-05 | 6.21E-03 | 3.90E-03 | -2.99E-03 | 2.86E-03 | -1.06E-03 | 3.50E-04 |

• **Test 9 Anisotropy with wells,** $\min = 0,\ \max = 1.$**, square uniform grid** `mesh9`

The implementational effort necessary to incorporate the required boundary conditions was considered too high. Therefore, the problem setting has been slightly changed. In particular, elements (4,6) and (8,6) have been omitted, resulting in two holes inside the domain. On the boundary of the two holes, the corresponding Dirichlet conditions have been set.

| nunkw | nnmat | sumflux | umin | umax |
|---|---|---|---|---|
| 2.64E+02 | 1.69E+03 | -7.03E-15 | -4.21E-02 | 1.04E+00 |

| i\j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -4.18E-02 | 1.01E-02 | 2.18E-01 | 5.56E-01 | 8.40E-01 | 9.74E-01 | 1.01E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 |
| 2 | -3.61E-02 | -2.28E-02 | 1.10E-01 | 4.03E-01 | 7.37E-01 | 9.38E-01 | 1.00E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 |
| 3 | -8.86E-03 | **-4.21E-02** | 3.95E-02 | 2.60E-01 | 6.01E-01 | 8.78E-01 | 9.91E-01 | 1.01E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 |
| 4 | -7.69E-03 | -2.32E-02 | -1.68E-02 | 1.53E-01 | 4.45E-01 | 7.82E-01 | 9.66E-01 | 1.01E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 |
| 5 | -2.19E-02 | -7.90E-03 | -2.74E-02 | 5.68E-02 | 3.13E-01 | 6.39E-01 | 9.19E-01 | 1.00E+00 | 1.02E+00 | 1.02E+00 | 1.02E+00 |
| 6 | -1.66E-02 | -1.84E-02 | -1.14E-02 | **0.00E+00** | 1.86E-01 | 5.00E-01 | 8.14E-01 | **1.00E+00** | 1.01E+00 | 1.02E+00 | 1.02E+00 |
| 7 | -1.75E-02 | -1.69E-02 | -1.61E-02 | -4.59E-03 | 8.11E-02 | 3.61E-01 | 6.87E-01 | 9.43E-01 | 1.03E+00 | 1.01E+00 | 1.02E+00 |
| 8 | -1.74E-02 | -1.73E-02 | -1.64E-02 | -1.11E-02 | 3.44E-02 | 2.18E-01 | 5.55E-01 | 8.47E-01 | 1.02E+00 | 1.02E+00 | 1.01E+00 |
| 9 | -1.74E-02 | -1.73E-02 | -1.69E-02 | -1.40E-02 | 9.22E-03 | 1.22E-01 | 3.99E-01 | 7.40E-01 | 9.60E-01 | 1.04E+00 | 1.01E+00 |
| 10 | -1.74E-02 | -1.74E-02 | -1.71E-02 | -1.56E-02 | -3.61E-03 | 6.18E-02 | 2.63E-01 | 5.97E-01 | 8.90E-01 | 1.02E+00 | **1.04E+00** |
| 11 | -1.74E-02 | -1.74E-02 | -1.73E-02 | -1.64E-02 | -1.02E-02 | 2.64E-02 | 1.60E-01 | 4.44E-01 | 7.82E-01 | 9.90E-01 | **1.04E+00** |

## 3. Comments on the results

Some more technical comments have already been made in section 2. Concerning the quality of the results, the mimetic finite difference method proves to be very robust and accurate in most tests. In particular, for tests 1, 2, 5, 6, and 7 with available exact solution, we observe the following:

– The error decay for the pressures always exhibits the typical quadratic (super)-convergence behavior.

– Concerning the error in the solution gradients, a standard linear decay can be observed for the simplicial meshes in tests 1 and 2, while for the quadrilateral grids, superconvergence results in a quadratic decay in tests 1 and 5. We point out that this superconvergence cannot only be observed for the uniform grids `mesh2`, but also for the distorted meshes `mesh4`.

– Concerning the fluxes, the errors `erflx0`, ..., `erflm` show at least the expected linear decay. Here, we like to add that, evaluated in the norm associated with the scalar product $(\cdot,\cdot)_{X_h}$, we usually also observe a superconvergent behavior for the error in the fluxes.

– Especially remarkable are the tests 6 and 7 where all calculated errors are practically zero since the piecewise linear exact solution, its gradient, and its fluxes are exactly reproduced at the barycenter of the elements and faces, respectively.

– Unfortunately, test 2 produces quite unsatisfactory results. After an already reasonable solution with respect to the mesh size is produced at the coarsest level, the solution as well as the errors blow up at the second one. Over the following levels, it becomes obvious that some higher order terms take effect, since the convergence is accelerated by half an order. It would be very interesting to obtain an explanation for this behavior.

For the other tests 3, 4, 8, and 9, the following observations are made:

– The sum of the fluxes over the boundary is always practically zero, which nicely illustrates the total mass conservation of the scheme.

– For test 3, the minimum and maximum pressure values, as well as the boundary fluxes and the discrete energies; converge fast towards the values of the reference solution. Already for quite coarse meshes, reasonable solutions are calculated.

– Test 4 shows that the solution on the regular $20\times20$ mesh is already very close to the reference solution.

– Tests 8 and 9 unfortunately show that the Hopf lemmas are violated. While test 9 still produces a good result and the violation is within an acceptable range, the oscillations in test 8 heavily reduce the quality of the solution. In order to improve the method, we could set up a suggestion of the developers made in [BRE 05b]. In particular, by making use of the freedom in selecting members of the proposed family of mimetic methods, we could try to enforce a discrete maximum principle. However, it is not yet clear how to explicitly tune the parameters of the method to achieve this.

Overall, the implemented mimetic finite difference method solves the majority of the proposed problems with convincing accuracy and robustness. It appears to be most promising for tackling real applications, and has already proved to be able to do so, [AAR 08]. This is not only due to accuracy and robustness, but also due to the remarkable ease of implementation.

## 4. References


[AAR 08] AARNES J. E., KROGSTAD S., LIE K.-A., "Multiscale mixed/mimetic methods on corner point grids", to appear in Comp. Geosc., 2008.

[BAS 07] BASTIAN P., BLATT M., DEDNER A., ENGWER C., KLÖFKORN R., KORNHUBER R., OHLBERGER M., SANDER O., "A Generic Grid Interface for Adaptive and Parallel Scientific Computing. Part II: implementation and Tests in DUNE", report num. 404, 2007, DFG Research Center MATHEON.

[BRE 05a] BREZZI F., LIPNIKOV K., SHASHKOV M., "Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes", *SIAM J. Numer. Anal.*, vol. 43, num. 5, 2005, p. 1872–1896.

[BRE 05b] BREZZI F., LIPNIKOV K., SIMONCINI V., "A family of mimetic finite difference methods on polygonal and polyhedral meshes", *Math. Models Methods Appl. Sci.*, vol. 15, num. 10, 2005, p. 1533–1551.

[FAV 81] FAVORSKII A., SAMARSKII A., SHASHKOV M. J., TISHKIN V., "Operational Finite-Difference Schemes", *Differential Equations*, vol. 17, 1981, p. 854–862.

[FLE 07] FLEMISCH B., FRITZ J., HELMIG R., NIESSNER J., WOHLMUTH B., "DUMUX: a multi-scale multi-physics toolbox for flow and transport processes in porous media", IBRAHIMBEGOVIC A., DIAS F., Eds., *ECCOMAS Thematic Conference on Multi-scale Computational Methods for Solids and Fluids*, Cachan, France, November 28-30, 2007, p. 82–87.

[HÆG 07] HÆGLAND H., DAHLE H., EIGESTAD G., LIE K.-A., AAVATSMARK I., "Improved streamlines and time-of-flight for streamline simulation on irregular grids", *Advances in Water Resources*, vol. 30, num. 4, 2007, p. 1027–1045.

[ROB ] ROBIDOUX N., "Mimetic discretizations bibliography homepage", http://www.math.sfu.ca/~nrobidou/public_html/mimetic/mimetic.html.

[SCH 04] SCHENK O., Gï¿½RTNER K., "Solving unsymmetric sparse systems of linear equations with PARDISO", *Journal of Future Generation Computer Systems*, vol. 20, num. 3, 2004, p. 475–487.

[SHA 96] SHASHKOV M. J., *Conservative finite-difference methods on general grids*, CRC Press, Boca Raton, FL, 1996.