Benchmark 3D: A Mimetic Finite Difference Method

Peter Bastian, Olaf Ippisch, and Sven Marnach

1 Presentation of the scheme

In the two-dimensional discretisation benchmark session at the FVCA5 conference, we participated with a Mimetic Finite Difference (MFD) method [7]. In this paper, we present results for the three-dimensional case using the same method. Since the previous conference, the equivalence of MFD, Hybrid Finite Volume and Mixed Finite Volume methods has been demonstrated in [6]. Our outline of the method as used in our computations follows the exposition in [5].

First, the diffusion problem is restated as a system of two first order PDEs

$$\begin{aligned} \operatorname{div}\mathbf{v} &= f \quad \text{in } \Omega, \\ \mathbf{v} &= -\mathbf{K}\nabla u \quad \text{in } \Omega, \\ u &= \bar{u} \quad \text{on } \Gamma_D, \\ \mathbf{K}\nabla u\cdot n &= g \quad \text{on } \Gamma_N. \end{aligned} \tag{1}$$

Our aim will be the definition of discrete analogues of the divergence operator div and the flux operator $-\mathbf{K}\nabla$. To this end, we first define the spaces of discrete scalar and vector functions. Let $\mathscr{T}_h$ denote a conforming triangulation of the domain Ω. The elements $E \in \mathscr{T}_h$ are assumed to be polyhedra. For details on the requirements of $\mathscr{T}_h$, see [4].

A *discrete scalar function* u is assumed to be constant on the elements E of the triangulation. The value of u in the element E is denoted by u_E. The dimension of the space Q_h of discrete scalar functions is equal to the number of elements of $\mathscr{T}_h$.

Peter Bastian, Olaf Ippisch, and Sven Marnach
Interdisciplinary Center for Scientific Computing, University of Heidelberg.
Corresponding author Sven Marnach, e-mail: sven.marnach@iwr.uni-heidelberg.de

J. Fořt et al. (eds.), *Finite Volumes for Complex Applications VI – Problems & Perspectives*, Springer Proceedings in Mathematics 4,
DOI 10.1007/978-3-642-20671-9_93, © Springer-Verlag Berlin Heidelberg 2011

A *discrete vector function* $\mathbf{v}$ is given by assigning a real number $\mathbf{v}_E^e$ to each face e of each element E. These numbers are regarded as the normal components of the vector function with respect to the outer normal $\mathbf{n}_E^e$ on the face e. For a face e that is shared by elements E_1 and E_2, we require the compatibility of the normal components

$$\mathbf{v}_{E_1}^e = -\mathbf{v}_{E_2}^e. \tag{2}$$

Thus the dimension of the space X_h of discrete vector functions is equal to the number of faces of $\mathscr{T}_h$.

We now introduce the discrete differential operators. Again, see [4] for the details. The *discrete divergence operator* $\mathrm{div}_h : X_h \to Q_h$ is defined to comply with the divergence theorem on each element,

$$(\mathrm{div}_h \mathbf{v})_E = \frac{1}{|E|} \sum_{e \subset \partial E} |e|\, v_E^e, \tag{3}$$

where the sum is over all faces e of the element E. For the definition of the *discrete flux operator*, we introduce additional scalar unknowns on each face e of the triangulation denoted by u^e, and define $(-\mathbf{K}\nabla)_h : Q_h \to X_h$ by

$$\big((-\mathbf{K}\nabla)_h u\big)_E^e = \sum_{f \subset \partial E} \mathbb{W}_E^{e,f} |f| (u_E - u^f), \tag{4}$$

where $\mathbb{W}_E$ is a symmetric and positive definite matrix defined below. The scalar unknowns on the faces can be eliminated by compatibility requirement (2) on the inner faces and by boundary conditions (1) on the outer faces.

Now we can give the whole linear system discretising the linear diffusion problem. For each element E, we have the equation

$$\mathrm{div}_h (-\mathbf{K}\nabla)_h u = \frac{1}{|E|} \sum_{e \subset \partial E} |e| \sum_{f \subset \partial E} \mathbb{W}_E^{e,f} |f| (u_E - u^f) = q_E. \tag{5}$$

The equation for an inner face e shared by the elements E_1 and E_2 reads

$$\begin{aligned} &\big((-\mathbf{K}\nabla)_h u\big)_{E_1}^e + \big((-\mathbf{K}\nabla)_h u\big)_{E_2}^e \\ &\qquad = \sum_{f \subset \partial E_1} \mathbb{W}_{E_1}^{e,f} |f| (u_{E_1} - u^f) + \sum_{f \subset \partial E_2} \mathbb{W}_{E_2}^{e,f} |f| (u_{E_2} - u^f) = 0. \end{aligned} \tag{6}$$

For each face e on the Neumann boundary Γ_N, we get

$$\big((-\mathbf{K}\nabla)_h u\big)_E^e = \sum_{f \subset \partial E} \mathbb{W}_E^{e,f} |f| (u_E - u^f) = g^e. \tag{7}$$

Finally, a face e on the Dirichlet boundary provides us with the trivial equation

$$u^e = \bar{u}^e. \tag{8}$$

To obtain a symmetric and positive definite stiffness matrix, we first eliminate the unknowns on the Dirichlet boundaries. Then, we scale (5) by $|E|$ and (6) as well as (7) by $-|e|$, see [8].

We finally give the definition of the matrix $\mathbb{W}_E$ for an element E. Let k_E denote the number of faces of E. Define the $k_E \times 3$ matrices $\mathbb{R}$ and $\mathbb{N}$ by

$$\mathbb{R}_{e,i} = \int_e (x_i - x_{E,i}), \quad \mathbb{N}_{e,i} = \mathbf{e}_i \cdot n_E^e, \tag{9}$$

where e ranges over the faces of E, $i = 1, 2, 3$, x_i is the i-th coordinate function, x_E is the "centre" of E and $\mathbf{e}_i$ is the unit vector in the direction of the i-th axis. The centre point x_E can be chosen on each cell individually (subject to some restrictions). A good choice is to use the centre of mass, which we used for the tetrahedral mesh. For the hexahedral meshes, we used the image of the centre of the unit cube under the usual trilinear coordinate mappings.

We construct a $k_E \times k_E$ matrix $\mathbb{W}_E$ according to algorithm 1 in [5]. In short, that means the following:

1. Orthonormalise the columns of the matrix $\mathbb{R}$ using the Gram–Schmidt algorithm and call the resulting matrix $\tilde{\mathbb{R}}$.
2. Set $\mathbb{D} = \mathbb{I} - \tilde{\mathbb{R}}\tilde{\mathbb{R}}^{\mathrm{T}}$, where $\mathbb{I}$ denotes the $k_E \times k_E$ unit matrix.
3. Define $\mathbb{W}_E$ by

$$\mathbb{W}_E = \frac{1}{|E|}\mathbb{N}\mathbf{K}\mathbb{N}^{\mathrm{T}} + \omega\mathbb{D}, \tag{10}$$

where ω is an arbitrary positive real number and $\mathbf{K}$ is simply evaluated at the cell centre x_E.

We used the common choice for ω

$$\omega = \frac{\operatorname{trace}\mathbf{K}}{|E|}, \tag{11}$$

which was suggested in [5].

2 Numerical results

For estimating the L_2 error, we compared the approximate solution u_E on a cell E with the average

$$u_{E,\text{exact}} = \frac{1}{|E|} \int_E u(x)dx$$

of the exact solution over the cell,

$$\text{erl2}^2 = \frac{\sum_E |E|(u_E - u_{E,\text{exact}})^2}{\sum_E |E|(u_{E,\text{exact}})^2}. \tag{12}$$

The MFD method provides values for the fluxes on the faces, but does not allow the direct computation of approximate gradients of the solution. In some cases it would be possible to get a reconstruction of the gradients in the interior of a cell using the Piola transformation, but this fails, for example, for cells with hanging nodes on some faces. To circumvent these problems, we substituted the H_1 semi-norm by the somewhat unnatural "flux norm"

$$\text{ergrad}^2 = \frac{\sum_e |e|(\mathbf{v}^e - \mathbf{v}^e_{\text{exact}})^2}{\sum_e |e|(\mathbf{v}^e_{\text{exact}})^2}, \tag{13}$$

where the exact average flux over the face e is given by

$$\mathbf{v}^e_{\text{exact}} = \frac{1}{|e|} \int_e \mathbf{n}(x) \cdot A(x) \nabla u(x) dx$$

We did not provide any values for the energy norm E. Though it is possible to give an approximation of the energy norm using the formulation

$$\begin{aligned} E &= \int_\Omega \mathbf{K} \nabla u \cdot \nabla u \\ &= \int_\Omega \mathbf{K}^{-1}(-\mathbf{K}\nabla u) \cdot (-\mathbf{K}\nabla u) \\ &= \sum_{E \in \mathcal{T}_h} \int_E \mathbf{K}^{-1}(-\mathbf{K}\nabla u) \cdot (-\mathbf{K}\nabla u) \end{aligned}$$

and the scalar product on X_h, this would not provide much information, since E would coincide with the exact energy norm up to the accuracy of the linear solver by construction of the method.

All numerical tests have been performed with the DUNE software framework [1, 3] using the `dune-pdelab` discretisation framework described in [2].

- **Test 1 Mild anisotropy,** $u(x, y, z) = 1 + \sin(\pi x) \sin\left(\pi\left(y + \frac{1}{2}\right)\right) \sin\left(\pi\left(z + \frac{1}{3}\right)\right)$ $\min = 0, \ \max = 2$**, Tetrahedral meshes**

i	nu	nmat	umin	uemin	umax	uemax	normg
1	6311	46371	2.26E-02	2.03E-02	1.986	1.989	0.000
2	12146	90106	5.50E-03	6.84E-03	1.989	1.989	0.000
3	23859	178079	8.50E-03	9.13E-03	1.994	1.994	0.000
4	46957	352277	5.10E-03	5.52E-03	1.997	1.997	0.000
5	93267	702867	1.91E-03	1.49E-03	1.996	1.997	0.000
6	186040	1407080	1.75E-03	1.83E-03	1.997	1.997	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	6311	4.55E-03	0.000	1.41E-01	0.000	0.00E+00	0.000
2	12146	2.88E-03	2.102	1.05E-01	1.376	0.00E+00	0.000
3	23859	1.82E-03	2.030	9.73E-02	0.328	0.00E+00	0.000
4	46957	1.20E-03	1.859	7.03E-02	1.440	0.00E+00	0.000
5	93267	7.38E-04	2.120	6.13E-02	0.602	0.00E+00	0.000
6	186040	4.65E-04	2.004	4.75E-02	1.102	0.00E+00	0.000

- **Test 1 Mild anisotropy,** $u(x, y, z) = 1 + \sin(\pi x) \sin\left(\pi\left(y + \frac{1}{2}\right)\right) \sin\left(\pi\left(z + \frac{1}{3}\right)\right)$ $\min = 0, \ \max = 2$**, Kershaw meshes**

i	nu	nmat	umin	uemin	umax	uemax	normg
1	2240	23744	-6.03E-01	3.03E-02	2.100	1.958	0.000
2	17152	189184	-5.83E-03	1.06E-02	2.008	1.993	0.000
3	134144	1510400	-1.11E-03	1.75E-03	2.000	1.997	0.000
4	1060864	12070912	1.65E-04	7.14E-04	2.000	1.999	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	2240	3.27E-01	0.000	1.19E+01	0.000	0.00E+00	0.000
2	17152	5.28E-02	2.685	3.37E+00	1.859	0.00E+00	0.000
3	134144	8.89E-03	2.600	5.26E-01	2.709	0.00E+00	0.000
4	1060864	2.02E-03	2.146	1.12E-01	2.245	0.00E+00	0.000

- **Test 1 Mild anisotropy,** $u(x,y,z) = 1 + \sin(\pi x)\sin\left(\pi\left(y+\frac{1}{2}\right)\right)\sin\left(\pi\left(z+\frac{1}{3}\right)\right)$ $\min = 0,\ \max = 2$**, Checkerboard meshes**

i	nu	nmat	umin	uemin	umax	uemax	normg
1	192	2496	1.27E-01	1.54E-01	1.883	1.846	0.000
2	1488	25248	-1.32E-01	4.01E-02	2.150	1.960	0.000
3	11712	225696	-5.37E-02	1.01E-02	2.053	1.990	0.000
4	92928	1905600	-1.40E-02	2.54E-03	2.014	1.997	0.000
5	740352	15655296	-3.52E-03	6.36E-04	2.004	1.999	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	192	2.81E-01	0.000	4.00E-01	0.000	0.00E+00	0.000
2	1488	1.19E-01	1.263	2.15E-01	0.906	0.00E+00	0.000
3	11712	3.44E-02	1.801	1.13E-01	0.936	0.00E+00	0.000
4	92928	9.39E-03	1.879	5.71E-02	0.992	0.00E+00	0.000
5	740352	2.46E-03	1.936	2.86E-02	1.001	0.00E+00	0.000

- **Test 3 Flow on random meshes,** $u(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$**,** $\min = -1,\ \max = 1$**, Random meshes**

i	nu	nmat	umin	uemin	umax	uemax	normg
1	304	2992	-1.02E+00	-7.59E-01	1.045	0.691	0.000
2	2240	23744	-9.79E-01	-9.39E-01	1.019	0.923	0.000
3	17152	189184	-1.02E+00	-9.85E-01	1.008	0.982	0.000
4	134144	1510400	-1.00E+00	-9.96E-01	1.000	0.996	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	304	8.38E-01	0.000	9.62E-01	0.000	0.00E+00	0.000
2	2240	1.58E-01	2.502	3.14E-01	1.679	0.00E+00	0.000
3	17152	3.91E-02	2.060	1.29E-01	1.314	0.00E+00	0.000
4	134144	1.15E-02	1.788	6.61E-02	0.975	0.00E+00	0.000

- **Test 4 Flow around a well, Well meshes,** min $= 0$, max $= 5.415$

i	nu	nmat	umin	uemin	umax	uemax	normg
1	3888	41268	5.74E-01	4.14E-01	5.317	5.317	0.000
2	9464	103208	2.96E-01	2.44E-01	5.328	5.328	0.000
3	20902	231574	1.75E-01	1.54E-01	5.329	5.329	0.000
4	46203	517443	1.30E-01	1.18E-01	5.330	5.330	0.000
5	94885	1069705	9.66E-02	8.99E-02	5.339	5.339	0.000
6	173515	1964101	7.67E-02	7.23E-02	5.345	5.345	0.000
7	303058	3439576	5.91E-02	5.65E-02	5.361	5.361	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	3888	5.53E-03	0.000	2.70E-01	0.000	0.00E+00	0.000
2	9464	1.64E-03	4.090	1.08E-01	3.089	0.00E+00	0.000
3	20902	8.43E-04	2.530	5.04E-02	2.888	0.00E+00	0.000
4	46203	8.27E-04	0.074	2.84E-02	2.173	0.00E+00	0.000
5	94885	6.83E-04	0.796	1.70E-02	2.142	0.00E+00	0.000
6	173515	4.84E-04	1.709	1.12E-02	2.060	0.00E+00	0.000
7	303058	4.07E-04	0.932	7.78E-03	1.965	0.00E+00	0.000

- **Test 5 Discontinuous permeability,** $u(x, y, z) = a_i \sin(2\pi x) \sin(2\pi y) \sin(2\pi z)$**,** min $= -100$, max $= 100$**, Locally refined meshes**

i	nu	nmat	umin	uemin	umax	uemax	normg
1	115	1231	-2.51E+02	-1.00E+02	250.808	100.000	0.000
2	812	8972	-4.44E+01	-3.54E+01	44.367	35.355	0.000
3	6064	68272	-8.32E+01	-7.89E+01	83.205	78.858	0.000
4	46784	532160	-9.56E+01	-9.43E+01	95.600	94.346	0.000
5	367360	4201216	-9.89E+01	-9.86E+01	98.887	98.562	0.000

i	nu	erl2	ratiol2	ergrad	ratiograd	ener	ratioener
1	115	8.77E+00	0.000	2.92E+00	0.000	0.00E+00	0.000
2	812	7.09E-01	3.860	7.88E-01	2.012	0.00E+00	0.000
3	6064	1.41E-01	2.413	3.15E-01	1.367	0.00E+00	0.000
4	46784	3.33E-02	2.116	2.13E-01	0.576	0.00E+00	0.000
5	367360	8.21E-03	2.038	1.55E-01	0.464	0.00E+00	0.000

References

1. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. part ii: Implementation and tests in dune. Computing **82**(2–3), 121–138 (2007)
2. Bastian, P., Heimann, F., Marnach, S.: Generic software components for finite elements (2009). To appear in the proceedings of Algorithmy 2009
3. Blatt, M., Bastian, P.: The iterative solver template library. In: B. Kagström, E. Elmroth, J. Dongarra, J. Wasniewski (eds.) Applied Parallel Computing. State of the Art in Scientific Computing, *Lecture Notes in Scientific Computing*, vol. 4699, pp. 666–675. Springer (2007)
4. Brezzi, F., Lipnikov, K., Shashkov, M.: Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. SIAM Journal on Numerical Analysis **43**(5), 1872–1896 (2005)
5. Brezzi, F., Lipnikov, K., Simoncini, V.: A family of mimetic finite difference methods on polygonal and polyhedral meshes. Mathematical Models and Methods in Applied Sciences **15**(10), 1533–1551 (2005)
6. Droniou, J., Eymard, R., Gallouët, T., Herbin, R.: A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods. Mathematical Models and Methods in Applied Sciences **20**(2), 265 – 295 (2010)
7. Marnach, S.: Benchmark on anisotropic problems – a mimetic finite difference method. In: J.M.H. Robert Eymard (ed.) Finite Volumes for Complex Applications V – Problems and Perspectives. ISTE Ltd and John Wiley & Sons Inc (2008)
8. Morel, J.E., Roberts, R.M., Shashkov, M.J.: A local support-operators diffusion discretization scheme for quadrilateral r-z meshes. Journal of Computational Physics **144**, 17–51 (1998)

The paper is in final form and no similar paper has been or is being submitted elsewhere.