V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

ENHANCED DIVERGENCE-FREE ELEMENTS FOR EFFICIENT INCOMPRESSIBLE FLOW SIMULATIONS IN THE PDE FRAMEWORK PEANO

Tobias Neckel, Miriam Mehl and Christoph Zenger

Fakultät für Informatik, TU München
Boltzmannstraße 3, 85748 Garching
e-mail: {neckel,mehl,zenger}@in.tum.de

Key words: incompressible flow, divergence-free elements, Cartesian grid, adaptivity, PDE framework

Abstract. *A new representation of two-dimensional divergence-free elements for incompressible flow simulations on adaptive Cartesian grids is derived. A rotated and mirrored coordinate system allows for a straightforward derivation of the divergence-free elements as well as a performance gain for the evaluation of the spatial operators due to a reduced number of computational operations. Several benchmark simulations have been performed in order to validate the derivation and implementation of the new type of elements in the PDE framework Peano.*

1 INTRODUCTION

Usual finite element discretisations for the incompressible Navier-Stokes equations lack the property of being (weakly) divergence-free. Therefore, a lot of (complicated, mathematical) workarounds are necessary and applied in order to overcome this drawback.

A different approach consists in using finite elements that are divergence-free by construction[1]. One important advantage of such divergence-free elements is the resulting skew-symmetric convection operator (at least up to outflow boundary contributions, cf. [3]) which allows to conserve both energy and momentum simultaneously. Hence, no instability can appear in the corresponding system of ordinary differential equations (ODE) which is important for turbulent simulations, e.g., where developers sometimes encounter "blow-ups" for non-divergence-free discretisations due to artificially introduced energy in the system. The derivation of such divergence-free elements typically is cumbersome and results in a coupling of the individual velocity components of the discrete solution.

In this paper, we present a straightforward way of constructing two-dimensional divergence-free elements. The derivation of these elements, which we call enhanced divergence-free elements, is based on [1] which in turn has roots in considerations for finite volume discretisations ([8, 9]). The advantage of the new derivation and representation in the proposed rotated system in this paper is a non-cumbersome construction of the elements. Furthermore, these elements are not coupled any more which results in a considerable reduction of necessary operations for evaluating spatial operators, in particular for the convection. In addition, the enhanced divergence-free elements allow for a straightforward usage in arbitrarily adaptive Cartesian grids.

The structure of this paper is as follows. In Section 2, we briefly describe our framework Peano used for simulations of partial differential equations (PDE). The structure of the enhanced divergence-free elements is derived in Section 3. We present numerical results for several benchmark scenarios in Section 4 before closing this discussion and giving a short outlook in Section 5.

2 THE PDE FRAMEWORK PEANO

Modern PDE simulations in general and CFD solvers in particular are still facing capital challenges concerning performance and re-usability. Therefore, we developed the C++ PDE framework Peano in a strictly object-oriented manner including successful solution approaches of Software Engineering such as automated tests and design patterns (see [5, 10]).

Peano works on adaptive Cartesian grids via spacetrees, space-filling curves and stack data structures. The grid construction starts with a hypercube root cell containing the complete computational domain and then, similar to the well-known octrees, subdivides the root in a recursive manner by splitting up each cell into three parts along each coor-

[1] Gresho and Sani present a nice historical overview on divergence-free discretisations in [3], p. 609 ff.

dinate axis. Hence, adaptive grids on complicated geometries can be generated automatically, easily, and fast.

In order to serialise the underlying spacetree for a run over the grid, a certain traversal order is necessary. The so called space-filling curves (cf. [6]) offer advantageous features concerning an efficient usage of caches. These self-similar recursively defined curves fit perfectly to our construction of the spacetree and grid. For accessing nodal data, we developed a very cache-efficient mechanism using simple stack data structures (see [4]) that allow push and pop operations only to access data.

Using the stacks in combination with the Peano curve assures temporal and spatial locality of data access. These two effects lead to a very efficient cache usage with cache-hit rates of 98 – 99%, independent of the dimension, the architecture or the concrete type of application in use. Furthermore, the strictly cell-wise operator evaluation, necessary for using space filling curves together with stacks, avoids the storage of any neighbourhood informations, in particular at boundaries between different refinement levels, and, thus, leads to very low memory requirements.

Peano's approach of adaptive Cartesian grids does neither depend on the concrete PDE to be solved nor on the type of temporal or spatial discretisation (FD, FE, FV, etc.). Therefore, Peano represents a general framework for different solvers, offering grid generation, data handling, etc. Currently, several solvers exist, such as for the Poisson equation, for heat equation simulations, for the continuity equation, and for the incompressible Navier-Stokes equations.

3 DIVERGENCE-FREE ELEMENTS

In this section, we first set up the general context of the discretisation of the Navier-Stokes equations in Peano. Then, we derive the divergence-free ansatz functions and list selected properties of the spatial operators before closing the section with hints on the additional enhanced degrees of freedom (DoF) on faces.

3.1 Navier-Stokes Equations

In order to approximate solutions for the incompressible Navier-Stokes Equations

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \, \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \, \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{g}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

we apply a semi-implicit approach (Chorin projection method) in Peano for the results in this paper. For a given spatial discretisation, the semi-discrete Navier-Stokes Equations have the following generic form

$$\begin{aligned} A\dot{\mathbf{u}}_h + D\mathbf{u}_h + C(\mathbf{u}_h)\mathbf{u}_h - M^T p_h &= f, \\ M\mathbf{u}_h &= 0, \end{aligned} \tag{3}$$

where $\mathbf{u}_h$ is the approximation of the solution, A is the mass matrix, and D and C represent the discrete diffusion and convection operator. As in the continuous case, the discrete divergence and gradient operators M and M^T are transposed counterparts. Gauss-Seidel iterations are currently used to compute the discrete pressure values p_h as Lagrangian multiplier values from the pressure Poisson equation (PPE); current work is on the PPE multigrid solver using Peano's adaptive grid hierarchy.

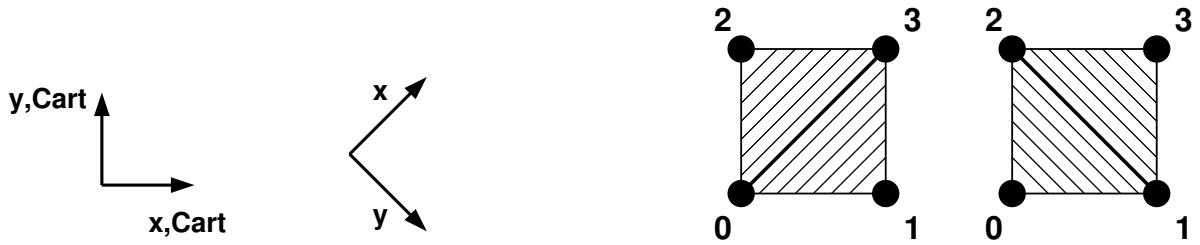
Note that for a standard discretisation, the approximate solution $\mathbf{u}_h$ is not divergence-free on all points within a given cell (which can easily be shown for bilinear ansatz functions, using the coordinate $\mathbf{x} = (h/3, h/3)$, e.g.).

3.2 Derivation of the Ansatz Functions

The basic idea of the divergence-free ansatz functions is to create functions for x and y components of the velocities that possess— individually—constant divergence on subparts of their support while having a simple structure. In a second step, the sum of all those divergence contributions on a Cartesian cell, which is still constant, will be shown to be equal to zero.

In order to derive the ansatz functions, we will use the following notation: Entities in the standard Cartesian coordinate system will hold an additional superscript $^{\text{Cart}}$ while those from the rotated and mirrored one (denoted 45 below) don't get additional superscripts (see Fig. 1(a) for a definition of the two representations). For example, $\varphi_i^{x,\text{Cart}}$ denotes the ansatz functions on node i for direction x in the standard Cartesian system, whereas φ_i^x represents the nodal base functions for the x-direction of the 45 system.

Lets assume that the subparts of the support of the ansatz functions for x and y in the 45 system are triangles on the squared grid cells defined as visualised in Fig. 1(b).



(a) (b)

Figure 1: (a) Coordinate system orientations for standard Cartesian (left) and 45 system (rotated and mirrored, right). (b) Triangle definition for $\partial_x u$ (left) and $\partial_y v$ (right) to be constant.

The partial derivative $\partial_x u$ of u is constant (with the same value) on both triangles if u is linear. In particular, $\partial_x u$ only depends on the values u_0 and u_3: $\partial_x u = u_3 - u_0$ (ignoring a constant factor involving the mesh size). This holds for v in an analogue manner: $\partial_y v = v_1 - v_2$. Hence, the overall divergence sums up to a constant value,

$$\nabla \cdot \mathbf{u}(\mathbf{x}) \;=\; \partial_x u(\mathbf{x}) + \partial_y v(\mathbf{x}) \;=\; u_3 - u_0 + v_1 - v_2 \;=\; const\ , \tag{4}$$

and is equal to zero for all points $\mathbf{x} = (x, y)^T$ if and only if the mass flux over the cell boundary is equal to zero, which is garantueed by the Chorin projection method. Hence, the resulting linear ansatz functions $\varphi_i^x(\mathbf{x})$ and $\varphi_i^y(\mathbf{x})$ are really divergence-free on all points in a cell.

Figure 2 visualises φ_i^x and φ_i^y on their support of four Cartesian grid cells (of the type of Fig. 1(b)). Using φ_i^x and φ_i^y to set up the approximate solution on N nodes with corresponding coefficients u_i and v_i,

$$\mathbf{u}_h := \sum_{i=1}^{N} (u_i \phi_i^x + v_i \phi_i^y), \tag{5}$$

$$\phi_i^x := \begin{pmatrix} \varphi_i^x \\ 0 \end{pmatrix}, \quad \phi_i^y := \begin{pmatrix} 0 \\ \varphi_i^y \end{pmatrix}, \tag{6}$$

shows the decoupling of the dimension components in ϕ_i^x and ϕ_i^y.

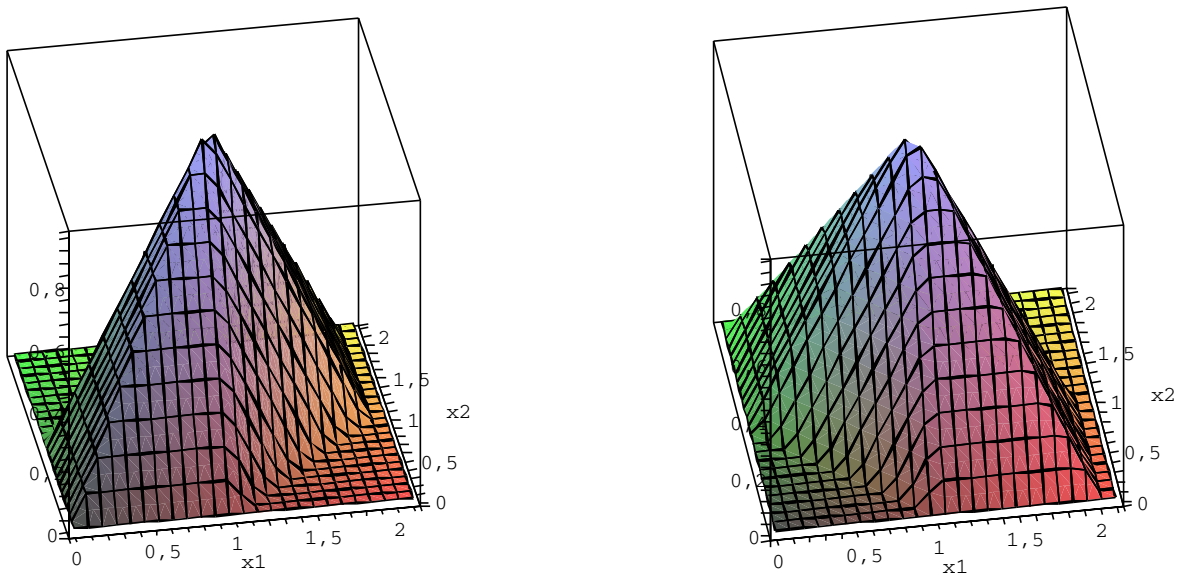


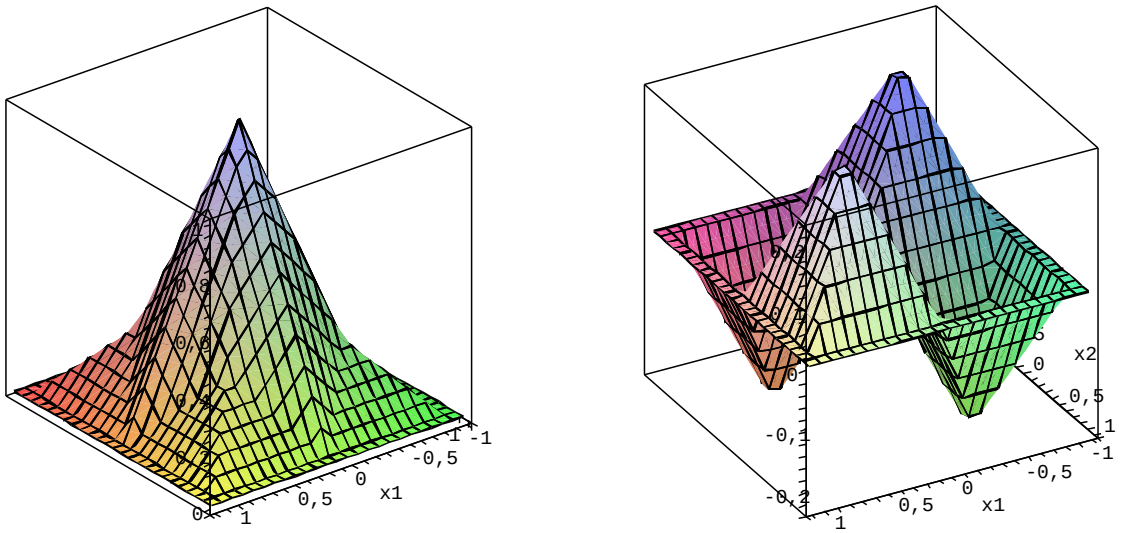
Figure 2: Divergence-free ansatz functions φ_i^x (a) and φ_i^y (b) in the 45 coordinate system (cf. Fig. 1). The two ansatz functions are linear and decoupled.

In order to check the structure of the ansatz functions in the standard rectangular Cartesian coordinate system,

$$\phi_i^{x,\mathrm{Cart}} := \begin{pmatrix} \varphi_i^{x,1,\mathrm{Cart}} \\ \varphi_i^{x,2,\mathrm{Cart}} \end{pmatrix}, \quad \phi_i^{y,\mathrm{Cart}} := \begin{pmatrix} \varphi_i^{y,1,\mathrm{Cart}} \\ \varphi_i^{y,2,\mathrm{Cart}} \end{pmatrix}, \tag{7}$$

we need to apply an orthogonal transformation matrix $A_{45} := \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Hence, the representation of the solution (5) results in

$$\begin{aligned}
\mathbf{u}_h^{\text{Cart}} &= A_{45} \cdot \mathbf{u}_h = \sum_{i=1}^{N} A_{45} \cdot (u_i \phi_i^x + v_i \phi_i^y) = \sum_{i=1}^{N} \frac{1}{\sqrt{2}} \begin{pmatrix} \varphi_i^x & \varphi_i^y \\ \varphi_i^x & -\varphi_i^y \end{pmatrix} \cdot \begin{pmatrix} u_i \\ v_i \end{pmatrix} \\
&= \sum_{i=1}^{N} \frac{1}{\sqrt{2}} \begin{pmatrix} \varphi_i^x & \varphi_i^y \\ \varphi_i^x & -\varphi_i^y \end{pmatrix} \cdot A_{45} \cdot \begin{pmatrix} u_i^{\text{Cart}} \\ v_i^{\text{Cart}} \end{pmatrix} \\
&= \sum_{i=1}^{N} \frac{1}{\sqrt{2}} \begin{pmatrix} \varphi_i^x & \varphi_i^y \\ \varphi_i^x & -\varphi_i^y \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} u_i^{\text{Cart}} + v_i^{\text{Cart}} \\ u_i^{\text{Cart}} - v_i^{\text{Cart}} \end{pmatrix} \\
&= \sum_{i=1}^{N} \frac{1}{2} \begin{pmatrix} \varphi_i^x (u_i^{\text{Cart}} + v_i^{\text{Cart}}) + \varphi_i^y (u_i^{\text{Cart}} - v_i^{\text{Cart}}) \\ \varphi_i^x (u_i^{\text{Cart}} + v_i^{\text{Cart}}) - \varphi_i^y (u_i^{\text{Cart}} - v_i^{\text{Cart}}) \end{pmatrix} \\
&= \sum_{i=1}^{N} \begin{pmatrix} u_i^{\text{Cart}} \frac{1}{2}(\varphi_i^x + \varphi_i^y) + v_i^{\text{Cart}} \frac{1}{2}(\varphi_i^x - \varphi_i^y) \\ u_i^{\text{Cart}} \frac{1}{2}(\varphi_i^x - \varphi_i^y) + v_i^{\text{Cart}} \frac{1}{2}(\varphi_i^x + \varphi_i^y) \end{pmatrix} \\
&= \sum_{i=1}^{N} \begin{pmatrix} \varphi_i^{x,1,\text{Cart}} & \varphi_i^{y,1,\text{Cart}} \\ \varphi_i^{x,2,\text{Cart}} & \varphi_i^{y,2,\text{Cart}} \end{pmatrix} \cdot \begin{pmatrix} u_i^{\text{Cart}} \\ v_i^{\text{Cart}} \end{pmatrix} = \sum_{i=1}^{N} \left(u_i^{\text{Cart}} \phi_i^{x,\text{Cart}} v_i^{\text{Cart}} \phi_i^{y,\text{Cart}} \right) . \qquad (8)
\end{aligned}$$



(a) (b)

Figure 3: Divergence-free ansatz functions in the standard Cartesian coordinate system (see Fig. 1(a)): (a) $\varphi_i^{x,1,\text{Cart}} = \varphi_i^{y,2,\text{Cart}} = \frac{1}{2}(\varphi_i^x + \varphi_i^y)$ and (b) $\varphi_i^{x,2,\text{Cart}} = \varphi_i^{y,1,\text{Cart}} = \frac{1}{2}(\varphi_i^x - \varphi_i^y)$.

We deduce from (8) that the Cartesian oriented ansatz functions have the form

$$\phi_i^{x,\text{Cart}} = \begin{pmatrix} \varphi_i^{x,1,\text{Cart}} \\ \varphi_i^{x,2,\text{Cart}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\varphi_i^x + \varphi_i^y) \\ \frac{1}{2}(\varphi_i^x - \varphi_i^y) \end{pmatrix} \qquad (9)$$

$$\phi_i^{y,\mathrm{Cart}} = \begin{pmatrix} \varphi_i^{y,1,\mathrm{Cart}} \\ \varphi_i^{y,2,\mathrm{Cart}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\varphi_i^x - \varphi_i^y) \\ \frac{1}{2}(\varphi_i^x + \varphi_i^y) \end{pmatrix} . \tag{10}$$

The representation (9)–(10) is visualised in Fig. 3 and is identic to the one derived in [1]. Note the coupling of both dimension components that appears in contrast to the 45 system: Horizontal velocities now depend on contributions from the vertical base functions and vice-versa.

3.3 Spatial Operators

In order to be able to implement the spatial discretisation in Peano, we need element matrices for all spatial operators M, M^T, D, and C in the new 45 system. The divergence operator M is straightforward as it appeared already in (4). It is clear that the new 45 system results in a reduced number of operations (3 additions instead of 7) on one cell compared to the standard Cartesian system. The same holds for the pressure gradient evaluation via M^T, used for the Gauß-Seidel iterations.

The diffusion operator D turns out to be the standard 5-point-stencil for each velocity component on the Cartesian grid also in the new 45 system. This is no surprise as it has the same form in the standard Cartesian coordinate system and, as a linear operator, is invariant to the transformation A_{45}.

The non-linear convection operator C, however, gets a considerably simpler representation in the new 45 system as no coupling between the different velocity components exists. In fact, the number of floating-point operations to be performed on one cells drops from 576 additions and 608 multiplications to 40 and 64, respectively, while keeping the important skew-symmetry property $C = -C^T$ (cf. [1, 5]).

3.4 Enhanced Divergence-Free Elements

In order to be able to capture additional flow over edges of a discrete cell, we enhance the divergence-free elements presented above by additional degrees of freedom $\tilde{u}_i, \tilde{v}_i$ on the centre of the faces. Figure 4(a) shows the additional DoF in red on the centres a, b, c, and

(a) (b)

Figure 4: Additional DoF $\tilde{u}_i, \tilde{v}_i$ on cell faces: (a) locations and directions; (b) example representation of $\tilde{v}_a$ (dashed red line) as hierarchical surplus with identic integral value as the parabola to approximate.

d of the faces of a cell. Note that for horizontal faces, only vertical velocities $(\tilde{v}_a, \tilde{v}_d)$ get flux contributions, whereas horizontal velocity DoF $(\tilde{u}_b, \tilde{u}_c)$ exist only on vertical edges. We intreprete $\tilde{u}_i, \tilde{v}_i$ as hierarchical surpluses to the velocity on the edge (cf. Fig. 4(b)).

On the one hand, the additional DoF $\tilde{u}_i, \tilde{v}_i$ come along with a bit more complicated discrete operators (element matrices) than those of Sec. 3.3. On the other hand, we now have the possibility to adjust those surpluses in such a manner that the exact flow over the edge is correct. This is, for example, useful for Dirichlet boundary conditions prescribing a parabolic velocity at the inlet of a channel scenario and may be advantageous for boundary layers, too, where parabolic profiles exist. Besides, since we always get pressure gradients at the same locations as velocity DoF, the stencil for the PPE changes: Instead of the usual 5-point-stencil rotated by 45 degrees, we now get a direct coupling of the "black" and the "white" cells via the faces, which avoids the well-known checkerboard mode in two dimensions.

4 NUMERICAL RESULTS

4.1 Runtime Comparisons of Convection and Diffusion

In this section, we compare the runtime results for one iteration over the whole adaptive grid to compute the diffusion and convection operators D and C for the standard Cartesian coordinate system and the new 45 system. The measurements for serial computations have been carried out on a Intel i7 architecture with 2.80 GHz per core and 8MB L2 cache using ubuntu 9.10 and gcc 4.4.1 with option `-O3`.

We can't expect the overall runtime for one iteration over the whole grid to compute D and C to scale directly like the mere number of operations listed in Sec. 3.3, due to several reasons: First, the compiler optimises the code in a non-straightforward manner, possibly using fused multiply-add commands. Second, cache effects may appear as we are using real production code runs on full grids with complete timestepping, etc. Third, besides the floating-point operations, general adaptivity computations etc. have to be performed (including integer arithmetics) which increase the overall computational workload.

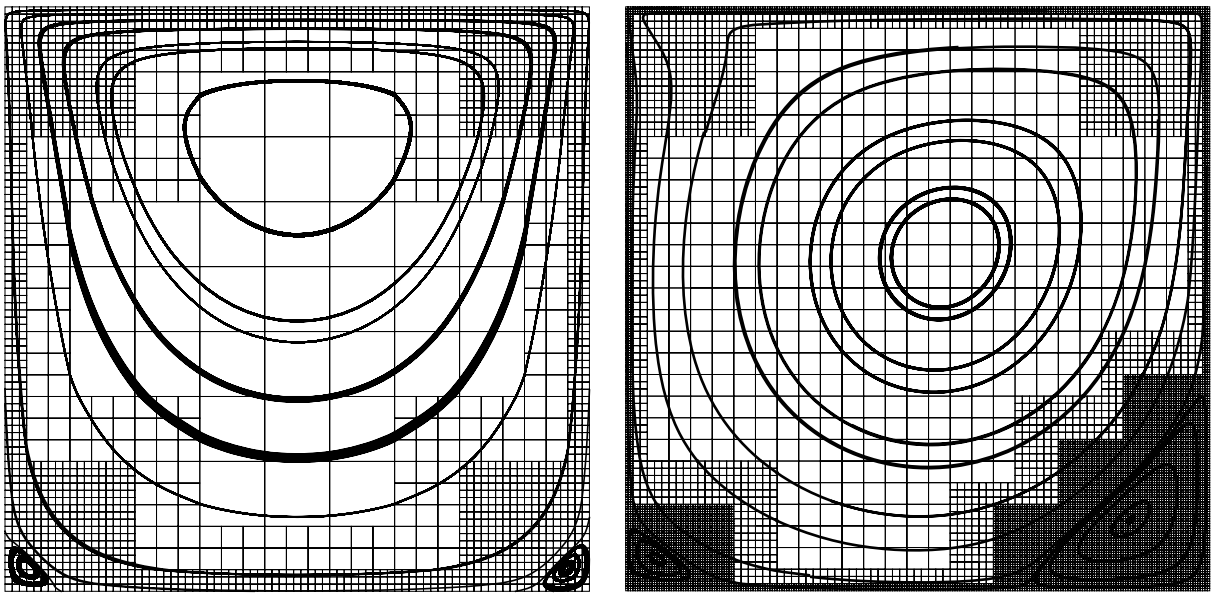
Run	level		# cells	CPU time in s		ratio
no.	min	max		Cart	45	45/Cart
1	6	8	29,406	0.06	0.05	0.83
2	7	8	57,261	0.10	0.08	0.80
3	8	8	351,469	0.63	0.52	0.82
4	9	9	3,163,829	5.45	4.17	0.76

Table 1: Comparison of runtimes for one iteration to compute convection and diffusion (D and C) in standard Cartesian coordinates (Cart) and in the 45 system of the cylinder benchmark (cf. Sec. 4.3) with different refinements. The last column shows the corresponding ratio.

Table 1 shows the results of the runtimes of one grid iteration for D and C for four different setups of the DFG cylinder scenario (see Sec. 4.3). The CPU times are indicated in seconds and have been averaged over several time steps to get reliable values. The first adaptive setup corresponds exactly to Fig. 6(a) (with a general spacetree refinement level of 6 and manual refinement up to level 8 around the cylinder, resulting in 29,406 cells), the other ones use an increased general refinement of one level each, up to level 9 (resulting in more than three million cells). The runtime saving for the diffusion and convection operator evaluation sums up to a considerable amount of about 17–24 per cent.

4.2 Driven Cavity

To demonstrate the qualitatively correct behaviour of the divergence-free elements in the 45 system, we computed two setups of the well-known lid-driven cavity with Reynolds numbers 1 and 1000, respectively. The streamlines visualised in Fig. 5 show the expected form. The adaptive Cartesian grids with maximum refinement level of 5 and 6 (cf. Fig. 5(a) and 5(b)) have been created by refinement boxes near the corners where the countervortices are located.



(a) (b)

Figure 5: Streamlines and Cartesian grid of the lid-driven cavity scenario with Re = 1 (a) and Re = 1000 (b). The maximum refinement level is 5 and 6, respectively, resulting in a total number of degrees of freedom of 5,713 and 32,641.

4.3 Laminar Flow around a Cylinder

The benchmark suite "laminar flow around a cylinder" [7] is used to validate also the quantitative correctness of the flow simulations using divergence-free elements. We computed the steady-state setup 2D-1 with Re = 20 and the time-dependent scenario 2D-2 with Re = 100. The adaptive grid in use, of which a zoomed part is visualised in Fig. 6(a), possesses a maximum spacetree refinement level of 8 resulting in 29,406 cells and 88,857 total DoF.

Table 2 shows a summary of the results for the two setups. We computed the force coefficient via the method of consistent forces (cf. [2, 3]). The (maximum) drag and lift coefficients c_d and c_l are in good accordance with the reference data of [7]. The Strouhal number of the time-dependent setup demonstrates the correct frequency of the vortex oscillations.

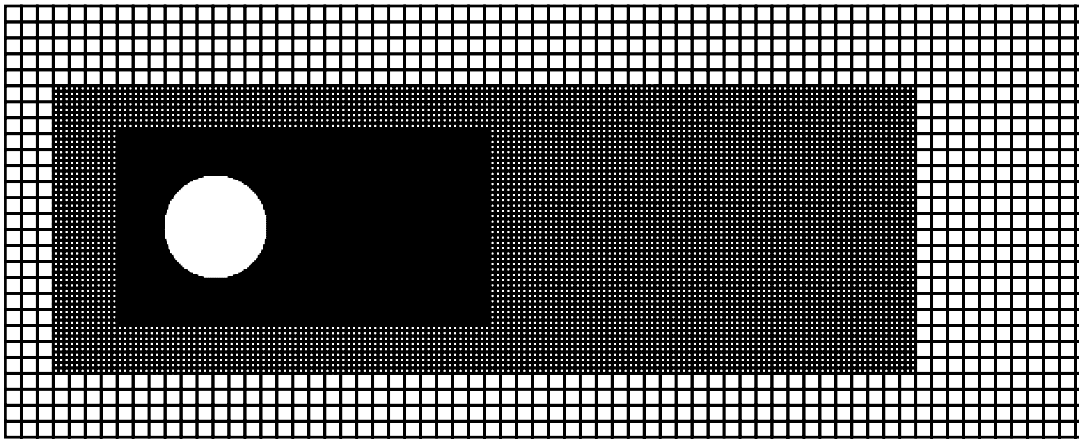
setup	# DoF	Re = 20		Re = 100		
		c_d	c_l	$c_{d,\max}$	$c_{l,\max}$	St
level 8 (box)	88,857	5.68	0.0151	3.225	0.94	0.299
reference data	–	5.58	0.0107	3.230	1.00	0.298

Table 2: Drag and lift coefficients c_d, c_l for the steady-state and time-dependent cylinder benchmarks [7] at Re = 20 and Re = 100. The Strouhal number St in the last column indicates the correct frequency of the oscillations for Re = 100.

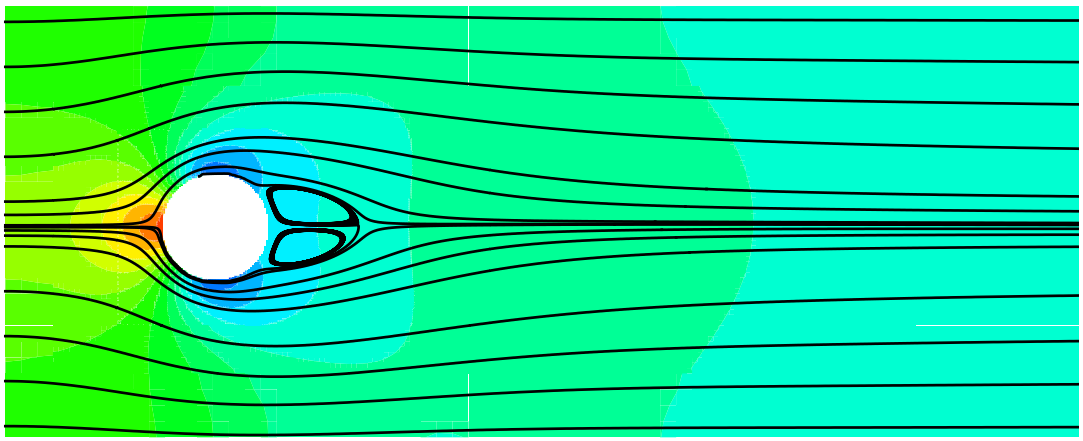
5 CONCLUSION AND OUTLOOK

In this contribution, we presented a new representation of two-dimensional divergence-free elements in a rotated and mirrored coordinate system that allows to easily develop and understand these elements. Considerable runtime savings up to 24% for evaluating the spatial operators, in particular the convection operator, can be achieved and have been demonstrated for a suite of meshes. Using the scenarios of the driven cavity and the flow around a cylinder with adaptively refined Cartesian meshes, we validated the correctness of the development and implementation of the new element type in the PDE framework Peano.

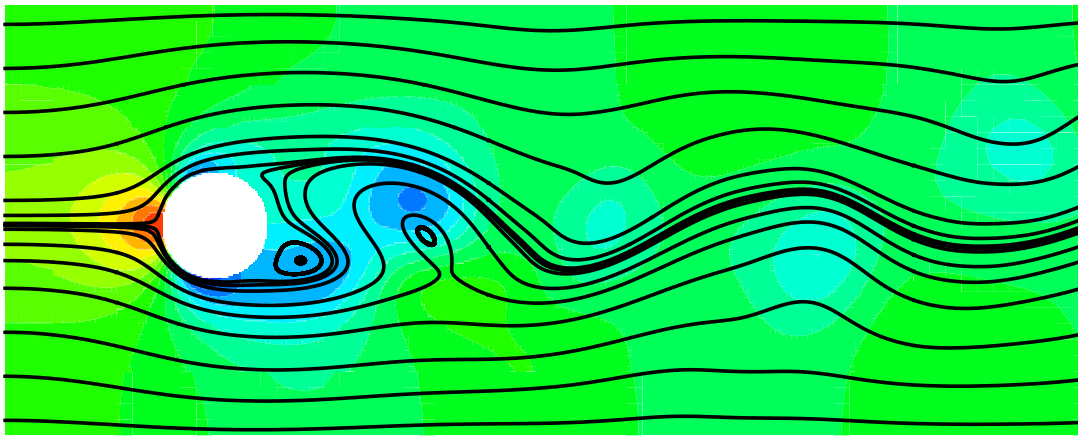
Current work is on the geometric multigrid solver for the pressure Poisson equation. Furthermore, we plan to extend the new type of elements to the challenging three-dimensional case.



(a)



(b)



(c)

Figure 6: Benchmark flow around a cylinder [7]: (a) adaptive grid with maximum level 8 and 29,406 fluid cells; (b) streamlines over pressure for setup 2D-1 (Re = 20) in the steady state; (c) streamlines over pressure for 2D-2 (Re = 100) at time 7.93.

REFERENCES

[1] C. Blanke. Kontinuitätserhaltende Finite-Element-Diskretisierung der Navier-Stokes-Gleichungen. Diploma Thesis, Technische Universität München, 2004.

[2] M. Brenk, H.-J. Bungartz, and T. Neckel. Cartesian discretisations for fluid-structure interaction – consistent forces. In P. Wesseling, E. Oñate, and J. Périaux, editors, *ECCOMAS CFD 2006, European Conference on Computational Fluid Dynamics*. TU Delft, 2006.

[3] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*. John Wiley & Sons, 1998.

[4] F. Günther, M. Mehl, M. Pögl, and C. Zenger. A cache-aware algorithm for PDEs on hierarchical data structures based on space-filling curves. *SIAM Journal on Scientific Computing*, 28(5):1634–1650, 2006.

[5] T. Neckel. *The PDE Framework Peano: An Environment for Efficient Flow Simulations*. Verlag Dr. Hut, 2009.

[6] H. Sagan. *Space-filling curves*. Springer-Verlag, New York, 1994.

[7] S. Turek and M. Schäfer. Benchmark computations of laminar flow around a cylinder. In E. H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, number 52 in NNFM. Vieweg, 1996.

[8] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent channel flow. In M. Breuer, F. Durst, and Ch. Zenger, editors, *High Performance Scientific and Engineering Computing (2001)*, pages 107–114. Springer-Verlag, 2001.

[9] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent channel flow. *J. Comp. Phys.*, 187:343–368, 2003.

[10] T. Weinzierl. *A Framework for Parallel PDE Solvers on Multiscale Adaptive Cartesian Grids*. Verlag Dr. Hut, 2009.