Numerical analysis and a-posteriori error control for a new nonconforming linear finite element on quadrilaterals

Matthias Grajewski Jaroslav Hron Stefan Turek*

August 6, 2003

Abstract

Starting with a short introduction of the new nonconforming linear quadrilateral $\tilde{P}_1$-finite element which has been recently proposed by Park ([13, 14]), we examine in detail the numerical behaviour of this element with special emphasis on the treatment of Dirichlet boundary conditions, efficient matrix assembly, solver aspects and the use as Stokes element in CFD. Furthermore, we compare the numerical characteristics of $\tilde{P}_1$ with other low order finite elements. Moreover, we derive a dual weighted residual-based a-posteriori error estimation procedure in the sense of Becker and Rannacher (c.f. [2]) for $\tilde{P}_1$. Several test examples show the efficiency and reliability of the proposed method for elliptic 2nd order problems.

1 Introduction

Nonconforming finite element methods are quite popular in the numerical treatment of partial differential equations (PDEs), in particular for incompressible flow problems due to their (often) excellent stability properties w.r.t. the 'inf-sup-condition' and anisotropic mesh deformations. Moreover, together with discrete projection techniques or Pressure-Schur-Complement methods they can provide very efficient FEM solvers for nonstationary problems [20]. And, finally, their mostly edge-oriented degrees of freedom lead to very compact data structures which have special advantages for parallel high-performance computations [6]. Altogether, they are quite natural candidates to combine modern error control mechanisms and concepts for adaptivity in space and time with optimized high performance computing techniques to simulate realistic problems, in particular for *Computational Fluid Dynamics* (CFD).

In the case of the incompressible Stokes- or Navier-Stokes equation the element pair (E_1, E_2), where E_1 discretizes the velocities and E_2 the pressure, is sometimes referred as Stokes-element. Examples of popular Stokes-element are the pairs (Q_1, Q_0) (conforming bilinear velocity, piecewise constant pressure), (Q_1, Q_1) (conforming bilinear velocity and pressure), (Q_2, P_1) (conforming biquadratic velocity, discontinuos linear pressure), (P_1^{nc}, P_0) (nonconforming linear velocity, piecewise constant pressure), $(\tilde{Q}_1, Q_0)$ (nonconforming rotated bilinear velocity, piecewise constant pressure) and many more. Besides the development of faster solvers for tackling real-world related problems with low

*Institute of Applied Mathematics, University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany, ture@featflow.de

computational costs, there is still an ongoing research for efficient and robust discretisations for such saddle point problems. The combination of the element with piecewise constant pressure which will be investigated in this paper turns out not to be BB-stable similar to the very popular (Q_1, Q_0) Stokes-element, but because of its simple structure (it is in fact the simplest nonconforming quadrilateral element known besides a piecewise constant approach) it plays the role of a new prototypial nonconforming finite element and merits therefore attention.

Let Ω be a bounded 2-dimensional domain with polygonal boundary (we restrict to 2D by technical reasons only), $\mathcal{P}_k$ denote the polynomial space with maximum total degree k and $\mathcal{Q}_k$ the polynomial space with maximum degree k in each unknown. In his PhD-thesis [13] and in [14] C. Park introduced a linear quadrilateral nonconforming finite element which will be called $\tilde{P}_1$ in this article. Let $\mathbb{T}_h$ denote a conforming mesh over Ω, which consists of convex quadrilaterals, and $\partial\mathbb{T}_h$ the set of all edges of $\mathbb{T}_h$. The construction of the element and the corresponding finite element space $\tilde{P}_1(\mathbb{T}_h)$ relies on the following simple facts. By elementary (vector) calculus one proves for an arbitrary convex quadrilateral T with vertices $v_1, \ldots, v_4$ and edge midpoints $m_1, \ldots, m_4$ that these midpoints form a parallelogram. By exploiting this fact, one can show that

$$\forall\, u \in \mathcal{P}_1(T) : u(m_1) + u(m_3) = u(m_2) + u(m_4) \tag{1}$$

and, vice versa,

$$\forall\, u_1, \ldots u_4 \text{ with } u_1 + u_3 = u_2 + u_4 \;\; \exists! u \in \mathcal{P}_1(T) :\; u_i = u(m_i). \tag{2}$$

Here we assume the vertices and midpoints of the edges to be orderd as in Figure 1. This allows the following definition:

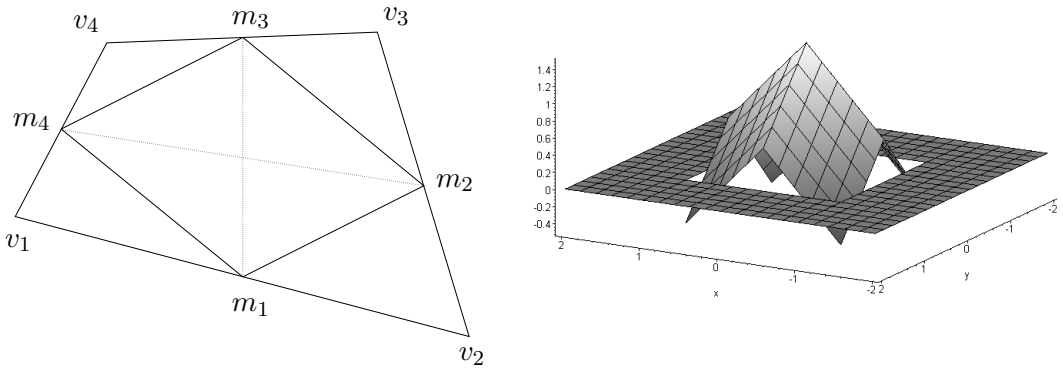

Figure 1: Edge midpoints of an arbitrary quadrilateral form a parallelogram; $\tilde{P}_1$ basis function Φ_j at node v_j

Definition 1.1 (Park). 1) *Let v_j be an arbitrary vertex in $\mathbb{T}_h$ and $\mathcal{M}(j) := \{i \in \mathbb{N} \,\big|\, \exists \Gamma \in \partial\mathbb{T}_h : m_i \in \Gamma \wedge v_j \in \Gamma\}$ "the neighbourhood of v_j". Then we can define Φ_j by $\Phi_j|_T \in \mathcal{P}_1(T)\, \forall\, T \in \mathbb{T}_h$ and*

$$\Phi_j(m_i) := \begin{cases} 1 & , \quad i \in \mathcal{M}(j) \\ 0 & , \quad else \end{cases} \tag{3}$$

2) *The finite element space* $\tilde{P}_1(\mathbb{T}_h)$ *is defined by*

$$\tilde{P}_1(\mathbb{T}_h) := \left\{ \varphi : \Omega \to \mathbb{R} \;\middle|\; \varphi|_T \in \mathcal{P}_1(T) \,\forall\, T \in \mathbb{T}_h \right\}. \tag{4}$$

On an edge $\Gamma \in \partial\mathbb{T}_h$ *the functional*

$$F_\Gamma(\varphi) = \varphi(m_\Gamma) \tag{5}$$

is well-defined, such that we can prescribe continuity in the midpoints m_Γ*.*

For this finite element space there holds the typical FEM approximation result:

Theorem 1.2 (Park). 1) *Let* $\Omega \subset \mathbb{R}^2$ *be simply connected convex domain with piecewise polygonal boundary. The triangulation* $\mathbb{T}_h$ *shall contain* N *vertices. Furthermore, to each inner edge, there shall belong at least one inner vertex. Then,*

$$\dim \tilde{P}_1(\mathbb{T}_h) = N - 1. \tag{6}$$

For arbitrary $0 < \hat{\jmath} \leq N$*, the set* $\{\Phi_j \mid j \in \{1 \ldots, N\} \setminus \{\hat{\jmath}\}\}$ *forms a basis of* $\tilde{P}_1(\mathbb{T}_h)$*.*
2) *Let* $f \in L_2$*,* $g \in H^{\frac{1}{2}}(\partial\Omega)$ *and* u *the solution of the Robin boundary value problem*

$$a(u,\varphi) := (\nabla u, \nabla\varphi)_\Omega + (cu,\varphi)_\Omega + (u,\varphi)_{\partial\Omega} = (f,\varphi)_\Omega + (g,\varphi)_{\partial\Omega} \quad \forall\, \varphi \in H^1(\Omega). \tag{7}$$

Then, there is a constant C *not depending on* h*, such that*

$$||u - u_h||_0 + h||u - u_h||_{1,h} \leq Ch^2||u||_2 \tag{8}$$

holds for the discrete solution u_h*.*

For a tensor product mesh, a typical basis function Φ_j is displayed in Figure 1 (right). It is important to notice that the finite element space $\tilde{P}_1(\mathbb{T}_h)$ may *not* be obtained by parametric transformation from some reference element, although it is possible to define such a finite element space. As in the case of the rotated bilinear $\tilde{Q}_1$-approach (see [16]) there arise problems for deformed meshes (compare also [1]). The transformation procedure and corresponding numerical results for various computational domains and meshes will be adressed in detail in the subsequent chapter.
For comparison of the approximation properties of $\tilde{P}_1$ with the approximation properties of the Rannacher-Turek-element $\tilde{Q}_1$ and the conforming Q_1-element, we compute on the coarse grid shown in Figure 2 the test problem $-\triangle u = f, u|_{\partial\Omega} = g$, where the right hand side f and the Dirichlet boundary condition g are chosen such that $u = x(x-1)(1-y)y^2\sin(x+2y)$ holds. The approximation results for this problem are collected in Table 2. Here, NEL denotes the total number of elements. Furthermore, we compute the test problem $-\triangle u = 10, u|_{\partial\Omega} = 0$ on the same grid. The exact solution is now unknown, therefore we computed a reference solution by using conforming biquadratic finite elements. To measure the error of the different discretisations, we use the l_2-norm defined by

$$||u - u_h||_{l_2} := \left(\frac{1}{NVT} \sum_{i=1}^{N} |u(v_i) - u_h(v_i)|^2 \right)^{1/2},$$

where v_i denotes a vertex and NVT the total number of vertices on the given grid. In the case of the nonconforming elements $\tilde{Q}_1$ and $\tilde{P}_1$ which are discontinuous in the vertices, first we computed a L_2-projection into the conforming space Q_1 and then proceeded as before.
The results (Tables 1 and 2) show that all three elements lead to very similar approximation results for this specific problem which obviously does not provide H^2-regularity in the second case.

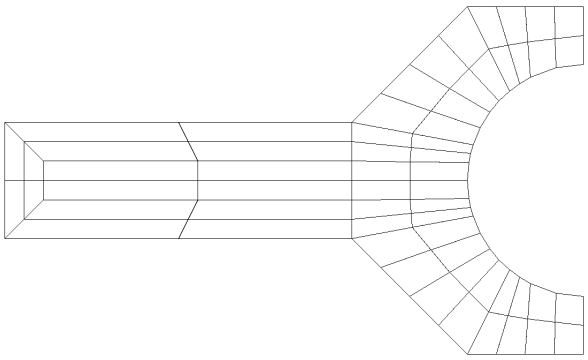

Figure 2: Computational domain with coarse mesh for the comparison of approximation results

NEL	Q_1	red.	$\tilde{Q}_1$	red.	$\tilde{P}_1$	red.
60	$1,48 \cdot 10^{-1}$	-	$3.28 \cdot 10^{-1}$	-	$4.30 \cdot 10^{-1}$	-
240	$3.81 \cdot 10^{-2}$	3.88	$3.81 \cdot 10^{-2}$	8.61	$4.68 \cdot 10^{-2}$	9.19
960	$9.69 \cdot 10^{-3}$	3.93	$8.95 \cdot 10^{-3}$	4.26	$8.83 \cdot 10^{-3}$	5.30
3840	$2.44 \cdot 10^{-3}$	3.97	$2.24 \cdot 10^{-3}$	4.00	$2.04 \cdot 10^{-3}$	4.33
15360	$6.11 \cdot 10^{-4}$	3.99	$5.60 \cdot 10^{-4}$	4.00	$4.99 \cdot 10^{-4}$	4.09

Table 1: Comparison of L_2-errors and reduction factors of the errors computed with Q_1, $\tilde{Q}_1$ and $\tilde{P}_1$

NEL	Q_1	red.	$\tilde{Q}_1$	red.	$\tilde{P}_1$	red.
60	$4.18 \cdot 10^{-2}$	-	$6.24 \cdot 10^{-2}$	-	$4.07 \cdot 10^{-2}$	-
240	$1.62 \cdot 10^{-2}$	2.58	$2.58 \cdot 10^{-2}$	2.42	$1.96 \cdot 10^{-2}$	2.08
960	$5.69 \cdot 10^{-3}$	2.85	$8.10 \cdot 10^{-3}$	3.19	$6.04 \cdot 10^{-3}$	3.25
3840	$1.96 \cdot 10^{-3}$	2.90	$2.59 \cdot 10^{-3}$	3.13	$1.81 \cdot 10^{-3}$	3.34
15360	$6.57 \cdot 10^{-4}$	2.98	$8.31 \cdot 10^{-4}$	3.12	$5.40 \cdot 10^{-4}$	3.35

Table 2: Comparison of l_2-errors and reduction factors of the errors computed with Q_1, $\tilde{Q}_1$ and $\tilde{P}_1$

2 Numerical aspects of the $\tilde{P}_1$-approach

Many common finite element spaces are spanned by basis functions φ_j which usually fulfill

$$F_i(\varphi_j) = \delta_{ij}$$

for appropriately given nodal functionals $F_i(\cdot)$. For instance, in the case of the midpoint orientated nonconforming Rannacher-Turek-element $\tilde{Q}_1$ with meanvalues over edges (Γ_i edges of $\mathbb{T}_h$),

$$F_i(\varphi_j) = \frac{1}{|\Gamma_i|} \int_{\Gamma_i} \varphi_j \, ds = \delta_{ij}$$

or $F_i(\varphi_j) = \varphi_j(m_{\Gamma_i})$, with m_{Γ_i} midpoint of Γ_i, in the case of the midpoint-orientated version (see [16]). Therefore, the implementation of Dirichlet boundary conditions can be typically reduced to finding those basis functions which are related to a boundary point or a boundary edge, respectively: The coefficient of such a basis function is set to the value of the corresponding nodal functional F. In contrast to this, in the case of $\tilde{P}_1$ there holds a non-local condition for determining the coefficients, namely

$$u(m_j) = u_j + u_{j+1} \tag{9}$$

where m_j is the midpoint of the edge $\overline{v_j v_{j+1}}$ and u_j, u_{j+1} are the coefficients of the basis functions corresponding to v_j and v_{j+1}. Note that although the continuity conditions refer to midpoints, the finite element space is organized vertex-oriented. In the case of simply connected domains, Park proposes due to Theorem 1.2 to fix one arbitrary basis function belonging to any boundary vertex and then to loop over all boundary edges successively setting the coefficients of the boundary basis functions by using (9). For arbitrary domains with multiple boundary components, it seems to be straightforward to repeat this process for every boundary component separately. By this, for a n-fold connected domain, totally n degrees of freedom will be fixed. This method will be regarded as *explicit boundary treatment*. Unfortunately, we observe a lack of convergence for this explicit boundary treatment as shown by the computation of Test Problem 2.1 on the grid in Figure 3 on the right (the results are collected in Table 3). Here and in similar situations we provide the relative L_2- and H_1-errors. The results lead to the conclusion that the explicit boundary treatment is unsuitable for multiply connected domains, which is very similar to the case of divergence-free FEM approaches which also require special techniques for determining the correct values for the streamfunction-like part in the case of more than one boundary component (see [19]).

Test Problem 2.1. *For $\Omega = [-3,3]^2 \backslash [-1,1]^2$, the right hand side f is chosen such that*

$$-\triangle u = f$$

holds with the exact solution

$$u(x,y) = x(x-1)(1-y)y^2 \sin(x+2y).$$

Dirichlet boundary conditions are set accordingly to $u|_{\partial\Omega}$.

NEL	$\|\|u-u_h\|\|_0$	$\|\|\nabla_h(u-u_h)\|\|_0$
128	$0,438$	$1,28$
512	$0,167$	$0,973$
2048	$7,69 \cdot 10^{-2}$	$0,905$
8192	$3,74 \cdot 10^{-2}$	$0,888$
32768	$1,85 \cdot 10^{-2}$	$0,884$
131072	$9,20 \cdot 10^{-3}$	$0,883$

Table 3: Test Problem 2.1 with the grid in Figure 3 (right), explicit boundary treatment

Careful examination of the error $u - u_h$ leads to the conjecture that the lack of convergence seen is caused by high-frequency oscillations. Therefore, the subsequent proposed implicit boundary treatment is motivated by the following Lemma.

Lemma 2.2. *Let $\mathbb{T}_h$ be a grid which shall be connected in the sense that the condition $\forall\, T', T'' \in \mathbb{T}_h\ \exists \{T_i\}_{i=1}^n : T_i \cap T_{i+1} \in \partial\mathbb{T}_h, T_1 = T', T_n = T''$ is fulfilled. Moreover, let be $T \in \mathbb{T}_h$ and $c_1, \ldots, c_4$ the coefficients of the solution vector corresponding to the vertices of T. The solution vector shall represent the constant solution α in $\tilde{P}_1(\mathbb{T}_h)$ and the nodes shall be ordered like in Figure 1. If for such an element T holds*

$$c_1 + c_3 = c_2 + c_4, \tag{10}$$

*then formula (10) holds for all $T \in \mathbb{T}_h$ und α is represented "*oscillationfree*", i.e. all coefficients have the same value $\alpha/2$.*

Proof: Let $T \in \mathbb{T}_h$ be an arbitrary convex quadrilateral and for the coefficients c_i of the representation (10) of α on T shall hold. This condition and the four conditions from (9) lead to the linear system

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} \alpha \\ \alpha \\ \alpha \\ \alpha \\ 0 \end{pmatrix}. \tag{11}$$

Its unique solution is $c_i = \alpha/2$. For a quadrilateral T' adjacent to T with nodes $v_1', \ldots, v_4'$ shall hold without restriction that $\overline{v_1 v_2} = T \cap T'$. Therefore for the corresponding coefficients c_1' and c_2' we have $c_1' = c_1 = \alpha/2 = c_2 = c_2'$. Because of $c_2' + c_3' = \alpha = c_1' + c_4'$ according to (9) we conclude $c_3' = c_4' = \alpha/2$. Due to the connectivity of the grid the statement follows by marching over all elements starting from T. □

Remark 2.3. *In the case of orthogonal tensor product meshes, even linear functions are represented oscillationfree by condition (10), i.e. the points $P_i := (v_{i,x}, v_{i,y}, c_i)$, where c_i denotes the coefficient associated to the vertex v_i, belong to an affine plane. This follows by a similar but more tedious argument like in the proof of Lemma 2.2. For arbitrary meshes linear functions are not represented oscillationfree by applying condition (10). A counterexample is the quadrilateral with vertices $v_1 = (0,0), v_2 = (1,0), v_3 = (1,1)$ and $v_4 = (0.5, 1)$. For the representation of $u = x$ the linear system arising from Lemma 2.2 leads to the coefficient vector $(1/16, 7/16, 9/16, 3/16)$. It is easy to verify that the point $P_4 = (v_{4,x}, v_{4,y}, 3/16)$ does not belong to the plane spanned by P_1, P_2 and P_3.*

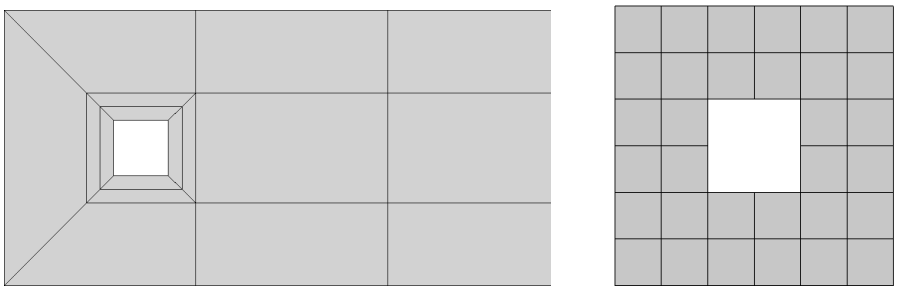

Figure 3: Two computational domains with coarse grids

Now we can define an implicit treatment of Dirichlet boundary conditions. Instead of fixing one basis function on each boundary component, i.e. replacing in the stiffness matrix the first row per boundary component by the corresponding row of the unit matrix, we first apply eq. (10) to these coefficients and then we loop over all following boundary vertices as before. As the starting point of this loop is not explicitly prescribed now, we define by this an implicit treatment of Dirichlet boundary conditions which couples the coefficients of the boundary components with the (unknown) solution in the interior. Alternatively, we can define a semi-implicit treatment of boundary conditions by applying the explicit treatment to the first boundary component only and the implicit treatment to all other boundary components. For a square in a channel, $\Omega = ([0,2] \times [0,1]) \backslash [0.4, 0.6]^2$ (Figure 3, left) we compute $\triangle u = 0, u|_{\partial\Omega} = x + 1$. The exact solution $u = x + 1$ should be represented exactly in the space $\tilde{P}_1(\mathbb{T}_h)$. Therefore any error occuring can be stated as consistency error coming from the treatment of Dirichlet boundary conditions. In the case of the semi-implicit and explicit boundary treatment, the coefficient belonging to the vertex located in the down left corner is set to 0. The consistency error $u - u_h$, which is listed in Table 4, seems to be of third order in the case of implicit boundary and of first order only in the case of semi-implicit and explicit boundary treatment. In fact, the explicit boundary treatment works fine, too, if the coefficients have been fixed to the "correct" values. Unfortunately, these values are not known a priori. We assume that the high order consistency error (it seems to be of third order) of the implicit boundary treatment is related to the fact that formula (10) does not guarantee the oscillationfree representation of linear functions on non-tensorproduct grids, as this error is not visible on tensorproduct grids. Similar results are observed for domains with three or even more boundary components. These results might disqualify the explicit and semi-implicit treatments for implementing Dirichlet boundary conditions on multiply connected domains unless one finds a way to predict the "right" coefficients.

In Figure 4 we show the structure of the stiffness matrices obtained by the explicit and implicit treatment in comparison with the matrix structure coming from a discretisation with $\tilde{Q}_1$ and/or Q_1 after suitable resorting (see also [21]): The grey blocks symbolize the matrix parts coming from inner nodes. In the case of the implicit treatment the resorting is chosen such that the coupling between inner nodes and boundary nodes takes place in the first row. As it can be seen, the (necessary) implicit treatment leads to the loss of diagonal dominance and symmetry which is quite non-standard for FEM discretizations of self-adjoint Poisson-like problems with Dirichlet boundary conditions.

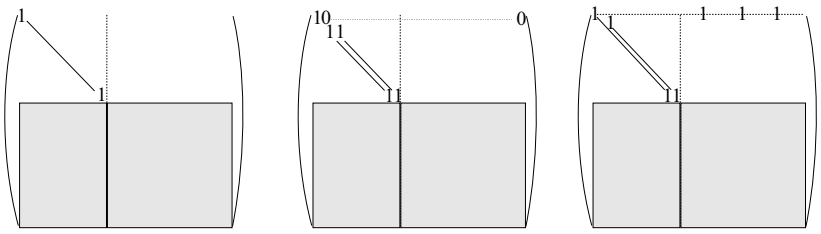

Figure 4: Matrix structure after applying Dirichlet boundary conditions in the case of different FEM discretisations: standard treatment via elimination (in rows only) in Q_1 and $\tilde{Q}_1$ (left), $\tilde{P}_1$ with explicit boundary treatment (middle) and $\tilde{P}_1$ with implicit boundary treatment (right)

NEL	implicit	red.	semi-implicit	red.	explicit	red.
68	$1.96 \cdot 10^{-4}$	-	$1,02 \cdot 10^{-2}$	-	$3.74 \cdot 10^{-3}$	-
272	$2.30 \cdot 10^{-5}$	8.52	$4.86 \cdot 10^{-3}$	2.10	$1.86 \cdot 10^{-3}$	2.01
1088	$2.79 \cdot 10^{-6}$	8.24	$2.38 \cdot 10^{-3}$	2.04	$9.29 \cdot 10^{-4}$	2.00
4352	$3.44 \cdot 10^{-7}$	8.11	$1.17 \cdot 10^{-3}$	2.03	$4.64 \cdot 10^{-4}$	2.00
17408	$4.28 \cdot 10^{-8}$	8.04	$5.84 \cdot 10^{-4}$	2.00	$2.32 \cdot 10^{-4}$	2.00
69632	$5.32 \cdot 10^{-9}$	8.05	$2.91 \cdot 10^{-4}$	2.01	$1.16 \cdot 10^{-4}$	2.00

Table 4: L_2-error and reduction rates for $u = x + 1$ on the grid in Figure 3 (right), comparison of the proposed treatments (the solution should be identically reproduced such that the resulting error should be 0)

Unfortunately, the implicit treatment of Dirichlet boundary conditions has severe impact on the behaviour of iterative solvers, too. The common combination BiCGStab-solver and SSOR-preconditioner faces difficulties when applying the implicit or semi-implicit treatment. This remains true when using ILU(0) (see [5]) as preconditioner. To demonstrate these facts, we compute Test Problem 2.1 on the grid in Figure 3 (right). The iteration is stopped when the norm of the relative residual falls below 10^{-11}, where 0 is the starting point of the iteration. The results are collected in Table 5. In this example, it is crucial to resort the complete linear system according to Cuthill-McKee or similar, as this can diminish the number of iterations by a factor up to 20 (!). By using GMRes instead of BiCGStab, one gets similar results to the ones presented here. However, in arbitrary situations, grid resorting does not always work as fine as it does in the presented example. To demonstrate this, we computed $-\triangle u = f$ with the same u as above, but on the "square in the channel" (Figure 3, left). Although resorting seems to improve the behaviour of the solver, these results (see Table 6) show that with simple ILU(0) as preconditioner the linear system cannot be solved on the finest level. To obtain the solution, one has at least to use ILU(1) instead where a higher fill-in is allowed. The preconditioner ILU(1) has been taken from the solver package SPLIB [5]. At the current state the problem of solving the linear systems which come from a discretisation with $\tilde{P}_1$ on multiply connected domains remains a future challenge, particularly for multigrid with corresponding grid transfer.

	explicit boundary treatment			implicit boundary treatment		
NEL	SSOR	ILU(0)	ILU(0), CMcK	SSOR	ILU(0)	ILU(0),CMcK
128	9	9	7	46	41	11
512	20	20	14	101	93	17
2048	44	44	28	227	203	30
8192	79	82	49	646	554	59
32768	154	154	93	1795	1522	103
131072	288	282	174	6206	4706	192

Table 5: Iteration numbers in the case of Test Problem 2.1 for solving with BiCGStab with SSOR preconditioner ($\omega = 1$), with ILU(0) preconditioner (standard FEM 2-level-ordering following [18]) and with ILU(0) preconditioner with resorting according to Cuthill-McKee; all computations on the grid in Figure 3 (right)

NEL	ILU(0)	ILU(0) + CMcK	ILU(1)	ILU(1) + CMcK
68	28	15	12	10
272	63	24	24	17
1088	171	46	48	26
4352	539	110	100	49
17408	2743	1060	292	97
69632	>10,000	> 10,000	1149	236

Table 6: Computation on the grid in Figure 3 (left) with standard FEM 2-level-ordering and resorting according to Cuthill-McKee; BiCGStab with ILU(0) and ILU(1) preconditioning

However, as a consequence of its simplicity, the matrix assembly performs outstandingly well for $\tilde{P}_1$. Park developed in [13] a nonparametric reference scheme which is as fast as a parametric transformation. Here, the bilinear transformation is divided up into two parts and intermediate coordinates are introduced which replace the reference coordinates of a parametric transformation (compare Figure 5). The element basis functions are explicitly given in these intermediate coordinates, therefore there is no need to compute explicitly local transformations like this is the case for many other nonconforming nonparametric finite elements (c.f. [19]). Note that *there is no way to avoid a nonparametric transformation scheme for nonconforming quadrilateral elements without loss of convergence on arbitrary meshes* (compare [1]). Unfortunately, the reference scheme described by Park is implicitly based upon the linearity of $\tilde{P}_1$, so that it is not suitable for other nonconforming quadrilateral elements like the Rannacher-Turek-element $\tilde{Q}_1$ which is spanned by $1, x, y, x^2 - y^2$. Experiments show that applying this reference scheme leads to comparable failure as using parametric transformation.

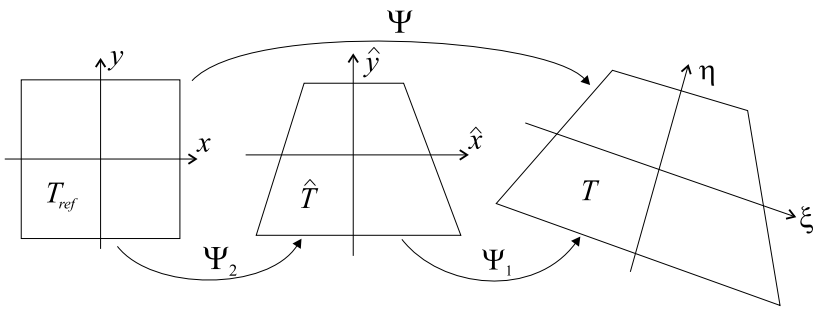

Figure 5: Nonparametric reference scheme for $\tilde{P}_1$

In the case of a PDE with constant coefficients, simple integration rules are sufficient to integrate the matrix entries. In the Laplacian case, the terms $\nabla\Phi_i \cdot \nabla\Phi_j$ are constant, so that the 1-point Gauss-rule will integrate exactly. For comparison, in the case of $\tilde{Q}_1$ one needs a third order rule for exact integration, e.g. the 2×2-Gauss-rule. Numerical experiments outline the difference in matrix assembly times between $\tilde{P}_1$ and $\tilde{Q}_1$ (compare Table 2). These results are obtained by assembling the matrices on the DFG-benchmark grid "flow around a cylinder" which is displayed in Figure 6 (for the DFG-benchmark, see [15]). For our tests, we used routines from the FEM-library FEAT (see [4]). The computations were performed on a COMPAQ ALPHA, 667 MHz. The measured time does not contain the time in which the pointer structure of the matrix is computed. The results show that the matrix can be assembled up to two times faster for $\tilde{P}_1$ than for $\tilde{Q}_1$. The optimal efficiency with respect to matrix assembly is one of the main advantages of $\tilde{P}_1$ compared with other approaches.

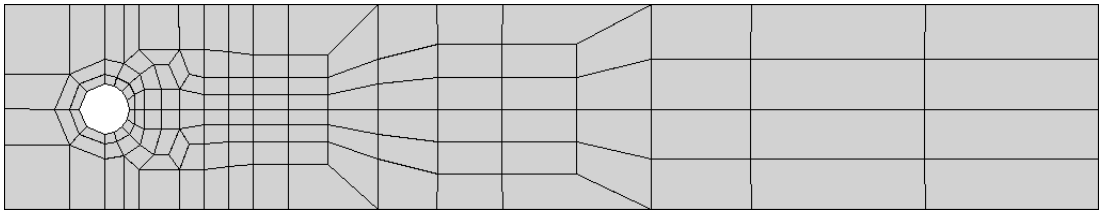

Figure 6: DFG-benchmark-grid "flow around a cylinder"

NEL	$\tilde{P}_1$ par., 1×1	$\tilde{P}_1$ par., 2×2	$\tilde{P}_1, 1\times1$	$\tilde{P}_1, 2\times2$	$\tilde{Q}_1$ par., 2×2	$\tilde{Q}_1, 2\times2$
133,120	1.5 s	2.3 s	1.6 s	2.4 s	2.4 s	3.1s
532,480	6.0 s	9.4 s	6.5 s	9.6 s	9.7 s	12.3s
2,129,920	24.7 s	37.4 s	25.5 s	38.6 s	38.9 s	48.2 s

Table 7: Computational times (in seconds) for assembling the linear system with the grid in Figure 6 on several refinement levels, see the resulting number of elements 'NEL' (the abbreviation 'par.' stands for parametric transformation)

The choice of an appropriate quadrature rule for the matrix assembly is crucial for the quality of the solution. To show this, we compute $\triangle u = f$, $u|_{\partial\Omega}$ on the unit square with a tensorproduct grid. The functions f and g are chosen such that there holds $u(x,y) = x(x-1)(1-y)y^2\sin(x+2y)$ as above. Note that on this grid *there is no difference between parametric and non-parametric transformation.* We compute this problem with $\tilde{P}_1$, the Rannacher-Turek element $\tilde{Q}_1$ and an extension $\breve{Q}_1$ proposed by Douglas et al. [7]. In this article, they replaced the ('rotated bilinear') function $x^2 - y^2$ in the Rannacher-Turek element by the fourth order polynomial $5x^4 - 3x^2$ to overcome problems with the midpoint-oriented version of the Rannacher-Turek-element on distorted meshes. The entries of the stiffness matrix are computed by the 1×1-Gauss and the 2×2-Gauss rule. The entries of the right hand side vector are always computed with the 4×4-Gauss rule. The results in Table 8 show that except in the case of $\tilde{P}_1$ the use of the 2×2-Gauss rule at least is necessary. As proven in [7], also $\breve{Q}_1$ shows full (quadratic) convergence when the stiffness matrix is computed with the 2×2-Gauss rule, but the L_2-errors exceed the corresponding errors from $\tilde{P}_1$ and $\tilde{Q}_1$ by a factor up to 20 (!). Experiments indicate that when using $\breve{Q}_1$ one has to apply at least the 3×3-Gauss-rule to get comparable results.

NEL	$\tilde{P}_1, 2\times 2$	$\tilde{P}_1, 1\times 1$	$\tilde{Q}_1, 2\times 2$	$\tilde{Q}_1, 1\times 1$	$\breve{Q}_1, 2\times 2$	$\breve{Q}_1, 1\times 1$
64	$2,22\cdot 10^{-2}$	$2,22\cdot 10^{-2}$	$1,96\cdot 10^{-2}$	$0,31\cdot 10^{0}$	$1,52\cdot 10^{-1}$	$2,41\cdot 10^{-1}$
256	$5,53\cdot 10^{-3}$	$5,53\cdot 10^{-3}$	$4,92\cdot 10^{-3}$	$0,31\cdot 10^{0}$	$7,24\cdot 10^{-2}$	$2,40\cdot 10^{-1}$
1024	$1,38\cdot 10^{-3}$	$1,38\cdot 10^{-3}$	$1,23\cdot 10^{-3}$	$0,31\cdot 10^{0}$	$2,38\cdot 10^{-2}$	$2,40\cdot 10^{-1}$
4096	$3,46\cdot 10^{-4}$	$3,46\cdot 10^{-4}$	$3,08\cdot 10^{-4}$	$0,31\cdot 10^{0}$	$6,53\cdot 10^{-3}$	$2,40\cdot 10^{-1}$
16384	$8,64\cdot 10^{-5}$	$8,64\cdot 10^{-5}$	$7,70\cdot 10^{-5}$	$0,31\cdot 10^{0}$	$1,68\cdot 10^{-3}$	$2,40\cdot 10^{-1}$
65536	$2,16\cdot 10^{-5}$	$2,16\cdot 10^{-5}$	$1,92\cdot 10^{-5}$	$0,31\cdot 10^{0}$	$4,22\cdot 10^{-4}$	$2,40\cdot 10^{-1}$

Table 8: L_2-error of computations on a tensorproduct grid in the case of applying the 1×1-Gauss and 2×2-Gauss rule for the integration of the entries of the stiffness matrix

Furthermore, we consider the application of $\tilde{P}_1$ to saddle point problems as the incompressible Navier-Stokes equation. Unfortunately, the combination $(\tilde{P}_1, Q_0)$ is *not* a stable Stokes-element which was already mentioned by Park in [14]. Here we give a simple example of a configuration where oscillations of the pressure occur and which demonstrates the failure of the Babuška-Brezzi-condition.

Example 2.4. *Let $\Omega = [0,1]^2$ and $\mathbb{T}_h$ be a grid consisting of four identical squares. The analytical solution of the homogenous Stokes equation on Ω with boundary conditions $\mathbf{u}|\partial\Omega = (y,0)$ is $\mathbf{u}(x,y) = (y,0)$, $p(x,y) = const$. If one computes the solution on $\mathbb{T}_h$, one gets a "checkerboard"-solution for p oscillating between $-c$ and c with c being arbitrarily large.*

Remark 2.5. *The reason for the element pair to be unstable is related to the fact that the space of velocities is "too small" related to the pressure space. As $\tilde{P}_1(\mathbb{T}_h)$ has one dimension less than the space $Q_1(\mathbb{T}_h)$ and the element pair (Q_1, Q_0) is known to be unstable, there is a strong evidence that the pair $(\tilde{P}_1, Q_0)$ is unstable, too.*

Also Korn's inequality, which is another important stability condition in CFD and CSM simulations, does not hold. To demonstrate this, we give the following example.

Example 2.6. *The sketched function in Figure 7 does not fulfill the discrete Korn's inequality*

$$|v|_{1,h} \leq c\,||\varepsilon(v)||^2_{0,h}, \quad c \neq c(h), \quad \forall\, v \in (\tilde{P}_1(\mathbb{T}_h))^2 \text{ with } v = 0 \text{ in boundary midpoints}, \tag{12}$$

where $\varepsilon(v)$ denotes the symmetric part of ∇v, $\varepsilon(v) = \frac{1}{2}(\nabla v + \nabla v^\top)$. Let Ω be the unit square and $\mathbb{T}$ a triangulation consisting of identical squares with diameter h_T. Let $\mathbb{T}_1 := \{T \in \mathbb{T} \mid \partial T \cap \partial\Omega \neq \emptyset\}$. On $\mathbb{T}\backslash\mathbb{T}_1$, we define the function $\mathbf{u} = (u, v) \in \tilde{P}_1(\mathbb{T})^2$ by

$$\mathbf{u}(x, y)|_T = (\pm h_T(y - y_T), \mp h_T(x - x_T))\,,$$

where (x_T, y_T) denotes the spatial midpoint of an element T. The function $\mathbf{u}$ is divergence-free on $\mathbb{T}\backslash\mathbb{T}_1$. Furthermore, we have

$$\nabla\mathbf{u}|_T = \begin{pmatrix} 0 & \pm h_T \\ \mp h_T & 0 \end{pmatrix}$$

and therefore $\varepsilon(\mathbf{u}) = 0$ auf $T \in \mathbb{T}\backslash\mathbb{T}_1$. On the boundary layer $\mathbb{T}_1$ we continue $\mathbf{u}$ linearly such that the function is continuous in the midpoints and there holds $\mathbf{u} = 0$ in all boundary midpoints. Globally, we do not have $||\varepsilon(\mathbf{u})|| = 0$, but on all inner quadrilaterals. As the area of the boundary layer $\mathbb{T}_1$ will vanish asymptotically, (12) cannot hold with c independent of $h = h_T$. The function $\mathbf{u}$ is the same "kernel function" as in [12], which is used there to show that Korn's inequality is not valid in the case of the Rannacher-Turek-element. However, see [10, 12] for a strategy of edge-oriented stabilization to repair this well-known fact for nonconforming FEM.

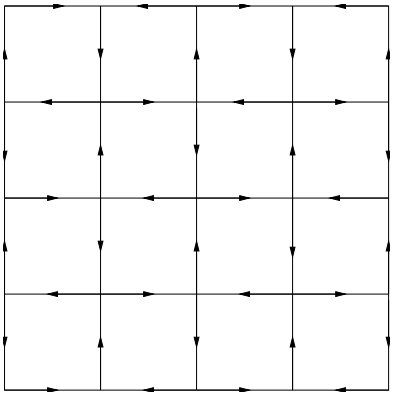

Figure 7: The shown velocity vector $\mathbf{u} \in (\tilde{P}_1(\mathbb{T})^2)$ does not fulfill the discrete Korn's inequality ($\mathbb{T}_1$ is not displayed)

3 Residual-based a posteriori error estimation for $\tilde{P}_1$

A posteriori error control and grid adaptation strategies are essential ingredients for a valuable and applicable simulation tool. However, it is often not sufficient to be able to estimate global error measures like L_2- or H_1-norms, but it is in view of practical simulation tasks crucial to control derived quantities like lift or drag coefficients or mean values over given curves or surfaces. The derivation of these user-specific quantities from the solution can be formalized by evaluating a corresponding *error functional* $J(\cdot)$ which is assumed here to be continuous and linear. Therefore, one needs upper (and also lower) bounds $\eta(e_h)$ for $|J(e_h)|$, where $e_h = u - u_h$ denotes the error function and u the unknown solution of the problem

$$a(u,\varphi) := (\nabla u, \nabla\varphi) = (f,\varphi) \quad \forall\varphi \in V := H_0^1(\Omega). \tag{13}$$

In this paper, we will restrict ourselves to the Poisson equation as prototypical problem. To obtain such an upper bound, Becker and Rannacher introduced in [3, 2] for a discretisation with conforming finite elements the *dual problem*

$$a(\varphi,z) := (\nabla\varphi, \nabla z) = J(\varphi) \quad \forall\varphi \in V, \tag{14}$$

where $z \in H_0^1(\Omega)$ is the 'dual solution'. Then, in the case of a domain with a polygonal boundary we derive by (14), the relation $V_h \subset V$ and the Galerkin orthogonality, $a(u - u_h, \varphi) = 0\,\forall\varphi \in V_h$, the following fundamental error identity

$$J(e_h) = a(e_h, z) = a(e_h, z - z_h) = \sum_{T\in\mathbb{T}_h} (\nabla e_h, \nabla(z - z_h))_T \tag{15}$$

for an arbitrary $z_h \in V_h$. Applying Green's theorem, we obtain the error representation

$$|J(e_h)| = \left| \sum_{T\in\mathbb{T}_h} \left\{ (\underbrace{-\triangle u}_{=f} + \triangle u_h, z - z_h)_T + \frac{1}{2}([\partial_{\mathfrak{n}} u_h], z - z_h)_{\partial T} \right\} \right|. \tag{16}$$

As the dual solution z is unknown as well as the primal solution u, it has to be replaced by a suitable discrete approximation z_I. As a consequence, the error representation (16) becomes an error estimation. To obtain asymptotic exactness of the error estimation, the replacement z_I should be of higher order than z_h being the FEM approximation of z. As this requirement can imply unacceptable high costs for computing the dual problem, we restrict ourselves to the condition of $||z - z_I|| \ll ||z - z_h||$. By this, we may loose asymptotic exactness. However, the latter requirement can be fulfilled by an interpolation approach analogously to the conforming case ([3, 2]).
To insert z_h, which from now on shall stand for the solution of the discretized dual problem, the Galerkin orthogonality, $a(u - u_h, \varphi) = 0\,\forall\varphi \in V$, is necessary in the derivation of eq. (16). For a discretisation with $\tilde{P}_1$, this does not hold any more and therefore eq. (16) is valid only for conforming finite elements. To extend the sketched method to nonconforming finite element methods, we modify the dual problem according to Suttmeier and Kanschat [11] such that

$$(\varphi, -\triangle z) = J(\varphi) \quad \forall\varphi \in L_2(\Omega) \tag{17}$$

and according to Suttmeier and Kanschat [11], too, we also modify the finite element space V_h by adding a bulb function φ_B which is transformed parametrically in contrast to the other basis functions. By this, the finite element space V_h is augmented to a new finite element space $V_h^E := V_h \cup \{\varphi_B\}$. This augments also the conforming subspace $V_h^C := V \cap V_h$ to the space $V_h^{E,C}$ to have appropriate approximation properties also in $V_h^{E,C} := V \cap V_h^E$. In fact, $V_h^{E,C}$ equals to the conforming space of parametric elementwise bilinear functions. Then, it holds the modified Galerkin orthogonality

$$a_h(u - u_h^E, \varphi_h) = \sum_{T \in \mathbb{T}_h} (\nabla(u - u_h^E), \nabla\varphi_h)_T = 0 \quad \forall \varphi_h \in V_h^{E,C} \tag{18}$$

where u_h^E denotes the solution of the discrete problem in the enhanced space V_h^E. Hence, we can insert some discrete function $z_h^C \in V_h^{E,C}$ analogously to the conforming case which does not have to be the solution of the discrete dual problem in $V_h^{E,C}$. By applying Green's theorem we obtain the final error representation:

$$J(e_h) = \sum_{T \in \mathbb{T}} \left\{ \left(f + \triangle u_h, z - z_h^C\right)_T - \frac{1}{2}\left([\partial_{\mathfrak{n}} u_h], z - z_h^C\right)_{\partial T} - \frac{1}{2}\left([u_h], \partial_{\mathfrak{n}} z\right)_{\partial T} \right\} + a_h(e_h, z_h^C) \tag{19}$$

The consistency error $a_h(e_h, z_h^C)$ (note that u_h is *not* taken from the enhanced space V_h^E but from V_h) which comes from the insertion of $z_h^C \in V_h^{E,C}$ and the enhancement of the function space turns out to be of fourth order and therefore does not affect the error estimation ([11, 8]). Because of this it will be neglected subsequently. All details of the derivation of formula (19) will be adressed in a forthcoming paper [9].

To compute z_h^C, we solve the discrete dual problem in the same nonconforming space V_h like the discrete primal problem and project afterwards the dual solution z_h into V_h^C. To do so, we define for a given vertex v_j the neighbourhood $\mathcal{N}(j) := \{i \in \mathbb{N} | \mathbb{T}_h \ni T_i \cap v_j \neq \emptyset\}$ and we set

$$z_h^C(v_j) := \sum_{i \in \mathcal{N}(j)} \frac{\alpha_i}{2\pi} \lim_{T_i \ni x \to v_j} z_h(x), \tag{20}$$

where α_i stands for the opening angle of T_i in v_j. Now we can define the global function z_h^C as the unique piecewise parametrically transformed bilinear function which matches these values. The discontinuity of $\tilde{P}_1$-functions in the vertices prevents a direct evaluation in the vertices for interpolation, so that it is replaced by a "smeared out" evaluation in the form of a mean value over some ε-ball B_ε around v_j. Formula (20) approximates this integration. It is also possible to define z_h^C as L_2-projection instead; experiments show that there is no substantial difference in these two projection types (compare [8]).

To replace z by a suitable approximation z_I, it is not possible to perform the same interpolation process as in the conforming case since the function z_h is discontinuous in the vertices. Using the conforming 'dual solution' z_h^C which has to be computed anyway, we perform a biquadratic interpolation of z_h^C on a patch of four elements. The edges serve as nodal points.

To evaluate the term $([u_h], \partial_{\mathfrak{n}} z)_{\partial T}$, we replace z by the arithmetic means of z_h on the two elements adjacent to the current edge to integrate over:

$$\partial_{\mathfrak{n}} z(x) \approx \overline{\partial_{\mathfrak{n}} z}(x) := \frac{1}{2}\big(\partial_{\mathfrak{n}} z_h|_{T_1}(x) + \partial_{\mathfrak{n}} z_h|_{T_2}(x)\big), \quad x \in T_1 \cap T_2. \tag{21}$$

Then, we obtain the following Lemma:

Lemma 3.1. *For arbitrary $T \in \mathbb{T}_h$, there holds*

$$\left([u_h], \overline{\partial_{\mathfrak{n}} z}\right)_{\partial T} = 0.$$

Proof: $z_h|_{T_1}$ und $z_h|_{T_2}$ are linear. Therefore, $\partial_{\mathfrak{n}} z_h|_{T_1} - \partial_{\mathfrak{n}} z_h|_{T_2}$ is constant and $[u_h] \cdot \frac{1}{2}(\partial_{\mathfrak{n}} z_h|_{T_1} - \partial_{\mathfrak{n}} z_h|_{T_2})$ is linear. As u_h is continuous in the midpoints, for a midpoint holds $[u_h](m_\Gamma) = 0$. As the midpoint rule integrates linear functions exactly, we obtain on the edge Γ

$$\int_\Gamma [u_h] \overline{\partial_{\mathfrak{n}} z}\, ds = [u_h](m_\Gamma) \cdot \overline{\partial_{\mathfrak{n}} z}(m_\Gamma) = 0.$$

This completes the proof. □

Since u_h is a piecewise linear function, there holds $\triangle u_h = 0$ on each element such that we get the final error estimation

$$|J(e_h)| \approx \left| \sum_{T \in \mathbb{T}_h} (f, z_I - z_h^C)_T - \frac{1}{2}([\partial_{\mathfrak{n}} u_h], z_I - z_h^C)_{\partial T} \right|. \tag{22}$$

To evaluate the quality of the error estimation (22) we estimate numerically the point error in $(0.35, 0.5)$ shortly before the square, i.e. $J(\varphi) = \varphi(0.35, 0.5)$, in the situation of a square in a channel (see Figure 3, left). We choose f and the Dirichlet boundary conditions such that the exact solution u is the same as in Test Problem 2.1. The results are collected in Table 9. As second example we use the DFG-benchmark grid "flow around a cylinder" (Figure 6) and estimate the error in $(0.28, 0.2)$ slightly behind the cylinder. Here, we prescribe the constant right hand side $f = 10$ and homogenous Dirichlet boundary conditions. As the exact solution is not known, we again refer to a reference solution computed with conforming biquadratic finite elements on a very fine mesh (Table 10). The error coming from the approximation of the curved boundary in the case of the cylinder in a channel is neglected here. As a quantitative measure for the quality of the error estimator we define the *efficiency index*

$$I_{\text{eff}} := \frac{\eta(e_h)}{|J(e_h)|}. \tag{23}$$

As $u_h \in \tilde{P}_1$ is not continuous, point values may not be well-defined such that we use instead of $J_{x_0}(\varphi) = \varphi(x_0)$, which represents the point evaluation, a regularized version

$$J_{x_0}^r(\varphi) := \frac{1}{|B_\varepsilon|} \int_{B_\varepsilon} \varphi \, dA. \tag{24}$$

For practical reasons, this regularized functional will be approximated by replacing the integral in the same way as we did in the interpolation process in the definition of z_h^C. Furthermore, we can estimate the error for the integral of the normal derivative on the interior boundary,

$$J_\Gamma(\varphi) := \int_\Gamma \partial_{\mathfrak{n}} \varphi \, ds. \tag{25}$$

NEL	$\left\|J^r_{(0.35,0.5)}(e_h)\right\|$	$\eta(e_h)$	I_{eff}
68	$1,97 \cdot 10^{-4}$	$2,13 \cdot 10^{-3}$	10,8
272	$2,00 \cdot 10^{-4}$	$4,70 \cdot 10^{-4}$	2,35
1088	$5,86 \cdot 10^{-5}$	$9,65 \cdot 10^{-5}$	1,65
4352	$1,51 \cdot 10^{-5}$	$2,31 \cdot 10^{-5}$	1,53
17408	$3,80 \cdot 10^{-6}$	$5,73 \cdot 10^{-6}$	1,51
69632	$9,51 \cdot 10^{-7}$	$1,43 \cdot 10^{-6}$	1,51

Table 9: Point error in $(0.35, 0.5)$ for the 'square in a channel'

NEL	$\left\|J^r_{(0.28,0.2)}(e_h)\right\|$	$\eta(e_h)$	I_{eff}
520	$3,56 \cdot 10^{-4}$	$1,10 \cdot 10^{-3}$	3,10
2080	$9,24 \cdot 10^{-5}$	$3,24 \cdot 10^{-4}$	3,51
8320	$2,22 \cdot 10^{-5}$	$7,82 \cdot 10^{-5}$	3,52
33280	$5,35 \cdot 10^{-6}$	$1,97 \cdot 10^{-5}$	3,69

Table 10: Point error in $(0.28, 0.2)$, 'DFG-benchmark' grid

This kind of error quantities resembles error functionals like lift and drag coefficients in the case of CFD solutions. The evaluation of the functional $J_\Gamma(T)$ where T represents the temperature equals the computation of the *Nusselt number* in thermodynamics (compare with [17]). For the corresponding results see Tables 11 and 12.
These results indicate that reliable and efficient error control is possible also for the nonconforming $\tilde{P}_1$-approach. On the other hand, this type of error control is not for free as the computation of the dual problem has approximately the same numerical amount as the computation of the problem itself and, therefore, applying the dual weighted residual based error control presented here doubles the total computational costs as shown by the results in Table 13 which contains the computational times for the example of the point error estimation in the situation of the 'square in the channel' presented above. As before, we use a Compaq ALPHA with 667 MHz. As solver, we take BiCGStab with ILU(1) as preconditioner. As the numerical amount of the iterative solver used here grows like $N^{1.5}$ where N denotes the number of unknowns, the computational time grows by a factor eight per refinement. The results show that the numerical amount for the primal and dual problem is almost the same. However, keep in mind that the growth of computational time coming from the error estimation can be compensated by adapting the grid to the desired user-specific quantity, so that instead of applying regular refinement we can refine the computational grid selectively as shown in the papers of Becker and Rannacher. In the presented work we restrict to regular refinement for technical reasons only.

NEL	$\lvert J_\Gamma(e_h)\rvert$	$\eta(e_h)$	I_{eff}
272	$1,18 \cdot 10^{-3}$	$3,47 \cdot 10^{-3}$	2,94
1088	$5,59 \cdot 10^{-4}$	$1,35 \cdot 10^{-3}$	2,41
4352	$3,12 \cdot 10^{-4}$	$5,27 \cdot 10^{-4}$	1,69
17408	$1,67 \cdot 10^{-4}$	$2,21 \cdot 10^{-4}$	1,33
69632	$8,63 \cdot 10^{-5}$	$9,91 \cdot 10^{-5}$	1,15

Table 11: Error estimation for the line integral for the 'square in the channel'

NEL	$\lvert J_\Gamma(e_h)\rvert$	$\eta(e_h)$	I_{eff}
520	$4,52 \cdot 10^{-2}$	$1,13 \cdot 10^{-1}$	2,50
2080	$2,34 \cdot 10^{-2}$	$1,53 \cdot 10^{-2}$	0,65
8320	$1,14 \cdot 10^{-2}$	$2,96 \cdot 10^{-3}$	2,61
33280	$5,16 \cdot 10^{-3}$	$1,75 \cdot 10^{-3}$	3,39

Table 12: Error estimation for the line integral for the 'cylinder in the channel'

NEL	primal problem	error estimation	total time
17408	3.9 (96)	4.0 (112)	7.9
69632	22.2 (218)	37.2 (246)	59.4
278528	290 (469)	322 (596)	612

Table 13: Computational times (number of iteration steps) for the primal and dual problem in the case of the square in the channel, BiCGStab with ILU(1) and resorting

4 Summary and outlook

In this article, we performed a numerical analysis regarding approximation properties, solution strategies and implementation details for the new nonconforming linear finite element $\tilde{P}_1$ on quadrilateral meshes. Among several treatments for Dirichlet boundary conditions, the fully implicit boundary treatment turned out to be suitable also for multiply connected domains. Unfortunately, it complicates the solving of the arising linear systems. This can be partially overcome by using ILU(0) and especially ILU(1) as preconditioners for an iterative method like BiCGStab or GMRes. In contrast to this, matrix assembly works outstandingly fast.Based upon the recent work of Becker and Rannacher we derived a dual weighted residual based error estimator with respect to user-specific error functionals. We proved this error estimation to be reliable and efficient for estimating user-specific quantities like point values as well as integrals of the normal derivative over parts of the boundary.
At current state, the velocity-pressure combination $(\tilde{P}_1, Q_0)$ is not a suitable Stokes-element (compare Example 2.4). Therefore, any further research on $\tilde{P}_1$ will be influenced by the development of fast and robust stabilisation techniques for saddle point problems. If the problem of stability is overcome, $\tilde{P}_1$ is a new prototypial nonconforming element and will be a interesting candidate for CFD.

References

[1] Arnold, D.N, Boffi, D., and Falk, R.S. Approximation by quadrilateral finite elements. *Math Comp.*, 71(239):909–922, 2002.

[2] Becker, R. and Rannacher, R. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math*, 4, 1996.

[3] Becker, R. and Rannacher, R. An optimal control approach to a posteriori error estimates in finite element methods. In *Acta Numerica 2001*, pages 1–102. Cambridge University Press, 2001.

[4] Blum, H., Harig, J., Müller, S., and Turek, S. Feat2D - Finite Element Analysis Tools User Manual Release 1.3. Technical report, University of Heidelberg, 1992.

[5] Bramley, R. and Wang, X. Splib: A library of iterative methods for sparse linear systems, 1995.

[6] Buijssen, S. and Turek, S. Sources of parallel inefficiency for incompressible CFD simulation. In Monien, B. and Feldmann, R., editors, *Proceedings 8th International Euro-Par Conference*, LNCS. Springer, 2002. Paderborn, Germany, August 27-30.

[7] Douglas, J., Santos, J., Sheen, D., and Ye, X. Nonconforming galerkin methods based on quadrilateral elements for second order elliptic problems. *Math. Model. Numer. Anal.*, 33(4):747–770, 1999.

[8] Grajewski, M. Numerische Analyse einer nichtkonformen linearen Finite-Elemente-Methode auf Viereckgittern. Diploma thesis, 2003. University of Dortmund.

[9] Grajewski, M. and Turek, S. Dual weighted a posteriori error estimation for nonconforming finite elements. to appear (2003).

[10] Hansbo, P. and Larson, G. A simple nonconforming bilinear element for the elasticity problem. In *CIMNE*, Barcelona, Spain, 2001.

[11] Kanschat, G. and Suttmeier, F.-T. A posteriori error estimates for nonconforming element schemens. *Calcolo*, 36:129–141, 1999.

[12] Ouazzi, A., Schmachtel, R., and Turek, S. Multigrid methods for stabilized nonconforming finite elements for incompressible flow involving the deformation tensor formulation. *J. Numer. Math.*, 10:235–248, 2002.

[13] Park, C. *A study on locking phenomena in finite element computations*. PhD thesis, Seoul National University, 2002.

[14] Park, C. and Sheen, D. P_1-nonconforming quadrilateral finite element methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 41:624–640, 2003.

[15] Rannacher, R., Schäfer, M., and Turek, S. Evaluation of a CFD benchmark for laminar flows. Technical report, University of Heidelberg, 1998. SFB 359.

[16] Rannacher, R. and Turek, S. A simple nonconforming quadrilateral Stokes element. *Numer. Meth. Part. Diff. Equ.*, 8:97–111, 1992.

[17] Schmachtel, R. and Turek, S. Fully coupled and operator-splitting approaches for natural convection. *Int. Numer. Meth. Fluids*, 40:1109–1119, 2002.

[18] Turek, S. On ordering strategies in a multigrid algorithm. In *Proc. 8th GAMM–Seminar*, volume 41 of *Notes on Numerical Fluid Mechanics*. Vieweg, 1992.

[19] Turek, S. Tools for simulating nonstationary incompressible flow via discretely divergence–free finite element models. *Int. J. Numer. Meth. Fluids*, 18:71–105, 1994.

[20] Turek, S. *Efficient solvers for incompressible flow problems: An algorithmic and computational approach*. Springer, 1999.

[21] Turek, S., Wan, D., and Rivkind, L. The fictitious boundary method for the implicit treatment of Dirichlet boundary conditions with applications to incompressible flow simulations. Technical report, University of Dortmund, 2003.