Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 4776–4796
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A direct solver for the least-squares spectral collocation system on rectangular elements for the incompressible Navier–Stokes equations

Wilhelm Heinrichs *, Thorsten Kattelans

University of Duisburg-Essen, Engineering Mathematics, Universitaetsstr. 3, D-45117 Essen, Germany

Received 14 September 2007; received in revised form 10 January 2008; accepted 16 January 2008
Available online 26 January 2008

Abstract

A least-squares spectral collocation scheme for the Stokes and incompressible Navier–Stokes equations is proposed. The original domain is decomposed into quadrilateral subelements and on the element interfaces continuity of the functions is enforced in the least-squares sense. The collocation conditions and the interface conditions lead to overdetermined systems. These systems are directly solved by QR decomposition of the underlying matrices. By numerical simulations it is shown that the direct method leads to better results than the approach with normal equations. Furthermore, it is shown that the condition numbers can be reduced by introducing the Clenshaw–Curtis quadrature rule for imposing the average pressure to be zero. Finally, our scheme is successfully applied to the regularized and lid-driven cavity flow problems.

Keywords: Navier–Stokes equations; Least-squares; Spectral collocation; Direct solvers; Condition numbers; Improved stability

1. Introduction

Spectral methods (see, e.g. [6,9,22] or [7]) employ global polynomials for the numerical solution of differential equations. Hence they give very accurate approximations for smooth solutions with relatively few degrees of freedom. For analytical data exponential convergence can be achieved. If one deals with problems with non-smooth solutions (e.g. discontinuities or layers) the usual (global) continuous spectral approach yields very poor approximation results. To avoid these difficulties the original domain has to be decomposed into several subdomains where jumps at the interfaces are allowed, i.e. there is the possibility to make the solution discontinuous between neighboring elements. These are the so called discontinuous Galerkin methods and as a very special case, there are the discontinuous least-squares approaches. Gerritsma and Proot showed in [10] the good performance of discontinuous least-squares spectral element methods. Furthermore, Bensow and Larson showed the good performance of discontinuous least-squares finite element methods in [1,2]. In

* Corresponding author. Tel.: +49 201 183 2534; fax: +49 201 183 4184.
E-mail addresses: wilhelm.heinrichs@uni-due.de (W. Heinrichs), thorsten.kattelans@uni-due.de (T. Kattelans).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.025

[13] we extended the above approach to one-dimensional singular perturbation problems where the least-squares spectral collocation schemes lead to a stabilization. Heinrichs extended the in [14] proposed least-squares spectral collocation method to a triangular decomposition [16] of the original domain and achieved good numerical results. Here we extend the method to a decomposition in quadrilaterals of the original domain and apply these scheme to the two-dimensional Stokes and incompressible Navier–Stokes equations. The collocation conditions together with the interface and boundary conditions lead to an overdetermined system that can be approximately solved by least-squares. The essential enhancements of the here introduced scheme is the increased accuracy because of the use of a direct solver. For the overdetermined system we compute the QR decomposition of the associated matrix and solve the system. Because of avoiding the normal equations we obtain linear systems of equations with dramatically reduced condition numbers and so round-off errors do not have such a big influence to the approximation results.

For the Stokes and Navier–Stokes problems the velocity and the pressure cannot be approximated independently due to the well known Babuška–Brezzi condition. If the velocity and the pressure are approximated by polynomials of the same degree eight spurious modes are introduced which lead to an unstable system (see [3]). A well-known compatible approximating velocity–pressure pair is the so-called $\mathbb{P}_N \times \mathbb{P}_{N-2}$ formulation (see e.g. [31]). Heinrichs [11,12] employed this technique for the splitting of the Stokes and Navier–Stokes equations. There the velocity components are approximated by polynomials in $\mathbb{P}_N$ and the pressure by two degrees lower order polynomials in $\mathbb{P}_{N-2}$. The resulting discrete system constitutes a saddle point problem which is difficult to solve numerically.

Least-squares techniques for such problems (see, e.g. [17–20]) offer theoretical and numerical advantages over the classical Galerkin type methods which fulfill the well-posedness (or stability) criterion, the so called LBB condition. One very special least-squares technique is the spectral least-squares method. These spectral least-squares methods for the Stokes problem were first introduced by Gerritsma and Proot in [27,28]. Spectral least-squares for the Navier–Stokes equations were first presented by Pontaza and Reddy in [24–26], followed by Gerritsma and Proot in [29]. Heinrichs investigated least-squares spectral collocation schemes in [14–16] that lead to symmetric and positive definite algebraic systems which circumvent the LBB stability condition. Since we here work on least-squares spectral collocation schemes we want to summarize some advantages of this approach:

- equal order interpolation polynomials can be employed;
- it is possible to vary the polynomial order from element to element;
- improved stability properties for small perturbation parameters in singular perturbation problems [8,13] and Stokes or Navier–Stokes equations [14–16,27–29];
- good performance in combination with domain decomposition techniques;
- direct and efficient iterative solvers for positive definite systems can be used;
- implementation is straightforward.

The paper is organized as follows. In Section 2, the first-order formulation of the Stokes and Navier–Stokes equations is introduced. Then we describe the least-squares spectral collocation scheme, specify the domain decomposition (Section 3) and describe the discrete linear system of equations (Section 4). Section 5 presents the numerical simulations with the results for the Stokes (Section 5.1 and 5.2) and for the Navier–Stokes equations (Section 5.3: smooth example, Section 5.4: regularised cavity flow, Section 5.5: lid-driven cavity flow). Finally, a conclusion is presented.

2. The Stokes and Navier–Stokes equations

In order to apply least-squares the Stokes and Navier–Stokes problem is transformed into an equivalent first-order system of partial differential equations. This is accomplished by introducing the vorticity $\omega = \nabla \times \mathbf{u}$ as an auxiliary variable. By using the identity

$$\nabla \times \nabla \times \mathbf{u} = -\Delta \mathbf{u} + \nabla(\nabla \cdot \mathbf{u})$$

and the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$ we obtain

$$\frac{\partial \mathbf{u}}{\partial t} + v\nabla \times \omega + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}], \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}], \tag{2}$$

$$\omega - \nabla \times \mathbf{u} = 0 \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}] \tag{3}$$

for the Stokes equations and for the Navier–Stokes equations we obtain

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + v\nabla \times \omega + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}], \tag{4}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}], \tag{5}$$

$$\omega - \nabla \times \mathbf{u} = 0 \quad \text{in } \Omega, \quad t \in [0, t_{\text{end}}], \tag{6}$$

where $\mathbf{u}^{\mathrm{T}} = [u_1, u_2]$ denotes the velocity vector, p the pressure, $\mathbf{f}^{\mathrm{T}} = [f_1, f_2]$ the forcing term and v the kinematic viscosity. Here it is assumed that the density equals unity. Since the pressure is through (1)–(3) or (4)–(6) only determined up to a constant for the Stokes or Navier–Stokes equations we have to introduce an additional condition for the pressure. One procedure is to impose the pressure at an arbitrary point of the given domain. In [29] the pressure constant was set to zero in the point $(0.5, 0.5)$. For simplicity we impose the pressure at the point $(1, 1)$. Another way of dealing with the pressure constant is imposing the average pressure to be zero; i.e.

$$\int_{\Omega} p \, \mathrm{d}\mathbf{x} = 0. \tag{7}$$

2.1. The Stokes equations

For the Stokes equations we use for time integration a second-order BDF scheme (see, e.g. [11]): If Δt denotes the step size in t and the index $n+1$ indicates that the functions are evaluated at the time step $t_{n+1} = (n+1)\Delta t$, $n = 0, 1, 2, \ldots$, the approximation of $\left(\frac{\partial \mathbf{u}}{\partial t}\right)^{n+1}$ can be written as

$$\frac{\frac{3}{2}\mathbf{u}^{n+1} - 2\mathbf{u}^{n} + \frac{1}{2}\mathbf{u}^{n-1}}{\Delta t}. \tag{8}$$

Now the complete system at time step t_{n+1} can explicitly be written as:

$$\begin{pmatrix} \frac{3}{2\Delta t} & 0 & v\frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \\ 0 & \frac{3}{2\Delta t} & -v\frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & -\frac{\partial}{\partial x_1} & 1 & 0 \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \omega^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} g_1^{n+1} \\ g_2^{n+1} \\ 0 \\ 0 \end{pmatrix} \quad \text{in } \Omega, \tag{9}$$

where

$$\mathbf{g}^{n+1} = \frac{2}{\Delta t}\mathbf{u}^{n} - \frac{1}{2\Delta t}\mathbf{u}^{n-1}.$$

2.2. The Navier–Stokes equations

For the Navier–Stokes equations we use an implicit and explicit scheme and compare both.

2.2.1. Implicit scheme

As proposed in [29], we apply a θ-integration scheme in time combined with the Picard linearization to the momentum equations of the unsteady Navier–Stokes equations. The subscript "0" corresponds to the results obtained at a previous integration time step. Now the momentum equations read as follows:

$$\frac{\mathbf{u} - \mathbf{u}_0}{\Delta t} + \theta(\mathbf{u}_0 \cdot \nabla \mathbf{u} + \nabla p + v\nabla \times \omega - \mathbf{f}) = (\theta - 1)(\mathbf{u}_0 \cdot \nabla \mathbf{u}_0 + \nabla p_0 + v\nabla \times \omega_0 - \mathbf{f}_0). \tag{10}$$

By taking $\theta = 1$, the time integration is carried out by the backward Euler method, which is only first-order accurate in time. The second-order time integration of Crank–Nicolson can be obtained by setting $\theta = 1/2$. Since the Crank–Nicolson scheme has no damping, one often takes $\theta = 1/2 + \mathcal{O}(\Delta t)$. The temporal accuracy remains second-order, and adding the small factor Δt effectively damps the small waves in spectral element simulations. Hence, in order to obtain time-accurate solutions, one should use $\theta = 1/2 + \mathcal{O}(\Delta t)$. The θ-scheme is unconditionally stable for $1/2 \leqslant \theta \leqslant 1$. Here we only consider stationary problems where it is recommended to use backward Euler ($\theta = 1$) with large time steps to obtain steady-state solutions. Now the complete system for each time step can explicitly be written as:

$$Az = r, \tag{11}$$

where

$$A = \begin{pmatrix} \frac{1}{\Delta t} + \theta u_{1,0}\frac{\partial}{\partial x_1} + \theta u_{2,0}\frac{\partial}{\partial x_2} & 0 & \theta v \frac{\partial}{\partial x_2} & \theta \frac{\partial}{\partial x_1} \\ 0 & \frac{1}{\Delta t} + \theta u_{1,0}\frac{\partial}{\partial x_1} + \theta u_{2,0}\frac{\partial}{\partial x_2} & -\theta v \frac{\partial}{\partial x_1} & \theta \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & -\frac{\partial}{\partial x_1} & 1 & 0 \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & 0 & 0 \end{pmatrix},$$

$$z = (u_1, u_2, \omega, p)^{\mathrm{T}} \quad \text{and}$$

$$r = \begin{pmatrix} f_1 + \frac{u_{1,0}}{\Delta t} + (\theta - 1)\left[u_{1,0}\frac{\partial u_{1,0}}{\partial x_1} + u_{2,0}\frac{\partial u_{1,0}}{\partial x_2} + v\frac{\partial \omega_0}{\partial x_2} + \frac{\partial p_0}{\partial x_1}\right] \\ f_2 + \frac{u_{2,0}}{\Delta t} + (\theta - 1)\left[u_{1,0}\frac{\partial u_{2,0}}{\partial x_1} + u_{2,0}\frac{\partial u_{2,0}}{\partial x_2} + v\frac{\partial \omega_0}{\partial x_1} + \frac{\partial p_0}{\partial x_2}\right] \\ 0 \\ 0 \end{pmatrix}.$$

2.2.2. Explicit scheme

For time integration we also employ a semi-implicit scheme where the second order backward differentiation scheme (8) for the viscous term is combined with a second order Adams–Bashforth scheme for the convective term. Hence the momentum equations (4) at time step $t_{n+1} = (n+1)\Delta t$, $n = 0, 1, 2, \ldots$ can be written as:

$$\frac{3}{2\Delta t}\mathbf{u}^{n+1} + v\nabla \times \omega^{n+1} + \nabla p^{n+1} = \mathbf{g}^{n+1}, \tag{12}$$

where

$$\mathbf{g}^{n+1} = \mathbf{f}^{n+1} - 2\mathbf{C}^{n} + \mathbf{C}^{n-1} + \frac{2}{\Delta t}\mathbf{u}^{n} - \frac{1}{2\Delta t}\mathbf{u}^{n-1} \tag{13}$$

with the convective term $\mathbf{C} = (\mathbf{u} \cdot \nabla)\mathbf{u}$.

Now the complete system at time step t_{n+1} can explicitly be written as:

$$\begin{pmatrix} \frac{3}{2\Delta t} & 0 & v\frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \\ 0 & \frac{3}{2\Delta t} & -v\frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & -\frac{\partial}{\partial x_1} & 1 & 0 \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \omega^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} g_1^{n+1} \\ g_2^{n+1} \\ 0 \\ 0 \end{pmatrix} \quad \text{in } \Omega. \tag{14}$$

The big advantage of the explicit scheme is that the system of equations must only be solved once. During time integration we only have to compute matrix-vector multiplications which are very fast. By numerical experiments we found out that for a well balanced system it is recommended to scale the momentum equations by Δt, as in [14]. Then for the least-squares scheme the incompressibility condition is well balanced against the momentum equations. In particular, we observed that without scaling the scheme becomes divergent for increasing Reynolds numbers since the diagonal entries $3/2\Delta t$ become large for decreasing step size. As our

simulations have shown the explicit scheme is much faster than the implicit one for Reynolds numbers up to 1000.

3. The least-squares spectral collocation scheme

For the spectral approximation we introduce the polynomial subspace

$$\mathbb{P}_N = \{\text{Polynomials of degree} \leqslant N \text{ in both variables } x_1, x_2\}.$$

Now all unknown functions are approximated by polynomials of the same degree N, i.e. u_1, u_2, ω, p are approximated by interpolating polynomials $u_1^N, u_2^N, \omega^N, p^N \in \mathbb{P}_N$. Furthermore, we have to introduce the standard Chebyshev Gauss–Lobatto collocation nodes which are explicitly given by

$$(\xi_i, \eta_j) = \left(-\cos\left(\frac{i\pi}{N}\right), -\cos\left(\frac{j\pi}{N}\right)\right), \quad i,j = 0,\ldots,N. \tag{15}$$

In the following we write the spectral derivatives. First one has to introduce the transformation matrices from physical space to coefficient space. Since we employ a Chebyshev expansion we obtain the following matrix:

$$T = (t_{i,j}) = \left(\cos\left(i\frac{j\pi}{N}\right)\right), \quad i,j = 0,\ldots,N.$$

Further we need the differentiation matrix in the Chebyshev coefficient space which is explicitly given by $\widehat{D} = (\hat{d}_{i,j}) \in \mathbb{R}^{N+1,N+1}$ with

$$\hat{d}_{i,j} = \begin{cases} \frac{2j}{c_i}, & j = i+1, i+3, \ldots, N, \\ 0, & \text{else} \end{cases}$$

and

$$c_i = \begin{cases} 2, & i = 0, \\ 1, & \text{else.} \end{cases}$$

Now we are able to write explicitly the spectral derivative matrix D for the first derivative which is given by

$$D = T\widehat{D}T^{-1} \in \mathbb{R}^{N+1,N+1}.$$

The spectral operator can be efficiently evaluated by Fast Fourier Transformations (FFTs) in $\mathcal{O}(N \log N)$ arithmetic operations. We further introduce the identity matrix $I \in \mathbb{R}^{N+1,N+1}$. By tensor product representation $A \otimes B = (Ab_{i,j})_{i,j}$ we are now able to write the spectral derivatives:

$$\frac{\partial}{\partial x} \cong D_1 := D \otimes I, \quad \frac{\partial}{\partial y} \cong D_2 := I \otimes D. \tag{16}$$

To decompose the domain Ω into quadrilateral elements $\Omega_{i,j} := (x_{i-1}, x_i) \times (y_{j-1}, y_j)$, $i,j = 1,\ldots,K$, where K^2 denotes the number of elements, we define the element borders for an equidistant decomposition by

$$x_i := -1 + i\frac{2}{K}, \quad y_j := -1 + j\frac{2}{K}, \quad i,j = 0,\ldots,K \tag{17}$$

for a Chebyshev Gauss–Lobatto (CGL) decomposition by

$$x_i = -\cos\left(\frac{i\pi}{K}\right), \quad y_j = -\cos\left(\frac{j\pi}{K}\right), \quad i,j = 0,\ldots,K$$

and for a 9-δ decomposition by

$$x_0 = y_0 = -1, \quad x_1 = y_1 = -1 + \delta, \quad x_2 = y_2 = 1 - \delta, \quad x_3 = y_3 = 1,$$

where $0 < \delta < 1$ is a given parameter (boundary distance).

Now the collocation nodes and the differentiation matrices on the kth element are given by

$$x_i^k := \frac{1}{2}[(x_k - x_{k-1})\xi_i + x_{k-1} + x_k], \quad y_j^k := \frac{1}{2}[(y_k - y_{k-1})\eta_j + y_{k-1} + y_k]$$

and

$$D_{1,k} := \frac{-2}{x_k - x_{k-1}} D_1, \quad D_{2,k} := \frac{-2}{y_k - y_{k-1}} D_2$$

with $i, j = 0, \ldots, N$, $k = 1, \ldots, K^2$.Fig. 1 shows the spectral element mesh for $K^2 = 9$ equidistant elements, $K^2 = 9$ CGL elements and 9-δ elements with $\delta = 10^{-1}$. In each case the polynomial degree is $N = 8$.

Next we have to realize the discrete formulation of Eqn. (7). This is performed by the Clenshaw–Curtis quadrature rule (see, e.g. [23]):

$$\int_\Omega p \, d\mathbf{x} \cong \sum_{i=0}^{N} \sum_{j=0}^{N} \omega_i \omega_j p(\xi_i, \eta_j),$$

where $\Omega = [-1, 1]^2$ denotes the standard domain, (ξ_i, η_j) the Chebyshev Gauss–Lobatto nodes on Ω and

$$\omega_i := \begin{cases} \frac{1}{N^2 - 1}, & i \in \{0, N\}, \\ \frac{4}{N} \sum_{j=0}^{\frac{N}{2}} \frac{1}{\bar{c}_j} \frac{\cos\left(\frac{2\pi i j}{N}\right)}{1 - 4j^2}, & 1 \leqslant i \leqslant N - 1 \end{cases}$$

with

$$\bar{c}_j := \begin{cases} 2, & j \in \{0, N/2\}, \\ 1, & 1 \leqslant j \leqslant N/2 - 1 \end{cases}$$

the integrations weights.

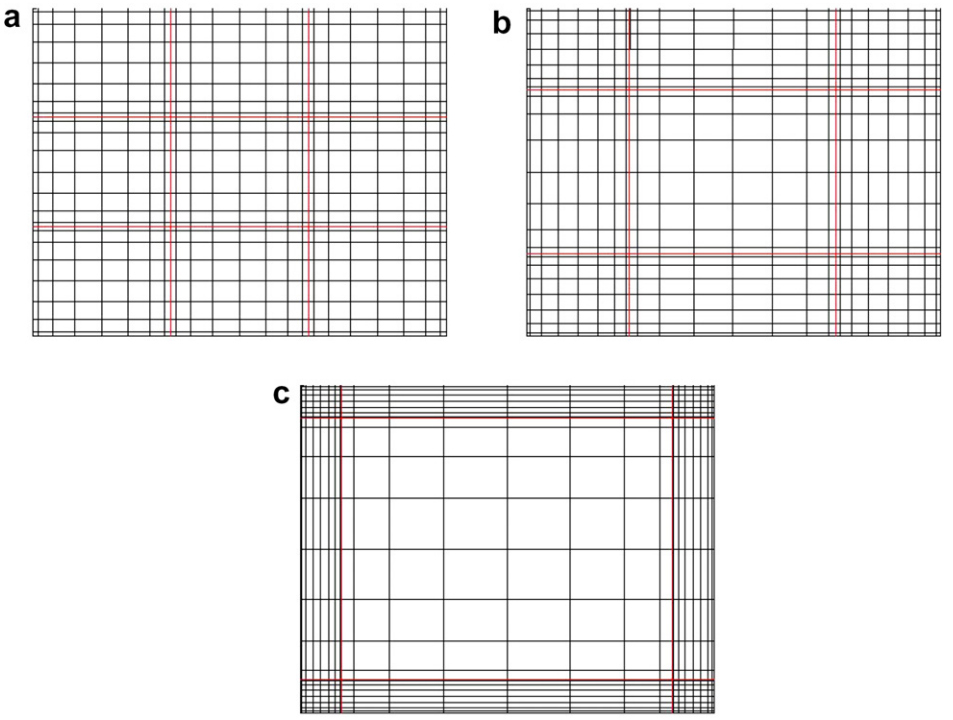

Fig. 1. (a) Equidistant; (b) CGL and (c) 9-δ ($\delta = 10^{-1}$) spectral element mesh for $K^2 = 9$ elements and polynomial degree $N = 8$.

We use the Clenshaw–Curtis quadrature rule since this is the appropriate quadrature rule for the Chebyshev Gauss–Lobatto nodes.

One could also use Gauss Legendre or Gauss Lobatto–Legendre nodes. In the numerical results there is no big difference. The advantage of the Chebyshev nodes is the fact that they are explicitly given and fast Fourier transforms (FFT) are available.

4. The discrete linear system of equations

At the interfaces between the elements, we require (as Heinrichs in [16]) continuity of both the functions and normal derivatives of u_1, u_2. For p we only require continuity and for ω we do not explicitly require interface conditions.

The corresponding discrete system of differential equations together with the discrete boundary and discrete interface conditions are written into a matrix A and compiled into an overdetermined system $Az = r$ where the matrix A is given by

$$A = \begin{pmatrix} A_1 & & & \\ & \ddots & & \\ & & A_{K^2} & \\ & M_I & & \\ & B & & \end{pmatrix}. \tag{18}$$

Here $A_k \in \mathbb{R}^{4(N+1)^2,4(N+1)^2}$, $k = 1,\ldots,K^2$, denotes the discrete version of the matrix in (9) or (11) or (14) depending on which equations are considered. The matrix $M_I \in \mathbb{R}^{10(N+1)(K^2-K),4K^2(N+1)^2}$ represents the discrete interface conditions and $B \in \mathbb{R}^{8NK,4K^2(N+1)^2}$ the given discrete boundary conditions for the velocity components u_1 and u_2 (see [21]). Since the pressure is only determined up to a constant we have to regularize the matrix A. One way is to cancel one condition for the pressure, i.e. we impose the pressure in one point. This matrix is denoted by $\widetilde{A}$. For simplicity we cancel the last row and the last column of A and so we impose the pressure in the point $(1,1)$ of the boundary. The disadvantage of this approach is, we do not know the pressure in one point for the regularized or lid-driven cavity flow. Nevertheless, in [29] the pressure constant was imposed to be zero in the middle of the cavity. This is a possible approach but since we want to discuss the incompressible (e.g. the integral over the pressure is equal zero) Navier–Stokes equations it seems to be the better way using (7). As our computations show we obtain smaller condition numbers of the linear system of equations using the discrete version of the additional pressure condition (7). Using this approach we obtain one additional row for the matrix A in (18) and denote this new matrix by $\widehat{A}$. The additional $1 \times 4K^2(N+1)^2$ matrix M_P is given by

$$M_P := (0,0,0,\bigstar,0,0,0,\bigstar,\ldots,0,0,0,\bigstar,0,0,0,\bigstar),$$

where $\bigstar$ (vector of length $(N+1)^2$) denotes the corresponding $(N+1)^2$ quadrature weights of the Clenshaw–Curtis quadrature formula and 0 a zero vector of length $(N+1)^2$. Since we particularly enforce the collocation conditions on the interfaces and on the boundary $\partial\Omega$ and we additionally enforce the pointwise continuity on the interfaces (represented by matrix M_I) and enforce the values of the velocity on the boundary (represented by matrix B) we get a really over-determined system. In order to verify that we achieve really over-determined linear systems of equations we compute the rank of $\widetilde{A}$ ($\operatorname{rank}(\widetilde{A})$) and the rank of $[\widetilde{A} \mid r]$ ($\operatorname{rank}([\widetilde{A} \mid r])$), i.e. the augmented matrix where we add the right hand side r as one column. The results of this computations are

$$\operatorname{rank}([\widetilde{A} \mid r]) = \operatorname{rank}(\widetilde{A}) + 1 > \operatorname{rank}(\widetilde{A})$$

and this shows $r \notin \mathcal{R}(\widetilde{A})$, where $\mathcal{R}(\widetilde{A})$ is the range of the matrix $\widetilde{A}$. For the matrix $\widehat{A}$ we obtain the same results. Hence, the systems are really over-determined and we need least-squares techniques to solve these linear systems of equations.

For the solution of the systems Heinrichs used, e.g. in [14–16] the normal equations $A^{\mathrm{T}}Az = A^{\mathrm{T}}r$.

It is well-known that the spectral derivative martices D have relatively large condition numbers

$$\kappa_2(D) = \frac{\max\limits_{\|x\|_2=1} \|Dx\|_2}{\min\limits_{\|x\|_2=1} \|Dx\|_2}$$

and the use of the corresponding normal equations lead to systems with even larger condition numbers ($\kappa_2(D^{\mathrm{T}}D) \cong \kappa_2(D)^2$). Because of the round-off errors and the large condition numbers of the systems one cannot obtain the best quality of approximations. Here we want to avoid the normal equations to get better approximations and so we make use of a direct solver for the system $Az = r$ by using the QR decomposition (computed with MATLAB 7.3.0), see, e.g. [30], of the matrix A where we achieved the following system:

$$Az = r \Longleftrightarrow QRz = r.$$

If $A \in \mathbb{R}^{m,n}$ with $m > n$ then $Q \in \mathbb{R}^{m,m}$ is an orthogonal matrix (i.e. $Q^{-1} = Q^{\mathrm{T}}$) and $R \in \mathbb{R}^{m,n}$ is an upper triangular matrix of the type

$$R = \begin{bmatrix} \widetilde{R} \\ \tilde{0} \end{bmatrix}, \quad \widetilde{R} \in \mathbb{R}^{n,n}, \quad \tilde{0} \in \mathbb{R}^{m-n,n}.$$

Consequently, we obtain

$$\psi := Q^{\mathrm{T}} r, \quad Rz = \psi.$$

Furthermore, we used the pseudoinverse, see, e.g. [30] (also known as Moore–Penrose inverse) A^+ of the matrix A to solve the system $Az = r$. The numerical experiments have shown that the approximation results by using the pseudoinverse are the same as using QR decomposition. The disadvantages of using pseudoinverses are higher computational costs (see Fig. 5). Form a theoretical point of view it is clear that all of the three solution techniques are equivalent if exact arithmetic is used. But since our linear systems of equations have very large condition numbers, the three techniques will not necessarily lead to the same results because of the influence of round-off errors.

The subject of a future project will be the development of some iterative solvers like incomplete QR decompositions. The problem of such iterative solvers is that they operate on the normal equations and that means the considered linear system of equations again have larger condition numbers.

5. Numerical simulations

We consider the steady and unsteady Stokes equations and the Navier–Stokes equations. For the Stokes equations we use our scheme only on equidistant elements since the other decompositions of the domain will not necessarily lead to better approximation results.

For the Navier–Stokes equations we first consider a smooth example on equidistant elements and then apply our scheme to the regularized and lid-driven cavity problem. For the regularized and lid-driven cavity problem we use our scheme on equidistant, CGL and 9-δ elements and compare the results.

5.1. The steady Stokes equations

First, we consider the steady case of the Stokes equations. Fig. 2 shows the condition numbers $\kappa_2(\widehat{A})$, $\kappa_2(\widetilde{A})$ and $\kappa_2(\widehat{A}^{\mathrm{T}}\widehat{A})$ for $K^2 = 4$, $K^2 = 36$ and $K^2 = 64$ elements with different polynomial degrees N. For higher element numbers and high polynomial degrees we do not list the condition numbers because of the influence of the round-off errors.

The condition numbers are rigorously reduced if we do not use the normal equations. A further reduction of the condition numbers is obtained if we use the Clenshaw–Curtis quadrature rule to avoid the pressure constant instead of cancelling one row and one column of the complete system matrix.

For other parameters K^2 and N we obtain similar results and so we do not list them here.

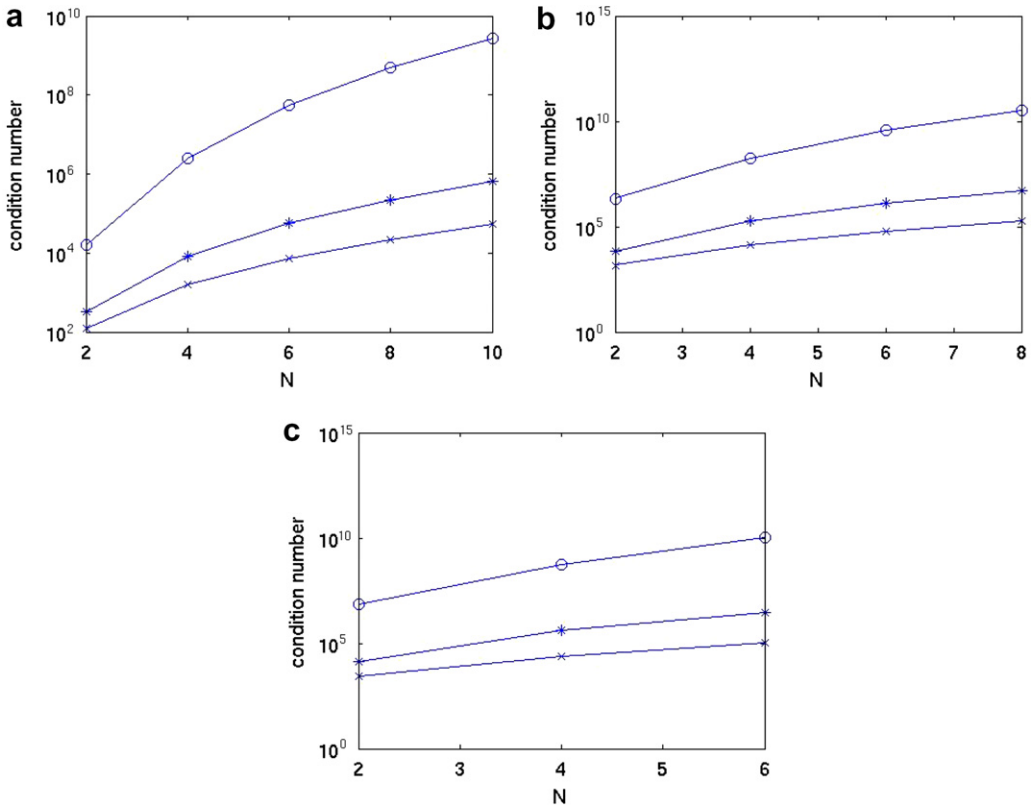

Fig. 2. Condition numbers for: (a) $K^2 = 4$; (b) $K^2 = 36$ and (c) $K^2 = 64$ equidistant elements and different polynomial degrees N ($\kappa_2(\widehat{A})$, $\times$; $\kappa_2(\widetilde{A})$, $*$; $\kappa_2(\widehat{A}^{\mathrm{T}}\widehat{A})$, $\circ$).

The convergence rates of the least-squares spectral collocation scheme are demonstrated by means of the model problem also introduced in [11] with $\nu = 1$. The exact velocity components and the pressure are defined on the square $\Omega := [-1,1]^2$ by

$$u_1(x,y,t) := \cos(\gamma t)\sin\left(\frac{\pi x}{2}\right)\cos\left(\frac{\pi y}{2}\right), \tag{19}$$

$$u_2(x,y,t) := -\cos(\gamma t)\cos\left(\frac{\pi x}{2}\right)\sin\left(\frac{\pi y}{2}\right), \tag{20}$$

$$p(x,y,t) := \frac{1}{4}\cos^2(\gamma t)(\cos(\pi x)+\cos(\pi y)) + 10(x+y)\cos(\gamma t). \tag{21}$$

This exact solution satisfies the Stokes equations if the following forcing term is used

$$\mathbf{f}(x,y,t) = \begin{pmatrix} \nu\frac{\pi^2}{2}\cos(\gamma t)\sin\left(\frac{\pi x}{2}\right)\cos\left(\frac{\pi y}{2}\right) \\ -\nu\frac{\pi^2}{2}\cos(\gamma t)\cos\left(\frac{\pi x}{2}\right)\sin\left(\frac{\pi y}{2}\right) \end{pmatrix} - \begin{pmatrix} \frac{\pi}{4}\cos^2(\gamma t)\sin(\pi x) - 10\cos(\gamma t) \\ \frac{\pi}{4}\cos^2(\gamma t)\sin(\pi y) - 10\cos(\gamma t) \end{pmatrix} + \begin{pmatrix} -\gamma\sin(\gamma t)\sin\left(\frac{\pi x}{2}\right)\cos\left(\frac{\pi y}{2}\right) \\ \gamma\sin(\gamma t)\cos\left(\frac{\pi x}{2}\right)\sin\left(\frac{\pi y}{2}\right) \end{pmatrix}. \tag{22}$$

For the steady case of the Stokes equations we set $\gamma = 0$.

We use the QR decomposition of the matrix $\widehat{A}$ to solve the discrete algebraic systems and numerically calculate the discrete L^2-error norms of the velocity components and the pressure. The corresponding numerical results are presented in the Tables 1–3.

Tables 1–3 show the high spectral accuracy of our scheme if the number of elements is constant and the polynomial degree increases. If we compare the approximation errors of the same polynomial degree with different numbers of elements we observe the expected slight improvement in the results.

Table 1
L^2-errors of the velocity components and the pressure on $K^2 = 4$ equidistant elements

N	$\|u_1 - u_1^N\|_{L^2}$	$\|u_2 - u_2^N\|_{L^2}$	$\|p - p^N\|_{L^2}$
$N = 2$	1.797×10^{-1}	1.809×10^{-1}	6.088×10^{-1}
$N = 4$	3.082×10^{-3}	3.042×10^{-3}	5.869×10^{-2}
$N = 6$	1.340×10^{-4}	1.311×10^{-4}	7.061×10^{-3}
$N = 8$	2.885×10^{-6}	2.844×10^{-6}	1.939×10^{-4}
$N = 10$	1.938×10^{-8}	1.936×10^{-8}	2.008×10^{-6}
$N = 12$	5.621×10^{-11}	5.622×10^{-11}	8.636×10^{-9}
$N = 14$	4.425×10^{-13}	4.689×10^{-13}	4.852×10^{-11}
$N = 16$	4.316×10^{-13}	4.041×10^{-13}	2.316×10^{-11}
$N = 18$	5.207×10^{-13}	4.977×10^{-13}	8.734×10^{-11}
$N = 20$	6.956×10^{-13}	7.618×10^{-13}	8.395×10^{-11}

Table 2
L^2-errors of the velocity components and the pressure on $K^2 = 36$ equidistant elements

N	$\|u_1 - u_1^N\|_{L^2}$	$\|u_2 - u_2^N\|_{L^2}$	$\|p - p^N\|_{L^2}$
$N = 2$	4.280×10^{-2}	4.434×10^{-2}	4.128×10^{-1}
$N = 4$	4.101×10^{-5}	4.448×10^{-5}	1.264×10^{-3}
$N = 6$	3.285×10^{-7}	3.147×10^{-7}	1.082×10^{-5}
$N = 8$	9.049×10^{-10}	8.933×10^{-10}	8.119×10^{-8}

Table 3
L^2-errors of the velocity components and the pressure on $K^2 = 64$ equidistant elements

N	$\|u_1 - u_1^N\|_{L^2}$	$\|u_2 - u_2^N\|_{L^2}$	$\|p - p^N\|_{L^2}$
$N = 2$	2.711×10^{-2}	2.884×10^{-2}	3.382×10^{-1}
$N = 4$	1.425×10^{-5}	1.562×10^{-5}	5.011×10^{-4}
$N = 6$	6.291×10^{-8}	6.072×10^{-8}	2.335×10^{-6}

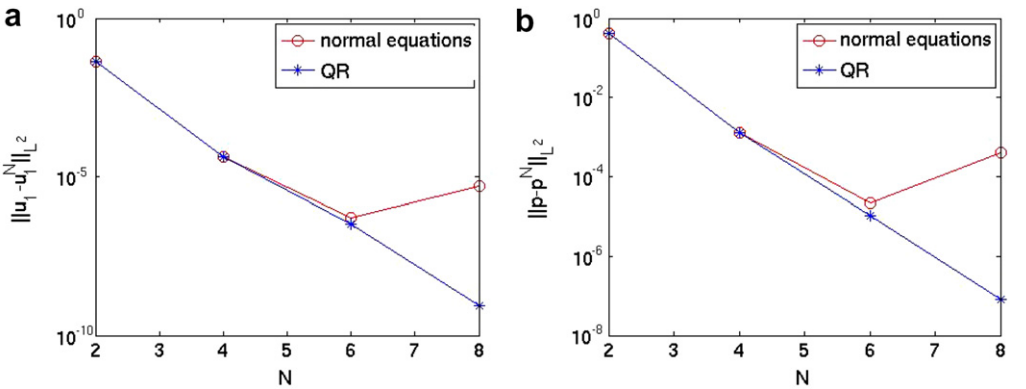

Fig. 3. L^2-error of: (a) the velocity component u_1 and (b) the pressure, obtained by the normal equations and by the QR decomposition on $K^2 = 36$ equidistant elements.

Table 1 obviously shows the influence of round-off errors for $N \geqslant 16$.

In the Figs. 3 and 4 we compare the approximation errors by solving the normal equations $\widehat{A}^{\mathrm{T}}\widehat{A}z = \widehat{A}^{T}r$ and by solving the system $\widehat{A}z = r$ with QR decomposition. We obtain the same errors if the polynomial degree is low, i.e. $N = 2, 4$, but if the polynomial degree increases the errors obtained by normal equations increase if a particular N is exceeded. The reason of this behaviour are the high condition numbers of the normal equations and thus the strong influence of round-off errors. Errors obtained by QR decomposition still decrease for increasing N. By using QR decomposition we avoid the very high condition numbers and so we obtain the improved approximations.

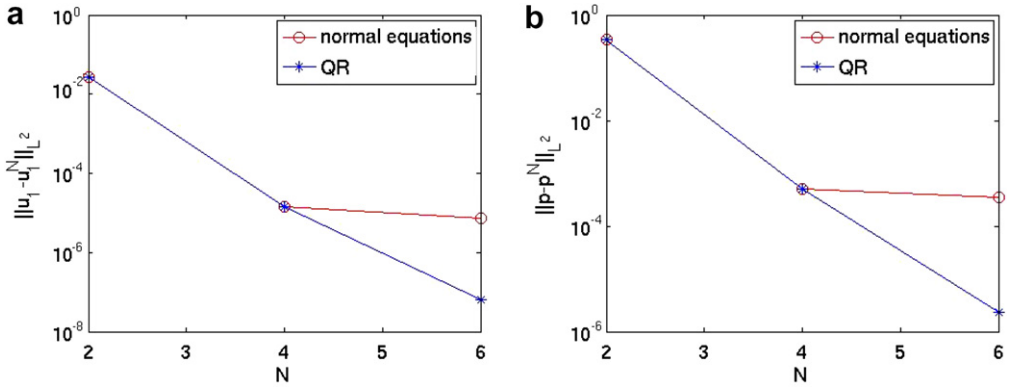

Fig. 4. L^2-error of: (a) the velocity component u_1 and (b) the pressure, obtained by the normal equations and by the QR decomposition on $K^2 = 64$ equidistant elements.

Since we have an analogue performance of the velocity component u_2 we here just show the results for u_1.

Next we check the numerical results obtained by solving the system of equations with the pseudoinverse $\widehat{A}^+$ of the matrix $\widehat{A}$. We got the same numerical results as by using QR decomposition and so we do not show the results here.

For comparing computational costs we show in Fig. 5 the required CPU-times for solving the system of equations by normal equations, QR decomposition and pseudoinverse. All results are computed with MATLAB 7.3.0 with the following code:

- normal equations: $\mathtt{z} = (\widehat{\mathtt{A}}' * \widehat{\mathtt{A}}) \setminus (\widehat{\mathtt{A}}' * \mathtt{r})$

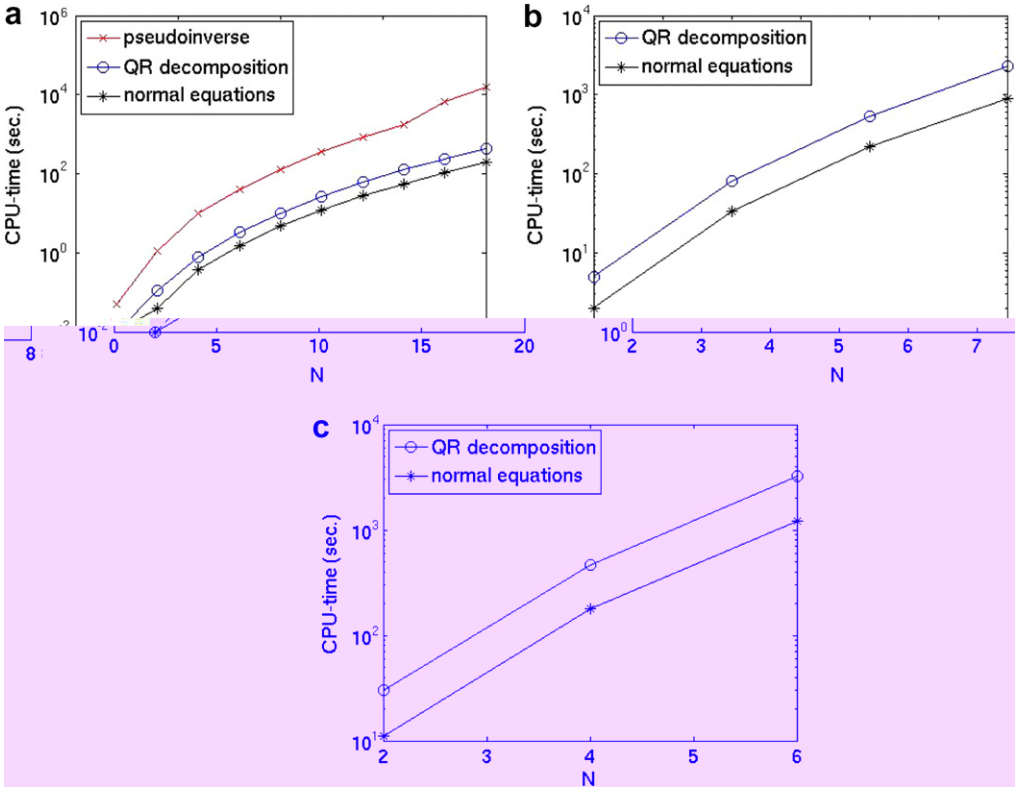

Fig. 5. CPU-time required for solving the system of linear equations with pseudoinverse (×), QR decomposition (○) and the normal equations (∗) on: (a) 4, (b) 36 and (c) 64 equidistant elements.

- QR decomposition: $\mathtt{z} = \widehat{\mathtt{A}} \setminus \mathtt{r}$
- pseudoinverse: $\mathtt{z} = \mathtt{pinv}(\widehat{\mathtt{A}}) * \mathtt{r}$

Since the use of the pseudoinverse needs extremely more time and produces same accuracy as the use of QR decomposition we only used QR decomposition for solving the systems. The application of normal equations is less expensive but the quality of approximation is worse than the one obtained by using QR decomposition. Furthermore, for time-dependent Stokes or Navier–Stokes problems we use an explicit scheme and so the QR decomposition of the linear system of equations must only be computed once. Hereafter, only matrix-vector multiplications are performed. In the following section we apply our scheme to the unsteady Stokes equations.

5.2. The unsteady Stokes equations

Now we consider the unsteady case of the Stokes equations. The exact velocity components and the pressure are defined as in (19)–(21), the corresponding forcing term by (22) and we recall that $v = \frac{1}{Re}$ where Re denotes the Reynolds number. Because we consider the unsteady case we set $\gamma = 5$ as in [11]. From [14] it is known that for a well balanced system it is recommended to scale the momentum equations by Δt. By numerical experiments we observed the same. Without scaling the incompressibility condition is no more fulfilled after time integration. Fig. 6 shows $\|\nabla \cdot \mathbf{u}\|$ on $K^2 = 4$ equidistant elements with $N = 10$, $Re = 1$ and $\Delta t = \frac{1}{58}$ in case without scaling. In the scaled case we obtain a stable scheme. Figs. 7 and 8 show the temporal evolution of the L^2-errors in the velocity components and in the pressure. We observe no enlargement of the oscillating errors in time, expressing stability of the numerical solution.

Since the numerical results in the unsteady case are similar to those in the steady case we here only show a few results. In Table 4 we demonstrate the condition numbers of the matrices $\widehat{A}$ and recall that $\kappa_2(\widehat{A}^{\mathrm{T}}\widehat{A}) \cong \kappa_2(\widehat{A})^2$.

In the Tables 5–7 we show the approximation errors for the unsteady case of the Stokes equations and we see the good performance of the here presented scheme for time-dependent Stokes problems. The time derivative is approximated by the second order BDF scheme (8). We set

$$E_{u_1} := \max\{\|u_1 - u_1^N\|_{L^2} : \ t \in [0,1]\},$$
$$E_{u_2} := \max\{\|u_2 - u_2^N\|_{L^2} : \ t \in [0,1]\},$$
$$E_p := \max\{\|p - p^N\|_{L^2} : \ t \in [0,1]\},$$

since Figs. 7 and 8 show that the maximum error is obtained in [0,1].

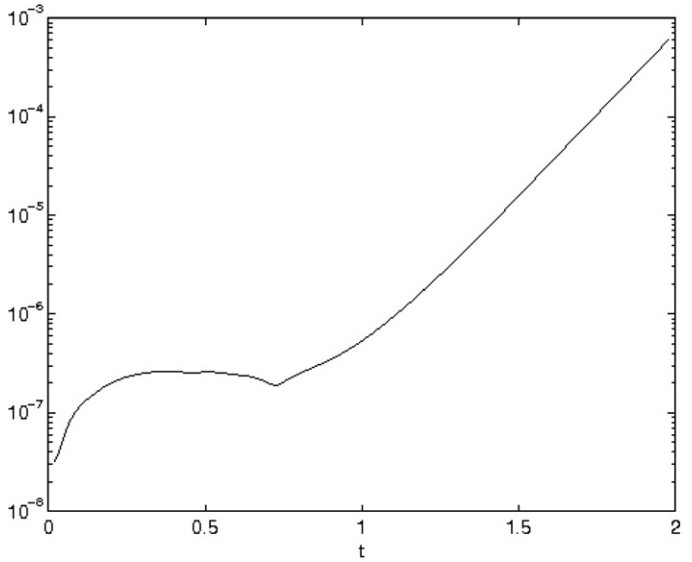

Fig. 6. Temporal evolution of $\|\nabla \cdot \mathbf{u}\|$ on $K^2 = 4$ equidistant elements with $N = 10$, $Re = 1$, $\Delta t = \frac{1}{58}$ in the case without scaling.

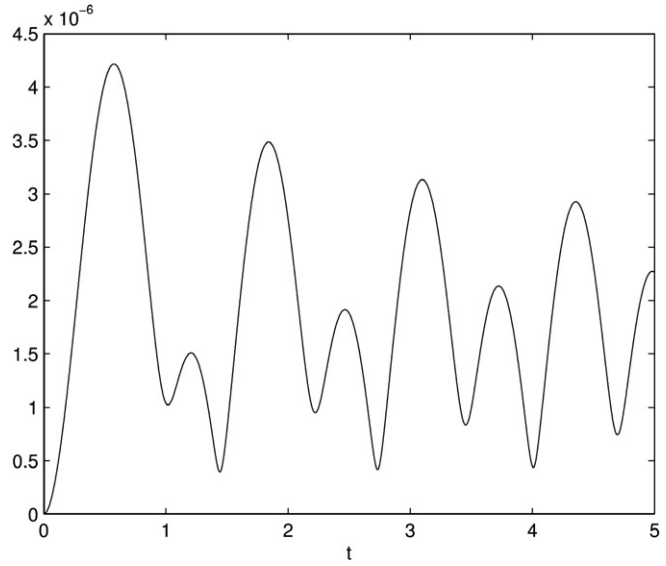

Fig. 7. Temporal evolution of the velocity error for $Re = 100$, $\Delta t = \frac{1}{1000}$ and $N = 10$ on $K^2 = 4$ equidistant elements.

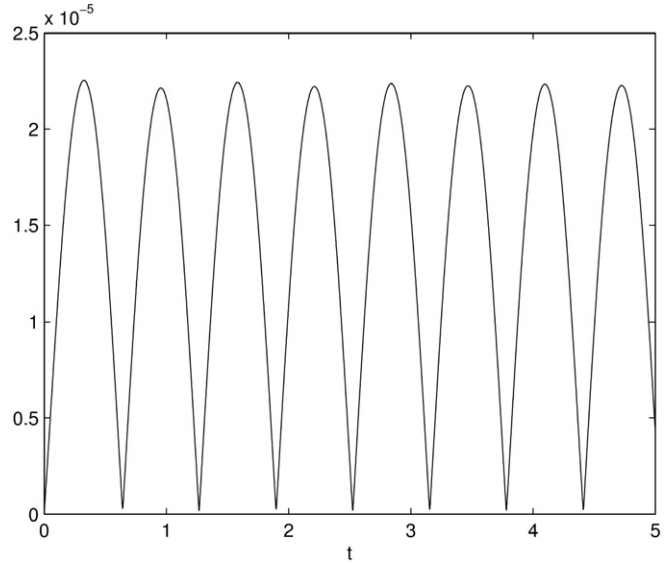

Fig. 8. Temporal evolution of the pressure error for $Re = 100$, $\Delta t = \frac{1}{1000}$ and $N = 10$ on $K^2 = 4$ equidistant elements.

Table 4
Condition numbers of the scaled matrix $\widehat{A}$ on $K^2 = 4$ equidistant elements with $N = 10$, different Reynolds numbers and different time steps

Re	Δt	$\kappa_2(\widehat{A})$
100	$\frac{1}{10}$	3.1067×10^4
100	$\frac{1}{100}$	9.0105×10^4
1000	$\frac{1}{100}$	8.7253×10^4

Since we use a second order scheme in time, the ratio of, e.g. E_{u_1} with time step size h and E_{u_1} with time step size $h/2$ must approximate $2^{\lambda} = 4$, where $\lambda = 2$ denotes the order of the scheme.

Table 5
Approximation errors on $K^2 = 4$ equidistant elements with $N = 10$ and $Re = 1$

Δt	$E_{u_1} = E_{u_2}$	Ratio	E_p	Ratio
$\frac{1}{10}$	2.885×10^{-3}	–	2.496×10^{-1}	–
$\frac{1}{20}$	7.386×10^{-4}	3.906	6.331×10^{-2}	3.943
$\frac{1}{40}$	1.858×10^{-4}	3.975	1.586×10^{-2}	3.992
$\frac{1}{80}$	4.657×10^{-5}	3.990	3.973×10^{-3}	3.992

Table 6
Approximation errors on $K^2 = 36$ equidistant elements with $N = 8$ and $Re = 100$

Δt	$E_{u_1} = E_{u_2}$	Ratio	E_p	Ratio
$\frac{1}{10}$	1.326×10^{-1}	–	5.354×10^{-1}	–
$\frac{1}{20}$	3.563×10^{-2}	3.722	1.358×10^{-1}	3.943
$\frac{1}{40}$	9.085×10^{-3}	3.922	3.404×10^{-2}	3.989
$\frac{1}{80}$	2.281×10^{-3}	3.983	8.532×10^{-3}	3.990

Table 7
Approximation errors on $K^2 = 64$ equidistant elements with $N = 6$ and $Re = 100$

Δt	$E_{u_1} = E_{u_2}$	Ratio	E_p	Ratio
$\frac{1}{10}$	1.890×10^{-1}	–	7.322×10^{-1}	–
$\frac{1}{20}$	5.067×10^{-2}	3.730	1.855×10^{-1}	3.947
$\frac{1}{40}$	1.292×10^{-2}	3.922	4.656×10^{-2}	3.984
$\frac{1}{80}$	3.244×10^{-3}	3.983	1.166×10^{-2}	3.993

5.3. The Navier–Stokes equations

Now we apply our scheme to a smooth example for the steady Navier–Stokes equations where the velocity and the pressure are given by (19)–(21) with $\gamma = 0$ and $\Omega = [0,1]^2$. Since we have seen that the additional pressure condition leads to linear system of equations with smaller condition numbers we will only use the matrices $\widehat{A}$. Furthermore, we have seen that solving the linear systems of equations with QR decomposition leads to better approximations. Hence, we will now only use the QR decomposition for solving the linear systems of equations.

Since the numerical results for the velocity component u_2 are similar to those of the velocity component u_1 we show only the results for u_1. All results are obtained by the explicit scheme.

In Fig. 9 we show the L^2-error norms of the velocity component u_1 and the pressure on 64 equidistant elements with $Re = 1$.

In Fig. 10 we show the L^2-errors for the velocity component u_1 and for the pressure on 64 CGL elements with $Re = 1$. Fig. 11 shows the L^2-errors for the velocity component u_1 and for the pressure on 9-δ elements with $Re = 1$ and $\delta = 10^{-1}$. Figs. 9–11 show the high spectral accuracy of our scheme on all here considered domain decompositions. In Fig. 11 we observe the influence of round-off errors for $N \geqslant 14$.

5.4. The regularized cavity flow

Now we set $\Omega := (0,1)^2$ and consider the regularized cavity flow (see [4]) where the fluid velocity on the edge $y = 1$ is given by

$$u_1(x,1) := -16x^2(1-x)^2, \quad u_2(x,1) := 0 \tag{23}$$

and $u_1 = u_2 = 0$ on the other three edges. The forcing term $\mathbf{f}$ is identical to zero. As Botella in [4] we consider that the steady state is reached when

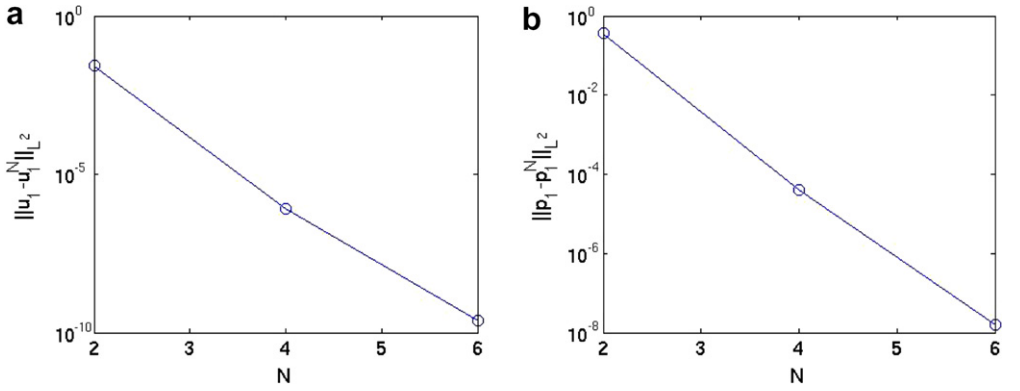

Fig. 9. L^2-error of: (a) the velocity component u_1 and the (b) the pressure on $K^2 = 64$ equidistant elements with $Re = 1$.

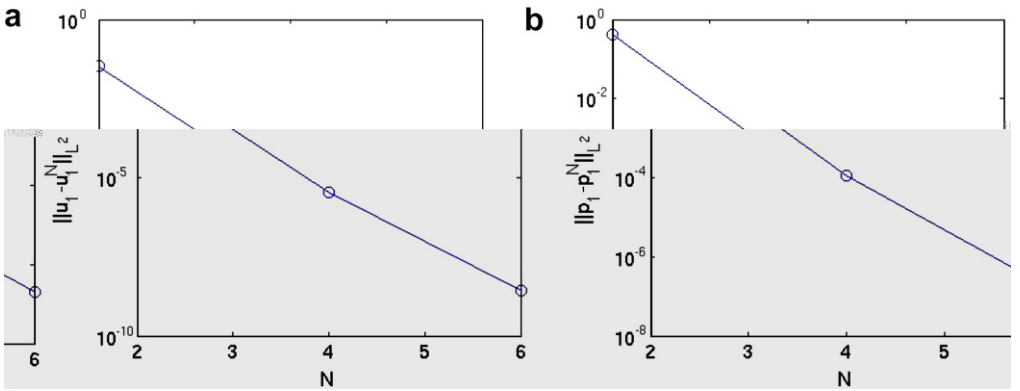

Fig. 10. L^2-error of: (a) the velocity component u_1 and the (b) the pressure on $K^2 = 64$ CGL elements with $Re = 1$.

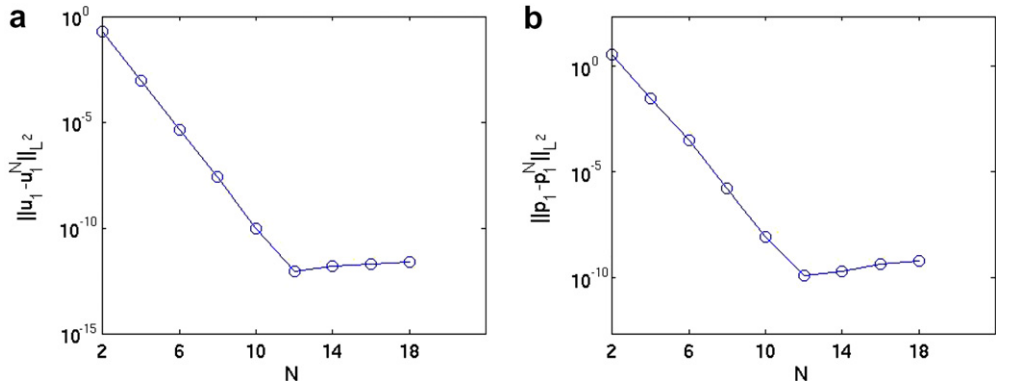

Fig. 11. L^2-error of: (a) the velocity component u_1 and the (b) the pressure on 9-δ elements with $Re = 1$ and $\delta = 10^{-1}$.

$$\frac{\max\limits_{i,j} \mid \omega_{i,j}^{n+1} - \omega_{i,j}^{n} \mid}{\Delta t \cdot \max\limits_{i,j} \mid \omega_{i,j}^{n+1} \mid} \leqslant 2 \times 10^{-6}. \tag{24}$$

Hereafter we calculate the stream function by solving the equation

$$\begin{aligned} &\Delta\psi = -\omega \quad \text{in } \Omega = (0,1)^2, \\ &\psi = 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{25}$$

Solving this equation we first transform (25) into an equivalent first-order system. This system with the pointwise enforced continuity on the element interfaces and the boundary conditions lead to an overdetermined system that is solved by our least-squares scheme. As the simulations have shown it is recommended to scale the boundary conditions of (25) by $\|A_1^{\text{stream}}\|_\infty$, otherwise the streamfunction is not accurately zero on the boundary $\partial\Omega$ since the boundary conditions are not well balanced against the other conditions. Here A_1^{stream} denotes the spectral system matrix of the first-order system of (25) on the first element (lower left corner). Computations to solve (25) are performed with the QR decomposition, too.

In order to compare our results with those of Botella in [4], we calculate the maximal value of ψ on the collocation points of Ω, denoted by M_1. Furthermore, we computed the maximal value of ω on the edge $y = 1$. This value is denoted by M_2. The benchmarks are shown in the last row of the tables.

Table 8 shows the results for $Re = 100$ on equidistant and CGL elements. The coordinates of the point where the values are reached are given in brackets.

Comparing our results with the results of Botella, we see the good performance of our scheme. Furthermore, we see that our scheme on CGL elements leads to better results than on equidistant elements. The reason is that the CGL elements are closer to the boundary and this is the area where the functions are not as smooth as in the middle of the cavity.

In the approach of Botella, there are only two unknown functions and one element with $N = 48$, which results in $2 \cdot 2401$ degrees of freedom. For a similar accuracy, the least-squares approach requires 64 elements with $N = 6$, which results in $4 \cdot 3136$ d.o.f. Hence our scheme is more expensive for the regularized cavity flow. But for more complex geometries and singularities (e.g. the lid-driven cavity, see Section 5.5), our approach is more favorable.

Next we perform our scheme on 9-δ elements for various δ. The results are shown in Table 9. For singular perturbation problems it is well-known that one has to take $\delta \cong N \cdot \varepsilon$ where ε denots the diffusion parameter. For the Navier–Stokes equations ($\varepsilon = \nu = 1/Re$) it is not clear how to choose δ and so we have chosen $\delta = N \cdot \nu = N/Re$ in the first row of Table 9 and in the other rows we have chosen different δ. As Table 9 shows it seems to be the best choice to choose $\delta = 0.18$ or $\delta = 0.1$ depending on what should be approximated better (M_1 or M_2). Comparing Tables 8 and 9 we see that the CGL elements are preferable. The approximations of the benchmarks show the well-known different performance for various δ.

In Table 10 we check our scheme for $Re = 400$ on equidistant and CGL elements and compare our approximations with the corresponding results of Botella in [4].

Since the vorticity ω has strong variation on the points of the edge $y = 1$ and is not as smooth as the stream function ψ we do not obtain such good approximations of M_2 compared to the approximations of M_1. Furthermore, the CGL elements lead to better results than the equidistant ones.

In Table 11 we show the results for the 9-δ decomposition for various δ with $Re = 400$. As in Table 9 the first row of Table 11 shows the results for $\delta = N/Re$. In the other rows of Table 11 we show the results for

Table 8
The regularized cavity flow with $Re = 100$

K^2	N	M_1	M_2	Δt	Elements
64	6	8.36×10^{-2} (0.38/0.76)	13.48 (0.62)	1/50	Equidistant
64	6	8.33×10^{-2} (0.40/0.77)	13.32 (0.60)	1/50	CGL
1	48	8.34×10^{-2} (0.40/0.74)	13.34 (0.60)	[4]	

Results with explicit time integration on equidistant and CGL elements with polynomial degree $N = 6$ and time step size $\Delta t = 1/50$.

Table 9
The regularized cavity flow with $Re = 100$

δ	M_1	M_2
0.18	8.36×10^{-2} (0.39/0.75)	13.42 (0.61)
0.1	8.31×10^{-2} (0.36/0.76)	13.38 (0.64)
0.075	8.26×10^{-2} (0.43/0.77)	13.30 (0.65)
[4]	8.34×10^{-2} (0.40/0.74)	13.34 (0.60)

Results with explicit time integration on 9-δ elements with polynomial degree $N = 18$ and time step size $\Delta t = 1/50$.

Table 10
The regularized cavity flow with $Re = 400$

K^2	N	M_1	M_2	Δt	Elements
64	6	8.63×10^{-2} (0.44/0.62)	25.61 (0.62)	1/100	Equidistant
64	6	8.55×10^{-2} (0.40/0.60)	24.81 (0.64)	1/100	CGL
1	48	8.55×10^{-2} (0.40/0.60)	24.78 (0.65)	[4]	

Results with explicit time integration on equidistant and CGL elements with polynomial degree $N = 6$ and time step size $\Delta t = 1/100$.

Table 11
The regularized cavity flow with $Re = 400$

δ	M_1	M_2
0.045	8.52×10^{-2} (0.42/0.58)	24.58 (0.66)
0.075	8.55×10^{-2} (0.43/0.65)	24.78 (0.65)
0.1	8.56×10^{-2} (0.43/0.64)	24.88 (0.64)
[4]	8.55×10^{-2} (0.40/0.60)	24.78 (0.65)

Results with explicit time integration on 9-δ elements with polynomial degree $N = 18$ and time step size $\Delta t = 1/100$.

various δ. Comparing the values of Table 11 it seems to be the best choice to choose $\delta = 0.075$. Again, the approximations of the benchmarks show the well-known different performance for various δ.

Comparing Tables 8, 9 and 10, 11 we see that the CGL elements are especially preferable for increasing Reynolds numbers.

Obviously, our explicit scheme works well for Reynolds numbers up to 400. Nevertheless, we have performed our computations with the implicit scheme introduced in Section 2.2 on 4, 9, 16 and 25 equidistant elements and compared the results obtained with the explicit scheme. We have seen that the implicit scheme leads to the same approximation results. The well-known disadvantage of the implicit scheme is that we have to solve the linear systems of equations in each time step. This requires a lot of CPU-time. The number of Picard iterations is 6 in the first time step and two in the last time step. From the literature it is well-known that these are the usual numbers of linearization steps for the implicit scheme.

5.5. The lid-driven cavity flow

Now we numerically solve the lid-driven cavity flow problem where the fluid velocity on the edge $y = 1$ is given by

$$u_1(x,1) := -1, \quad u_2(x,1) := 0$$

and $u_1 = u_2 = 0$ on the other three edges. The source term $\mathbf{f}$ is identical to zero.

Since the velocity is discontinuous at the two upper corners $A(0,1)$ and $B(1,1)$ of the square box $\Omega = (0,1)^2$ the solution of the Navier–Stokes equations becomes singular at these corners. In particular, the vorticity and the pressure becomes infinite at these two points. Because of the sharp gradients of the variables and the singularities at these corners the lid-driven cavity flow is a difficult test case. Furthermore, there are singularities at the points $C(0,0)$ and $D(1,0)$ but they are much more weaker than the ones at A and B since only the second derivatives of the pressure and the vorticity are unbounded.

Here we understand the lid-driven cavity flow as an stationary problem and, as in [5], we assume that the steady state is reached if

$$\frac{\max \mid \phi^{n+1} - \phi^n \mid}{\Delta t \cdot \max \mid \phi^{n+1} \mid} \leqslant 10^{-8}, \tag{26}$$

where $\phi := (u_1, u_2)^{\mathrm{T}}$. When (26) is fulfilled we calculate the stream function ψ by

$$\begin{aligned} -\Delta\psi &= \omega \quad \text{in } \Omega = (0,1)^2, \\ \psi &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{27}$$

Computations to solve (27) are performed as in the previous section.

For the regularized cavity flow we have seen that the explicit scheme is preferable compared to the implicit scheme since it leads to the same approximation results with an enormous less amount of computing time. Hence, we here present only the results obtained with the explicit scheme.

In Table 12 we show the extrema of the velocity components on the centerlines $x = 0.5$ and $y = 0.5$ and $\omega(0.5, 0.5)$ for $Re = 100$ on equidistant and CGL elements with $N = 6$. In the last row of the tables we show the corresponding benchmarks of Botella and Peyret in [5]. Comparing our results with the benchmarks in Table 12 we again see the good performance of our scheme. Furthermore, we see that for $Re = 100$ we do not obtain analog improvements on CGL elements as for the regularized cavity flow. The reason is that the CGL elements are bigger than the equidistant ones in the middle of the cavity and if the Reynolds number is relatively low the solution is not as smooth as for higher Reynolds numbers in the middle of the cavity. That means, the non-smooth parts extend more into the cavity. As a result of this, we expect better approximations on CGL elements than on equidistant elements if we increase the Reynolds number since then the non-smooth parts do not extend so far into the cavity.

In Table 13 we show the results for the 9-δ decomposition for various δ. As for the regularized cavity flow we have chosen $\delta = N/Re$ in the first row of Table 13. In the other rows we show the results for various δ. Table 13 shows that it seems to be the best choice to choose $\delta = 0.10$ to obtain good approximations for all values. The approximations of the benchmarks show the well-known different performance for various δ.

Comparing Tables 12 and 13 we see the better performance of our scheme on equidistant elements.

Next, we check our scheme for $Re = 1000$. Table 14 shows the extrema of the velocity components on the centerlines $x = 0.5$ and $y = 0.5$ for $Re = 100$ on equidistant and CGL elements with $N = 6$. Comparing our results with the benchmarks we see the good approximations obtained by our scheme. Furthermore, we see that the CGL elements lead to extremely better results. The reason that our values are not as good as the benchmarks is that we have placed all variables in the singularities. However, in [5] a subtraction method was used. More precisely, a subtraction method of the leading terms of the asymptotic expansion of the solution in the vincinity of the corners, where the velocity is discontinuous, was used.

Table 12
Extrema of the velocity for the lid-driven cavity flow through the centerlines $x = 0.5$ and $y = 0.5$ of the cavity and $\omega(0.5, 0.5)$ for $Re = 100$ on 64 equidistant and CGL elements

$u_{1,\max}$, $y_{\max}$	$u_{2,\max}$, $x_{\max}$	$u_{2,\min}$, $y_{\min}$	$\omega(0.5, 0.5)$	Δt	Elements
0.211, 0.469	0.178, 0.758	−0.253, 0.188	1.173	1/50	Equidistant
0.212, 0.452	0.178, 0.772	−0.251, 0.187	1.160	1/50	CGL
0.214, 0.458	0.180, 0.763	−0.254, 0.190	1.174	[5]	

Table 13
Extrema of the velocity for the lid-driven cavity flow through the centerlines $x = 0.5$ and $y = 0.5$ of the cavity and the value of the vorticity in the middle of the cavity for $Re = 100$ on 9-δ elements with polynomial degree $N = 18$ and time step size $\Delta t = 1/50$

δ	$u_{1,\max}$, $y_{\max}$	$u_{2,\max}$, $x_{\max}$	$u_{2,\min}$, $y_{\min}$	$\omega(0.5, 0.5)$
0.18	0.212, 0.444	0.178, 0.777	−0.252, 0.185	1.168
0.10	0.211, 0.431	0.179, 0.757	−0.253, 0.194	1.171
0.075	0.211, 0.426	0.179, 0.773	−0.252, 0.174	1.172
0.05	0.210, 0.422	0.178, 0.789	−0.250, 0.211	1.172
[5]	0.214, 0.458	0.180, 0.763	−0.254, 0.190	1.174

Table 14
Extrema of the velocity for the lid-driven cavity flow through the centerlines $x = 0.5$ and $y = 0.5$ of the cavity for $Re = 1000$ on 64 equidistant and CGL elements

$u_{1,\max}$, $y_{\max}$	$u_{2,\max}$, $x_{\max}$	$u_{2,\min}$, $y_{\min}$	Δt	Elements
0.500, 0.188	0.497, 0.867	−0.640, 0.094	1/300	Equidistant
0.397, 0.157	0.391, 0.843	−0.538, 0.092	1/300	CGL
0.389, 0.172	0.377, 0.842	−0.527, 0.091	[5]	

Table 15
Extrema of the velocity for the lid-driven cavity flow through the centerlines $x = 0.5$ and $y = 0.5$ of the cavity for $Re = 1000$ on 9-δ elements with polynomial degree $N = 18$ and time step size $\Delta t = 1/500$

δ	$u_{1,\max}$, $y_{\max}$	$u_{2,\max}$, $x_{\max}$	$u_{2,\min}$, $y_{\max}$
0.018	0.383, 0.190	0.367, 0.810	−0.519, 0.083
0.075	0.385, 0.174	0.371, 0.826	−0.517, 0.101
0.10	0.383, 0.194	0.377, 0.846	−0.528, 0.093
[5]	0.389, 0.172	0.377, 0.842	−0.527, 0.091

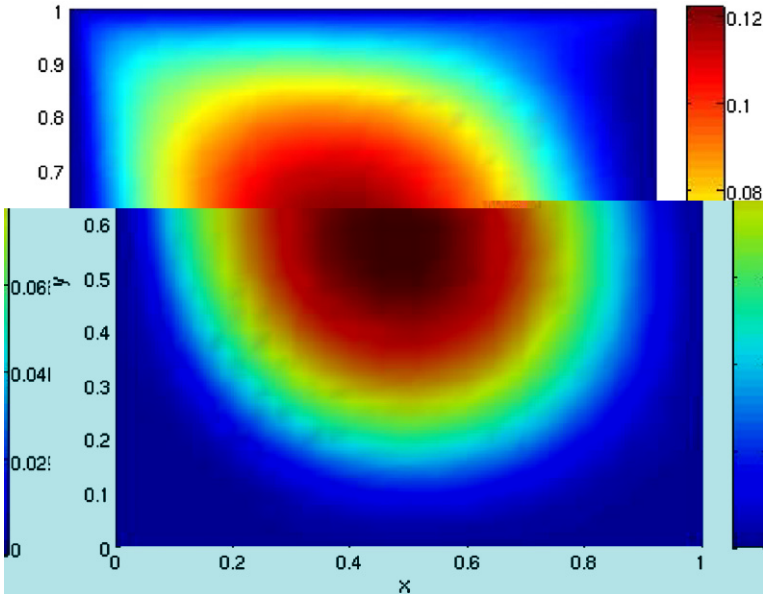

Fig. 12. Stream function of the lid-driven cavity flow for $Re = 1000$ on 64 CGL elements with polynomial degree 6 and time step size 1/300.

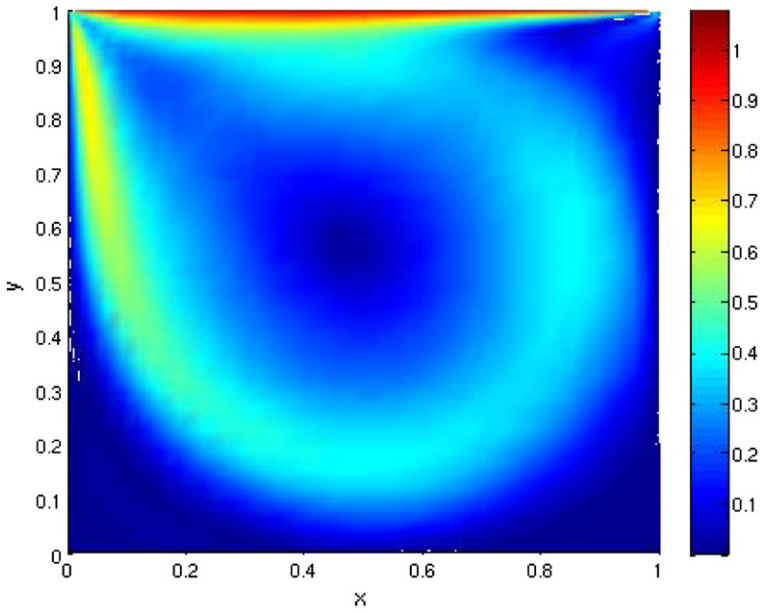

Fig. 13. Velocity profile of the lid-driven cavity flow for $Re = 1000$ on 64 CGL elements with polynomial degree 6 and time step size 1/300.

Table 15 shows the results obtained on 9-δ elements with polynomial degree $N = 18$ and time step size $\Delta t = 1/500$ for $Re = 1000$. Choosing bigger time step sizes leads to a divergent scheme. Since we consider stationary problems, the time step size does not influence the values of our computations.

The first row of Table 15 shows the results for $\delta = N/Re$. Here it seems to be the best choice to choose $\delta = 0.10$ to obtain good approximations for all values. Again, the approximations of the benchmarks show the well-known different performance for various δ.

Comparing Tables 14 and 15 we see the better performance of our scheme on 9-δ elements. The disadvantage of 9-δ elements is that we have to choose δ carefully.

Comparing the results for the regularized and lid-driven cavity flow we recommend to use CGL elements since they lead to good approximations and they are not depending on a parameter that has to be chosen as for the 9-δ elements.

In Fig. 12 we show the stream function of the lid-driven cavity flow for $Re = 1000$ on 64 CGL elements with polynomial degree $N = 6$ and time step size $\Delta t = 1/300$. In Fig. 13 we show the velocity profile of the lid-driven cavity flow for $Re = 1000$ on 64 CGL elements with polynomial degree $N = 6$ and time step size $\Delta t = 1/300$.

6. Conclusion

We presented a least-squares spectral collocation scheme for the steady and unsteady Stokes equations and for the incompressible Navier–Stokes equations where the original domain has been decomposed into quadrilateral subelements. To avoid high condition numbers of normal equations a direct solver (QR decomposition of the matrices) was used for the overdetermined systems. The numerical simulations confirm the high accuracy of the proposed spectral least-squares scheme and solving the overdetermined systems with QR decomposition yields better approximation results. The computational cost of QR decomposition is higher than using normal equations. Using pseudoinverse shows same results as using QR decomposition but causes much higher computational costs.

In the unsteady case of the Stokes equations we have shown the good performance of the here presented scheme for different Reynolds numbers, different time steps and various numbers of elements.

Furthermore, we applied our scheme successfully to the regularized and lid-driven cavity flow problems where we achieved good results.

References

[1] R. Bensow, M.G. Larson, Discontinuous/continuous least-squares finite element methods for elliptic problems, Math. Models Methods Appl. Sci. 15 (6) (2005) 825–842.
[2] R. Bensow, M.G. Larson, Discontinuous least-squares finite element method for the div–curl problem, Num. Math. 101 (4) (2005) 601–617.
[3] C. Bernardi, C. Canuto, Y. Maday, Generalized inf–sup condition for Chebyshev approximations to Navier–Stokes equations, C.R. Acad. Sci. Paris 303 (serie I) (1986) 971–974 (Springer, 1988).
[4] O. Botella, On the solution of the Navier–Stokes equations using Chebyshev projection schemes with third-order accuracy in time, Comput. Fluids 26 (2) (1997) 107–116.
[5] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Comput. Fluids 27 (4) (1998) 421–433.
[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer Series in Computational Physics, Springer, 1988.
[7] M.O. Deville, P.F. Fischer, E.H. Mund, High-Order Methods for Incompressible Fluid Flow, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2002.
[8] H. Eisen, W. Heinrichs, A new method of stabilization for singular perturbation problems with spectral methods, SIAM J. Numer. Anal. 29 (1992) 107–122.
[9] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, in: CBMS-NSF Regional Conference Series in Applied Mathematics, no. 26. SIAM, 1977.
[10] M.I. Gerritsma, M.M.J. Proot, Analysis of a discontinuous least-squares spectral element method, J. Sci. Comput. 17 (2002) 297–306.
[11] H. Haschke, W. Heinrichs, Splitting techniques with staggered grids for the Navier–Stokes equations in the 2D case, J. Comput. Phys. 168 (1) (2001) 131–154.
[12] W. Heinrichs, Splitting techniques for the pseudospectral approximation of the unsteady Stokes equations, SIAM J. Numer. Anal. 30 (1) (1993) 19–39.

[13] W. Heinrichs, Least-squares spectral collocation for discontinuous and singular perturbation problems, J. Comput. Appl. Math. 157 (2) (2003) 329–345.
[14] W. Heinrichs, Least-squares spectral collocation for the Navier–Stokes equations, J. Sci. Comput. 21 (1) (2004) 81–90.
[15] W. Heinrichs, Least-squares spectral collocation with the overlapping Schwarz method for the incompressible Navier–Stokes equations, Numer. Algorithms (43) (2006) 61–73.
[16] W. Heinrichs, An adaptive least-squares spectral collocation method with triangular elements for the incompressible Navier–Stokes equations, J. Eng. Math. 56 (3) (2006) 337–350.
[17] B.-N. Jiang, C.L. Chang, Least-squares finite elements for the Stokes problem, Comput. Methods Appl. Mech. Eng. 78 (3) (1990) 297–311.
[18] B.-N. Jiang, L. Povinelli, Least-squares finite element method for fluid dynamics, Comput. Methods Appl. Mech. Eng. 81 (1) (1990) 13–37.
[19] B.-N. Jiang, A least-squares finite element method for incompressible Navier–Stokes problems, Int. J. Numer. Methods Fluids 14 (7) (1992) 843–859.
[20] B.-N. Jiang, On the least-squares method, Comput. Methods Appl. Mech. Eng. 152 (1-2) (1998) 239–257.
[21] T. Kattelans, Spektrale Least-Squares Verfahren für inkompressible Navier–Stokes-Gleichungen, Diploma thesis (in German), University of Duisburg-Essen, 2007.
[22] S.A. Orszag, Spectral methods for problems in complex geometries, J. Comput. Phys. 37 (1980) 70–92.
[23] R. Peyret, Spectral Methods for Incompressible Viscous Flow, Springer, 2002.
[24] J.P. Pontaza, J.N. Reddy, Spectral/hp least-squares finite element formulation for the Navier–Stokes equations, J. Comput. Phys. 190 (2) (2003) 523–549.
[25] J.P. Pontaza, J.N. Reddy, Space-time coupled spectral/hp least-squares finite element formulation for the incompressible Navier–Stokes equations, J. Comput. Phys. 197 (2) (2004) 418–459.
[26] J.P. Pontaza, J.N. Reddy, Least-squares finite element formulations for viscous incompressible and compressible fluid flows, Comput. Methods Appl. Mech. Eng. 195 (19-22) (2006) 2454–2494.
[27] M.M.J. Proot, M.I. Gerritsma, A least-squares spectral element formulation for the Stokes problem, J. Sci. Comput. 17 (1–4) (2002) 285–296.
[28] M.M.J. Proot, M.I. Gerritsma, Least-squares spectral elements applied to the Stokes problem, J. Comput. Phys. 181 (2) (2002) 454–477.
[29] M.M.J. Proot, M.I. Gerritsma, Application of the least-squares spectral element method using Chebyshev polynomials to solve the incompressible Navier–Stokes equations, Numer. Algorithms 38 (2005) 155–172.
[30] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer, 1994.
[31] E. Rønquist, Optimal spectral element methods for the unsteady three dimensional incompressible Navier–Stokes equations, Phd. Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1988.