

Diskretisierung der stationären inkompressiblen Navier-Stokes- Gleichungen in 3D auf unstrukturierten Tetraedergittern

Bachelorarbeit

Freie Universität Berlin

Fachbereich Mathematik und Informatik

Institut für Mathematik

Florian Litzinger

November 2013

Dem Fachbereich Mathematik und Informatik
der Freien Universität Berlin
zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)
vorgelegt am 21. November 2013.

Angefertigt am
Weierstraß-Institut für Angewandte Analysis und Stochastik
bei Dr. Alexander Linke und Prof. Dr. Volker John.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der numerischen Lösung der stationären inkompressiblen Navier-Stokes-Gleichungen im \mathbb{R}^3 mit der Finite-Elemente-Methode. Dazu werden die Navier-Stokes-Gleichungen zunächst in der starken Formulierung vorgestellt und die schwache Formulierung motiviert. Im Folgenden werden Sätze zur Existenz und Eindeutigkeit einer Lösung unter bestimmten Bedingungen hergeleitet. Alsdann wird eine Diskretisierung des Navier-Stokes-Problems unter Benutzung von nichtkonformen finiten Elementen entwickelt, wozu eine gemischte $\mathcal{P}_1^{\text{nc}}/\mathcal{P}_0$ -Methode mit dem nichtkonformen Crouzeix-Raviart-Element verwendet wird. Die Existenz und Eindeutigkeit der diskreten Lösung wird gezeigt und Abschätzungen sowohl für den Approximations- als auch den Konsistenzfehler werden hergeleitet. Ferner wird ein modifiziertes Verfahren unter Benutzung eines Rekonstruktionsoperators eingeführt, das eine echt divergenzfreie Approximation der Lösung liefert. Zuletzt wird die im Rahmen dieser Arbeit erstellte Implementierung beider Verfahren in C++ genauer erläutert und die Ergebnisse von vier Testbeispielen werden präsentiert. Den Abschluss bildet eine zusammenfassende Betrachtung, die auch einen Ausblick auf weiterführende Fragestellungen gibt.

Inhaltsverzeichnis

1. Einleitung	1
2. Vorbereitungen aus der Analysis	3
2.1. Lebesgue-Räume	3
2.2. Schwache Ableitung, Sobolev-Räume	4
2.3. Helmholtz-Zerlegung	10
3. Die inkompressiblen Navier-Stokes-Gleichungen	13
3.1. Das kontinuierliche Problem	13
3.1.1. Klassische Formulierung des Problems	13
3.1.2. Schwache Formulierung des Problems	14
3.1.3. Existenz und Eindeutigkeit einer Lösung	19
3.2. Diskretisierung des Problems	25
3.2.1. Diskretisierung mit nichtkonformen finiten Elementen	26
3.2.2. Existenz und Eindeutigkeit der diskreten Lösung	29
3.2.3. Fehlerabschätzungen	34
3.2.4. Der divergenzfreie Rekonstruktionsansatz	43
4. Implementierung und Studie des numerischen Verfahrens	50
4.1. Beispiel 1: Ethier/Steinman-Benchmark	52
4.2. Beispiel 2: Rotierendes Geschwindigkeitsfeld	56
4.3. Beispiel 3: Rohrströmung	61
4.4. Beispiel 4: Lid-driven cavity	67
5. Fazit und Ausblick	71

Inhaltsverzeichnis

Literatur	73
Abbildungsverzeichnis	76
Tabellenverzeichnis	77
A. Quelltexte	78

1. Einleitung

Es wird allgemein angenommen, dass die Navier-Stokes-Gleichungen eine gute Beschreibung der Strömung von inkompressiblen Flüssigkeiten darstellen. Insofern sind diese in vielen wissenschaftlichen Gebieten von großer Bedeutung. Im Kontext dieser Arbeit ist der stationäre Fall von primärem Interesse. Da es im Allgemeinen nicht möglich ist, für ein konkretes Problem eine analytische Lösung der Navier-Stokes-Gleichungen zu finden, wird diese gewöhnlich mit numerischen Verfahren approximiert. Einen populären Ansatz dazu stellt die Finite-Elemente-Methode dar.

In der vorliegenden Arbeit wird eine nichtkonforme Diskretisierung der Navier-Stokes-Gleichungen vorgestellt, implementiert und numerisch untersucht. Dabei wird auch eine divergenzfreie Rekonstruktion entwickelt, die einen Nachteil von gemischten Finite-Elemente-Methoden zu vermeiden sucht; namentlich die Tatsache, dass der Fehler, der bei der Berechnung des Geschwindigkeitsfeldes auftritt, von dem Druck abhängig ist, was den Tatsachen des kontinuierlichen Falles und der zugrunde liegenden Physik nicht entspricht. Diese Rekonstruktion wurde bereits in Beiträgen von A. Linke [Lin13] und C. Brennecke [Bre13] in zwei Raumdimensionen untersucht und soll in dieser Arbeit nun auf den dreidimensionalen Fall übertragen werden.

Die vorliegende Arbeit gliedert sich wie folgt: Zunächst werden die in dieser Arbeit vorrangig benötigten Resultate der Analysis zitiert. Im folgenden Kapitel werden die Navier-Stokes-Gleichungen vorgestellt und Existenz und Eindeutigkeit einer analytischen Lösung geklärt. Danach wird die Diskretisierung mit dem Crouzeix-Raviart-Element entwickelt und die grundlegende Fehlerabschätzung bewiesen. Ferner wird die bereits angesprochene divergenzfreie Rekonstruktion präsentiert. Beide Methoden werden hernach einer numerischen Studie unterzogen, die die Vorteile des modifizier-

1. Einleitung

ten Verfahrens deutlich machen soll. Dazu werden vier unterschiedliche Testbeispiele herangezogen, die die Robustheit der Methode mit dem Rekonstruktionsoperator untermauern sollen. Abschließend werden die Hauptergebnisse resümiert und Fragen, die weiter ergründet werden könnten, kurz erörtert.

Im Rahmen dieser Arbeit wurden die angesprochenen Methoden in einem C++-Programm implementiert. Da dieses recht umfangreich ist, befindet es sich lediglich stark gekürzt im Anhang.

2. Vorbereitungen aus der Analysis

In dieser Arbeit werden einige Definitionen und Sätze der Analysis benötigt, die im Folgenden zusammengestellt werden. Diese wurden, sofern keine weitere Angabe gemacht wird, den Büchern von Evans [Eva10, Ch. 5, Appx. C] und Werner [Wer11, Kap. I] entnommen.

2.1. Lebesgue-Räume

Sei $\Omega \subset \mathbb{R}^n$, $n \in \mathbb{N}$ offen und beschränkt. Wir betrachten die Funktionenräume

$$L^p(\Omega) := \left\{ f: \Omega \rightarrow \mathbb{R} : f \text{ } \mathcal{L}^n\text{-messbar, } \int_{\Omega} |f|^p < \infty \right\}, \quad 1 \leq p < \infty,$$
$$L^\infty(\Omega) := \left\{ f: \Omega \rightarrow \mathbb{R} : f \text{ } \mathcal{L}^n\text{-messbar, } \operatorname{ess\,sup}_{x \in \Omega} |f(x)| < \infty \right\}.$$

Mit den Normen

$$\|f\|_{L^p} := \left(\int_{\Omega} |f|^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty,$$
$$\|f\|_{L^\infty} := \operatorname{ess\,sup}_{x \in \Omega} |f(x)|$$

sind $L^p(\Omega)$ und $L^\infty(\Omega)$ Banachräume.

Bemerkung (1). Eigentlich sind die Elemente von $L^p(\Omega)$, $1 \leq p \leq \infty$, Äquivalenzklassen von Funktionen. Die Gleichheit zweier Funktionen $f, g \in L^p(\Omega)$ ist eine Gleichheit fast überall. Trotzdem betrachten wir die Elemente von $L^p(\Omega)$ als Funktionen.

Bemerkung (2). Für $u \in L^\infty(\Omega)$ gilt, dass $u \in L^p(\Omega)$ für alle $1 \leq p < \infty$.

Bemerkung (3). Mit $\int_{\Omega} f$ ist das Integral bezüglich des Lebesgue-Maßes \mathcal{L}^n gemeint.

2. Vorbereitungen aus der Analysis

Für den Spezialfall $p = 2$ wird die Norm $\|\cdot\|_{L^2(\Omega)}$ durch das Skalarprodukt

$$\langle f, g \rangle_{L^2(\Omega)} := \int_{\Omega} fg$$

induziert, mit welchem $L^2(\Omega)$ zu einem Hilbertraum wird. Entsprechend definieren wir für vektorwertige Funktionen $\mathbf{f}: \Omega \rightarrow \mathbb{R}^n$ den Raum $L^2(\Omega)^n$ durch

$$L^2(\Omega)^n := \left\{ \mathbf{f} = (f_1, \dots, f_n)^T : f_i \in L^2(\Omega) \forall i = 1, \dots, n \right\}$$

mit der Norm

$$\|\mathbf{f}\|_{L^2(\Omega)^n} := \left(\sum_{i=1}^n \|f_i\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}.$$

Definiere nun zuletzt die Räume

$$L_0^2(\Omega) := \left\{ f \in L^2(\Omega) : \int_{\Omega} f = 0 \right\}$$

sowie den Raum der lokal integrierbaren Funktionen

$$L_{\text{loc}}^1(\Omega) := \left\{ f \in L^1(K) : K \subset\subset \Omega \right\}.$$

2.2. Schwache Ableitung, Sobolev-Räume

Sei $u \in C^1(\Omega)$. Dann gilt nach der partiellen Integration mit $\phi \in C_c^\infty(\Omega)$

$$\int_{\Omega} u \phi_{x_i} = - \int_{\Omega} u_{x_i} \phi, \quad i = 1, \dots, n, \quad (2.1)$$

wobei $u_{x_i} = \frac{\partial u}{\partial x_i}$ die partielle Ableitung nach x_i bezeichnet und kein Randterm auftritt, da ϕ aufgrund des kompakten Trägers auf $\partial\Omega$ verschwindet. Allgemeiner gilt für ein $k \in \mathbb{N}$, $u \in C^k(\Omega)$ und einen Multiindex $\alpha \in \mathbb{N}_0^n$ der Ordnung $|\alpha| = \alpha_1 + \dots + \alpha_n = k$ nach k -maliger Anwendung von (2.1)

$$\int_{\Omega} u D^\alpha \phi = (-1)^{|\alpha|} \int_{\Omega} D^\alpha u \phi, \quad (2.2)$$

2. Vorbereitungen aus der Analysis

wobei

$$D^\alpha u := \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \dots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}} u.$$

Auf der linken Seite von (2.2) muss u nur lokal integrierbar sein, da ϕ einen kompakten Träger hat. Auf der rechten Seite müsste dann aber $D^\alpha u$ für solch ein u , das nicht in C^k ist, erklärt werden. Das tun wir in einer späteren Definition. Zunächst zeigt aber das folgende Lemma einen Grund dafür auf, warum man Funktionen $\phi \in C_c^\infty(\Omega)$ auch als *Testfunktionen* bezeichnet:

Lemma 2.1. *Sei $\Omega \subset \mathbb{R}^n$ offen, beschränkt und $u \in L^1_{\text{loc}}(\Omega)$. Es gelte für alle $\phi \in C_c^\infty(\Omega)$*

$$\int_{\Omega} u \phi = 0.$$

Dann gilt $u = 0$ in $L^1_{\text{loc}}(\Omega)$.

Beweis. Da wir für ein $K \subset\subset \Omega$ von Ω zu $\overset{\circ}{K}$ übergehen können, betrachten wir nur $u \in L^1(\Omega)$. Setze

$$\tilde{u}(x) := \begin{cases} \frac{u(x)}{|u(x)|}, & u(x) \neq 0, \\ 0, & u(x) = 0. \end{cases}$$

Dann gilt $|\tilde{u}| \leq 1$ und $u\tilde{u} = |u|$. Damit ist $\tilde{u} \in L^\infty(\Omega)$ und wegen der Bemerkung in Abschnitt 2.1 also auch $\tilde{u} \in L^2(\Omega)$. Da $C_c^\infty(\Omega)$ in $L^2(\Omega)$ dicht liegt (siehe [Wer11, S. 209]), betrachten wir nun eine Folge von Glättungen $(\tilde{u}_\epsilon) \subset C_c^\infty(\Omega)$ mit $\tilde{u}_\epsilon \xrightarrow{L^2(\Omega)} \tilde{u}$. Diese hat eine Teilfolge, die punktweise fast überall gegen \tilde{u} konvergiert. Sofern diese Eigenschaft nicht schon für (\tilde{u}_ϵ) gilt, benutzen wir die gleiche Bezeichnung jetzt für die Teilfolge. Es gilt mit dem üblichen Glättungskern η_ϵ

$$\begin{aligned} |\tilde{u}_\epsilon(x)| &= \left| \int_{B_\epsilon(x)} \eta_\epsilon(x-y) \tilde{u}(y) \, dy \right| \\ &\leq \int_{B_\epsilon(x)} \eta_\epsilon(x-y) |\tilde{u}(y)| \, dy \\ &\leq \|\tilde{u}\|_{L^\infty} \int_{B_\epsilon(x)} \eta_\epsilon(x-y) \, dy = \|\tilde{u}\|_{L^\infty} \end{aligned} \tag{2.3}$$

2. Vorbereitungen aus der Analysis

und damit

$$\|\tilde{u}_\epsilon\|_{L^\infty} \leq \|\tilde{u}\|_{L^\infty} \leq 1. \quad (2.4)$$

Mit dem Satz von der dominierten Konvergenz folgt

$$0 = \int_{\Omega} u \tilde{u}_\epsilon \xrightarrow{\epsilon \rightarrow 0} \int_{\Omega} u \tilde{u} = \int_{\Omega} |u|. \quad (2.5)$$

Damit ist $u = 0$ fast überall in Ω . □

Nun wird der Begriff der schwachen Ableitung definiert.

Definition 2.2 (Schwache Ableitung). Seien $u, v \in L^1_{\text{loc}}(\Omega)$ und α ein Multiindex. Wir nennen v die *schwache α -partielle Ableitung* von u ,

$$D^\alpha u = v,$$

falls für alle Testfunktionen $\phi \in C_c^\infty(\Omega)$ gilt

$$\int_{\Omega} u D^\alpha \phi = (-1)^{|\alpha|} \int_{\Omega} v \phi.$$

Bemerkung. Eine schwache α -partielle Ableitung von u ist, falls sie existiert, bis auf eine Nullmenge eindeutig definiert. Dazu betrachte zwei schwache α -partielle Ableitungen $v, \tilde{v} \in L^1_{\text{loc}}(\Omega)$. Für diese gilt dann $\int_{\Omega} (v - \tilde{v}) \phi = 0$ für alle Testfunktionen $\phi \in C_c^\infty(\Omega)$ und damit (vgl. Lemma 2.1) $v - \tilde{v} = 0$ fast überall.

Das Konzept der schwachen Ableitung führt zur Definition der Sobolev-Räume.

Definition 2.3 (Sobolev-Raum). Der *Sobolev-Raum*

$$W^{k,p}(\Omega)$$

besteht aus allen lokal integrierbaren Funktionen $u: \Omega \rightarrow \mathbb{R}$, sodass für jeden Multiindex α mit $|\alpha| \leq k$ die Ableitung $D^\alpha u$ im schwachen Sinne existiert und es gilt $D^\alpha u \in L^p(\Omega)$.

2. Vorbereitungen aus der Analysis

Für ein $u \in W^{k,p}(\Omega)$ ist eine Norm gegeben durch

$$\|u\|_{W^{k,p}(\Omega)} := \begin{cases} \left(\sum_{|\alpha| \leq k} \int_{\Omega} |D^{\alpha}u|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty, \\ \sum_{|\alpha| \leq k} \operatorname{ess\,sup}_{\Omega} |D^{\alpha}u|, & p = \infty. \end{cases}$$

Bemerkung. Die $W^{k,p}(\Omega)$ sind Banachräume. Für den Spezialfall $p = 2$ schreiben wir

$$H^k(\Omega) = W^{k,2}(\Omega), \quad k = 0, 1, \dots$$

$H^k(\Omega)$ ist ein Hilbertraum, dessen Norm durch das Skalarprodukt

$$\langle u, v \rangle_{H^k(\Omega)} := \sum_{|\alpha| \leq k} \langle D^{\alpha}u, D^{\alpha}v \rangle_{L^2(\Omega)}$$

induziert wird. Weiter ist

$$\|u\|_{H^k(\Omega)} := \left(\sum_{|\alpha| \leq k} \|D^{\alpha}u\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}$$

eine Halbnorm auf $H^k(\Omega)$. Die Sobolev-Räume von vektorwertigen Funktionen, wie zum Beispiel $H^k(\Omega)^n$, sind analog zu den Lebesgue-Räumen $L^p(\Omega)^n$ definiert.

Definition 2.4. Wir bezeichnen mit

$$H_0^k(\Omega)$$

den Abschluss von $C_c^{\infty}(\Omega)$ in $H^k(\Omega)$.

Insbesondere von $H_0^1(\Omega)$ wird im Folgenden reger Gebrauch gemacht. Daher wollen wir ihn noch auf eine andere Weise charakterisieren. Dazu benötigen wir zunächst die folgende Definition und den Satz 2.6.

Definition 2.5 (Lipschitz-Gebiet). Sei $\Omega \subset \mathbb{R}^n$ offen und beschränkt. Ω heißt *Lipschitz-Gebiet*, falls für jeden Punkt $x^* \in \partial\Omega$ ein $r > 0$ und eine Lipschitz-stetige

2. Vorbereitungen aus der Analysis

Abbildung $\gamma: \mathbb{R}^{n-1} \rightarrow \mathbb{R}^n$ existiert, sodass nach Umbenennung und Umorientierung der Koordinatenachsen, falls erforderlich, gilt (vgl. [Eva10, S. 710])

$$\Omega \cap B_r(x^*) = \{x \in B_r(x^*) : x_n > \gamma(x_1, \dots, x_{n-1})\}.$$

Also kann $\partial\Omega$ lokal als Graph einer Lipschitz-stetigen Funktion betrachtet werden.

Im Allgemeinen stellt sich die Frage, wie man einer Funktion $u \in W^{1,p}(\Omega)$ Randwerte auf $\partial\Omega$ zuordnen soll. Falls u stetig auf $\bar{\Omega}$ ist, hat u Randwerte im normalen Sinne. Eine Funktion $u \in W^{1,p}(\Omega)$ muss aber nicht stetig sein und ist gegebenenfalls sogar nur fast überall in Ω definiert. Aufgrund der Tatsache, dass $\mathcal{L}^n(\partial\Omega) = 0$ gilt, ist nicht klar, was $u|_{\partial\Omega}$ bedeuten soll. Dieses Problem löst der *Spuroperator* (siehe auch [Alt12, S. 279] und [EG91, S. 133]).

Satz 2.6 (Spursatz). *Sei $\Omega \subset \mathbb{R}^n$ ein beschränktes Lipschitz-Gebiet und $1 \leq p < \infty$. Dann existiert genau ein beschränkter linearer Operator*

$$T: W^{1,p}(\Omega) \rightarrow L^p(\partial\Omega)^1,$$

sodass für alle $u \in W^{1,p}(\Omega) \cap C(\bar{\Omega})$ gilt

$$Tu = u|_{\partial\Omega}.$$

Bemerkung. Wir bezeichnen Tu als die *Spur* von u und schreiben für $x \in \partial\Omega$ statt $(Tu)(x)$ einfach $u(x)$.

Es erweist sich, dass die Funktionen $u \in H_0^1(\Omega)$ wie in Definition 2.4, also die Funktionen, zu denen es eine Folge $(u_j)_{j=1}^\infty \subset C_c^\infty(\Omega)$ gibt mit $u_j \xrightarrow{j \rightarrow \infty} u$ in $H^1(\Omega)$ (d. h. $\lim_{j \rightarrow \infty} \|u_j - u\|_{H^1(\Omega)} = 0$), genau die Funktionen sind, für die $u|_{\partial\Omega} = 0$ im Spursinne gilt.

¹In der Norm dieses Raumes ist das Integral bezüglich des Hausdorff-Maßes \mathcal{H}^{n-1} zu nehmen.

2. Vorbereitungen aus der Analysis

Satz 2.7. Sei $\Omega \subset \mathbb{R}^n$ ein beschränktes Lipschitz-Gebiet. Dann gilt

$$H_0^1(\Omega) = \left\{ u \in H^1(\Omega) : Tu = 0 \right\}.$$

Beweis. Siehe bei Alt [Alt12, S. 284].

Weiterhin benötigen wir zwei Versionen des Divergenzsatzes für Funktionen in Sobolev-Räumen.

Satz 2.8 (Divergenzsatz). Sei $\Omega \subset \mathbb{R}^n$ ein beschränktes Lipschitz-Gebiet, $u \in W^{1,p}(\Omega)$, $1 \leq p \leq \infty$, $\mathbf{v} \in C^1(\Omega)^n$. Dann gilt

$$\int_{\Omega} u \operatorname{div} \mathbf{v} = - \int_{\Omega} \nabla u \cdot \mathbf{v} + \int_{\partial\Omega} (\mathbf{v} \cdot \mathbf{n}) u \, d\mathcal{H}^{n-1},$$

mit der äußeren Normale \mathbf{n} an $\partial\Omega$. Insbesondere gilt für $u \in H_0^1(\Omega)$

$$\int_{\Omega} u \operatorname{div} \mathbf{v} = - \int_{\Omega} \nabla u \cdot \mathbf{v}.$$

Beweis. Siehe bei Evans und Gariepy [EG91, S. 133].

Nach [Soh01, S. 49] gilt auch die folgende Verallgemeinerung, in der auch \mathbf{v} nur schwach differenzierbar zu sein braucht.

Satz 2.9 (Schwacher Divergenzsatz). Sei $\Omega \subset \mathbb{R}^n$, $n \geq 2$, ein beschränktes Lipschitz-Gebiet und $1 < q < \infty$, $q' := \frac{q}{q-1}$. Dann gilt für alle $u \in W^{1,q}(\Omega)$ und $\mathbf{v} \in W^{1,q'}(\Omega)^n$

$$\int_{\Omega} u \operatorname{div} \mathbf{v} = - \int_{\Omega} \nabla u \cdot \mathbf{v} + \int_{\partial\Omega} (\mathbf{v} \cdot \mathbf{n}) u \, d\mathcal{H}^{n-1}.$$

Insbesondere gilt diese Aussage also für $u \in H^1(\Omega)$ und $\mathbf{v} \in H^1(\Omega)^n$.

Es gilt außerdem eine sehr hilfreiche Sobolev-Ungleichung.

2. Vorbereitungen aus der Analysis

Lemma 2.10 (Sobolev-Ungleichung). Sei $\Omega \subset \mathbb{R}^n$ beschränkt. Sei $u \in W_0^{1,p}(\Omega)$ für ein $1 \leq p < n$. Dann gilt die Abschätzung

$$\|u\|_{L^q(\Omega)} \leq C \|\nabla u\|_{L^p(\Omega)^n}$$

für alle $q \in [1, p^*]$ mit $p^* = \frac{np}{n-p}$ und einer Konstante $C = C(p, q, n, \Omega)$.

Beweis. Siehe bei Evans [Eva10, S. 279].

Bemerkung. Im Speziellen gilt diese Ungleichung also für $n = 3$, $p = 2$ und $q = 4$, das heißt, dass für $\mathbf{u} \in H_0^1(\Omega)^n$ gilt

$$\|\mathbf{u}\|_{L^4(\Omega)^n} \leq C \|\nabla \mathbf{u}\|_{L^2(\Omega)^{n^2}}. \quad (2.6)$$

Die Ungleichung wird auch eine Poincaré-Friedrichs-Ungleichung genannt. Außerdem folgt, da $\|u\|_{H^1} = \|\nabla u\|_{L^2}$, dass $\|\cdot\|_{H^1}$ und $|\cdot|_{H^1}$ auf $H_0^1(\Omega)$ äquivalent sind.

2.3. Helmholtz-Zerlegung

Ein Vektorfeld \mathbf{w} bezeichnet man als *divergenzfrei*, sofern gilt $\operatorname{div} \mathbf{w} = 0$. Entsprechend heißt das Vektorfeld \mathbf{v} *rotationsfrei*, falls $\nabla \times \mathbf{v} = \mathbf{0}$. Der Gradient einer skalaren Funktion $\psi: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ ist rotationsfrei, sofern man die partiellen Ableitungen vertauschen darf:

$$\nabla \times (\nabla \psi) = \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} \times \begin{pmatrix} \psi_x \\ \psi_y \\ \psi_z \end{pmatrix} = \begin{pmatrix} (\psi_z)_y - (\psi_y)_z \\ (\psi_x)_z - (\psi_z)_x \\ (\psi_y)_x - (\psi_x)_y \end{pmatrix} = \mathbf{0}. \quad (2.7)$$

Im Folgenden nehmen wir stets eine hinreichende (ggf. auch schwache) Differenzierbarkeit der beteiligten Funktionen an. Für ein divergenzfreies Vektorfeld \mathbf{w} und den rotationsfreien Gradienten $\nabla \psi$ gilt dann unter der Voraussetzung, dass

$$(\mathbf{w} \cdot \mathbf{n})|_{\partial\Omega} = 0 \quad (2.8)$$

2. Vorbereitungen aus der Analysis

aufgrund des Divergenzsatzes 2.8, dass diese orthogonal bezüglich des L^2 -Skalarproduktes sind (vgl. [Lin13, S. 5]):

$$\begin{aligned} \int_{\Omega} \nabla \psi \cdot \mathbf{w} &= \int_{\Omega} \operatorname{div}(\psi \mathbf{w}) - \int_{\Omega} \psi \operatorname{div} \mathbf{w} \\ &= \int_{\partial \Omega} (\mathbf{w} \cdot \mathbf{n}) \psi \, d\mathcal{H}^{n-1} \\ &= 0. \end{aligned} \tag{2.9}$$

Zum folgenden Satz siehe auch bei Sohr [Soh01, S. 81ff].

Satz 2.11 (Helmholtz-Zerlegung). *Sei $\Omega \subset \mathbb{R}^n$ mit $n \geq 2$ ein beschränktes Lipschitz-Gebiet. Dann gibt es zu jedem $\mathbf{f} \in L^2(\Omega)^n$ eine eindeutige Zerlegung*

$$\mathbf{f} = \mathbf{w} + \nabla \psi$$

mit $\mathbf{w} \in L^2(\Omega)^n$ mit $\operatorname{div} \mathbf{w} = 0$ und $(\mathbf{w} \cdot \mathbf{n})|_{\partial \Omega} = 0$ sowie $\nabla \psi \in L^2(\Omega)^n$. Außerdem sind $\nabla \psi$ und \mathbf{w} bezüglich des L^2 -Skalarproduktes orthogonal, es gilt also

$$\int_{\Omega} \nabla \psi \cdot \mathbf{w} = 0.$$

Nach [Lin13, S. 6] ist ψ eindeutig als Lösung des folgenden Problems bestimmt: Finde $\psi \in H^1(\Omega)$, sodass für alle $\chi \in H^1(\Omega)$ gilt

$$\begin{aligned} \int_{\Omega} \nabla \psi \cdot \nabla \chi &= \int_{\Omega} \mathbf{f} \cdot \nabla \chi, \\ \int_{\Omega} \psi &= 0. \end{aligned} \tag{2.10}$$

Dann berechnet sich der divergenzfreie Anteil \mathbf{w} von \mathbf{f} durch

$$\mathbf{w} = \mathbf{f} - \nabla \psi. \tag{2.11}$$

Bemerkung. Satz 2.11 definiert einen beschränkten linearen Operator

$$\mathbb{P}: L^2(\Omega)^n \rightarrow \left\{ \mathbf{w} \in L^2(\Omega)^n : \operatorname{div} \mathbf{w} = 0, (\mathbf{w} \cdot \mathbf{n})|_{\partial \Omega} = 0 \right\}, \tag{2.12}$$

2. Vorbereitungen aus der Analysis

der durch $\mathbf{f} \mapsto \mathbf{w}$ wie oben gegeben ist. Dieser wird *Helmholtz-Projektion* genannt. Hinreichende Glattheit vorausgesetzt, folgt aus

$$(\mathbf{f} \cdot \mathbf{n})|_{\partial\Omega} = 0 \tag{2.13}$$

dann

$$(\mathbf{w} \cdot \mathbf{n})|_{\partial\Omega} + (\nabla\psi \cdot \mathbf{n})|_{\partial\Omega} = 0 \tag{2.14}$$

und somit wegen der homogenen Neumann-Randbedingung $(\nabla\psi \cdot \mathbf{n})|_{\partial\Omega} = 0$ auch

$$(\mathbf{w} \cdot \mathbf{n})|_{\partial\Omega} = 0. \tag{2.15}$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Im Folgenden werden die für diese Arbeit zentralen Navier-Stokes-Gleichungen vorgestellt und Existenz und Eindeutigkeit einer Lösung untersucht. Diese Einführung orientiert sich vor allem an [Lay08; Lin13; Soh01; Tem79] sowie [Ver11] und [Ver98]. Alsdann wird das Problem diskretisiert und das numerische Verfahren zur Lösung der Gleichungen entwickelt und diskutiert.

3.1. Das kontinuierliche Problem

3.1.1. Klassische Formulierung des Problems

Sei $\Omega \subset \mathbb{R}^n$ ein beschränktes Lipschitz-Gebiet. Dann nehmen wir an, dass die Strömung einer Flüssigkeit innerhalb des Gebietes durch das folgende System nichtlinearer partieller Differentialgleichungen mit homogenen Dirichlet-Randbedingungen beschrieben wird:

$$\begin{aligned} -\nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega, \\ \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{0} && \text{auf } \partial\Omega. \end{aligned} \tag{3.1}$$

Diese Gleichungen werden als *Navier-Stokes-Gleichungen* bezeichnet. Im Folgenden wählen wir stets $n = 3$. Da wir annehmen, dass die Strömung nicht von der Zeit abhängt und die Dichte der betrachteten Flüssigkeit konstant ist, sprechen wir von dem stationären inkompressiblen Fall. In (3.1) beschreibt $\mathbf{u}: \Omega \rightarrow \mathbb{R}^n$ die Geschwindigkeit und $p: \Omega \rightarrow \mathbb{R}$ den Druck innerhalb der Strömung. Die ausgeübte Kraftdichte bezeichnen wir mit $\mathbf{f}: \Omega \rightarrow \mathbb{R}^n$ und $\nu > 0$ bezeichnet die kinematische Viskosität.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Das Problem besteht nun darin, bei vorgegebener Kraft \mathbf{f} das Geschwindigkeitsfeld \mathbf{u} und den Druck p zu bestimmen. Zunächst fällt auf, dass in den Navier-Stokes-Gleichungen anstelle von p nur ∇p vorkommt und in (3.1) keine Randbedingung an den Druck gemacht wird. Also wird p von (3.1) nur bis auf eine Konstante $C \in \mathbb{R}$ bestimmt, denn es gilt offenbar

$$\nabla(p + C) = \nabla p. \quad (3.2)$$

Damit ist für jede Lösung (\mathbf{u}, p) von (3.1) auch $(\mathbf{u}, p + C)$ eine Lösung. Daher führen wir als weitere Bedingung an p ein:

$$\int_{\Omega} p = 0. \quad (3.3)$$

Die obige Formulierung der Navier-Stokes-Gleichungen macht implizit Annahmen darüber, wie glatt Lösungen (\mathbf{u}, p) sein müssen: man möchte $\mathbf{u} \in C^2(\Omega)^n$ sowie $p \in C^1(\Omega)$ fordern. In vielen interessanten Anwendungen ist das allerdings nicht gewährleistet. Aus diesem Grund versuchen wir, die Anforderungen abzuschwächen.

3.1.2. Schwache Formulierung des Problems

Wir multiplizieren (3.1) mit einer Testfunktion $\mathbf{v} \in H_0^1(\Omega)^n$ und integrieren über Ω . Dann ergibt sich

$$-\nu \int_{\Omega} \Delta \mathbf{u} \cdot \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} + \int_{\Omega} \nabla p \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}. \quad (3.4)$$

Mit dem schwachen Divergenzsatz 2.9 angewendet auf die Komponentenfunktionen und der Gleichheit $\Delta \mathbf{u} = \operatorname{div}(\nabla \mathbf{u})$ folgt, da \mathbf{v} auf $\partial\Omega$ verschwindet,

$$\nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div} \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}.^1 \quad (3.5)$$

¹Mit $\nabla \mathbf{u} : \nabla \mathbf{v}$ ist hier das Tensorprodukt $\mathbf{T} : \mathbf{S} := \sum_{i,j=1}^3 \mathbf{T}_{ij} \mathbf{S}_{ij}$ gemeint.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Ebenso ergibt sich mit $q \in L_0^2(\Omega)$

$$\int_{\Omega} q \operatorname{div} \mathbf{u} = 0. \quad (3.6)$$

Damit ist eine klassische Lösung (\mathbf{u}, p) von (3.1) mit $\mathbf{u} \in C^2(\Omega)^n$, $\mathbf{u}|_{\partial\Omega} = \mathbf{0}$ und $p \in C^1(\Omega) \cap L_0^2(\Omega)$ eine Lösung des folgenden Problems:

Gegeben seien ein beschränktes Lipschitz-Gebiet $\Omega \subset \mathbb{R}^n$ und ein $\mathbf{f} \in L^2(\Omega)^n$. Finde $\mathbf{u} \in X := H_0^1(\Omega)^n$, $p \in Q := L_0^2(\Omega)$, so dass für alle $\mathbf{v} \in X$, $q \in Q$ gilt

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div} \mathbf{v} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v}, \\ \int_{\Omega} q \operatorname{div} \mathbf{u} &= 0. \end{aligned} \quad (3.7)$$

Die Bedingung $p \in Q$ gewährleistet hier, dass die Bedingung (3.3) erfüllt ist. Insofern bezeichnen wir das Problem (3.7) als *schwache Formulierung* des Navier-Stokes-Problems.

Als nächstes wollen wir Existenz und Eindeutigkeit von Lösungen $(\mathbf{u}, p) \in X \times Q$ klären. Dazu führen wir eine abstraktere Formulierung des Navier-Stokes-Problems ein.

Definition 3.1. Wir definieren die Bilinearformen $a: X \times X \rightarrow \mathbb{R}$, $b: X \times Q \rightarrow \mathbb{R}$, die Trilinearform $c: X \times X \times X \rightarrow \mathbb{R}$ und die Linearform $l: X \rightarrow \mathbb{R}$ vermöge

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &:= \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v}, \\ b(\mathbf{u}, q) &:= - \int_{\Omega} q \operatorname{div} \mathbf{u}, \\ c(\mathbf{g}, \mathbf{u}, \mathbf{v}) &:= \int_{\Omega} (\mathbf{g} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}, \\ l(\mathbf{v}) &:= \int_{\Omega} \mathbf{f} \cdot \mathbf{v}. \end{aligned} \quad (3.8)$$

Dass es sich hier tatsächlich um lineare Abbildungen handelt, ist aufgrund der Linearität der beteiligten Differentialoperatoren ∇ und div , des Skalar- bzw. Tensorprodukts und des Integrals gegeben.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Mit diesen Definitionen lässt sich das Problem (3.7) äquivalent wie folgt formulieren:
Finde $(\mathbf{u}, p) \in X \times Q$, sodass für alle $(\mathbf{v}, q) \in X \times Q$ gilt

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= l(\mathbf{v}), \\ b(\mathbf{u}, q) &= 0. \end{aligned} \tag{3.9}$$

Die in (3.8) definierten Abbildungen haben weitere nützliche Eigenschaften, wie das folgende Lemma zeigt (siehe auch [Lay08, S. 106f]).

Lemma 3.2. *Sei $\Omega \subset \mathbb{R}^n$, $n = 3$, ein beschränktes Lipschitz-Gebiet. Dann sind die Bilinearformen a und b stetig auf $X \times X$ bzw. $X \times Q$, a ist symmetrisch und koerziv und die Trilinearform c ist stetig auf $X \times X \times X$.*

Beweis. Die Symmetrie von a ergibt sich aus der Symmetrie des Tensorproduktes, die Koerzivität aus der Bemerkung nach Lemma 2.10. Für die Stetigkeit von c ist zu zeigen, dass für $\mathbf{g}, \mathbf{u}, \mathbf{v} \in X$ gilt

$$|c(\mathbf{g}, \mathbf{u}, \mathbf{v})| \leq C \|\mathbf{g}\|_X \|\mathbf{u}\|_X \|\mathbf{v}\|_X. \tag{3.10}$$

Mit der Hölder-Ungleichung gilt

$$|c(\mathbf{g}, \mathbf{u}, \mathbf{v})| = \left| \int_{\Omega} (\mathbf{g} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \right| \leq \|\mathbf{g}\|_{L^p} \|\nabla \mathbf{u}\|_{L^2} \|\mathbf{v}\|_{L^q}, \tag{3.11}$$

mit $\frac{1}{p} + \frac{1}{2} + \frac{1}{q} = 1$. Mit der Sobolev-Ungleichung aus dem Lemma 2.10 folgt nun

$$\|\mathbf{g}\|_{L^p} \|\nabla \mathbf{u}\|_{L^2} \|\mathbf{v}\|_{L^q} \leq C \|\nabla \mathbf{g}\|_{L^2} \|\nabla \mathbf{u}\|_{L^2} \|\nabla \mathbf{v}\|_{L^2} \leq C \|\mathbf{g}\|_X \|\mathbf{u}\|_X \|\mathbf{v}\|_X, \tag{3.12}$$

wobei wir $p, q = 4$ gewählt haben. Der Nachweis der Stetigkeit von a und b erfolgt analog. \square

Bemerkung. Die kleinste Konstante, die (3.10) erfüllt, ist offenbar

$$M := \sup_{\mathbf{g}, \mathbf{u}, \mathbf{v} \in X} \frac{|c(\mathbf{g}, \mathbf{u}, \mathbf{v})|}{\|\mathbf{g}\|_X \|\mathbf{u}\|_X \|\mathbf{v}\|_X}. \tag{3.13}$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Nehmen wir das Supremum nur über einen Unterraum $V \subset X$, so erhalten wir

$$N := \sup_{\mathbf{g}, \mathbf{u}, \mathbf{v} \in V} \frac{|c(\mathbf{g}, \mathbf{u}, \mathbf{v})|}{\|\mathbf{g}\|_X \|\mathbf{u}\|_X \|\mathbf{v}\|_X}, \quad (3.14)$$

$$0 < N \leq M < \infty, \quad (3.15)$$

da beim Übergang zu einer Obermenge das Supremum höchstens größer werden kann. Also gilt für $\mathbf{g}, \mathbf{u}, \mathbf{v} \in V$

$$|c(\mathbf{g}, \mathbf{u}, \mathbf{v})| \leq N \|\mathbf{g}\|_X \|\mathbf{u}\|_X \|\mathbf{v}\|_X. \quad (3.16)$$

Das *Sattelpunktproblem* (3.9) lässt sich auch als *elliptisches Problem* schreiben, in dem p nicht mehr auftaucht. Dazu nehmen wir die Bedingung der (schwachen) Divergenzfreiheit von \mathbf{u} in den Lösungsraum mit auf.

Lemma 3.3. *Es sei*

$$V := \{\mathbf{v} \in X : \operatorname{div} \mathbf{v} = 0\}.$$

Dann ist V ein abgeschlossener Unterraum von X und somit ist V ebenfalls ein Hilbertraum.

Beweis. Der Divergenzoperator ist linear und stetig, daher ist V ein Unterraum von X . Sei $(\mathbf{v}_n)_{n=1}^\infty \subset V$ eine Folge mit $\mathbf{v}_n \xrightarrow{n \rightarrow \infty} \mathbf{v} \in X$. Wegen der Stetigkeit gilt $\operatorname{div} \mathbf{v}_n \xrightarrow{n \rightarrow \infty} \operatorname{div} \mathbf{v}$. Aus $\mathbf{v}_n \in V$ folgt $\operatorname{div} \mathbf{v}_n = 0$ und damit $\operatorname{div} \mathbf{v} = 0$. \square

Nun gilt aufgrund der Koerzivität von a , dass eine Lösung \mathbf{u} des Problems (3.9) auch das folgende Problem löst: Finde $\mathbf{u} \in V$, sodass für alle $\mathbf{v} \in V$ gilt

$$a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = l(\mathbf{v}). \quad (3.17)$$

In Abschnitt 3.1.3 werden wir die Existenz und Eindeutigkeit von Lösungen des elliptischen Problems (3.17) untersuchen. Der folgende Satz garantiert dann die Existenz eines dazugehörigen Drucks $p \in Q$ und somit einer Lösung $(\mathbf{u}, p) \in X \times Q$ des Navier-Stokes-Problems (3.9), ergo sind beide Probleme äquivalent.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Lemma 3.4 (inf-sup-Bedingung). *Es gibt eine Konstante $\beta > 0$, sodass*

$$\inf_{q \in Q \setminus \{0\}} \sup_{\mathbf{v} \in X \setminus \{0\}} \frac{b(\mathbf{v}, q)}{\|\nabla \mathbf{v}\|_{L^2} \|q\|_{L^2(\Omega)}} \geq \beta > 0. \quad (3.18)$$

Beweis. Siehe bei Verfürth [Ver98, S. 41].

Satz 3.5. *Die Bilinearform a sei symmetrisch und koerziv auf V und die Bilinearform b erfülle die inf-sup-Bedingung. Dann gibt es zu einer Lösung $\mathbf{u} \in V$ des Problems (3.17) auch ein $p \in Q$, sodass (3.9) erfüllt ist.*

Beweis. Siehe bei Girault und Raviart [GR79, S. 109f], vgl. auch [Ver98, S. 109].

Ist eine Lösung $\mathbf{u} \in V$ von (3.17) gefunden, so lässt sich $p \in Q$ aus diesem Problem bestimmen: Für alle $\mathbf{v} \in V^\perp$, das heißt dem orthogonalen Komplement von V bezüglich a , also

$$V^\perp := \{\mathbf{v} \in X : a(\mathbf{v}, \mathbf{w}) = 0 \ \forall \mathbf{w} \in V\}, \quad (3.19)$$

gelte

$$b(\mathbf{v}, p) = l(\mathbf{v}) - c(\mathbf{u}, \mathbf{u}, \mathbf{v}). \quad (3.20)$$

Die inf-sup-Bedingung ist äquivalent dazu, dass der Divergenzoperator $\operatorname{div} : V^\perp \rightarrow Q$ bijektiv ist (vgl. [Gun02, S. 101]). Daraus folgt, dass das obige Problem eine Lösung besitzt.

Die Trilinearform c ist antisymmetrisch bezüglich der beiden letzteren Argumente. Dies wird im folgenden Abschnitt benötigt.

Lemma 3.6. *Es gelten für $\mathbf{u} \in V$, $\mathbf{v}, \mathbf{w} \in X$*

$$c(\mathbf{u}, \mathbf{v}, \mathbf{v}) = 0, \quad (3.21)$$

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -c(\mathbf{u}, \mathbf{w}, \mathbf{v}). \quad (3.22)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Beweis. Um (3.21) zu zeigen, seien $\mathbf{u} \in V \cap C_c^\infty(\Omega)^n$ und $\mathbf{v} \in X \cap C_c^\infty(\Omega)^n$, ein Approximationsargument ergibt dann die Behauptung. Dann gilt (vgl. [Tem79, S. 163])

$$\int_{\Omega} \mathbf{u}_i \left(\frac{\partial \mathbf{v}_j}{\partial x_i} \right) \mathbf{v}_j = \int_{\Omega} \mathbf{u}_i \frac{\partial}{\partial x_i} \left(\frac{\mathbf{v}_j^2}{2} \right) = -\frac{1}{2} \int_{\Omega} \left(\frac{\partial \mathbf{u}_i}{\partial x_i} \right) \mathbf{v}_j^2 \quad (3.23)$$

und damit

$$c(\mathbf{u}, \mathbf{v}, \mathbf{v}) = -\frac{1}{2} \sum_{j=1}^n \int_{\Omega} \mathbf{v}_j^2 \operatorname{div} \mathbf{u} = 0, \quad (3.24)$$

da $\operatorname{div} \mathbf{u} = 0$. Gehen wir in (3.21) von \mathbf{v} zu $\mathbf{v} + \mathbf{w}$ über, so folgt (3.22). \square

3.1.3. Existenz und Eindeutigkeit einer Lösung

Zunächst werden wir die Eindeutigkeit einer etwaigen Lösung von (3.17) nachweisen, das heißt also, dass es, wenn überhaupt, *höchstens* eine Lösung \mathbf{u} geben kann. Danach erst wird der Existenzbeweis geführt.

Eindeutigkeit.

Definition 3.7. Sei $\mathbf{f} \in L^2(\Omega)^n$. Definiere die V^* -Norm von \mathbf{f} durch

$$\|\mathbf{f}\|_* := \sup_{\mathbf{v} \in V} \frac{\langle \mathbf{f}, \mathbf{v} \rangle_{L^2}}{\|\nabla \mathbf{v}\|_{L^2}}.$$

Dann nennen wir $V^* := \overline{V}^{\|\cdot\|_*}$ den Dualraum von V :

$$V^* = \{F: V \rightarrow \mathbb{R} : F \text{ ist linear und stetig}\}.$$

Die Eindeutigkeit einer Lösung von (3.17) ist bei den stationären Navier-Stokes-Gleichungen nur unter gewissen Bedingungen an die Daten ν , \mathbf{f} gegeben. Dies zeigt der folgende Satz (siehe auch [Lay08, S. 107]).

Satz 3.8 (Eindeutigkeit für „kleine Daten“). *Es gelte die folgende Bedingung an ν und \mathbf{f} mit N wie in (3.14):*

$$N\nu^{-2} \|\mathbf{f}\|_* < 1. \quad (3.25)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Dann hat das Problem (3.17) höchstens eine Lösung.

Beweis. Seien \mathbf{u} und $\tilde{\mathbf{u}}$ zwei Lösungen von (3.17). Daraus folgt

$$a(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) - c(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{v}) = 0. \quad (3.26)$$

Mit $\phi := \mathbf{v} = \mathbf{u} - \tilde{\mathbf{u}} \in V$ ergibt sich

$$\nu \|\nabla \phi\|_{L^2}^2 + c(\mathbf{u}, \phi, \phi) + c(\phi, \tilde{\mathbf{u}}, \phi) = 0 \quad (3.27)$$

und unter Benutzung von (3.21)

$$\nu \|\nabla \phi\|_{L^2}^2 + c(\phi, \tilde{\mathbf{u}}, \phi) = 0. \quad (3.28)$$

Wenden wir (3.16) an, so folgt, da $\phi, \tilde{\mathbf{u}} \in V$

$$\nu \|\nabla \phi\|_{L^2}^2 = -c(\phi, \tilde{\mathbf{u}}, \phi) \leq N \|\nabla \phi\|_{L^2}^2 \|\nabla \tilde{\mathbf{u}}\|_{L^2}. \quad (3.29)$$

Die Norm der Lösung $\tilde{\mathbf{u}}$ können wir abschätzen: Setze $\mathbf{v} = \tilde{\mathbf{u}}$ in (3.17), dann ergibt sich mit (3.21) und der Definition von $\|\cdot\|_*$

$$\nu \|\nabla \tilde{\mathbf{u}}\|_{L^2}^2 = \langle \mathbf{f}, \tilde{\mathbf{u}} \rangle_{L^2} \leq \|\mathbf{f}\|_* \|\nabla \tilde{\mathbf{u}}\|_{L^2}. \quad (3.30)$$

Also haben wir

$$\nu \|\nabla \phi\|_{L^2}^2 \leq N \nu^{-1} \|\mathbf{f}\|_* \|\nabla \phi\|_{L^2}^2 \quad (3.31)$$

und schließlich mit (3.25)

$$(1 - N \nu^{-2} \|\mathbf{f}\|_*) \|\nabla \phi\|_{L^2}^2 \leq 0. \quad (3.32)$$

Daraus folgt $\phi = \mathbf{0}$ in X und damit die Behauptung. \square

3. Die inkompressiblen Navier-Stokes-Gleichungen

Existenz.

Für ein Vektorfeld $\mathbf{b} \in V$ betrachten wir zunächst das folgende Problem: Finde $(\mathbf{u}, p) \in X \times Q$, sodass gilt

$$\begin{aligned} -\nu \Delta \mathbf{u} + \mathbf{b} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega, \\ \operatorname{div} \mathbf{u} &= 0 && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{0} && \text{auf } \partial\Omega, \\ \int_{\Omega} p &= 0. \end{aligned} \tag{3.33}$$

Dieses Problem nennen wir das *Oseen-Problem* (vgl. [Lay08, S. 108-110]). Seine variationelle Formulierung mit den Abbildungen aus (3.8) lautet: Finde $\mathbf{u} \in V$, sodass für alle $\mathbf{v} \in V$ gilt

$$\tilde{a}(\mathbf{u}, \mathbf{v}) := a(\mathbf{u}, \mathbf{v}) + c(\mathbf{b}, \mathbf{u}, \mathbf{v}) = l(\mathbf{v}). \tag{3.34}$$

Lemma 3.9. *Sei $\mathbf{b} \in V$ gegeben. Dann ist die Bilinearform \tilde{a} stetig und koerziv, das heißt, es gilt*

$$\tilde{a}(\mathbf{u}, \mathbf{u}) \geq \nu \|\nabla \mathbf{u}\|_{L^2}^2 \quad \forall \mathbf{u} \in V, \tag{3.35}$$

$$\tilde{a}(\mathbf{u}, \mathbf{v}) \leq (\nu + N \|\mathbf{b}\|_V) \|\mathbf{u}\|_V \|\mathbf{v}\|_V \quad \forall \mathbf{u}, \mathbf{v} \in V. \tag{3.36}$$

Beweis. Die Stetigkeit folgt mit der Cauchy-Schwarz-Ungleichung und Lemma 3.2:

$$\begin{aligned} |\tilde{a}(\mathbf{u}, \mathbf{v})| &\leq |a(\mathbf{u}, \mathbf{v})| + |c(\mathbf{b}, \mathbf{u}, \mathbf{v})| \\ &\leq \nu \|\mathbf{u}\|_V \|\mathbf{v}\|_V + N \|\mathbf{b}\|_V \|\mathbf{u}\|_V \|\mathbf{v}\|_V. \end{aligned} \tag{3.37}$$

Die Koerzivität ist klar, da nach (3.21) gilt, dass $c(\mathbf{b}, \mathbf{u}, \mathbf{u}) = 0$ und somit

$$\tilde{a}(\mathbf{u}, \mathbf{u}) = a(\mathbf{u}, \mathbf{u}) + c(\mathbf{b}, \mathbf{u}, \mathbf{u}) \geq \nu \|\nabla \mathbf{u}\|_{L^2}^2. \tag{3.38}$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Damit sind beide Behauptungen gezeigt. \square

Das Oseen-Problem (3.34) hat eine eindeutige Lösung:

Satz 3.10. *Sei $\mathbf{f} \in X^*$ fest. Dann hat das Problem (3.34) für alle $\mathbf{b} \in V$ eine eindeutige Lösung $\mathbf{u} \in V$. Außerdem gilt die Abschätzung*

$$\nu \|\nabla \mathbf{u}\|_{L^2} \leq \|\mathbf{f}\|_* . \quad (3.39)$$

Beweis. Existenz und Eindeutigkeit der Lösung folgen mit Lemma 3.9 und Lemma 3.2 aus dem Satz von Lax-Milgram [Lay08, S. 28]. Die Abschätzung (3.39) folgt ebenfalls, wegen Lemma 3.9 und Definition 3.7 mit $a_0 = \nu$ und $C = \|\mathbf{f}\|_*$. \square

Definiere die Abbildung $\mathbf{T}: V \rightarrow V$ durch $\mathbf{b} \mapsto \mathbf{u}$, wobei \mathbf{u} die Lösung des Oseen-Problems (3.34) für dieses gegebene \mathbf{b} ist. Wenn nun $\mathbf{u}^* \in V$ ein Fixpunkt von \mathbf{T} ist, das heißt

$$\mathbf{T}(\mathbf{u}^*) = \mathbf{u}^* , \quad (3.40)$$

dann ist \mathbf{u}^* offenbar eine Lösung des Navier-Stokes-Problems (3.17). Mit der Abbildung \mathbf{T} lässt sich die Existenz von Lösungen von (3.17) zeigen.

Satz 3.11 (Existenz für „kleine Daten“). *\mathbf{T} wie oben definiert ist auf V gleichmäßig beschränkt. Für ein $\mathbf{u} = \mathbf{T}(\mathbf{b})$ gilt also*

$$\|\nabla \mathbf{u}\|_{L^2} \leq C < \infty ,$$

wobei $C = C(\nu, \Omega, \|\mathbf{f}\|_*)$ nicht von $\|\nabla \mathbf{b}\|_{L^2}$ abhängt. Weiterhin ist \mathbf{T} unter der Voraussetzung (3.25) eine Kontraktion auf V . Dann existiert eine Lösung von (3.17).

Beweis. Die Existenz einer oberen Schranke an $\|\nabla \mathbf{u}\|_{L^2}$ haben wir bereits mit Satz 3.10 nachgewiesen. Nehme also an, dass (3.25) gilt, d. h.

$$N\nu^{-2} \|\mathbf{f}\|_* < 1 .$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Seien $\mathbf{b}, \tilde{\mathbf{b}} \in V$ gegeben und $\mathbf{u} = \mathbf{T}(\mathbf{b})$, $\tilde{\mathbf{u}} = \mathbf{T}(\tilde{\mathbf{b}})$. Dann gilt mit $\phi := \mathbf{u} - \tilde{\mathbf{u}}$

$$\|\nabla \mathbf{u} - \nabla \tilde{\mathbf{u}}\|_{L^2} = \|\nabla(\mathbf{u} - \tilde{\mathbf{u}})\|_{L^2} = \|\nabla \phi\|_{L^2}. \quad (3.41)$$

Zunächst sehen wir ein, dass

$$\begin{aligned} c(\mathbf{b}, \mathbf{u}, \mathbf{v}) - c(\tilde{\mathbf{b}}, \tilde{\mathbf{u}}, \mathbf{v}) &= \int_{\Omega} (\mathbf{b} \cdot \nabla \mathbf{u} - \tilde{\mathbf{b}} \cdot \nabla \tilde{\mathbf{u}}) \cdot \mathbf{v} \\ &= \int_{\Omega} [(\mathbf{b} - \tilde{\mathbf{b}}) \cdot \nabla \mathbf{u} + \tilde{\mathbf{b}} \cdot (\nabla \mathbf{u} - \nabla \tilde{\mathbf{u}})] \cdot \mathbf{v} \\ &= c(\mathbf{b} - \tilde{\mathbf{b}}, \mathbf{u}, \mathbf{v}) + c(\tilde{\mathbf{b}}, \mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}). \end{aligned} \quad (3.42)$$

Da \mathbf{u} und $\tilde{\mathbf{u}}$ Lösungen von (3.34) zu \mathbf{b} respektive $\tilde{\mathbf{b}}$ sind, gilt

$$a(\mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) + c(\mathbf{b} - \tilde{\mathbf{b}}, \mathbf{u}, \mathbf{v}) + c(\tilde{\mathbf{b}}, \mathbf{u} - \tilde{\mathbf{u}}, \mathbf{v}) = 0. \quad (3.43)$$

Wir setzen wieder $\mathbf{v} = \phi = \mathbf{u} - \tilde{\mathbf{u}}$ und wie im Beweis von Satz 3.8 ergibt sich

$$\begin{aligned} \nu \|\nabla \phi\|_{L^2}^2 &= -c(\mathbf{b} - \tilde{\mathbf{b}}, \mathbf{u}, \phi) \\ &\leq N \left\| \nabla(\mathbf{b} - \tilde{\mathbf{b}}) \right\|_{L^2} \|\nabla \mathbf{u}\|_{L^2} \|\nabla \phi\|_{L^2} \\ &\leq N\nu^{-1} \|\mathbf{f}\|_* \|\nabla \phi\|_{L^2} \left\| \nabla \mathbf{b} - \nabla \tilde{\mathbf{b}} \right\|_{L^2} \end{aligned} \quad (3.44)$$

und schließlich folgt daraus

$$\|\nabla \phi\|_{L^2} \leq N\nu^{-2} \|\mathbf{f}\|_* \left\| \nabla \mathbf{b} - \nabla \tilde{\mathbf{b}} \right\|_{L^2}. \quad (3.45)$$

Also ist \mathbf{T} eine Kontraktion. Mit dem Banachschen Fixpunktsatz (vgl. [Wer11, S. 166]) folgen die Existenz und Eindeutigkeit eines Fixpunktes und somit einer Lösung des Navier-Stokes-Problems (3.17). \square

Zur numerischen Lösung der Navier-Stokes-Gleichungen in den folgenden Abschnitten werden wir es uns zunutze machen, dass die folgende *Picard-Iteration* konvergiert: Gegeben sei mit $\mathbf{u}^n \in V$ eine Lösung des Oseen-Problems ohne Berücksichtigung des

3. Die inkompressiblen Navier-Stokes-Gleichungen

nichtlinearen Terms $\mathbf{u} \cdot \nabla \mathbf{u}$. Dann berechnen wir $\mathbf{T}(\mathbf{u}^n) := \mathbf{u}^{n+1} \in V$ vermöge

$$a(\mathbf{u}^{n+1}, \mathbf{v}) + c(\mathbf{u}^n, \mathbf{u}^{n+1}, \mathbf{v}) = l(\mathbf{v}), \quad \mathbf{v} \in V. \quad (3.46)$$

Satz 3.12 (Picard-Iteration). *Unter der Voraussetzung (3.25) bilden die Iterierten $(\mathbf{u}^n)_{n=1}^\infty$ eine Cauchy-Folge, die aufgrund der Vollständigkeit von V konvergiert. Der Grenzwert*

$$\mathbf{u} = \lim_{n \rightarrow \infty} \mathbf{u}^n$$

ist die eindeutige Lösung des Navier-Stokes-Problems (3.17).

Beweis. Zunächst sei festgestellt, dass eine Lösung $\mathbf{u} \in V$ des elliptischen Problems (3.17) nach Satz 3.11 existiert und nach Satz 3.8 auch eindeutig ist.

Wir definieren $\alpha < 1$ durch $N\nu^{-2} \|\mathbf{f}\|_* \leq \alpha$. Da \mathbf{T} eine Kontraktion ist, gilt für alle $n \in \mathbb{N}$:

$$\left\| \nabla(\mathbf{u}^{n+1} - \mathbf{u}^n) \right\|_{L^2} \leq \alpha \left\| \nabla(\mathbf{u}^n - \mathbf{u}^{n-1}) \right\|_{L^2}. \quad (3.47)$$

Für $n > m \geq 1$ gilt daher mit einem Teleskopsummentrick

$$\left\| \nabla(\mathbf{u}^n - \mathbf{u}^m) \right\|_{L^2} \leq \sum_{k=m}^{n-1} \alpha^k \left\| \nabla(\mathbf{u}^1 - \mathbf{u}^0) \right\|_{L^2} \leq \frac{\alpha^m}{1 - \alpha} \left\| \nabla(\mathbf{u}^1 - \mathbf{u}^0) \right\|_{L^2}. \quad (3.48)$$

Wir schließen

$$\forall \epsilon > 0 \exists N \in \mathbb{N} \forall n > m \geq N : \left\| \nabla(\mathbf{u}^n - \mathbf{u}^m) \right\|_{L^2} < \epsilon, \quad (3.49)$$

und damit ist $(\mathbf{u}^n)_{n=1}^\infty$ eine Cauchy-Folge und somit konvergent.

Nun ist noch zu zeigen, dass der Grenzwert der Folge auch tatsächlich gleich \mathbf{u} ist. Wir setzen $\phi^n := \mathbf{u} - \mathbf{u}^n \in V$ und $\mathbf{v} := \phi^{n+1}$. Mit ähnlichen Umformungen wie im Beweis von Satz 3.11 erhalten wir

$$\left\| \nabla \phi^{n+1} \right\|_{L^2} \leq \alpha \left\| \nabla \phi^n \right\|_{L^2}. \quad (3.50)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Dabei gilt die gleichmäßige Schranke $\nu \|\nabla \mathbf{u}^n\|_{L^2} \leq \|\mathbf{f}\|_*$, da \mathbf{u}^n für alle $n \in \mathbb{N}$ nach Konstruktion eine Lösung der Iterationsvorschrift (3.46) ist. Also gilt

$$\left\| \nabla \phi^{n+1} \right\|_{L^2} \xrightarrow{n \rightarrow \infty} 0, \quad (3.51)$$

woraus wiederum $\mathbf{u} = \lim_{n \rightarrow \infty} \mathbf{u}^n$ in X folgt. \square

Man kann zeigen, dass auch ohne die Bedingung an die Eingabedaten (3.25) eine Lösung des Navier-Stokes-Problems (3.17) existiert. Der Beweis ist allerdings etwas komplizierter, unter anderem benötigt man den *Fixpunktsatz von Schauder*. Daher sei hier nur das Ergebnis zitiert.

Satz 3.13 (Existenz für beliebige Daten). *Für alle $\nu > 0$ und $\mathbf{f} \in X^*$ existiert eine Lösung $\mathbf{u} \in V$ des Navier-Stokes-Problems (3.17).*

Beweis. Siehe bei Layton [Lay08, S. 110-114] oder auch bei Temam [Tem79, S. 164].

3.2. Diskretisierung des Problems

In diesem Abschnitt wird das Navier-Stokes-Problem in seiner schwachen Formulierung numerisch mittels der Finite-Elemente-Methode gelöst. Da das Problem gemäß (3.17) in unendlichdimensionalen Funktionenräumen gestellt ist, ist es für einen Computer aufgrund seiner endlichen Ressourcen nicht möglich, das Problem in endlicher Zeit zu lösen. Daher sollen die beteiligten Funktionenräume durch geeignete endlichdimensionale Ansatzräume diskretisiert und somit auch die Lösung „möglichst gut“ approximiert werden.

Dazu wird zunächst ein *nichtkonformes* Verfahren vorgestellt und die zugehörigen Fehlerabschätzungen hergeleitet. Außerdem wird eine Modifikation des Verfahrens präsentiert, die unter gewissen Voraussetzungen ein besseres Ergebnis als das klassische Verfahren liefern soll. Eine Studie des Verfahrens und der Implementierung durch geeignete Beispiele und ferner der Vergleich mit den Fehlerabschätzungen und zwischen dem klassischen und dem modifizierten Verfahren erfolgen in Kapitel 4.

3.2.1. Diskretisierung mit nichtkonformen finiten Elementen

Sei $\Omega \subset \mathbb{R}^3$ ein konvexes polyedrisches Lipschitz-Gebiet. Um das Gebiet zu diskretisieren, verwenden wir eine reguläre *Triangulierung* \mathcal{T}_h von Ω in Tetraeder $T \in \mathcal{T}_h$. Weiter sei \mathcal{F}_h^* die Menge der Flächen (Dreiecke) in \mathcal{T}_h und \mathcal{F}_h die Menge der inneren Flächen $F \in \mathcal{F}_h$, für die gilt $\partial\Omega \cap F = \emptyset$. Diese seien offene Mengen. Der Schwerpunkt des Dreiecks $F \in \mathcal{F}_h^*$ sei \mathbf{x}_F . Der Normalenvektor \mathbf{n}_F an eine Fläche F hat für $F \in \mathcal{F}_h$ eine beliebige, aber feste Orientierung und für $F \in \mathcal{F}_h^* \setminus \mathcal{F}_h$ ist er nach außen gerichtet.

Für einen Tetraeder $T \in \mathcal{T}_h$ ist \mathbf{x}_T das Baryzentrum, \mathcal{F}_T die Menge der Flächen dieses Tetraeders und $\mathbf{n}_{T,F}$ der nach außen zeigende Normalenvektor an seine Fläche F . Im Falle von zwei angrenzenden Tetraedern $T_1, T_2 \in \mathcal{T}_h$ mit $T_1 \neq T_2$ bezeichnet $T_1|T_2 := F \in \mathcal{F}_h$ die gemeinsame Fläche. Für eine stückweise polynomiale Funktion $\phi \in L^\infty(\Omega)$ ist für jede innere Fläche $F = T_1|T_2 \in \mathcal{F}_h$ der Sprung in einem $\mathbf{x} \in F$ definiert durch (vgl. [Lin13, S. 8])

$$[\phi](\mathbf{x}) := \left(\lim_{\substack{\mathbf{y} \rightarrow \mathbf{x} \\ \mathbf{y} \in T_1}} \phi(\mathbf{y}) \mathbf{n}_{T_1,F} + \lim_{\substack{\mathbf{y} \rightarrow \mathbf{x} \\ \mathbf{y} \in T_2}} \phi(\mathbf{y}) \mathbf{n}_{T_2,F} \right) \cdot \mathbf{n}_F. \quad (3.52)$$

Wie im vorherigen Abschnitt definieren wir $X := H_0^1(\Omega)^n$ und $Q := L_0^2(\Omega)$. Zur Approximation der Geschwindigkeit $\mathbf{u} \in X$ verwenden wir als Ansatzraum den Raum X_h der *Crouzeix-Raviart-Elemente*:

$$\begin{aligned} X_h := \left\{ \mathbf{v}_h \in L^2(\Omega)^n : \mathbf{v}_h|_T \in \mathcal{P}_1^n \ \forall T \in \mathcal{T}_h, \right. \\ \left. [\mathbf{v}_h](\mathbf{x}_F) = \mathbf{0} \ \forall F \in \mathcal{F}_h, \right. \\ \left. \mathbf{v}_h(\mathbf{x}_F) = \mathbf{0} \ \forall F \in \mathcal{F}_h^* \setminus \mathcal{F}_h \right\}. \end{aligned} \quad (3.53)$$

Es bezeichnet \mathcal{P}_1^n den Raum der linearen Polynome $\mathbf{p}: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Die Bedingung $[\mathbf{v}_h](\mathbf{x}_F) = \mathbf{0}$ für alle $F \in \mathcal{F}_h$ bedeutet die Stetigkeit von \mathbf{v}_h im Schwerpunkt der Fläche F . Die homogenen Randbedingungen an die Geschwindigkeit berücksichtigen wir, indem wir fordern, dass \mathbf{v}_h jeweils im Schwerpunkt der Randflächen verschwindet. Auf einem Tetraeder ist ein lineares Polynom $\mathbf{p} \in \mathcal{P}_1^3$ durch die Werte in den Schwer-

3. Die inkompressiblen Navier-Stokes-Gleichungen

punkten der Flächen $F \in \mathcal{F}_T$ eindeutig bestimmt: Diese vier Bedingungen bestimmen genau die vier Freiheitsgrade des Polynoms.

Definition 3.14. Wir definieren den *gebrochenen Gradienten* $\nabla_h: X_h \rightarrow L^2(\Omega)^{n^2}$, die *gebrochene Divergenz* $\operatorname{div}_h: X_h \rightarrow L^2(\Omega)$ und die *gebrochene Rotation* $\nabla_h \times (\cdot): X_h \rightarrow L^2(\Omega)^n$ so, dass für alle $T \in \mathcal{T}_h$ gilt

$$\begin{aligned} (\nabla_h \mathbf{v}_h)|_T &:= \nabla(\mathbf{v}_h|_T), \\ (\operatorname{div}_h \mathbf{v}_h)|_T &:= \operatorname{div}(\mathbf{v}_h|_T), \\ (\nabla_h \times \mathbf{v}_h)|_T &:= \nabla \times \mathbf{v}_h|_T. \end{aligned}$$

Außerdem sei eine Norm auf X_h gegeben:

$$|\mathbf{v}_h|_{1,h} := \left(\int_{\Omega} \nabla_h \mathbf{v}_h : \nabla_h \mathbf{v}_h \right)^{\frac{1}{2}} = \left(\sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{v}_h : \nabla \mathbf{v}_h \right)^{\frac{1}{2}}.$$

Es fällt auf, dass im Allgemeinen eine Funktion $\mathbf{v}_h \in X_h$ am Übergang zwischen zwei Tetraedern Sprünge aufweisen kann, da wir die Stetigkeit nur im Schwerpunkt der Flächen gefordert haben. Außerdem muss eine diskret divergenzfreie Funktion $\mathbf{v}_h \in X_h$ nicht unbedingt eine Divergenz in L^2 besitzen und somit insgesamt divergenzfrei sein. Dementsprechend gilt $X_h \not\subset X$ und, wenn wir

$$V_h := \{\mathbf{v}_h \in X_h : b_h(\mathbf{v}_h, q_h) = 0 \forall q_h \in Q_h\} \subset X_h, \quad (3.54)$$

$$H(\operatorname{div}; \Omega) := \left\{ \mathbf{u} \in L^2(\Omega)^n : \operatorname{div} \mathbf{u} \in L^2(\Omega) \text{ existiert im schwachen Sinne} \right\} \quad (3.55)$$

definieren, auch $V_h \not\subset H(\operatorname{div}; \Omega)$. Wegen $X_h \not\subset X$ bezeichnen wir das Crouzeix-Raviart-Element als *nichtkonform*.

Die Basisfunktionen von X_h haben einen lokalen Träger: Betrachtet man ein Basiselement ϕ_i , so ist nur eine der Komponentenfunktionen ungleich null, und zwar nur auf zwei benachbarten Tetraedern $T_1, T_2 \in \mathcal{T}_h$. Für eine Komponente ϕ_i^j mit $j \in \{1, 2, 3\}$ soll dann gelten $\phi_i^j(\mathbf{x}_{T_1|T_2}) = 1$.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Den Ansatzraum Q_h definieren wir wie folgt:

$$Q_h := \left\{ q_h \in L_0^2(\Omega) : q_h|_T \in \mathcal{P}_0 \forall T \in \mathcal{T}_h \right\}. \quad (3.56)$$

Der Druck wird also durch auf jedem Tetraeder stückweise konstante Funktionen approximiert. Wir stellen fest, dass $Q_h \subset Q$ und somit auch $\|\cdot\|_{Q_h} = \|\cdot\|_Q$ gilt.

Ebenso wie im vorherigen Abschnitt formulieren wir das diskrete Problem in schwacher Form: Finde $(\mathbf{u}_h, p_h) \in X_h \times Q_h$, sodass für alle $\mathbf{v}_h \in X_h$, $q_h \in Q_h$ gilt

$$\begin{aligned} a_h(\mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b_h(\mathbf{v}_h, p_h) &= l_h(\mathbf{v}_h), \\ b_h(\mathbf{u}_h, q_h) &= 0. \end{aligned} \quad (3.57)$$

Dabei seien die diskreten Analoga der Linearformen aus (3.8) wie folgt definiert:

$$\begin{aligned} a_h(\mathbf{u}_h, \mathbf{v}_h) &:= \nu \sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{u}_h : \nabla \mathbf{v}_h, \\ b_h(\mathbf{u}_h, q_h) &:= - \sum_{T \in \mathcal{T}_h} \int_T q_h \operatorname{div} \mathbf{u}_h, \\ c_h(\mathbf{g}_h, \mathbf{u}_h, \mathbf{v}_h) &:= \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{g}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h, \\ l_h(\mathbf{v}_h) &:= \sum_{T \in \mathcal{T}_h} \int_T \mathbf{f} \cdot \mathbf{v}_h. \end{aligned} \quad (3.58)$$

Man beachte, dass diese so definiert sind, dass sie für $\mathbf{u}, \mathbf{v} \in X$ und $q \in Q$ mit denen aus (3.8) übereinstimmen. Allerdings ist die Trilinearform c_h nicht wie die kontinuierliche Version c antisymmetrisch. Das liegt daran, dass wie oben schon bemerkt $X_h \not\subset X$ gilt (nichtkonforme Methode) und eine Funktion $\mathbf{v}_h \in V_h$ im Allgemeinen nicht divergenzfrei ist. Dies hatten wir aber im Beweis der Antisymmetrie von c benutzt. Daher betrachten wir die folgende Variante von c_h :

$$\tilde{c}_h(\mathbf{g}_h, \mathbf{u}_h, \mathbf{v}_h) := \frac{1}{2} [c_h(\mathbf{g}_h, \mathbf{u}_h, \mathbf{v}_h) - c_h(\mathbf{g}_h, \mathbf{v}_h, \mathbf{u}_h)]. \quad (3.59)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Dann stimmen \tilde{c}_h , c_h und c auf $V \times X \times X$ überein und \tilde{c}_h ist antisymmetrisch bezüglich der beiden letzteren Argumente. Insofern lassen wir im Folgenden die Tilde einfach weg und nehmen c_h als antisymmetrisch an.

Sind dann Basen $\{\mathbf{v}_k(\mathbf{x})\}_{k=1}^K$, $\{q_j(\mathbf{x})\}_{j=1}^J$ von X_h und Q_h gegeben, so gilt mit gewissen Koeffizienten λ_k , μ_j (vgl. [Gun02, S. 101f]):

$$\mathbf{u}_h(\mathbf{x}) = \sum_{k=1}^K \lambda_k \mathbf{v}_k(\mathbf{x}), \quad p_h(\mathbf{x}) = \sum_{j=1}^J \mu_j q_j(\mathbf{x}). \quad (3.60)$$

Dann ist das diskrete Navier-Stokes-Problem (3.57) äquivalent zu dem *nichtlinearen* Gleichungssystem

$$\begin{aligned} \sum_{k=1}^K \lambda_k a_h(\mathbf{v}_k, \mathbf{v}_l) + \sum_{k,m=1}^K \lambda_k \lambda_m c_h(\mathbf{v}_m, \mathbf{v}_k, \mathbf{v}_l) + \sum_{j=1}^J \mu_j b_h(\mathbf{v}_l, q_j) &= l_h(\mathbf{v}_l), \\ \sum_{k=1}^K \lambda_k b_h(\mathbf{v}_k, q_i) &= 0 \end{aligned} \quad (3.61)$$

für alle $l = 1, \dots, K$ und $i = 1, \dots, J$.

3.2.2. Existenz und Eindeutigkeit der diskreten Lösung

Wie im kontinuierlichen Fall sind auch für das diskrete Problem zunächst die Existenz und Eindeutigkeit einer Lösung zu klären. Dazu weisen wir die Koerzivität von a_h und die inf-sup-Bedingung für die Finite-Element-Räume X_h und Q_h nach.

Lemma 3.15. *Für alle $h > 0$ und $\mathbf{v}_h \in X_h$ gilt*

$$a_h(\mathbf{v}_h, \mathbf{v}_h) \geq \nu |\mathbf{v}_h|_{1,h}^2. \quad (3.62)$$

Außerdem erfüllt die Bilinearform b_h die inf-sup-Bedingung: Es gibt ein $\beta > 0$, das nicht von h abhängt, sodass für alle $h > 0$ gilt

$$\inf_{q_h \in Q_h \setminus \{0\}} \sup_{\mathbf{v}_h \in X_h \setminus \{0\}} \frac{b_h(\mathbf{v}_h, q_h)}{|\mathbf{v}_h|_{1,h} \|q_h\|_{L^2(\Omega)}} \geq \beta. \quad (3.63)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Schließlich ist $V_h \neq \{\mathbf{0}\}$ und es gilt

$$V_h = \{\mathbf{v}_h \in X_h : \operatorname{div}_h \mathbf{v}_h = 0\}. \quad (3.64)$$

Beweis. Es gilt für $\mathbf{v}_h \in X_h$

$$a_h(\mathbf{v}_h, \mathbf{v}_h) = \nu \sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{v}_h : \nabla \mathbf{v}_h = \nu \int_{\Omega} \nabla_h \mathbf{v}_h : \nabla_h \mathbf{v}_h = \nu |\mathbf{v}_h|_{1,h}^2. \quad (3.65)$$

Um die inf-sup-Bedingung nachzuweisen, definieren wir einen Operator $\pi_h^{\text{CR}}: X \rightarrow X_h$ (siehe [Ver98, S. 71-73]) durch

$$\left(\pi_h^{\text{CR}} \mathbf{v}\right)(\mathbf{x}_F) := \begin{cases} \frac{1}{|F|} \int_F \mathbf{v} \, d\mathcal{H}^{n-1}, & F \in \mathcal{F}_h, \\ \mathbf{0}, & F \in \mathcal{F}_h^* \setminus \mathcal{F}_h. \end{cases} \quad (3.66)$$

Dadurch, dass ein $\mathbf{v}_h \in X_h$ durch den Wert in \mathbf{x}_F schon bestimmt ist, ist π_h^{CR} wohldefiniert. Der Operator ist aufgrund der Linearität des Integrals linear. Seien nun $\mathbf{v} \in X$ und $q_h \in Q_h$ beliebig. Da q_h auf jedem $T \in \mathcal{T}_h$ stückweise konstant ist, folgt

$$\begin{aligned} b_h(\pi_h^{\text{CR}} \mathbf{v}, q_h) &= - \sum_{T \in \mathcal{T}_h} \int_T q_h \operatorname{div}(\pi_h^{\text{CR}} \mathbf{v}) \\ &= - \sum_{T \in \mathcal{T}_h} q_h \int_T \operatorname{div}(\pi_h^{\text{CR}} \mathbf{v}). \end{aligned} \quad (3.67)$$

Nun ist $\pi_h^{\text{CR}} \mathbf{v}$ für jedes $T \in \mathcal{T}_h$ auf $F \in \mathcal{F}_T$ linear und wird somit durch Anwendung der Mittelpunktsregel exakt integriert. Daher gilt

$$\begin{aligned} b_h(\pi_h^{\text{CR}} \mathbf{v}, q_h) &= - \sum_{T \in \mathcal{T}_h} q_h \left(\sum_{F \in \mathcal{F}_T} \int_F \mathbf{n}_{T,F} \cdot \pi_h^{\text{CR}} \mathbf{v} \, d\mathcal{H}^{n-1} \right) \\ &= - \sum_{T \in \mathcal{T}_h} q_h \left(\sum_{F \in \mathcal{F}_T} |F| \mathbf{n}_{T,F} \cdot \pi_h^{\text{CR}} \mathbf{v}(\mathbf{x}_F) \right) \end{aligned} \quad (3.68)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

und mit dem Divergenzsatz folgt

$$\begin{aligned}
b_h(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}, q_h) &= - \sum_{T \in \mathcal{T}_h} q_h \left(\sum_{F \in \mathcal{F}_T} \int_F \mathbf{n}_{T,F} \cdot \mathbf{v} \, d\mathcal{H}^{n-1} \right) \\
&= - \sum_{T \in \mathcal{T}_h} \int_T q_h \operatorname{div} \mathbf{v} \\
&= b_h(\mathbf{v}, q_h) \\
&= b(\mathbf{v}, q_h).
\end{aligned} \tag{3.69}$$

Aufgrund der Linearität von $\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}$ auf den Flächen $F \in \mathcal{F}_T$ eines Tetraeders $T \in \mathcal{T}_h$ ist der Gradient $\nabla(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})$ und damit auch die Ableitung in Normalenrichtung $\frac{\partial \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}}{\partial \mathbf{n}_{T,F}}$ dort jeweils konstant. Wenden wir erneut die Mittelpunktsregel, den Divergenzsatz und die Cauchy-Schwarz-Ungleichung an, gilt auf einem Tetraeder $T \in \mathcal{T}_h$

$$\begin{aligned}
\left| \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}} \right|_{H^1(T)^n}^2 &= \int_T \nabla(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}) : \nabla(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}) \\
&= \int_{\partial T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_T} \cdot \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}} \, d\mathcal{H}^{n-1} - \int_T \Delta(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}) \cdot \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}} \\
&= \sum_{F \in \mathcal{F}_T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_{T,F}} \cdot \int_F \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}} \, d\mathcal{H}^{n-1} \\
&= \sum_{F \in \mathcal{F}_T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_{T,F}} \cdot |F| (\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})(\mathbf{x}_F) \\
&= \sum_{F \in \mathcal{F}_T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_{T,F}} \cdot \int_F \mathbf{v} \, d\mathcal{H}^{n-1} \\
&= \int_{\partial T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_T} \cdot \mathbf{v} \, d\mathcal{H}^{n-1} \\
&= \int_{\partial T} \frac{\partial(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}})}{\partial \mathbf{n}_T} \cdot \mathbf{v} \, d\mathcal{H}^{n-1} - \int_T \Delta(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}) \cdot \mathbf{v} \\
&= \int_T \nabla(\boldsymbol{\pi}_h^{\text{CR}\mathbf{v}}) : \nabla \mathbf{v} \\
&\leq \left| \boldsymbol{\pi}_h^{\text{CR}\mathbf{v}} \right|_{H^1(T)^n} |\mathbf{v}|_{H^1(T)^n}.
\end{aligned} \tag{3.70}$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Summation über alle $T \in \mathcal{T}_h$ und die Cauchy-Schwarz-Ungleichung liefern

$$\left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{1,h}^2 \leq \sum_{T \in \mathcal{T}_h} \left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{H^1(T)^n} |\mathbf{v}|_{H^1(T)^n} \leq \left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{1,h} |\mathbf{v}|_{H^1(\Omega)^n} \quad (3.71)$$

und damit

$$\left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{1,h} \leq |\mathbf{v}|_{H^1(\Omega)^n}. \quad (3.72)$$

Für das kontinuierliche Navier-Stokes-Problem gilt die inf-sup-Bedingung (Lemma 3.4). Es gibt also ein $\beta > 0$ mit

$$\inf_{q \in Q \setminus \{0\}} \sup_{\mathbf{v} \in X \setminus \{0\}} \frac{b(\mathbf{v}, q)}{|\mathbf{v}|_{H^1(\Omega)^n} \|q\|_{L^2(\Omega)}} \geq \beta. \quad (3.73)$$

Sei nun $q_h \in Q_h$ mit $\|q_h\|_{L^2(\Omega)} = 1$. Wir können ein $\mathbf{v} \in X$ mit $|\mathbf{v}|_{H^1(\Omega)^n} \leq 1$ wählen, sodass

$$b(\mathbf{v}, q_h) \geq \beta. \quad (3.74)$$

Mit (3.72) und (3.69) gilt also $\left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{1,h} \leq 1$ und damit

$$\frac{b_h(\boldsymbol{\pi}_h^{\text{CR}} \mathbf{v}, q_h)}{\left| \boldsymbol{\pi}_h^{\text{CR}} \mathbf{v} \right|_{1,h}} \geq b_h(\boldsymbol{\pi}_h^{\text{CR}} \mathbf{v}, q_h) = b(\mathbf{v}, q_h) \geq \beta. \quad (3.75)$$

Damit ist die inf-sup-Bedingung auch für den diskreten Fall gezeigt. Für den Beweis von $V_h \neq \{0\}$ sei auf [Ver98, S. 73] verwiesen. Die letzte Behauptung sehen wir wie folgt ein: Sei $\mathbf{v}_h \in V_h$. Dann ist auch $\mathbf{v}_h \in X_h$ und \mathbf{v}_h ist stückweise linear. Damit ist $\text{div}_h \mathbf{v}_h$ auf jedem Tetraeder $T \in \mathcal{T}_h$ konstant und es gilt zunächst $\text{div}_h \mathbf{v}_h \in L^2(\Omega)$. Mit der Mittelpunktsregel folgt

$$\begin{aligned} \sum_{T \in \mathcal{T}_h} \int_T \text{div} \mathbf{v}_h &= \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \int_F \mathbf{v}_h \cdot \mathbf{n}_{T,F} \\ &= \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} |F| \mathbf{v}_h(\mathbf{x}_F) \cdot \mathbf{n}_{T,F} \\ &= \sum_{F \in \mathcal{F}_h} |F| [\mathbf{v}_h \cdot \mathbf{n}_F]_F(\mathbf{x}_F) + \sum_{F \in \mathcal{F}_h^* \setminus \mathcal{F}_h} |F| \mathbf{v}_h(\mathbf{x}_F) \cdot \mathbf{n}_F = 0. \end{aligned} \quad (3.76)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Dabei bezeichnet $[\mathbf{v}_h \cdot \mathbf{n}_F]_F$ den Sprung der Normalenkomponente von \mathbf{v}_h auf einer Fläche $F = T_1|T_2$ zwischen zwei Tetraedern $T_1, T_2 \in \mathcal{T}_h$. Dieser ist nach Voraussetzung im Schwerpunkt \mathbf{x}_F der Fläche gerade gleich Null. Der letztere Summand verschwindet ebenfalls nach Definition von X_h . Somit gilt $\int_{\Omega} \operatorname{div}_h \mathbf{v}_h = 0$ und daher $\operatorname{div}_h \mathbf{v}_h \in Q_h$, woraus folgt

$$0 = b_h(\mathbf{v}_h, \operatorname{div}_h \mathbf{v}_h) = - \sum_{T \in \mathcal{T}_h} \int_T (\operatorname{div} \mathbf{v}_h)^2, \quad (3.77)$$

also $\operatorname{div}_h \mathbf{v}_h = 0$. Nehmen wir nun an, dass $\mathbf{v}_h \in X_h$ und $\operatorname{div} \mathbf{v}_h = 0$. Dann ergibt sich nach Multiplikation mit einem beliebigen $q_h \in Q_h$, Integration über $T \in \mathcal{T}_h$ und Summation

$$0 = - \sum_{T \in \mathcal{T}_h} \int_T q_h \operatorname{div} \mathbf{v}_h = b_h(\mathbf{v}_h, q_h) \quad (3.78)$$

und somit $\mathbf{v}_h \in V_h$. □

Bemerkung. Die inf-sup-Bedingung ist eine „Verträglichkeitsbedingung“ an die Finite-Element-Räume X_h und Q_h . Sie impliziert auch, dass der diskrete Divergenzoperator stabil invertierbar ist, das heißt, dass das Problem $b_h(\mathbf{v}_h, q_h) = \langle q, q_h \rangle_{L^2(\Omega)}$, $q_h \in Q_h$ stets eine Lösung $\mathbf{v}_h \in V_h$ besitzt mit (vgl. [Gun02, S. 103])

$$|\mathbf{v}_h|_{1,h} \leq C \|q\|_{L^2(\Omega)}, \quad C > 0. \quad (3.79)$$

Wie auch im kontinuierlichen Fall garantiert die inf-sup-Bedingung die Existenz eines assoziierten Drucks p_h zu einem $\mathbf{u}_h \in V_h$. Der folgende Satz weist die Existenz und Eindeutigkeit einer Lösung des diskreten Navier-Stokes-Problems unter der bereits bekannten Bedingung an die Eingabedaten nach. Der Beweis ist analog zum kontinuierlichen Fall. Die Stetigkeit von a_h , b_h und c_h wird ebenso nachgewiesen.

Satz 3.16 (Existenz und Eindeutigkeit für das diskrete Navier-Stokes-Problem). *Seien $\mathbf{f} \in L^2(\Omega)^n$ und $\nu > 0$, sodass die folgende Bedingung erfüllt ist:*

$$N_h \nu^{-2} \|\mathbf{f}\|_{*,h} < 1, \quad (3.80)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

wobei wir definieren

$$N_h := \sup_{\mathbf{g}_h, \mathbf{u}_h, \mathbf{v}_h \in V_h} \frac{|c_h(\mathbf{g}_h, \mathbf{u}_h, \mathbf{v}_h)|}{|\mathbf{g}_h|_{1,h} |\mathbf{u}_h|_{1,h} |\mathbf{v}_h|_{1,h}}, \quad (3.81)$$

$$\|\mathbf{f}\|_{*,h} := \sup_{\mathbf{v}_h \in V_h} \frac{\langle \mathbf{f}, \mathbf{v}_h \rangle_{L^2}}{|\mathbf{v}_h|_{1,h}}. \quad (3.82)$$

Dann hat das diskrete Navier-Stokes-Problem

$$a_h(\mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) = l_h(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h \quad (3.83)$$

genau eine Lösung $\mathbf{u}_h \in V_h$ und diese ist gleich dem Grenzwert der Folge $(\mathbf{u}_h^n)_{n=1}^\infty$ in V_h definiert durch

$$a_h(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + c_h(\mathbf{u}_h^n, \mathbf{u}_h^{n+1}, \mathbf{v}_h) = l_h(\mathbf{v}_h), \quad \mathbf{v}_h \in V_h. \quad (3.84)$$

Bemerkung. Man kann zeigen, dass unter Voraussetzungen an die Approximationseigenschaften von X_h und Q_h (diskrete Poincaré-Ungleichung) $N_h \xrightarrow{h \rightarrow 0} N$ gilt (siehe Girault und Raviart [GR79, S. 123]). Daher nehmen wir, falls nötig, implizit an, dass h klein genug ist.

3.2.3. Fehlerabschätzungen

Aus Gründen der Übersicht fassen wir die bisherigen Ergebnisse noch einmal zusammen:

1. a_h ist koerziv und c_h antisymmetrisch,
2. a_h und b_h sind stetig, c_h ist stetig mit der Konstante N_h ,
3. es gilt die inf-sup-Bedingung für X_h und Q_h ,
4. es existiert ein beschränkter linearer Operator $\pi_h^{\text{CR}}: X \rightarrow X_h$, sodass für alle $q_h \in Q_h$ gilt $b_h(\mathbf{v} - \pi_h^{\text{CR}} \mathbf{v}, q_h) = 0$.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Weiter haben wir angenommen, dass

$$\begin{aligned} N_h \nu^{-2} \|\mathbf{f}\|_{*,h} < 1 &\Leftrightarrow \exists \delta_1 \in (0, 1) : N_h \nu^{-2} \|\mathbf{f}\|_{*,h} \leq 1 - \delta_1, \\ N \nu^{-2} \|\mathbf{f}\|_* < 1 &\Leftrightarrow \exists \delta_2 \in (0, 1) : N \nu^{-2} \|\mathbf{f}\|_* \leq 1 - \delta_2. \end{aligned} \quad (3.85)$$

Die Fehlerterme der folgenden Lemmata bestehen aus mehreren Komponenten: Der Approximationsfehler $\left| \mathbf{u} - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u} \right|_{1,h}$ entsteht durch die Diskretisierung von X durch X_h und ist von der zu lösenden Gleichung unabhängig, ebenso für Q und Q_h im zweiten Term. Der Konsistenzfehler E_h im letzten Term entsteht durch die Diskretisierung mit nichtkonformen Elementen. Mit G_h wird der Fehler bezeichnet, der durch die nichtkonforme Approximation des nichtlinearen Terms der Navier-Stokes-Gleichungen eingebracht wird.

Wir nehmen nun stets an, dass $(\mathbf{u}, p) \in X \times Q$ die eindeutige schwache Lösung des Navier-Stokes-Problems (3.9) und $(\mathbf{u}_h, p_h) \in X_h \times Q_h$ die eindeutige Lösung des diskreten Navier-Stokes-Problems (3.57) bezeichnen. Außerdem sind die vorkommenden Konstanten C unabhängig von h . Die Beweise dieses und des darauffolgenden Lemmas halten sich an [Han84, S. 87f], wo die analogen Abschätzungen für ein nichtkonformes Rechteckelement hergeleitet werden.

Lemma 3.17. *Mit den vorstehenden Resultaten und Annahmen gilt*

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{1,h} \leq C &\left[\|\mathbf{u} - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}\|_{1,h} + \frac{1}{\nu} \inf_{q \in Q_h} \|p - q_h\|_{L^2(\Omega)} \right. \\ &\left. + \frac{1}{\nu} \sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} + \frac{1}{\nu} \sup_{\mathbf{v}_h \in X_h} \frac{|E_h(\mathbf{u}, p, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \right], \end{aligned} \quad (3.86)$$

$$\begin{aligned} \|p - p_h\|_{L^2} \leq C &\left[\|\mathbf{u} - \mathbf{u}_h\|_{1,h} + \inf_{q \in Q_h} \|p - q_h\|_{L^2(\Omega)} \right. \\ &\left. + \sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \mathbf{u}_h, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} + \sup_{\mathbf{v}_h \in X_h} \frac{|E_h(\mathbf{u}, p, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \right]. \end{aligned} \quad (3.87)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Dabei sind G_h und E_h definiert durch

$$E_h(\mathbf{u}, p, \mathbf{v}_h) := a_h(\mathbf{u}, \mathbf{v}_h) + c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) + b_h(\mathbf{v}_h, p) - l_h(\mathbf{v}_h), \quad (3.88)$$

$$G_h(\mathbf{u}, \mathbf{g}_h, \mathbf{v}_h) := c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) - c_h(\mathbf{g}_h, \mathbf{g}_h, \mathbf{v}_h). \quad (3.89)$$

Beweis. Nach Voraussetzung erfüllen \mathbf{u}_h und p_h die Gleichung

$$a_h(\mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b_h(\mathbf{v}_h, p_h) = l_h(\mathbf{v}_h). \quad (3.90)$$

Wir setzen $\mathbf{v}_h = \mathbf{u}_h$ und erhalten wegen $c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) = 0$ und $b_h(\mathbf{u}_h, p_h) = 0$

$$a_h(\mathbf{u}_h, \mathbf{u}_h) = l_h(\mathbf{u}_h). \quad (3.91)$$

Mit der Koerzivitat von a_h und der Definition der Norm $\|\cdot\|_{*,h}$ ergibt sich

$$|\mathbf{u}_h|_{1,h} \leq \frac{l_h(\mathbf{u}_h)}{\nu |\mathbf{u}_h|_{1,h}} \leq \frac{\|\mathbf{f}\|_{*,h}}{\nu}. \quad (3.92)$$

Nach der Dreiecksungleichung gilt

$$|\mathbf{u} - \mathbf{u}_h|_{1,h} \leq \left| \mathbf{u} - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u} \right|_{1,h} + \left| \mathbf{u}_h - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u} \right|_{1,h}, \quad (3.93)$$

also mussen wir noch den zweiten Term abschatzen. Setze dazu $\mathbf{w}_h = \mathbf{u}_h - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u} \in V_h$ und auerdem

$$\begin{aligned} R &= a_h(\mathbf{u}_h, \mathbf{w}_h) - a_h(\boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) \\ &\quad + c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{w}_h) - c_h(\boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}, \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h). \end{aligned} \quad (3.94)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Daraus folgt mit der Antisymmetrie von c_h , der Koerzivitat von a_h , der Definition von N_h , Ungleichung (3.92) und der Annahme (3.85)

$$\begin{aligned}
R &= a_h(\mathbf{w}_h, \mathbf{w}_h) + c_h(\mathbf{w}_h, \mathbf{u}_h, \mathbf{w}_h) \\
&\geq \nu |\mathbf{w}_h|_{1,h}^2 - N_h |\mathbf{u}_h|_{1,h} |\mathbf{w}_h|_{1,h}^2 \\
&\geq \nu |\mathbf{w}_h|_{1,h}^2 \left(1 - \frac{N_h \|\mathbf{f}\|_{*,h}}{\nu^2} \right) \\
&\geq \delta_1 \nu |\mathbf{w}_h|_{1,h}^2.
\end{aligned} \tag{3.95}$$

Mit (3.90) und der Definition von E_h gilt aber auch

$$\begin{aligned}
R &= l_h(\mathbf{w}_h) - b_h(\mathbf{w}_h, p_h) - a_h(\pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - c_h(\pi_h^{\text{CR}} \mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) \\
&= a_h(\mathbf{u}, \mathbf{w}_h) + c_h(\mathbf{u}, \mathbf{u}, \mathbf{w}_h) + b_h(\mathbf{w}_h, p) - E_h(\mathbf{u}, p, \mathbf{w}_h) \\
&\quad - a_h(\pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - c_h(\pi_h^{\text{CR}} \mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - b_h(\mathbf{w}_h, p_h) \\
&= a_h(\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) + c_h(\mathbf{u}, \mathbf{u}, \mathbf{w}_h) - c_h(\pi_h^{\text{CR}} \mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) \\
&\quad + b_h(\mathbf{w}_h, p - p_h) - E_h(\mathbf{u}, p, \mathbf{w}_h).
\end{aligned} \tag{3.96}$$

Da $\mathbf{w}_h \in V_h$ gilt, ist $b_h(\mathbf{w}_h, q_h) = 0$ fur beliebiges $q_h \in Q_h$. Also konnen wir p_h durch solch ein q_h ersetzen und erhalten mit der Definition von G_h

$$\begin{aligned}
R &= a_h(\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) + b_h(\mathbf{w}_h, p - q_h) \\
&\quad + G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - E_h(\mathbf{u}, p, \mathbf{w}_h).
\end{aligned} \tag{3.97}$$

Insgesamt haben wir also

$$\begin{aligned}
\delta_1 \nu |\mathbf{w}_h|_{1,h}^2 &\leq a_h(\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) + b_h(\mathbf{w}_h, p - q_h) \\
&\quad + G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - E_h(\mathbf{u}, p, \mathbf{w}_h) \\
&\leq C_1 \nu \left| \mathbf{u} - \pi_h^{\text{CR}} \mathbf{u} \right|_{1,h} |\mathbf{w}_h|_{1,h} + C_2 |\mathbf{w}_h|_{1,h} \|p - q_h\|_{L^2} \\
&\quad + G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{w}_h) - E_h(\mathbf{u}, p, \mathbf{w}_h).
\end{aligned} \tag{3.98}$$

Durch Einsetzen von \mathbf{w}_h , Division und ubergang zum Infimum (da $q_h \in Q_h$ beliebig

3. Die inkompressiblen Navier-Stokes-Gleichungen

war) beziehungsweise zu den Suprema über X_h ergibt sich

$$\begin{aligned} \left| \mathbf{u}_h - \pi_h^{\text{CR}} \mathbf{u} \right|_{1,h} \leq C & \left[\left| \mathbf{u} - \pi_h^{\text{CR}} \mathbf{u} \right|_{1,h} + \frac{1}{\nu} \inf_{q_h \in Q_h} \|p - q_h\|_{L^2(\Omega)} \right. \\ & \left. + \frac{1}{\nu} \sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} + \frac{1}{\nu} \sup_{\mathbf{v}_h \in X_h} \frac{|E_h(\mathbf{u}, p, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \right], \end{aligned} \quad (3.99)$$

wobei wir die Stetigkeit von a_h und b_h ausgenutzt haben. In Verbindung mit der Dreiecksungleichung (3.93) ist damit die Abschätzung (3.86) gezeigt. Für die zweite Abschätzung (3.87) sei $\mathbf{v}_h \in X_h$ beliebig. Dann gilt wieder mit (3.90) und der Definition von E_h

$$\begin{aligned} b_h(\mathbf{v}_h, p_h) &= -a_h(\mathbf{u}_h, \mathbf{v}_h) - c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + l_h(\mathbf{v}_h) \\ &= a_h(\mathbf{u} - \mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) - c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) \\ &\quad + b_h(\mathbf{v}_h, p) - E_h(\mathbf{u}, p, \mathbf{v}_h). \end{aligned} \quad (3.100)$$

Sei weiter $q_h \in Q_h$ beliebig. Dann ergibt sich mit der Definition von G_h

$$b_h(\mathbf{v}_h, p_h - q_h) = a_h(\mathbf{u} - \mathbf{u}_h, \mathbf{v}_h) + G_h(\mathbf{u}, \mathbf{u}_h, \mathbf{v}_h) - E_h(\mathbf{u}, p, \mathbf{v}_h) + b_h(\mathbf{v}_h, p - q_h). \quad (3.101)$$

Die diskrete inf-sup-Bedingung (3.63) liefert mit der Stetigkeit von a_h und b_h

$$\begin{aligned} \|p_h - q_h\|_{L^2(\Omega)} &\leq \frac{1}{\beta} \sup_{\mathbf{v}_h \in X_h} \frac{b_h(\mathbf{v}_h, p_h - q_h)}{|\mathbf{v}_h|_{1,h}} \\ &\leq C \left[|\mathbf{u} - \mathbf{u}_h|_{1,h} + \|p - q_h\|_{L^2(\Omega)} \right. \\ &\quad \left. + \sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \mathbf{u}_h, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} + \sup_{\mathbf{v}_h \in X_h} \frac{|E_h(\mathbf{u}, p, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \right], \end{aligned} \quad (3.102)$$

da $(p_h - q_h) \in Q_h$. Mit der Dreiecksungleichung

$$\|p - p_h\|_{L^2(\Omega)} \leq \|p - q_h\|_{L^2(\Omega)} + \|p_h - q_h\|_{L^2(\Omega)} \quad (3.103)$$

folgt dann die Abschätzung (3.87). \square

3. Die inkompressiblen Navier-Stokes-Gleichungen

Um das Hauptergebnis dieses Abschnitts beweisen zu können, werden die Terme, in denen G_h und E_h vorkommen, nun einzeln abgeschätzt.

Lemma 3.18. *Es seien außerdem $\mathbf{u} \in H^2(\Omega)^n$ und $p \in H^1(\Omega)$. Dann gibt es eine Konstante C unabhängig von h und \mathbf{u} , sodass gilt*

$$\sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \leq Ch |\mathbf{u}|_{H^2(\Omega)^n} |\mathbf{u}|_{H^1(\Omega)^n}. \quad (3.104)$$

Weiter gibt es für jedes $\mathbf{u}_h \in X_h$ eine Konstante C unabhängig von h , \mathbf{u} und \mathbf{u}_h , sodass gilt

$$\sup_{\mathbf{v}_h \in X_h} \frac{|G_h(\mathbf{u}, \mathbf{u}_h, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \leq C |\mathbf{u} - \mathbf{u}_h|_{1,h} (|\mathbf{u}|_{1,h} + |\mathbf{u}_h|_{1,h}). \quad (3.105)$$

Außerdem gibt es eine Konstante C unabhängig von h , sodass gilt

$$\sup_{\mathbf{v}_h \in X_h} \frac{|E_h(\mathbf{u}, p, \mathbf{v}_h)|}{|\mathbf{v}_h|_{1,h}} \leq Ch \left[|\mathbf{u}|_{H^2(\Omega)^n} (|\mathbf{u}|_{H^1(\Omega)^n} + \nu) + |p|_{H^1(\Omega)} \right]. \quad (3.106)$$

Beweis. Wir nutzen die folgenden Interpolationsabschätzungen (vgl. [CR73]):

$$\begin{aligned} |\mathbf{v} - \pi_h^{\text{CR}} \mathbf{v}|_{1,h} &\leq Ch |\mathbf{v}|_{H^2(\Omega)^n}, \quad \mathbf{v} \in H^2(\Omega)^n, \\ |\mathbf{v} - \pi_h^{\text{CR}} \mathbf{v}|_{1,h} &\leq C |\mathbf{v}|_{H^1(\Omega)^n}, \quad \mathbf{v} \in X. \end{aligned} \quad (3.107)$$

Seien $\mathbf{u} \in X_h \cap H^2(\Omega)^n$ und $\mathbf{v}_h \in X_h$. Dann gilt mit der Dreiecksungleichung

$$\begin{aligned} |G_h(\mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)| &= |c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) - c_h(\pi_h^{\text{CR}} \mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)| \\ &\leq |c_h(\mathbf{u}, \mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)| + |c_h(\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}, \pi_h^{\text{CR}} \mathbf{u}, \mathbf{v}_h)| \\ &\leq C |\mathbf{v}_h|_{1,h} \left[|\mathbf{u}|_{1,h} |\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}|_{1,h} + |\mathbf{u} - \pi_h^{\text{CR}} \mathbf{u}|_{1,h} |\pi_h^{\text{CR}} \mathbf{u}|_{1,h} \right] \\ &\leq Ch |\mathbf{u}|_{H^2(\Omega)^n} |\mathbf{u}|_{H^1(\Omega)^n} |\mathbf{v}_h|_{1,h}. \end{aligned} \quad (3.108)$$

Division durch $|\mathbf{v}_h|_{1,h}$ und Übergang zum Supremum über $\mathbf{v}_h \in X_h$ ergibt dann die

3. Die inkompressiblen Navier-Stokes-Gleichungen

Abschätzung (3.104). Sei weiter $\mathbf{u}_h \in X_h$. Es gilt ähnlich wie zuvor

$$\begin{aligned}
|G_h(\mathbf{u}, \mathbf{u}_h, \mathbf{v}_h)| &= |c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) - c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)| \\
&\leq |c_h(\mathbf{u}, \mathbf{u} - \mathbf{u}_h, \mathbf{v}_h)| + |c_h(\mathbf{u} - \mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)| \\
&\leq C |\mathbf{v}_h|_{1,h} \left(|\mathbf{u} - \mathbf{u}_h|_{1,h} |\mathbf{u}|_{1,h} + |\mathbf{u} - \mathbf{u}_h|_{1,h} |\mathbf{u}_h|_{1,h} \right) \\
&= C |\mathbf{v}_h|_{1,h} |\mathbf{u} - \mathbf{u}_h|_{1,h} \left(|\mathbf{u}|_{1,h} + |\mathbf{u}_h|_{1,h} \right).
\end{aligned} \tag{3.109}$$

Nach Definition von E_h und (3.90) gilt

$$E_h(\mathbf{u}, p, \mathbf{v}_h) = a_h(\mathbf{u} - \mathbf{u}_h, \mathbf{v}_h) + c_h(\mathbf{u}, \mathbf{u}, \mathbf{v}_h) - c_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b_h(\mathbf{v}_h, p - p_h). \tag{3.110}$$

Wegen $\mathbf{u} \in H^2(\Omega)^n$ gilt $\Delta \mathbf{u} \in L^2(\Omega)^n$. Da wir vorausgesetzt haben, dass $(\mathbf{u}, p) \in X \times Q$ die schwache Lösung des Navier-Stokes-Problems ist, folgt

$$-\nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \mathbf{f} = \mathbf{0}. \tag{3.111}$$

Aus der Definition von E_h erhalten wir folglich durch Einsetzen, Anwendung des Divergenzsatzes oder der partiellen Integration auf jedem $T \in \mathcal{T}_h$ und Umsortieren

$$\begin{aligned}
E_h(\mathbf{u}, p, \mathbf{v}_h) &= \sum_{T \in \mathcal{T}_h} \int_T (-\nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \mathbf{f}) \cdot \mathbf{v}_h \\
&\quad + \nu \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \int_F \frac{\partial \mathbf{u}}{\partial \mathbf{n}_{T,F}} \cdot \mathbf{v}_h \, d\mathcal{H}^{n-1} \\
&\quad - \frac{1}{2} \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \int_F [(\mathbf{u} \cdot \mathbf{n}_{T,F}) \mathbf{u}] \cdot \mathbf{v}_h \, d\mathcal{H}^{n-1} \\
&\quad - \sum_{T \in \mathcal{T}_h} \sum_{F \in \mathcal{F}_T} \int_F p \mathbf{v}_h \cdot \mathbf{n}_{T,F} \, d\mathcal{H}^{n-1} \\
&=: 0 + I_1 + I_2 + I_3.
\end{aligned} \tag{3.112}$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Es gelten (vgl. [CR73]) die Abschätzungen

$$|I_1| \leq Ch\nu |\mathbf{u}|_{H^2(\Omega)^n} |\mathbf{v}_h|_{1,h}, \quad (3.113)$$

$$|I_3| \leq Ch |p|_{H^1(\Omega)} |\mathbf{v}_h|_{1,h}. \quad (3.114)$$

Gehen wir in (3.114) von p zu $\sum_{i,j=1}^n \mathbf{u}_i \mathbf{u}_j$ über, erhalten wir (vgl. [Han84, S. 87])

$$\begin{aligned} |I_2| &\leq Ch \left(\sum_{i,j=1}^n |\mathbf{u}_i \mathbf{u}_j|_{H^1(\Omega)} \right) |\mathbf{v}_h|_{1,h} \\ &\leq Ch |\mathbf{u}|_{H^2(\Omega)^n} |\mathbf{u}|_{H^1(\Omega)^n} |\mathbf{v}_h|_{1,h} \end{aligned} \quad (3.115)$$

und daraus folgt die Behauptung. \square

Jetzt kann das zentrale Ergebnis dieses Abschnitts bewiesen werden.

Satz 3.19. *Sei $\Omega \subset \mathbb{R}^3$ ein konvexes polyedrisches Lipschitz-Gebiet. Seien $(\mathbf{u}, p) \in X \times Q$ die eindeutige schwache Lösung des Navier-Stokes-Problems (3.9) und außerdem $(\mathbf{u}_h, p_h) \in X_h \times Q_h$ die eindeutige Lösung des diskreten Navier-Stokes-Problems (3.57). Weiter seien $\mathbf{u} \in H^2(\Omega)^n$ und $p \in H^1(\Omega)$. Dann gelten die Fehlerabschätzungen*

$$\begin{aligned} |\mathbf{u} - \mathbf{u}_h|_{1,h} &\leq Ch \left[|\mathbf{u}|_{H^2(\Omega)^n} \left(\frac{1}{\nu} |\mathbf{u}|_{H^1(\Omega)^n} + \nu + \frac{1}{\nu} \right) + \frac{1}{\nu} |p|_{H^1(\Omega)} \right], \\ \|p - p_h\|_{L^2(\Omega)} &\leq C \left(|\mathbf{u}|_{H^2(\Omega)^n}, |\mathbf{u}|_{H^1(\Omega)^n}, |p|_{H^1(\Omega)}, \frac{1}{\nu} \right) h. \end{aligned} \quad (3.116)$$

sowie

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)^n} \leq C \left(|\mathbf{u}|_{H^2(\Omega)^n}, |\mathbf{u}|_{H^1(\Omega)^n}, |p|_{H^1(\Omega)}, \frac{1}{\nu} \right) h^2. \quad (3.117)$$

Beweis. Zunächst ergibt sich die Abschätzung

$$|\mathbf{u} - \mathbf{u}_h|_{1,h} \leq Ch \left[|\mathbf{u}|_{H^2(\Omega)^n} \left(\frac{1}{\nu} |\mathbf{u}|_{H^1(\Omega)^n} + \nu + \frac{1}{\nu} \right) + \frac{1}{\nu} |p|_{H^1(\Omega)} \right] \quad (3.118)$$

aus der Ungleichung (3.86), den Abschätzungen für G_h (3.104) und E_h (3.106), den

3. Die inkompressiblen Navier-Stokes-Gleichungen

Interpolationsabschätzungen (3.107) und

$$\inf_{q_h \in Q_h} \|p - q_h\|_{L^2(\Omega)} \leq Ch |p|_{H^1(\Omega)}, \quad p \in H^1(\Omega). \quad (3.119)$$

Mit den Abschätzungen (3.87), (3.105) und (3.106) erhalten wir

$$\begin{aligned} \|p - p_h\|_{L^2} &\leq C \left[|\mathbf{u} - \mathbf{u}_h|_{1,h} + h |p|_{H^1(\Omega)} + |\mathbf{u} - \mathbf{u}_h|_{1,h} \left(|\mathbf{u}|_{1,h} + |\mathbf{u}_h|_{1,h} \right) \right. \\ &\quad \left. + h |\mathbf{u}|_{H^2(\Omega)^n} \left(|\mathbf{u}|_{H^1(\Omega)^n} + \nu \right) \right] \\ &\leq C \left[|\mathbf{u} - \mathbf{u}_h|_{1,h} \left(|\mathbf{u}|_{H^1(\Omega)^n} + |\mathbf{u}_h - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}|_{1,h} + 1 \right) \right. \\ &\quad \left. + h |\mathbf{u}|_{H^2(\Omega)^n} \left(|\mathbf{u}|_{H^1(\Omega)^n} + \nu \right) + h |p|_{H^1(\Omega)} \right]. \end{aligned} \quad (3.120)$$

Einsetzen von (3.99) und der bereits gezeigten Abschätzung für $|\mathbf{u} - \mathbf{u}_h|_{1,h}$ liefert

$$\begin{aligned} \|p - p_h\|_{L^2} &\leq Ch \left[\left(|\mathbf{u}|_{H^2(\Omega)^n} \left[\frac{1}{\nu} |\mathbf{u}|_{H^1(\Omega)^n} + \nu + \frac{1}{\nu} \right] + \frac{1}{\nu} |p|_{H^1(\Omega)} \right) \right. \\ &\quad \left(|\mathbf{u}|_{H^1(\Omega)^n} + |\mathbf{u}_h - \boldsymbol{\pi}_h^{\text{CR}} \mathbf{u}|_{1,h} + 1 \right) \\ &\quad \left. + |\mathbf{u}|_{H^2(\Omega)^n} \left(|\mathbf{u}|_{H^1(\Omega)^n} + \nu \right) + |p|_{H^1(\Omega)} \right] \\ &= C \left(|\mathbf{u}|_{H^2(\Omega)^n}, |\mathbf{u}|_{H^1(\Omega)^n}, |p|_{H^1(\Omega)}, \frac{1}{\nu} \right) h. \end{aligned} \quad (3.121)$$

Für den Beweis der Abschätzung in der L^2 -Norm (3.117) sei auf [DXS00, S. 146ff] verwiesen. \square

Bemerkung. Für $\mathbf{f} \in L^2(\Omega)^n$ und ein konvexes polyedrisches Lipschitz-Gebiet $\Omega \subset \mathbb{R}^n$ gilt: Jede schwache Lösung (\mathbf{u}, p) des Navier-Stokes-Problems ist in $H^2(\Omega)^n \times H^1(\Omega)$ (vgl. [Ver98, S. 113f]). Deswegen sind die Annahmen in Satz 3.19 gerechtfertigt.

3.2.4. Der divergenzfreie Rekonstruktionsansatz

Wie bereits bemerkt, ist ein Crouzeix-Raviart-Element $\mathbf{v}_h \in V_h \subset X_h$ zwar diskret divergenzfrei, aber nicht insgesamt divergenzfrei. Gegebenenfalls existiert noch nicht einmal eine Divergenz in L^2 , es gilt also $V_h \not\subset H(\text{div}; \Omega)$. Dies äußert sich darin, dass die obere Schranke für den Geschwindigkeitsfehler (Satz 3.19) offenbar von dem Druck p abhängt. Den Grund dafür sehen wir durch die folgenden Betrachtungen ein (siehe dazu auch [Lin13, S. 6ff]).

Die schwache Formulierung des Navier-Stokes-Problems lautet wie in (3.7): Zu gegebenem $\mathbf{f} \in L^2(\Omega)^n$ finde $\mathbf{u} \in X$, $p \in Q$, so dass für alle $\mathbf{v} \in X$, $q \in Q$ gilt

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_{\Omega} p \operatorname{div} \mathbf{v} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v}, \\ \int_{\Omega} q \operatorname{div} \mathbf{u} &= 0. \end{aligned} \quad (3.122)$$

Um das Geschwindigkeitsfeld \mathbf{u} zu bestimmen, reicht es aus, das Problem (3.17) zu lösen: Finde $\mathbf{u} \in V$, sodass für alle $\mathbf{v} \in V$ gilt

$$\nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}. \quad (3.123)$$

Dann lässt sich $p \in Q$ wie in (3.20) bestimmen: Für alle $\mathbf{v} \in V^\perp$ soll gelten

$$- \int_{\Omega} p \operatorname{div} \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} - \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}. \quad (3.124)$$

Der Einfachheit halber nehmen wir nun $\mathbf{u} \in L^\infty(\Omega)^n$ an, denn dann gilt auch $\mathbf{u} \cdot \nabla \mathbf{u} \in L^2(\Omega)^n$. In Abschnitt 2.3 wurde bereits festgestellt, dass ein divergenzfreies Vektorfeld $\mathbf{v} \in V$ und ein rotationsfreies Vektorfeld $\nabla \psi$, $\psi \in L_0^2(\Omega)$, orthogonal bezüglich des L^2 -Skalarproduktes sind. Daher ist (3.123) äquivalent zu: Finde $\mathbf{u} \in V$, sodass für alle $\mathbf{v} \in V$ gilt

$$\nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} \mathbb{P}(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} = \int_{\Omega} \mathbb{P}(\mathbf{f}) \cdot \mathbf{v}, \quad (3.125)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

mit der in Abschnitt 2.3 eingeführten Helmholtz-Projektion \mathbb{P} , die einem Vektorfeld $\mathbf{f} \in L^2(\Omega)^n$ den divergenzfreien Anteil \mathbf{w} der Helmholtz-Zerlegung $\mathbf{f} = \mathbf{w} + \nabla\psi$ von \mathbf{f} zuordnet. Gehen wir von \mathbf{f} zu $\mathbf{f} + \nabla\psi$ über, so ergibt sich

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} \mathbb{P}(\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} &= \int_{\Omega} \mathbb{P}(\mathbf{f} + \nabla\psi) \cdot \mathbf{v} \\ &= \int_{\Omega} \mathbb{P}(\mathbf{f}) \cdot \mathbf{v} \end{aligned} \quad (3.126)$$

für alle $\mathbf{v} \in V$. Trotz der Transformation des Gradientenanteils von \mathbf{f} nach $\mathbf{f} \rightarrow \mathbf{f} + \nabla\psi$ bleibt das Geschwindigkeitsfeld \mathbf{u} also gleich. Es gilt nämlich nach dem Divergenzansatz

$$\begin{aligned} \int_{\Omega} (\mathbf{f} + \nabla\psi) \cdot \mathbf{v} &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Omega} \nabla\psi \cdot \mathbf{v} \\ &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} - \int_{\Omega} \psi \operatorname{div} \mathbf{v} + \int_{\partial\Omega} (\mathbf{v} \cdot \mathbf{n})\psi \, d\mathcal{H}^{n-1} \\ &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \end{aligned} \quad (3.127)$$

und daher auch $\mathbb{P}(\nabla\psi) = \mathbf{0}$. Wenden wir die Transformation auf das Problem (3.124) an, erhalten wir für alle $\mathbf{v} \in V^\perp$

$$\begin{aligned} - \int_{\Omega} p \operatorname{div} \mathbf{v} &= \int_{\Omega} (\mathbf{f} + \nabla\psi) \cdot \mathbf{v} - \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} \\ &= - \int_{\Omega} \psi \operatorname{div} \mathbf{v} + \int_{\Omega} \mathbf{f} \cdot \mathbf{v} - \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v}. \end{aligned} \quad (3.128)$$

Aus der Transformation $\mathbf{f} \rightarrow \mathbf{f} + \nabla\psi$ folgt also $p \rightarrow p + \psi$. Insgesamt bewirkt die Transformation also eine Änderung der Lösung gemäß $(\mathbf{u}, p) \rightarrow (\mathbf{u}, p + \psi)$, das heißt, dass die zusätzlich eingebrachte rotationsfreie Kraft $\nabla\psi$ in den Navier-Stokes-Gleichungen durch den Druckgradienten ∇p bilanziert wird.

Die hier vorgestellte Diskretisierung der Navier-Stokes-Gleichungen weist diese Invarianz des Geschwindigkeitsfeldes allerdings nicht auf. Das diskrete Navier-Stokes-Problem lautet: Finde $\mathbf{u}_h \in V_h$, sodass für alle $\mathbf{v}_h \in V_h$ gilt

$$\nu \sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{u}_h : \nabla \mathbf{v}_h + \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h = \sum_{T \in \mathcal{T}_h} \int_T \mathbf{f} \cdot \mathbf{v}_h. \quad (3.129)$$

3. Die inkompressiblen Navier-Stokes-Gleichungen

Wenden wir die Transformation der rechten Seite wie oben an, so ergibt sich

$$\begin{aligned} \nu \sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{u}_h : \nabla \mathbf{v}_h + \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h &= \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{f} + \nabla \psi) \cdot \mathbf{v}_h \quad (3.130) \\ &= \sum_{T \in \mathcal{T}_h} \int_T \mathbf{f} \cdot \mathbf{v}_h + \sum_{T \in \mathcal{T}_h} \int_T \nabla \psi \cdot \mathbf{v}_h. \end{aligned}$$

Auf den letzteren Summanden wenden wir nun den Divergenzansatz an. Wir erhalten, da $(\operatorname{div} \mathbf{v}_h)|_T = 0$ wegen $\mathbf{v}_h \in V_h$ gilt

$$\begin{aligned} \sum_{T \in \mathcal{T}_h} \int_T \nabla \psi \cdot \mathbf{v}_h &= - \sum_{T \in \mathcal{T}_h} \int_T \psi \operatorname{div} \mathbf{v}_h + \sum_{T \in \mathcal{T}_h} \int_T \operatorname{div} (\psi \mathbf{v}_h) \quad (3.131) \\ &= \sum_{T \in \mathcal{T}_h} \int_T \operatorname{div} (\psi \mathbf{v}_h) \\ &= \sum_{F \in \mathcal{F}_h} \int_F [\psi \mathbf{v}_h \cdot \mathbf{n}_F]_F \, d\mathcal{H}^{n-1} + \sum_{F \in \mathcal{F}_h^* \setminus \mathcal{F}_h} \int_F \psi \mathbf{v}_h \cdot \mathbf{n}_F \, d\mathcal{H}^{n-1}. \end{aligned}$$

Die Randintegrale verschwinden im Allgemeinen nicht, da für das Crouzeix-Raviart-Element der Sprung auf einer Fläche $F \in \mathcal{F}_h$ nur im Schwerpunkt \mathbf{x}_F in jedem Fall verschwindet. Daher ist das diskrete Navier-Stokes-Problem (3.129) *nicht* äquivalent zu: Finde $\mathbf{u}_h \in V_h$, sodass für alle $\mathbf{v}_h \in V_h$ gilt

$$\nu \sum_{T \in \mathcal{T}_h} \int_T \nabla \mathbf{u}_h : \nabla \mathbf{v}_h + \sum_{T \in \mathcal{T}_h} \int_T \mathbb{P}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h = \sum_{T \in \mathcal{T}_h} \int_T \mathbb{P}(\mathbf{f}) \cdot \mathbf{v}_h. \quad (3.132)$$

Die Diskretisierung erhält also nicht die Trennung von divergenzfreien und rotationsfreien Kräften, die das kontinuierliche Problem aufweist. Daher ist die obere Schranke des Geschwindigkeitsfehlers abhängig von dem Druck p und die Methode liefert unter Umständen ein insofern der zugrunde liegenden Physik nicht entsprechendes Ergebnis.

Im Folgenden wird daher ein Rekonstruktionsoperator π_h^{RT} eingeführt, der einem diskret divergenzfreien Vektorfeld ein echt divergenzfreies Vektorfeld zuweist. Dann

3. Die inkompressiblen Navier-Stokes-Gleichungen

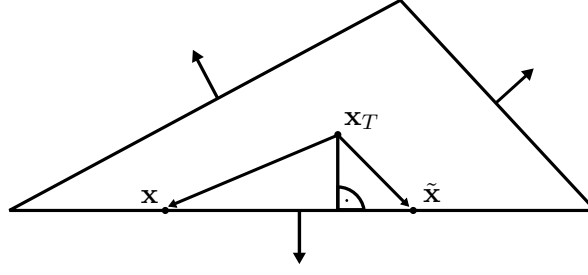


Abbildung 1: Für das Raviart-Thomas-Element ist die Normalkomponente $(\mathbf{x} - \mathbf{x}_T) \cdot \mathbf{n}_F$ jeweils konstant

gilt nämlich wie im kontinuierlichen Fall (vgl. [Lin13, S. 8])

$$\begin{aligned} \int_{\Omega} \mathbf{f} \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h &= \int_{\Omega} \mathbb{P}(\mathbf{f}) \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h, \\ \int_{\Omega} (\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h &= \int_{\Omega} \mathbb{P}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h. \end{aligned} \quad (3.133)$$

Dazu definieren wir den Raum der *Raviart-Thomas-Elemente* niedrigster Ordnung durch

$$\begin{aligned} \text{RT}_h := \left\{ \mathbf{v}_h \in L^2(\Omega)^n : \mathbf{v}_h|_T(\mathbf{x}) &= \mathbf{a}_T + \frac{b_T}{n}(\mathbf{x} - \mathbf{x}_T), T \in \mathcal{T}_h, \mathbf{a}_T \in \mathbb{R}^n, b_T \in \mathbb{R}, \right. \\ &[\mathbf{v}_h \cdot \mathbf{n}_F](\mathbf{x}_F) = 0 \quad \forall F \in \mathcal{F}_h, \\ &\left. \mathbf{v}_h(\mathbf{x}_F) \cdot \mathbf{n}_F = 0 \quad \forall F \in \mathcal{F}_h^* \setminus \mathcal{F}_h \right\}. \end{aligned} \quad (3.134)$$

Zunächst ist zu bemerken, dass auf jeder Fläche $F \in \mathcal{F}_T$ eines Tetraeders $T \in \mathcal{T}_h$ die Normalkomponente $\mathbf{v}_h(\mathbf{x}) \cdot \mathbf{n}_F$ eines Elements $\mathbf{v}_h \in \text{RT}_h$ für jedes $\mathbf{x} \in F$ konstant ist, denn für $\mathbf{x} \in F$ ist $(\mathbf{x} - \mathbf{x}_T) \cdot \mathbf{n}_F$ konstant (siehe Abbildung 1). Also gilt

$$\begin{aligned} \mathbf{v}_h(\mathbf{x}) \cdot \mathbf{n}_F &= \left[\mathbf{a}_T + \frac{b_T}{n}(\mathbf{x} - \mathbf{x}_T) \right] \cdot \mathbf{n}_F \\ &= \mathbf{a}_T \cdot \mathbf{n}_F + \frac{b_T}{n}(\mathbf{x} - \mathbf{x}_T) \cdot \mathbf{n}_F \\ &= \text{const.} \end{aligned} \quad (3.135)$$

Weiterhin gilt nach Definition auch $[\mathbf{v}_h \cdot \mathbf{n}_F](\mathbf{x}_F) = 0$. Da, wie eben gesehen, die Normalkomponente konstant auf einer Fläche ist, gilt somit auch $[\mathbf{v}_h \cdot \mathbf{n}_F](\mathbf{x}) = 0$ für

3. Die inkompressiblen Navier-Stokes-Gleichungen

alle $\mathbf{x} \in F$. Insofern sind die Normalkomponenten von \mathbf{v}_h auf den Flächen $F \in \mathcal{F}_h$ stetig, weshalb $\text{RT}_h \subset H(\text{div}; \Omega)$ gilt (vgl. [BF91, S. 95]).

Der Rekonstruktionsoperator $\pi_h^{\text{RT}} : X \cup X_h \rightarrow \text{RT}_h$ ist dann definiert durch

$$\mathbf{n}_F \cdot (\pi_h^{\text{RT}} \mathbf{v})(\mathbf{x}_F) := \begin{cases} \frac{1}{|F|} \int_F \mathbf{v} \cdot \mathbf{n}_F \, d\mathcal{H}^{n-1}, & F \in \mathcal{F}_h, \\ 0, & F \in \mathcal{F}_h^* \setminus \mathcal{F}_h. \end{cases} \quad (3.136)$$

Die Normalkomponente eines Elements $\pi_h^{\text{RT}} \mathbf{v}$ auf einer Fläche $F \in \mathcal{F}_h$ entspricht also genau dem Mittelwert der Normalkomponente von \mathbf{v} auf F .

Für einen gegebenen Tetraeder $T \in \mathcal{T}_h$ müssen die Parameter $\mathbf{a}_T \in \mathbb{R}^n$, $\mathbf{b}_T \in \mathbb{R}$ bestimmt werden. Offenbar reicht es aus, dies für die Basiselemente von X_h zu tun, da ein $\mathbf{v}_h \in X_h$ ja als Linearkombination dieser Basiselemente dargestellt werden kann. Es sei (ϕ_i) eine Basis von X_h und $\pi_h^{\text{RT}} \phi_i = \mathbf{a}_T + \frac{\mathbf{b}_T}{n}(\mathbf{x} - \mathbf{x}_T)$. Zunächst wählen wir

$$b_T := \text{div } \phi_i, \quad (3.137)$$

denn dann gilt

$$\text{div } \pi_h^{\text{RT}} \phi_i = \frac{b_T}{3} \sum_{k=1}^3 \frac{\partial}{\partial x_k} (x_k) = b_T = \text{div } \phi_i. \quad (3.138)$$

Aufgrund der Linearität von ϕ_i ist $\text{div } \phi_i$ konstant. Also gilt unter Anwendung des Divergenzsatzes und der Mittelpunktsregel

$$\begin{aligned} \text{div } \phi_i &= \frac{1}{|T|} \int_T \text{div } \phi_i \\ &= \frac{1}{|T|} \sum_{F \in \mathcal{F}_T} \int_F \phi_i \cdot \mathbf{n}_{T,F} \, d\mathcal{H}^{n-1} = \frac{1}{|T|} \sum_{F \in \mathcal{F}_T} |F| \phi_i(\mathbf{x}_F) \cdot \mathbf{n}_{T,F}. \end{aligned} \quad (3.139)$$

Um \mathbf{a}_T zu berechnen, sehen wir zunächst ein, dass $\mathbf{x}_T = \frac{1}{|T|} \int_T \mathbf{x}$ für jedes $T \in \mathcal{T}_h$ gilt. Da jeder Tetraeder T durch eine affine Transformation in den Tetraeder T_0 mit den Eckpunkten $(0, 0, 0)^T$, $(1, 0, 0)^T$, $(0, 1, 0)^T$ und $(0, 0, 1)^T$ überführt werden kann,

3. Die inkompressiblen Navier-Stokes-Gleichungen

wird die Aussage nur für diesen nachgewiesen. Es gilt

$$\frac{1}{|T_0|} \int_{T_0} \mathbf{x} = \frac{1}{|T_0|} \int_0^1 \int_0^{1-x_1} \int_0^{1-x_1-x_2} (x_1, x_2, x_3)^T dx_3 dx_2 dx_1 = \frac{1}{4}(1, 1, 1)^T = \mathbf{x}_{T_0}. \quad (3.140)$$

Daraus folgt durch Einsetzen, dass

$$\mathbf{a}_T = \frac{1}{|T|} \int_T \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i. \quad (3.141)$$

Mit der Produktregel gilt für jede Komponente $j = 1, \dots, 3$

$$\mathbf{a}_T^j = \frac{1}{|T|} \int_T (\boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i)^j = \frac{1}{|T|} \int_T \operatorname{div} (x^j \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i) - \frac{1}{|T|} \int_T x^j \operatorname{div} \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i. \quad (3.142)$$

Wie schon bemerkt, sind $\operatorname{div} \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i$ und die Normalkomponente auf einer Fläche eines Tetraeders konstant. Wir erhalten mit dem Divergenzsatz und der Mittelpunktsregel

$$\begin{aligned} \mathbf{a}_T^j &= \frac{1}{|T|} \sum_{F \in \mathcal{F}_T} \left(\int_F x^j \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i \cdot \mathbf{n}_{T,F} d\mathcal{H}^{n-1} \right) - \mathbf{x}_T^j \operatorname{div} \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i \\ &= \frac{1}{|T|} \sum_{F \in \mathcal{F}_T} \left(|F| \mathbf{x}_F^j \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i \cdot \mathbf{n}_{T,F} \right) - \mathbf{x}_T^j \operatorname{div} \boldsymbol{\pi}_h^{\text{RT}} \boldsymbol{\phi}_i. \end{aligned} \quad (3.143)$$

Daher folgt mit (3.138) und (3.139), dass

$$\mathbf{a}_T = \frac{1}{|T|} \sum_{F \in \mathcal{F}_T} |F| \boldsymbol{\phi}_i(\mathbf{x}_F) \cdot \mathbf{n}_{T,F} (\mathbf{x}_F - \mathbf{x}_T). \quad (3.144)$$

Durch die Wahl der Parameter gilt insbesondere für alle $\mathbf{v}_h \in V_h$

$$\operatorname{div} \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h = \operatorname{div}_h \mathbf{v}_h. \quad (3.145)$$

Ein diskret divergenzfreies Vektorfeld wird also durch den Rekonstruktionsoperator $\boldsymbol{\pi}_h^{\text{RT}}$ auf ein divergenzfreies Vektorfeld abgebildet. Daher gelten die Gleichungen (3.133) und es ist zu erwarten, dass die obere Schranke an den Geschwindigkeitsfehler nicht mehr vom Druck abhängt.

3. Die inkompressiblen Navier-Stokes-Gleichungen

Bemerkung. Für die Diskretisierung des Stokes-Problems mit dem Rekonstruktionsoperator wird eine verbesserte Fehlerabschätzung in [Lin13, S. 14] gezeigt. Das Stokes-Problem lautet: Finde $(\mathbf{u}_h, p_h) \in X_h \times Q_h$, sodass für alle $(\mathbf{v}_h, q_h) \in X_h \times Q_h$ gilt

$$\begin{aligned} \nu \int_{\Omega} \nabla_h \mathbf{u}_h : \nabla_h \mathbf{v}_h - \int_{\Omega} p_h \operatorname{div}_h \mathbf{v}_h &= \int_{\Omega} \mathbf{f} \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h, \\ - \int_{\Omega} q_h \operatorname{div}_h \mathbf{u}_h &= 0. \end{aligned} \quad (3.146)$$

Gilt nun für die Lösung $(\mathbf{u}, p) \in H^2(\Omega)^n \times H^1(\Omega)$, dann gelten für die diskrete Lösung $(\mathbf{u}_h, p_h) \in X_h \times Q_h$ die Fehlerabschätzungen

$$\begin{aligned} |\mathbf{u} - \mathbf{u}_h|_{1,h} &\leq Ch |\mathbf{u}|_{H^2(\Omega)^n}, \\ \|p - p_h\|_{L^2(\Omega)} &\leq Ch \left(\nu |\mathbf{u}|_{H^2(\Omega)^n} + |p|_{H^1(\Omega)} \right). \end{aligned} \quad (3.147)$$

Durch den Rekonstruktionsoperator wird also im Vergleich zur klassischen Crouzeix-Raviart-Diskretisierung wie erwartet die obere Schranke an den Geschwindigkeitsfehler von der Norm des Drucks p entkoppelt.

4. Implementierung und Studie des numerischen Verfahrens

Um die Verfahren und die theoretisch hergeleiteten Fehlerabschätzungen numerisch zu überprüfen, sollen nun die Ergebnisse einiger Beispiele vorgestellt werden. Die Beispiele sind dabei überwiegend so konstruiert, dass die exakte Lösung der Navier-Stokes-Gleichungen bekannt ist und so mit der numerisch berechneten Lösung verglichen werden kann. Die Programmierung beider Verfahren erfolgte in C++ unter Benutzung der Bibliothek `PDELIB2`.¹ Diese stellt unter anderem Routinen zur Ansteuerung von Gittergeneratoren, Gewichte numerischer Quadraturformeln und ein Werkzeug zur graphischen Anzeige des Geschwindigkeitsfeldes sowie des Drucks zur Verfügung. Zur Lösung von linearen Gleichungssystemen wurde `PARDISO`² verwendet.

Die Implementierung sowohl der klassischen Crouzeix-Raviart-Diskretisierung als auch der modifizierten Diskretisierung mit dem Rekonstruktionsoperator folgen demselben groben Ablauf: Assemblierung der Steifigkeitsmatrix, Assemblierung der rechten Seite, Lösung des Gleichungssystems und Berechnung der Fehlernormen. Zur Lösung des in (3.61) bereits vorgestellten nichtlinearen Gleichungssystems setzen wir die Picard-Iteration ein. Der Ablauf des Programms ist in Algorithmus 1 dargestellt. Der Programmcode befindet sich auszugsweise in Anhang A.

Der nichtlineare Term $\mathbf{u} \cdot \nabla \mathbf{u}$ wird in der *Rotationsform* diskretisiert. Diese beruht auf der Vektoridentität

$$\mathbf{u} \cdot \nabla \mathbf{u} \equiv (\nabla \times \mathbf{u}) \times \mathbf{u} + \frac{1}{2} \nabla (\mathbf{u} \cdot \mathbf{u}). \quad (4.1)$$

¹URL: <https://www.wias-berlin.de/software/pdelib/> (Zuletzt abgerufen am 23.09.2013)

²URL: <http://www.pardiso-project.org/> (Zuletzt abgerufen am 23.09.2013)

4. Implementierung und Studie des numerischen Verfahrens

Algorithmus 1 Numerische Lösung der Navier-Stokes-Gleichungen

```

1   $n_{\text{dof}} \leftarrow 3 |\mathcal{F}_h^*| + |\mathcal{T}_h|$ 
2   $\text{SMat} \leftarrow \text{MATRIX}(n_{\text{dof}}, n_{\text{dof}})$ 
3   $\text{RHS} \leftarrow \text{VECTOR}(n_{\text{dof}})$ 
4  for all  $T \in \mathcal{T}_h$  do → Assemblierung der Stokes-Matrix
5    Bestimme Koeffizienten  $a_k, b_k, c_k, d_k$  der linearen Basisfunktionen  $\phi_k$  auf  $T$ 
6    Assembliere den Laplace- und Divergenzoperator sowie den Druckgradienten
7    Assembliere die rechte Seite
8  end for
9  for all  $F \in \mathcal{F}_h^* \setminus \mathcal{F}_h$  do → Assemblierung der Randwerte
10    $\text{RHS}(F) \leftarrow \frac{1}{|F|} \int_F \mathbf{u} \, d\mathcal{H}^{n-1}$ 
11 end for
12 Löse lineares Gleichungssystem  $\text{SMat} \cdot (\mathbf{u}_h^0, p_h^0)^T = \text{RHS}$ 
13 for  $i \leftarrow 1, \dots, i_{\text{max}}$  do → Picard-Iteration
14    $\text{NIMat} \leftarrow \text{SMat}$ 
15   for all  $T \in \mathcal{T}_h$  do
16     Assembliere nichtlinearen Term in  $\text{NIMat}$  mithilfe  $\mathbf{u}_h^{i-1}$ 
17   end for
18   Löse lineares Gleichungssystem  $\text{NIMat} \cdot (\mathbf{u}_h^i, p_h^i)^T = \text{RHS}$ 
19   Berechne nichtlineares Residuum  $r$ 
20   if  $r < \epsilon_{\text{nl}}$  then → Abbruchbedingung für die Picard-Iteration
21     break
22   end if
23    $i \leftarrow i + 1$ 
24 end for
25  $p_h \leftarrow p_h - \int_{\Omega} p_h$  → Normierung des diskreten Drucks
26 Berechne Fehlernormen  $|\mathbf{u} - \mathbf{u}_h|_{1,h}, \|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)^n}, \|p - p_h\|_{L^2(\Omega)}$ 

```

Setzen wir den *Bernoulli-Druck* $P := p + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$, dann lauten die Navier-Stokes-Gleichungen äquivalent

$$\begin{aligned}
 -\nu \Delta \mathbf{u} + (\nabla \times \mathbf{u}) \times \mathbf{u} + \nabla P &= \mathbf{f}, \\
 \operatorname{div} \mathbf{u} &= 0.
 \end{aligned}
 \tag{4.2}$$

Für das Crouzeix-Raviart-Element wird der nichtlineare Term dann diskretisiert als

$$\int_{\Omega} \left((\nabla_h \times \mathbf{u}_h^i) \times \mathbf{u}_h^{i-1} \right) \cdot \mathbf{v}_h
 \tag{4.3}$$

4. Implementierung und Studie des numerischen Verfahrens

und im Falle der Diskretisierung mit dem Rekonstruktionsoperator als

$$\int_{\Omega} \left((\nabla_h \times \mathbf{u}_h^i) \times \mathbf{u}_h^{i-1} \right) \cdot \boldsymbol{\pi}_h^{\text{RT}} \mathbf{v}_h. \quad (4.4)$$

Daher wird bei Verwendung des Rekonstruktionsoperators sowohl die rechte Seite als auch die Steifigkeitsmatrix modifiziert.

4.1. Beispiel 1: Ethier/Steinman-Benchmark

Für das erste Testbeispiel sei $\Omega := [0, 1]^3$. Das Geschwindigkeitsfeld \mathbf{u} sei wie folgt gewählt:

$$\mathbf{u}(x, y, z) := \begin{pmatrix} be^{a(x-z)+b(y-z)} - ae^{a(z-y)+b(x-y)} \\ be^{a(y-x)+b(z-x)} - ae^{a(x-z)+b(y-z)} \\ be^{a(z-y)+b(x-y)} - ae^{a(y-x)+b(z-x)} \end{pmatrix}, \quad (4.5)$$

mit $a := \frac{3}{2}$ und $b := \frac{3}{4}$. Der Druck p ist definiert durch

$$p(x, y, z) := (a^2 + b^2 + ab) \left(e^{a(x-y)+b(x-z)} + e^{a(y-z)+b(y-x)} + e^{a(z-x)+b(z-y)} \right). \quad (4.6)$$

Der Bernoulli-Druck berechnet sich dann nach $\tilde{P} := p + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$. Damit $P \in L_0^2(\Omega)$ gilt, definieren wir $P := \tilde{P} - \int_{\Omega} \tilde{P}$. Für alle Randflächen $F \in \mathcal{F}_h^* \setminus \mathcal{F}_h$ setzen wir die inhomogenen Dirichletrandwerte $\frac{1}{|F|} \int_F \mathbf{u} \, d\mathcal{H}^{n-1}$ fest. Die rechte Seite \mathbf{f} der Navier-Stokes-Gleichungen wird dann gemäß

$$\mathbf{f} := -\nu \Delta \mathbf{u} + (\nabla \times \mathbf{u}) \times \mathbf{u} + \nabla P \quad (4.7)$$

berechnet. Nach Konstruktion gilt auch $\text{div } \mathbf{u} = 0$. Da diese Lösung (\mathbf{u}, p) der Navier-Stokes-Gleichungen hinreichend glatt ist, entspricht sie auch der schwachen Lösung, welche eine zwar physikalisch nicht unbedingt realistische, aber dennoch echt dreidimensionale Strömung darstellt, die sich daher gut zum Test der numerischen Methoden eignet (vgl. [ES94]). Außerdem gilt offenbar $\mathbf{u} \notin X_h$ und $p \notin Q_h$, sodass auch die Approximationseigenschaften der Diskretisierung untersucht werden können.

4. Implementierung und Studie des numerischen Verfahrens

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$1.3372808 \cdot 10^0$	$3.9197849 \cdot 10^{-2}$	$1.6259071 \cdot 10^1$
56 155	$1.0456448 \cdot 10^0$	$2.4360342 \cdot 10^{-2}$	$1.6248562 \cdot 10^1$
120 493	$8.1603062 \cdot 10^{-1}$	$1.5083601 \cdot 10^{-2}$	$1.6243178 \cdot 10^1$
249 418	$6.4155673 \cdot 10^{-1}$	$9.4273099 \cdot 10^{-3}$	$1.6240242 \cdot 10^1$

Tabelle 1: Beispiel 1 (Ethier/Steinman-Benchmark), $\nu = 1$. Numerischer Fehler bei der Lösung des Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$1.5633007 \cdot 10^0$	$4.7869196 \cdot 10^{-2}$	$1.6257882 \cdot 10^1$
56 155	$1.2312312 \cdot 10^0$	$2.9986576 \cdot 10^{-2}$	$1.6248323 \cdot 10^1$
120 493	$9.6576696 \cdot 10^{-1}$	$1.8557660 \cdot 10^{-2}$	$1.6243007 \cdot 10^1$
249 418	$7.5592956 \cdot 10^{-1}$	$1.1476220 \cdot 10^{-2}$	$1.6240062 \cdot 10^1$

Tabelle 2: Beispiel 1 (Ethier/Steinman-Benchmark), $\nu = 1$. Numerischer Fehler bei der Lösung des Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$3.4204201 \cdot 10^0$	$1.3600756 \cdot 10^{-1}$	$1.9696821 \cdot 10^1$
56 155	$2.7448915 \cdot 10^0$	$8.5532553 \cdot 10^{-2}$	$2.0126041 \cdot 10^1$
120 493	$2.1571423 \cdot 10^0$	$5.3893717 \cdot 10^{-2}$	$2.0436042 \cdot 10^1$
249 418	$1.7169442 \cdot 10^0$	$3.4050907 \cdot 10^{-2}$	$2.0645710 \cdot 10^1$

Tabelle 3: Beispiel 1 (Ethier/Steinman-Benchmark), $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$1.5624752 \cdot 10^0$	$4.7834258 \cdot 10^{-2}$	$2.1003972 \cdot 10^1$
56 155	$1.2313692 \cdot 10^0$	$2.9969292 \cdot 10^{-2}$	$2.1051351 \cdot 10^1$
120 493	$9.6581003 \cdot 10^{-1}$	$1.8552319 \cdot 10^{-2}$	$2.1078946 \cdot 10^1$
249 418	$7.5584558 \cdot 10^{-1}$	$1.1468962 \cdot 10^{-2}$	$2.1094342 \cdot 10^1$

Tabelle 4: Beispiel 1 (Ethier/Steinman-Benchmark), $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens

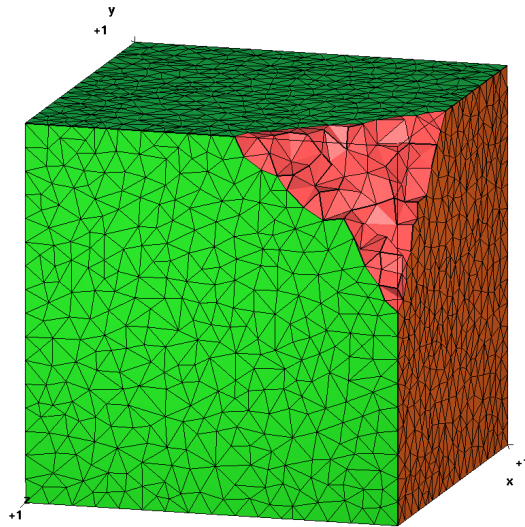


Abbildung 2: Triangulierung von $\Omega = [0, 1]^3$ mit $n_{\text{dof}} = 249\,418$

Die numerischen Ergebnisse sind für $\nu = 1$ und das Stokes- wie auch das Navier-Stokes-Problem jeweils unter Verwendung der klassischen Crouzeix-Raviart-Diskretisierung respektive der Diskretisierung mit dem Rekonstruktionsoperator für vier Gitterweiten in den Tabellen 1–4 dargestellt. Die Zahl der Freiheitsgrade n_{dof} ist hier ein Maß für die Feinheit des Gitters. Das feinste verwendete Gitter ist in Abbildung 2 dargestellt. Angegeben werden der Geschwindigkeitsfehler in der diskreten H^1 - wie auch der L^2 -Norm sowie der Druckfehler in der L^2 -Norm.

Zunächst fällt auf, dass bei der Lösung des Stokes-Problems die Ergebnisse von klassischer Crouzeix-Raviart-Diskretisierung und der Diskretisierung mit dem Rekonstruktionsoperator vergleichbar sind. Bei der Lösung des Navier-Stokes-Problems jedoch zeigen sich deutliche Vorteile für die Diskretisierung mit dem Rekonstruktionsoperator, so ist der Geschwindigkeitsfehler in beiden Normen im Vergleich zu der klassischen Diskretisierung mindestens halbiert. Der Druckfehler ist in allen Fällen absolut gesehen recht hoch, da der Druck ja durch auf jedem Tetraeder konstante Funktionen approximiert wird. Da die exakte Lösung p jedoch ein exponentielles Verhalten hat, ist dies nachvollziehbar.

4. Implementierung und Studie des numerischen Verfahrens

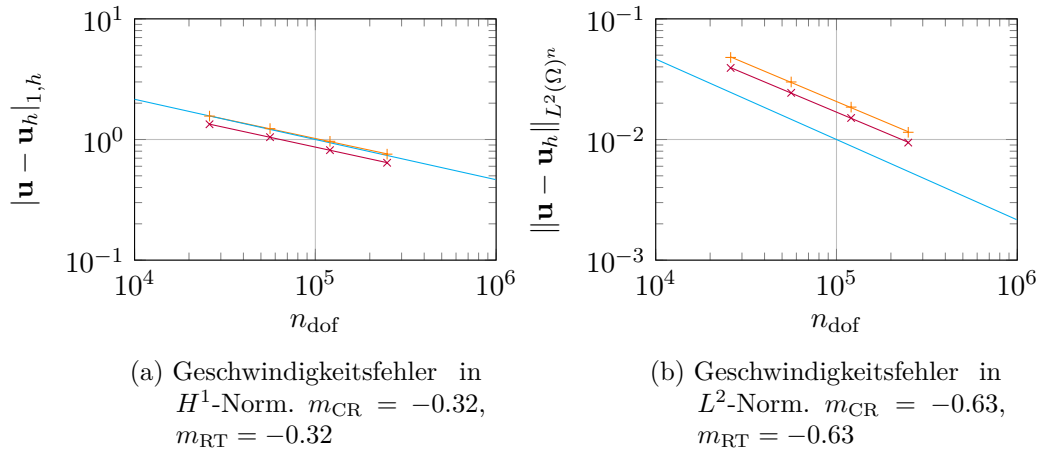


Abbildung 3: Geschwindigkeitsfehler für das Beispiel 1 bei der Lösung des Stokes-Problems mit $\nu = 1$ in Abhängigkeit von der Anzahl der Freiheitsgrade.
 × Crouzeix-Raviart, + Crouzeix-Raviart mit Rekonstruktionsoperator

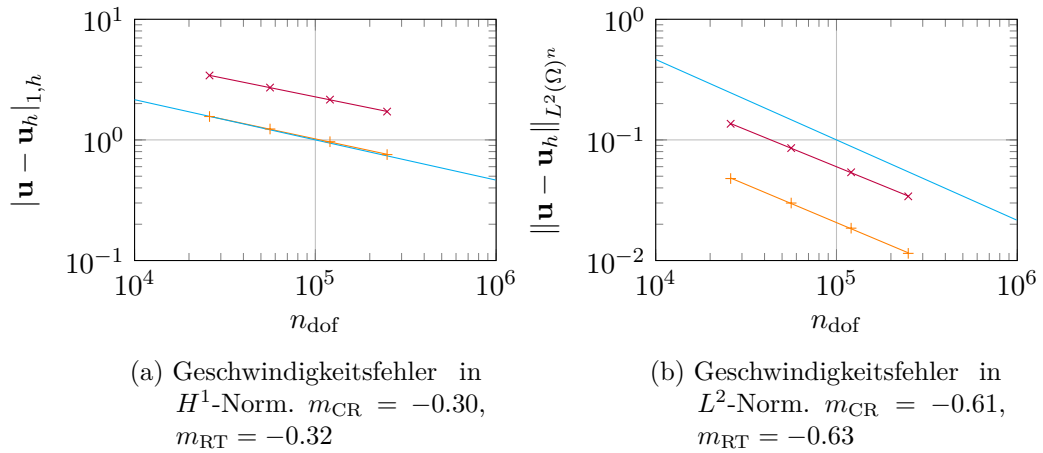


Abbildung 4: Geschwindigkeitsfehler für das Beispiel 1 bei der Lösung des Navier-Stokes-Problems mit $\nu = 1$ in Abhängigkeit von der Anzahl der Freiheitsgrade.
 × Crouzeix-Raviart, + Crouzeix-Raviart mit Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens

In Abbildung 3 werden die Geschwindigkeitsfehler $|\mathbf{u} - \mathbf{u}_h|_{1,h}$ in der diskreten H^1 -Norm und $\|p - p_h\|_{L^2(\Omega)}$ in der L^2 -Norm der klassischen Crouzeix-Raviart-Diskretisierung und der Diskretisierung mit dem Rekonstruktionsoperator für das Stokes-Problem miteinander verglichen, für das Navier-Stokes-Problem in Abbildung 4. Dabei zeigt sich annähernd die optimale Konvergenzordnung. Bei der Diskretisierung des Navier-Stokes-Problems liefert die Methode mit Rekonstruktionsoperator ein leicht besseres Ergebnis als das klassische Verfahren. Da im dreidimensionalen Fall gilt

$$\frac{1}{\sqrt[3]{n_{\text{dof}}}} \sim h, \quad (4.8)$$

liegt die optimale Konvergenzordnung bezüglich der Anzahl der Freiheitsgrade n_{dof} für die diskrete H^1 -Norm bei $\frac{1}{3}$ und für die L^2 -Norm bei $\frac{2}{3}$. Insofern zeigen beide Methoden das gewünschte Verhalten, mit Vorteilen für die Diskretisierung mit Rekonstruktionsoperator im nichtlinearen Fall.

4.2. Beispiel 2: Rotierendes Geschwindigkeitsfeld

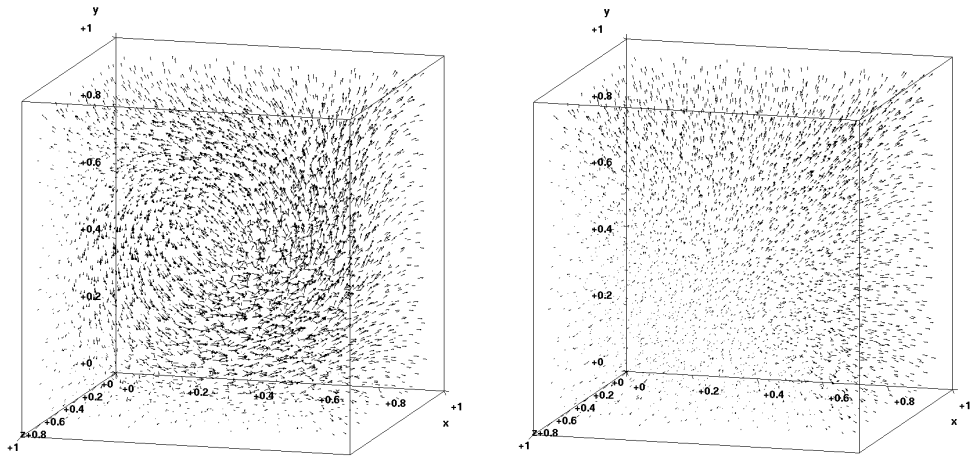
Es sei wieder $\Omega := [0, 1]^3$. Wir verwenden dieselben Triangulierungen von Ω wie im ersten Beispiel und definieren die exakte Lösung (\mathbf{u}, p) durch

$$\begin{aligned} \mathbf{u}(x, y, z) &:= \nabla \times \begin{pmatrix} \xi(x, y, z) \\ \xi(x, y, z) \\ \xi(x, y, z) \end{pmatrix}, \\ p(x, y, z) &:= x^3 + y^3 + z^3 - \frac{3}{4}, \end{aligned} \quad (4.9)$$

mit $\xi(x, y, z) := x^2(1-x)^2y^2(1-y)^2z^2(1-z)^2$. Als Rotation eines Vektorfeldes ist \mathbf{u} divergenzfrei. Weiter gilt $\mathbf{u}|_{\partial\Omega} = \mathbf{0}$. Für das Navier-Stokes-Problem verwenden wir den Bernoulli-Druck $\tilde{P} := p + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}$, den wir durch $P = \tilde{P} - \int_{\Omega} \tilde{P}$ normieren.

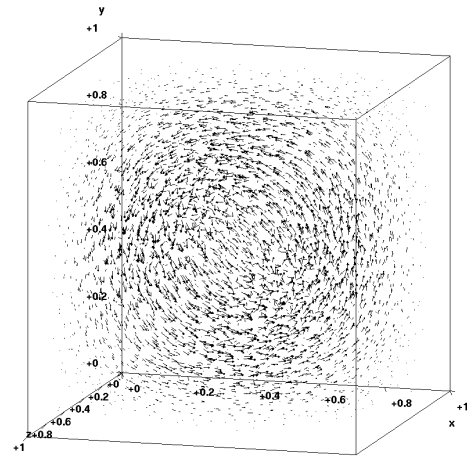
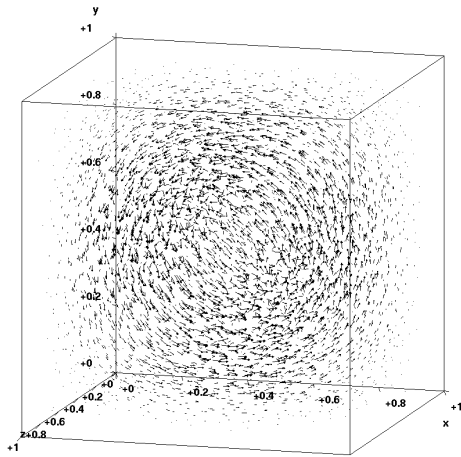
In Abbildung 5 sind die numerisch berechneten Geschwindigkeitsfehler für das feinste Gitter dargestellt. Für $\nu = 1$ liefern sowohl die klassische Crouzeix-Raviart-

4. Implementierung und Studie des numerischen Verfahrens



(a) Crouzeix-Raviart-Diskretisierung,
 $\nu = 1$

(b) Crouzeix-Raviart-Diskretisierung,
 $\nu = 10^{-3}$



(c) Diskretisierung mit Rekonstruktionsoperator, $\nu = 1$

(d) Diskretisierung mit Rekonstruktionsoperator, $\nu = 10^{-3}$

Abbildung 5: Numerisch berechnete Geschwindigkeitsfelder des Navier-Stokes-Problems aus Beispiel 2 mit $n_{\text{dof}} = 249\,418$

Diskretisierung als auch die modifizierte Diskretisierung das charakteristische rotierende Geschwindigkeitsfeld. Im Fall $\nu = 10^{-3}$ zeigt sich allerdings der klare Vorteil der Diskretisierung mit dem Rekonstruktionsoperator. Dem klassischen Verfahren gelingt es nicht, trotz des Auftretens der zusätzlichen rotationsfreien Kräfte noch eine zufriedenstellende Approximation der analytischen Lösung zu liefern. Die modifizierte Methode produziert dagegen ein optisch nicht von dem Fall $\nu = 1$ unterscheidbares

4. Implementierung und Studie des numerischen Verfahrens

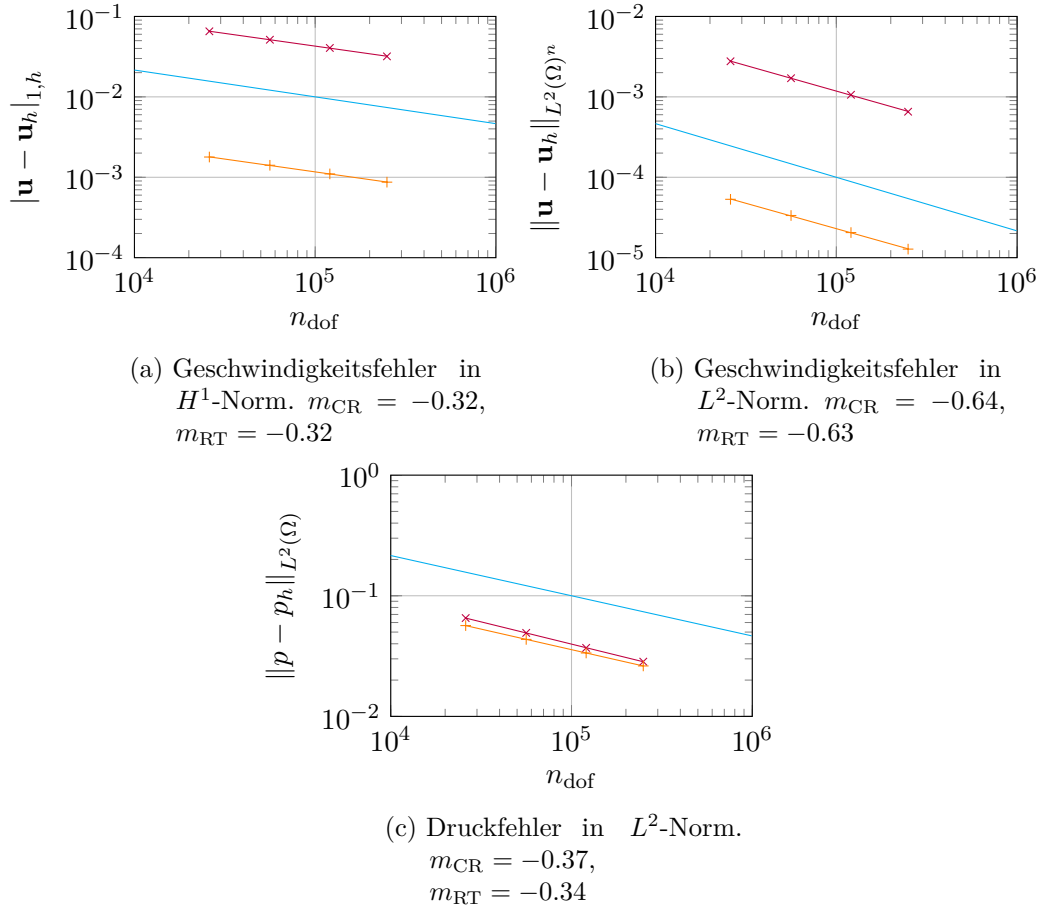
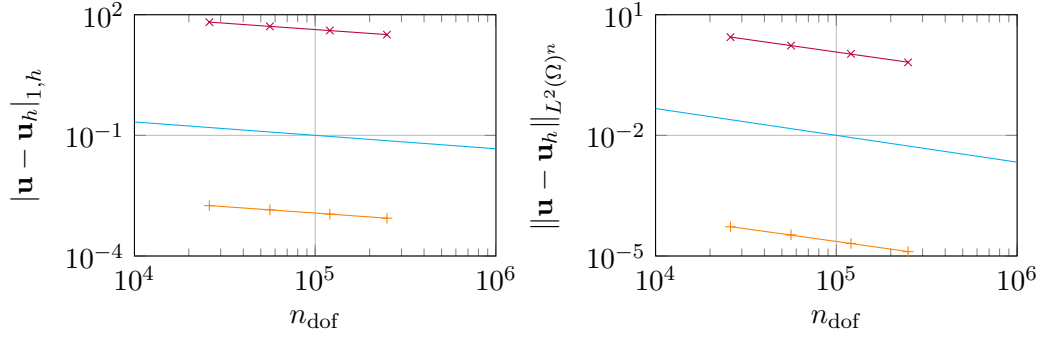


Abbildung 6: Geschwindigkeits- und Druckfehler für das Beispiel 2 bei der Lösung des Navier-Stokes-Problems mit $\nu = 1$ in Abhängigkeit von der Anzahl der Freiheitsgrade. \times Crouzeix-Raviart, $+$ Crouzeix-Raviart mit Rekonstruktionsoperator

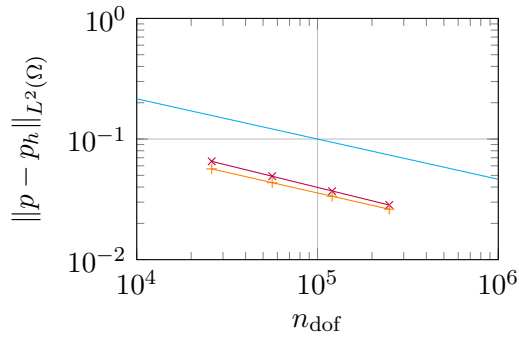
Ergebnis. Der Blick auf die in Tabelle 5–8 dargestellten numerischen Ergebnisse bestätigt diesen Eindruck. Der Geschwindigkeitsfehler ist für $\nu = 1$ bei der Diskretisierung mit dem Rekonstruktionsoperator bis zu zwei Ordnungen geringer als bei der klassischen Diskretisierung. Setzt man $\nu = 10^{-3}$, steigt der Geschwindigkeitsfehler bei der Crouzeix-Raviart-Diskretisierung dramatisch an und macht das Ergebnis unbrauchbar. Für das modifizierte Verfahren dagegen ist der Fehler, wie aufgrund der verbesserten Fehlerabschätzung (3.147) für das Stokes-Problem bereits erhofft, praktisch mit dem des Falls $\nu = 1$ identisch. Die Abbildungen 6 und 7 zeigen die annähernd optimalen Konvergenzordnungen beider Verfahren in Geschwindigkeits- und Druckfehler.

4. Implementierung und Studie des numerischen Verfahrens



(a) Geschwindigkeitsfehler in H^1 -Norm. $m_{\text{CR}} = -0.32$,
 $m_{\text{RT}} = -0.32$

(b) Geschwindigkeitsfehler in L^2 -Norm. $m_{\text{CR}} = -0.64$,
 $m_{\text{RT}} = -0.63$



(c) Druckfehler in L^2 -Norm.
 $m_{\text{CR}} = -0.37$,
 $m_{\text{RT}} = -0.34$

Abbildung 7: Geschwindigkeits- und Druckfehler für das Beispiel 2 bei der Lösung des Navier-Stokes-Problems mit $\nu = 10^{-3}$ in Abhängigkeit von der Anzahl der Freiheitsgrade. \times Crouzeix-Raviart, $+$ Crouzeix-Raviart mit Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$6.5313689 \cdot 10^{-2}$	$2.7695580 \cdot 10^{-3}$	$6.5343524 \cdot 10^{-2}$
56 155	$5.1329663 \cdot 10^{-2}$	$1.7049588 \cdot 10^{-3}$	$4.9222816 \cdot 10^{-2}$
120 493	$4.0551737 \cdot 10^{-2}$	$1.0619385 \cdot 10^{-3}$	$3.6991735 \cdot 10^{-2}$
249 418	$3.1876175 \cdot 10^{-2}$	$6.5401257 \cdot 10^{-4}$	$2.8440758 \cdot 10^{-2}$

Tabelle 5: Beispiel 2, $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$6.5305919 \cdot 10^1$	$2.7696122 \cdot 10^0$	$6.5344682 \cdot 10^{-2}$
56 155	$5.1322299 \cdot 10^1$	$1.7049040 \cdot 10^0$	$4.9223808 \cdot 10^{-2}$
120 493	$4.0547056 \cdot 10^1$	$1.0619362 \cdot 10^0$	$3.6991948 \cdot 10^{-2}$
249 418	$3.1870573 \cdot 10^1$	$6.5393078 \cdot 10^{-1}$	$2.8440239 \cdot 10^{-2}$

Tabelle 6: Beispiel 2, $\nu = 10^{-3}$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$1.7866391 \cdot 10^{-3}$	$5.3232360 \cdot 10^{-5}$	$5.6610101 \cdot 10^{-2}$
56 155	$1.4105785 \cdot 10^{-3}$	$3.3485314 \cdot 10^{-5}$	$4.3471206 \cdot 10^{-2}$
120 493	$1.1015270 \cdot 10^{-3}$	$2.0547718 \cdot 10^{-5}$	$3.3549421 \cdot 10^{-2}$
249 418	$8.7011644 \cdot 10^{-4}$	$1.2795294 \cdot 10^{-5}$	$2.6207321 \cdot 10^{-2}$

Tabelle 7: Beispiel 2, $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

n_{dof}	$ \mathbf{u} - \mathbf{u}_h _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
25 976	$1.7864985 \cdot 10^{-3}$	$5.3212168 \cdot 10^{-5}$	$5.6608884 \cdot 10^{-2}$
56 155	$1.4105453 \cdot 10^{-3}$	$3.3482173 \cdot 10^{-5}$	$4.3470295 \cdot 10^{-2}$
120 493	$1.1015272 \cdot 10^{-3}$	$2.0547513 \cdot 10^{-5}$	$3.3548754 \cdot 10^{-2}$
249 418	$8.7012367 \cdot 10^{-4}$	$1.2796006 \cdot 10^{-5}$	$2.6206838 \cdot 10^{-2}$

Tabelle 8: Beispiel 2, $\nu = 10^{-3}$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

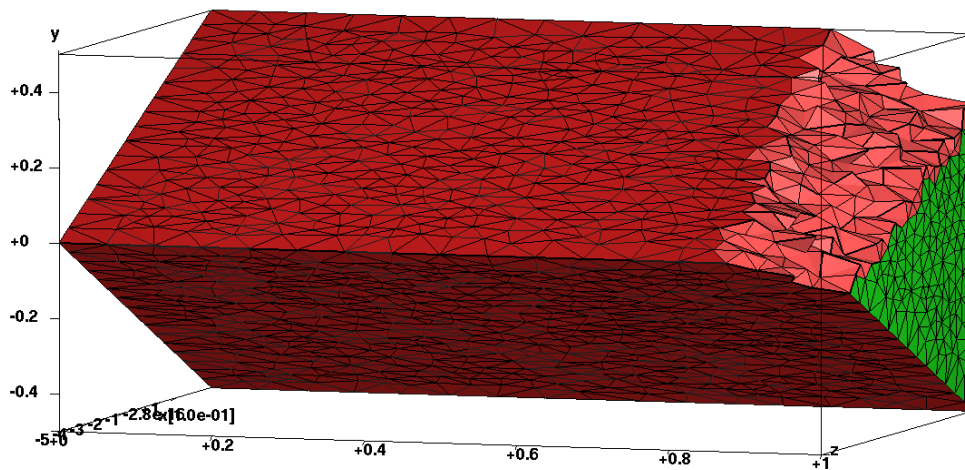


Abbildung 8: Triangulierung des dreieckigen Rohrs mit $n_{\text{dof}} = 229\,126$

4.3. Beispiel 3: Rohrströmung

Im folgenden Beispiel soll die Strömung einer Flüssigkeit durch ein Rohr untersucht werden, dessen Querschnittsfläche ein gleichseitiges Dreieck ist. Dabei handelt es sich um eine Abwandlung der bekannten *Hagen-Poiseuille-Strömung*. Das feinste verwendete Gitter ist in Abbildung 8 dargestellt. Die Lösung (\mathbf{u}, p) ist gegeben durch

$$\begin{aligned} \mathbf{u}(x, y, z) &:= (1 - 9x^2 - 6\sqrt{3}x^3 - 9y^2 + 18\sqrt{3}xy^2) \cdot (0, 0, 1)^T, \\ p(x, y, z) &:= 18 - 36z. \end{aligned} \quad (4.10)$$

Die mithilfe der Diskretisierung mit dem Rekonstruktionsoperator berechnete Lösung \mathbf{u}_h ist in Abbildung 10 dargestellt. Hier zeigt sich deutlich das quadratische, von z unabhängige Profil des Geschwindigkeitsfeldes und der dreieckige Querschnitt des Rohres. In Abbildung 11 wird der diskrete Druck p_h visualisiert. Zu erkennen ist der Verlauf des Drucks zwischen dem Einfluss bei $z = 0$ und dem Ausfluss bei $z = 1$ sowie das sich bei kleiner werdender Viskosität ändernde Druckprofil. Die in den Tabellen 9–12 angegebenen numerischen Ergebnisse zeigen wiederum die Überlegenheit der Diskretisierung mit dem Rekonstruktionsoperator im Vergleich mit der

4. Implementierung und Studie des numerischen Verfahrens

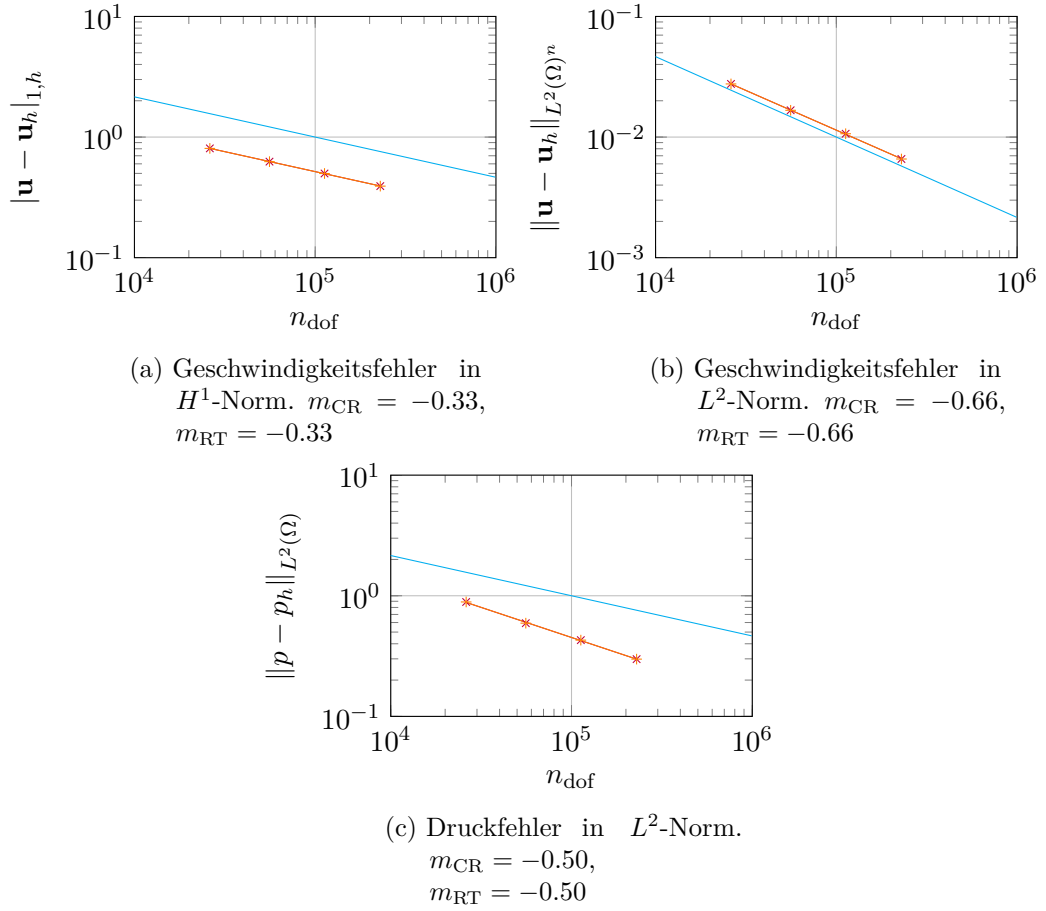
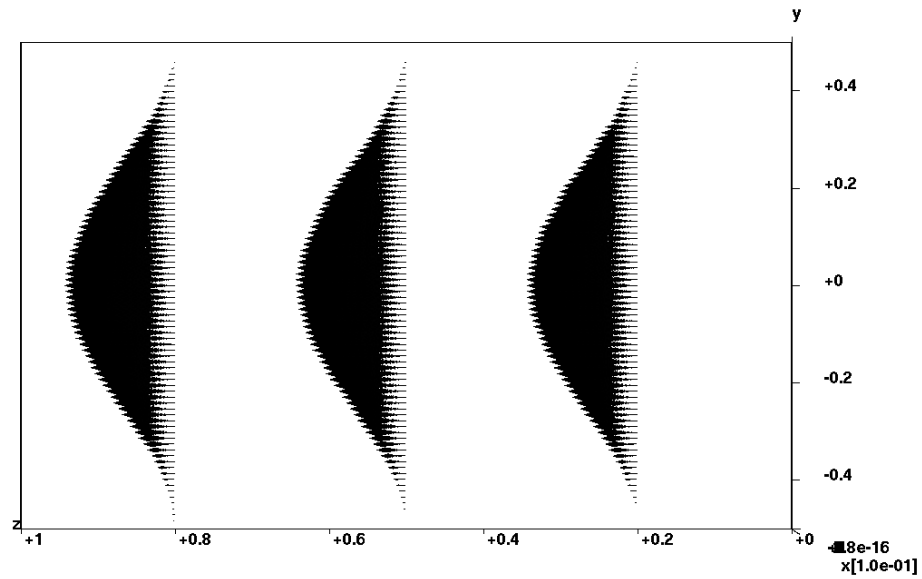


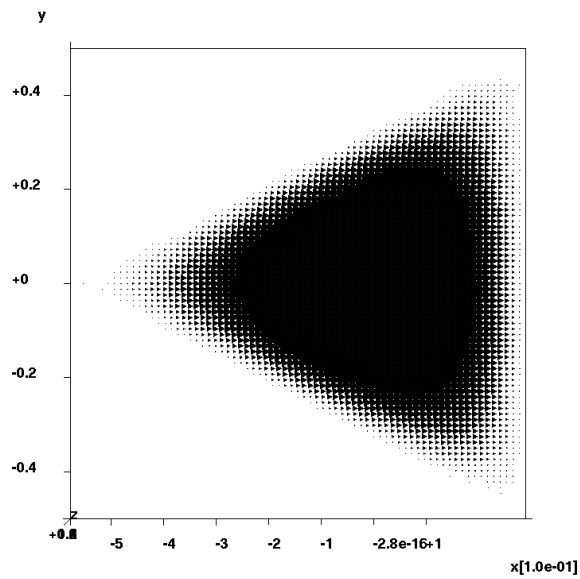
Abbildung 9: Geschwindigkeits- und Druckfehler für das Beispiel 3 bei der Lösung des Navier-Stokes-Problems mit $\nu = 1$ in Abhängigkeit von der Anzahl der Freiheitsgrade. \times Crouzeix-Raviart, $+$ Crouzeix-Raviart mit Rekonstruktionsoperator

klassischen Methode, bei der der Fehler in der diskreten H^1 -Norm bei kleinerem ν stark ansteigt. Für $\nu = 1$ zeigt die Abbildung 9 für beide Verfahren die optimale Konvergenzordnung des Geschwindigkeitsfehlers, Abbildung 12 illustriert das deutlich bessere Abschneiden der modifizierten Methode bei $\nu = 10^{-2}$. Zu bemerken ist weiter, dass die Konvergenzordnung des Druckfehlers für dieses Beispiel in beiden Fällen sogar höher ausfällt als theoretisch zu erwarten wäre.

4. Implementierung und Studie des numerischen Verfahrens



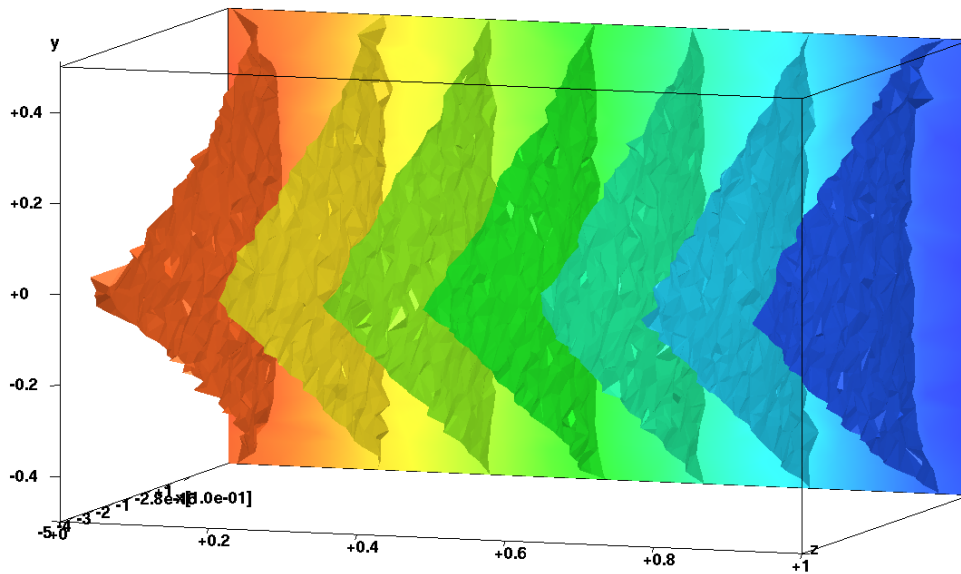
(a) Durchsicht auf der y - z -Ebene



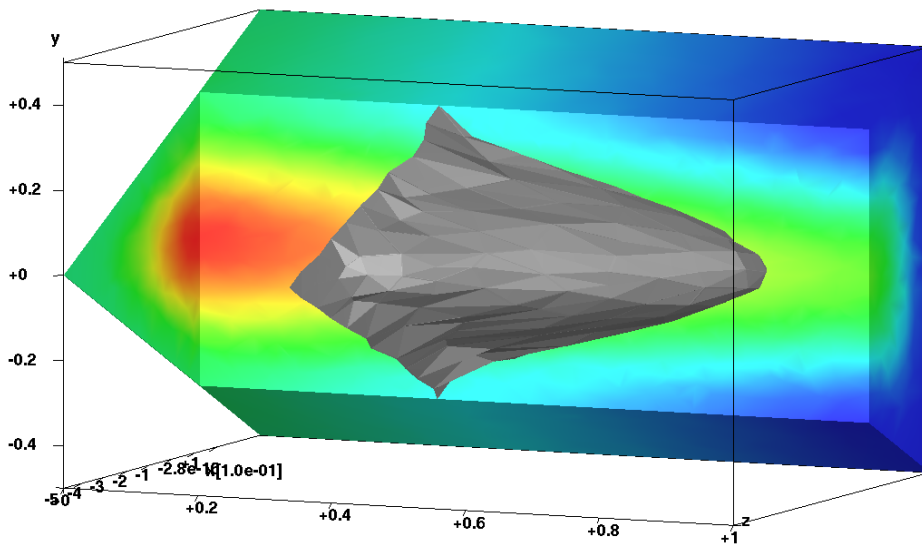
(b) Durchsicht auf der x - y -Ebene

Abbildung 10: Numerisch berechnetes Geschwindigkeitsfeld des Navier-Stokes-Problems aus Beispiel 3 mit $n_{\text{dof}} = 229\,126$. Diskretisierung mit Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens



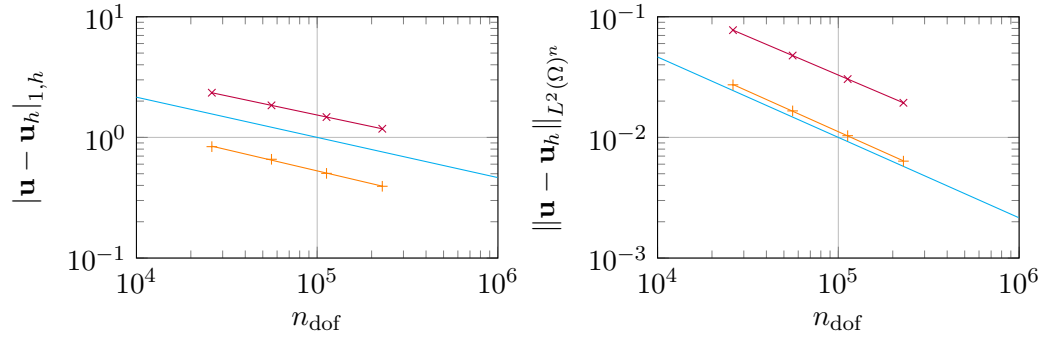
(a) $\nu = 1$. Isoflächen bei $p_h = -15, -10, \dots, 10, 15$



(b) $\nu = 10^{-2}$. Schnittebene bei $z = 0$

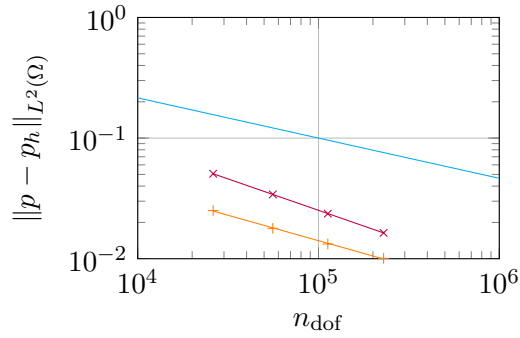
Abbildung 11: Numerisch berechnete Druckfelder (Bernoulli-Druck) des Navier-Stokes-Problems aus Beispiel 3 mit $n_{\text{dof}} = 229\,126$. Diskretisierung mit Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens



(a) Geschwindigkeitsfehler in H^1 -Norm. $m_{\text{CR}} = -0.32$,
 $m_{\text{RT}} = -0.35$

(b) Geschwindigkeitsfehler in L^2 -Norm. $m_{\text{CR}} = -0.64$,
 $m_{\text{RT}} = -0.67$



(c) Druckfehler in L^2 -Norm.
 $m_{\text{CR}} = -0.52$,
 $m_{\text{RT}} = -0.42$

Abbildung 12: Geschwindigkeits- und Druckfehler für das Beispiel 3 bei der Lösung des Navier-Stokes-Problems mit $\nu = 10^{-2}$ in Abhängigkeit von der Anzahl der Freiheitsgrade.

× Crouzeix-Raviart, + Crouzeix-Raviart mit Rekonstruktionsoperator

4. Implementierung und Studie des numerischen Verfahrens

n_{dof}	$\ \mathbf{u} - \mathbf{u}_h\ _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
26 137	$8.0595407 \cdot 10^{-1}$	$2.7585566 \cdot 10^{-2}$	$8.8801597 \cdot 10^{-1}$
55 872	$6.2591662 \cdot 10^{-1}$	$1.6679021 \cdot 10^{-2}$	$5.9248083 \cdot 10^{-1}$
112 602	$4.9915348 \cdot 10^{-1}$	$1.0660510 \cdot 10^{-2}$	$4.2867492 \cdot 10^{-1}$
229 126	$3.9232660 \cdot 10^{-1}$	$6.5822745 \cdot 10^{-3}$	$2.9835113 \cdot 10^{-1}$

Tabelle 9: Beispiel 3, $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$\ \mathbf{u} - \mathbf{u}_h\ _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
26 137	$2.3470447 \cdot 10^0$	$7.7464235 \cdot 10^{-2}$	$5.0659625 \cdot 10^{-2}$
55 872	$1.8437065 \cdot 10^0$	$4.7716597 \cdot 10^{-2}$	$3.4135785 \cdot 10^{-2}$
112 602	$1.4744359 \cdot 10^0$	$3.0444367 \cdot 10^{-2}$	$2.3646370 \cdot 10^{-2}$
229 126	$1.1797430 \cdot 10^0$	$1.9365210 \cdot 10^{-2}$	$1.6385955 \cdot 10^{-2}$

Tabelle 10: Beispiel 3, $\nu = 10^{-2}$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$\ \mathbf{u} - \mathbf{u}_h\ _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
26 137	$8.0567995 \cdot 10^{-1}$	$2.7577202 \cdot 10^{-2}$	$8.8846777 \cdot 10^{-1}$
55 872	$6.2570071 \cdot 10^{-1}$	$1.6675345 \cdot 10^{-2}$	$5.9266696 \cdot 10^{-1}$
112 602	$4.9903309 \cdot 10^{-1}$	$1.0659869 \cdot 10^{-2}$	$4.2879877 \cdot 10^{-1}$
229 126	$3.9215468 \cdot 10^{-1}$	$6.5798799 \cdot 10^{-3}$	$2.9841285 \cdot 10^{-1}$

Tabelle 11: Beispiel 3, $\nu = 1$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

n_{dof}	$\ \mathbf{u} - \mathbf{u}_h\ _{1,h}$	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2(\Omega)^n}$	$\ p - p_h\ _{L^2(\Omega)}$
26 137	$8.3834886 \cdot 10^{-1}$	$2.7296393 \cdot 10^{-2}$	$2.5077503 \cdot 10^{-2}$
55 872	$6.5883181 \cdot 10^{-1}$	$1.6586110 \cdot 10^{-2}$	$1.7870420 \cdot 10^{-2}$
112 602	$5.0338410 \cdot 10^{-1}$	$1.0338704 \cdot 10^{-2}$	$1.3256489 \cdot 10^{-2}$
229 126	$3.9360166 \cdot 10^{-1}$	$6.3585453 \cdot 10^{-3}$	$1.0070856 \cdot 10^{-2}$

Tabelle 12: Beispiel 3, $\nu = 10^{-2}$. Numerischer Fehler bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

4.4. Beispiel 4: Lid-driven cavity

Das letzte Beispiel ist eine Strömung, die vollständig über die Randbedingungen definiert ist, es gilt also $\mathbf{f} \equiv \mathbf{0}$. Wir betrachten das Gebiet $\Omega := [0, 1]^3$ und nehmen homogene Dirichletrandwerte an. Für $z = 1$ setzen wir $\mathbf{u} \equiv (1, 0, 0)^T$. Die Flüssigkeit wird am oberen Rand des Würfels also gleichmäßig in Bewegung versetzt. Der nichtlineare Term in Rotationsform $(\nabla \times \mathbf{u}) \times \mathbf{u}$ hat hier sowohl einen divergenzfreien als auch einen rotationsfreien Anteil aus der Helmholtz-Zerlegung, wovon der letztere für Probleme bei der Verwendung des klassischen Crouzeix-Raviart-Elements zur Diskretisierung sorgen könnte. Man beachte auch, dass aufgrund der Randbedingungen $\mathbf{u} \notin H_0^1(\Omega)^n$ gilt. Wir untersuchen dieses Beispiel dennoch, da es ein in der übrigen Literatur weit verbreitetes Beispiel zum Test von numerischen Verfahren zur Lösung der Navier-Stokes-Gleichungen ist. Da keine analytische Lösung verfügbar ist, überprüfen wir die Methoden neben der optischen Kontrolle auch durch den Vergleich der L^2 -Norm des Geschwindigkeitsfeldes. Dazu berechnen wir eine diskrete Lösung des Problems und deren L^2 -Norm zusätzlich mithilfe des Softwarepakets FENICS³. Der Python-Programmcode befindet sich in Anhang A.

In Abbildung 13 zeigt sich das für dieses Beispiel typische Geschwindigkeitsfeld, das hier der Übersicht halber jeweils nur auf den drei Ebenen $x = 1$, $y = 1$ und $z = 1$ zu sehen ist. Dem optischen Eindruck nach scheinen beide Methoden ein gutes Ergebnis zu liefern. So verschiebt sich der Mittelpunkt des primären Wirbels bei kleiner werdendem ν und bei $\nu = 10^{-2}$ sind Ansätze der typischen kleinen Wirbel zu sehen. Das auf dem feinsten Gitter berechnete Druckfeld zeigt Abbildung 14. Zumindest für $\nu = 1$ und $\nu = 10^{-1}$ scheint die in den Tabellen 13 und 14 angegebene L^2 -Norm des Geschwindigkeitsfeldes gegen einen Wert von etwa 0.231 zu konvergieren. Diese Größenordnung wird durch das FENICS-Programm auch bestätigt (Tabelle 15). Somit ist anzunehmen, dass die Implementierung beider Verfahren keinen prinzipiellen Fehler aufweist. Im Falle von $\nu = 10^{-2}$ ist zu berücksichtigen, dass hier das nichtlineare Gleichungssystem in beiden Fällen nur bis auf ein nichtlineares Residuum in der

³URL: <http://fenicsproject.org/> (Zuletzt abgerufen am 25.09.2013)

4. Implementierung und Studie des numerischen Verfahrens

n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$	n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$
25 976	$2.2984766 \cdot 10^{-1}$	25 976	$2.2780290 \cdot 10^{-1}$
56 155	$2.3014658 \cdot 10^{-1}$	56 155	$2.2876801 \cdot 10^{-1}$
120 493	$2.3068052 \cdot 10^{-1}$	120 493	$2.2986813 \cdot 10^{-1}$
249 418	$2.3133194 \cdot 10^{-1}$	249 418	$2.3079324 \cdot 10^{-1}$
(a) $\nu = 1$		(b) $\nu = 10^{-1}$	

Tabelle 13: Beispiel 4 (Lid-driven cavity). L^2 -Norm des Geschwindigkeitsfeldes bei der Lösung des Navier-Stokes-Problems mit der klassischen Crouzeix-Raviart-Diskretisierung

n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$	n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$
25 976	$2.2988024 \cdot 10^{-1}$	25 976	$2.2965361 \cdot 10^{-1}$
56 155	$2.3015817 \cdot 10^{-1}$	56 155	$2.2995361 \cdot 10^{-1}$
120 493	$2.3068123 \cdot 10^{-1}$	120 493	$2.3053511 \cdot 10^{-1}$
249 418	$2.3133650 \cdot 10^{-1}$	249 418	$2.3123708 \cdot 10^{-1}$
(a) $\nu = 1$		(b) $\nu = 10^{-1}$	

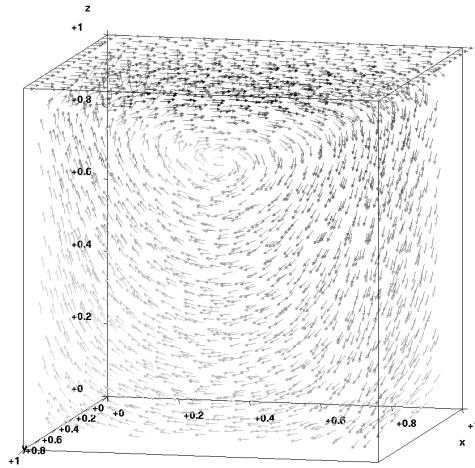
Tabelle 14: Beispiel 4 (Lid-driven cavity). L^2 -Norm des Geschwindigkeitsfeldes bei der Lösung des Navier-Stokes-Problems mit der Crouzeix-Raviart-Diskretisierung und dem Rekonstruktionsoperator

n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$	n_{dof}	$\ \mathbf{u}_h\ _{L^2(\Omega)^n}$
22 656	$2.8761054 \cdot 10^{-1}$	22 656	$2.8751972 \cdot 10^{-1}$
43 800	$2.6938405 \cdot 10^{-1}$	43 800	$2.6915761 \cdot 10^{-1}$
118 776	$2.4943598 \cdot 10^{-1}$	118 776	$2.4913852 \cdot 10^{-1}$
250 776	$2.3977305 \cdot 10^{-1}$	250 776	$2.3948387 \cdot 10^{-1}$
(a) $\nu = 1$		(b) $\nu = 10^{-1}$	

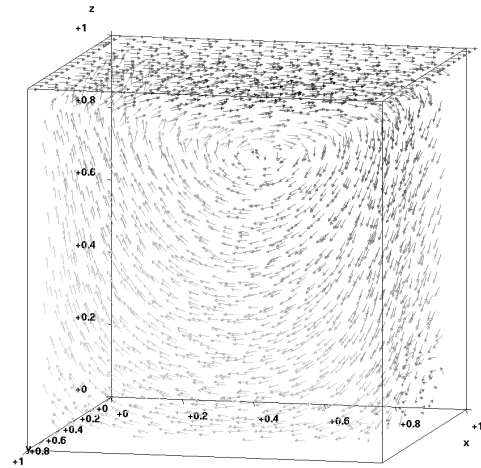
Tabelle 15: Beispiel 4 (Lid-driven cavity). L^2 -Norm des Geschwindigkeitsfeldes bei der Lösung des Navier-Stokes-Problems mit FENICS

Größenordnung von 10^{-2} gelöst wurde, was die diskrete Lösung unbrauchbar werden lässt. In den anderen Fällen wurde das Residuum bis auf maximal 10^{-12} reduziert. Für kleine ν ist die Picard-Iteration offenbar ungeeignet und man müsste entweder ein anderes Verfahren zur Lösung des nichtlinearen Gleichungssystems anwenden oder vielleicht sogar die dominante Konvektion stabilisieren. Abbildung 15 zeigt zumindest für $\nu = 10^{-1}$ wieder die typische Überlegenheit des modifizierten Verfahrens gegenüber der klassischen Crouzeix-Raviart-Diskretisierung.

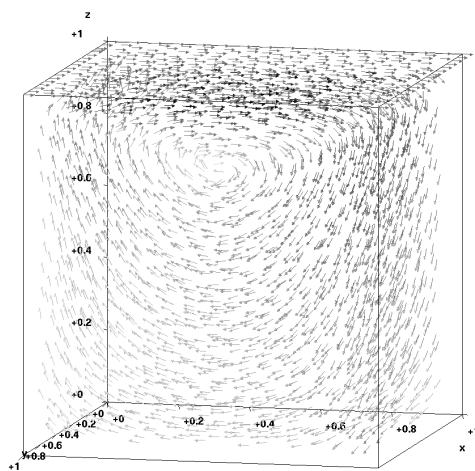
4. Implementierung und Studie des numerischen Verfahrens



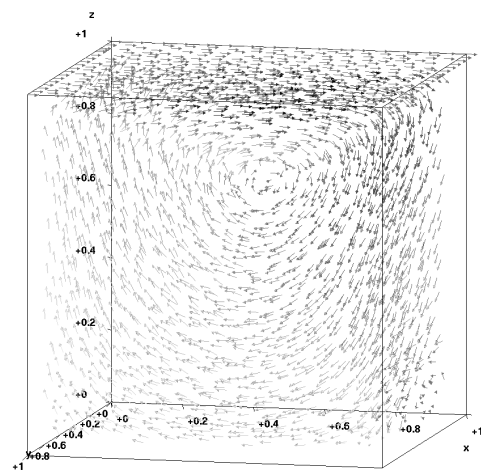
(a) Crouzeix-Raviart-Diskretisierung,
 $\nu = 1$



(b) Crouzeix-Raviart-Diskretisierung,
 $\nu = 10^{-2}$



(c) Diskretisierung mit Rekonstruktionsoperator,
 $\nu = 1$



(d) Diskretisierung mit Rekonstruktionsoperator,
 $\nu = 10^{-2}$

Abbildung 13: Numerisch berechnete Geschwindigkeitsfelder des Navier-Stokes-Problems aus Beispiel 4 (Lid-driven cavity) mit $n_{\text{dof}} = 249\,418$

4. Implementierung und Studie des numerischen Verfahrens

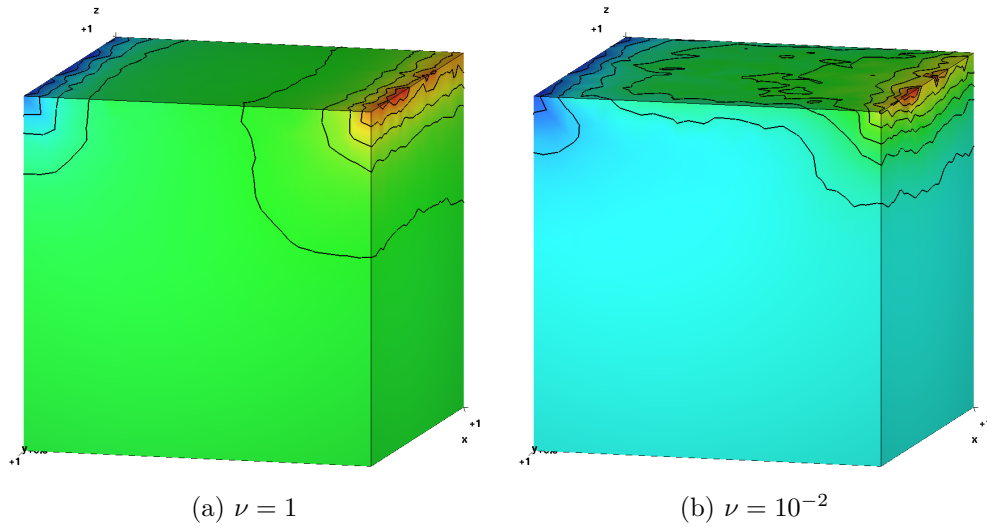


Abbildung 14: Numerisch berechnete Druckfelder des Navier-Stokes-Problems aus Beispiel 4 mit $n_{\text{dof}} = 249\,418$. Klassische Crouzeix-Raviart-Diskretisierung

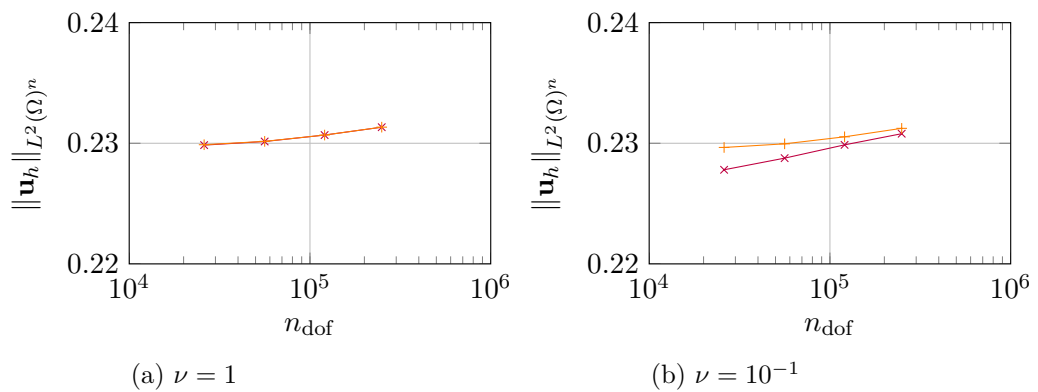


Abbildung 15: Beispiel 4 (Lid-driven cavity). L^2 -Norm der diskreten Lösung \mathbf{u}_h in Abhängigkeit von der Anzahl der Freiheitsgrade.

— \times — Crouzeix-Raviart, — $+$ — Crouzeix-Raviart mit Rekonstruktionsoperator

5. Fazit und Ausblick

In der vorliegenden Arbeit wurde eine gemischte Diskretisierung der stationären inkompressiblen Navier-Stokes-Gleichungen vorgestellt. Außerdem wurde eine divergenzfreie Rekonstruktion präsentiert, die den Geschwindigkeitsfehler von der Kopplung an den Druck befreit, welche bei der Approximation der Lösung mit dem klassischen Crouzeix-Raviart-Element auftritt. Die Resultate der numerischen Studie zeigen deutlich, dass das modifizierte Verfahren mit dem Rekonstruktionsoperator für eine kleine kinematische Viskosität ν die Strömung wesentlich besser approximiert als die klassische Methode. Übertragen auf die Simulation eines physikalischen Modells bedeutet dies, dass die neue Methode die Simulation mancher Anwendungen erst sinnvoll möglich macht.

Die Ergebnisse legen außerdem für die divergenzfreie Rekonstruktion eine quadratische Konvergenz des Geschwindigkeitsfehlers in der L^2 -Norm nahe. Um eine entsprechende Fehlerabschätzung zu beweisen, müssten allerdings umfassende theoretische Betrachtungen unternommen werden. Gegebenenfalls wäre auch der Gebrauch eines anderen finiten Elements ratsam, für das sich die Fehlerabschätzung leichter beweisen lässt (vgl. [Lin13, S. 16]). Zudem wäre zunächst eine Fehlerabschätzung des Konsistenzfehlers vonnöten, um zumindest eine lineare Konvergenz wie für die Rekonstruktion im Falle des Stokes-Problems zu zeigen.

Eine weitere mögliche Verbesserung stellt die Verwendung des Newton-Verfahrens anstelle der Picard-Iteration zur Lösung des nichtlinearen Gleichungssystems dar, das allerdings einen erhöhten implementativen Aufwand mit sich bringt. Insgesamt kann die hier vorliegende Implementierung sicherlich noch an einigen Stellen verbessert werden, um eine Steigerung der Effizienz des Programms zu erreichen. Zur Lösung

5. Fazit und Ausblick

komplexerer Probleme, die die Kopplung der Navier-Stokes-Gleichungen mit weiteren partiellen Differentialgleichungen erfordern, ist das vorliegende Programm aber eine gute Ausgangsbasis.

Ebenfalls von Interesse ist gewiss die Erweiterung des numerischen Verfahrens auf die instationären Navier-Stokes-Gleichungen, um eine zeitlich variierende Strömung simulieren zu können. Dazu wäre die Idee der divergenzfreien Rekonstruktion auf bereits existierende Verfahren zur Lösung des instationären Problems zu übertragen.

Alles in allem ist festzuhalten, dass eine vielseitig einsetzbare Methode zur divergenzfreien Lösung der Navier-Stokes-Gleichungen fraglos noch einige Entwicklung benötigen wird. Das hier vorgestellte Verfahren stellt aber einen sehr vielversprechenden Ansatz dar.

Literatur

- [Alt12] Alt, Hans Wilhelm: *Lineare Funktionalanalysis*. 6., überarb. Aufl. Berlin, Heidelberg: Springer, 2012.
- [Bra13] Braess, Dietrich: *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. 5. Aufl. Berlin, Heidelberg: Springer, 2013.
- [Bre13] Brennecke, Christian: „Eine divergenzfreie Rekonstruktion für eine nicht-konforme Diskretisierung der inkompressiblen Stokes-Gleichungen“. Bachelorarbeit. Freie Universität Berlin, 2013.
- [BF91] Brezzi, Franco und Michel Fortin: *Mixed and Hybrid Finite Element Methods*. New York, NY: Springer, 1991.
- [CR73] Crouzeix, Michel und Pierre-Arnaud Raviart: „Conforming and nonconforming finite element methods for solving the stationary Stokes equations I“. In: *Revue française d'automatique, informatique, recherche opérationnelle. Mathématique* 7.3 (1973), S. 33–75.
- [DXS00] Deng, Qing-ping, Xue-jun Xu und Shu-min Shen: „Maximum Norm Error Estimates Of Crouzeix-Raviart Nonconforming Finite Element Approximation Of Navier-Stokes Problem“. In: *Journal of Computational Mathematics* 18.2 (2000), S. 141–156.
- [ES94] Ethier, C. Ross und D. A. Steinman: „Exact Fully 3D Navier-Stokes Solutions For Benchmarking“. In: *International Journal for Numerical Methods in Fluids* 19.5 (1994), S. 369–375.
- [Eva10] Evans, Lawrence C.: *Partial Differential Equations*. Second Edition. Providence, RI: American Mathematical Society, 2010.

Literatur

- [EG91] Evans, Lawrence C. und Ronald F. Gariepy: *Measure Theory and Fine Properties of Functions*. Boca Raton, FL: CRC Press, 1991.
- [GR79] Girault, Vivette und Pierre-Arnaud Raviart: *Finite Element Approximations of the Navier-Stokes Equations*. Lecture Notes in Mathematics 749. Berlin, Heidelberg: Springer, 1979.
- [Gun02] Gunzburger, Max D.: „The inf-sup Condition in Mixed Finite Element Methods with Application to the Stokes System“. In: *Collected Lectures on the Preservation of Stability Under Discretization*. Hrsg. von Donald Estep und Simon Tavener. Philadelphia, PA: Society for Industrial und Applied Mathematics, 2002, S. 93–121.
- [Han84] Han, Hou-de: „A Finite Element Approximation Of Navier-Stokes Equations Using Nonconforming Elements“. In: *Journal of Computational Mathematics* 2.1 (1984), S. 77–88.
- [Joh09] John, Volker: *Numerik Partieller Differentialgleichungen: Eine elementare Einführung*. Vorlesungsskriptum. Freie Universität Berlin, 2009.
URL: http://www.wias-berlin.de/people/john/LEHRE/NUM_PDE/SS09/num_pde_intro.pdf.
- [Lay08] Layton, William: *Introduction to the Numerical Analysis of Incompressible Viscous Flows*. Philadelphia, PA: Society for Industrial und Applied Mathematics, 2008.
- [Lin13] Linke, Alexander: „On the Role of the Helmholtz-Decomposition in Mixed Methods for Incompressible Flows and a New Variational Crime“. In: *Computer Methods in Applied Mechanics and Engineering* (angenommen, 2013).
- [Soh01] Sohr, Hermann: *The Navier-Stokes Equations: An Elementary Functional Analytic Approach*. Basel, Boston, Berlin: Birkhäuser, 2001.
- [Tem79] Temam, Roger: *Navier-Stokes Equations*. Revised Edition. Amsterdam, New York, Oxford: North-Holland Publishing Company, 1979.

Literatur

- [Ver98] Verfürth, Rüdiger: *Numerische Strömungsmechanik*. Vorlesungsskriptum. Ruhr-Universität Bochum, 1998.
URL: <http://www.ruhr-uni-bochum.de/num1/files/lectures/SM.pdf>.
- [Ver11] Verfürth, Rüdiger: *Numerik II: Finite Elemente*. Vorlesungsskriptum. Ruhr-Universität Bochum, 2011.
URL: <http://www.ruhr-uni-bochum.de/num1/files/lectures/NumDg12.pdf>.
- [Wer11] Werner, Dirk: *Funktionalanalysis*. 7., korr. und erw. Aufl. Berlin, Heidelberg: Springer, 2011.

Abbildungsverzeichnis

1. Normalkomponente des Raviart-Thomas-Elements	46
2. Triangulierung von $\Omega = [0, 1]^3$ mit $n_{\text{dof}} = 249\,418$	54
3. Konvergenzordnungen, Ethier/Steinman, Stokes, $\nu = 1$	55
4. Konvergenzordnungen, Ethier/Steinman, Navier-Stokes, $\nu = 1$	55
5. Geschwindigkeitsfelder, Beispiel 2, Navier-Stokes	57
6. Konvergenzordnungen, Beispiel 2, Navier-Stokes, $\nu = 1$	58
7. Konvergenzordnungen, Beispiel 2, Navier-Stokes, $\nu = 10^{-3}$	59
8. Triangulierung des dreieckigen Rohrs mit $n_{\text{dof}} = 229\,126$	61
9. Konvergenzordnungen, Beispiel 3, Navier-Stokes, $\nu = 1$	62
10. Geschwindigkeitsfeld, Beispiel 3, Navier-Stokes, $\nu = 10^{-2}$	63
11. Druckfelder, Beispiel 3, Navier-Stokes, $\nu = 1, 10^{-2}$	64
12. Konvergenzordnungen, Beispiel 3, Navier-Stokes, $\nu = 10^{-2}$	65
13. Geschwindigkeitsfelder, Beispiel 4 (Lid-driven cavity), Navier-Stokes	69
14. Druckfelder, Beispiel 4 (Lid-driven cavity), Navier-Stokes, $\nu = 1, 10^{-2}$. . .	70
15. L^2 -Norm der diskreten Lösung \mathbf{u}_h , Beispiel 4 (Lid-driven cavity), Navier- Stokes, $\nu = 1, 10^{-1}$	70

Tabellenverzeichnis

1. Ethier/Steinman, $\nu = 1$, Stokes, Crouzeix-Raviart	53
2. Ethier/Steinman, $\nu = 1$, Stokes, Rekonstruktionsoperator	53
3. Ethier/Steinman, $\nu = 1$, Navier-Stokes, Crouzeix-Raviart	53
4. Ethier/Steinman, $\nu = 1$, Navier-Stokes, Rekonstruktionsoperator	53
5. Beispiel 2, $\nu = 1$, Navier-Stokes, Crouzeix-Raviart	60
6. Beispiel 2, $\nu = 10^{-3}$, Navier-Stokes, Crouzeix-Raviart	60
7. Beispiel 2, $\nu = 1$, Navier-Stokes, Rekonstruktionsoperator	60
8. Beispiel 2, $\nu = 10^{-3}$, Navier-Stokes, Rekonstruktionsoperator	60
9. Beispiel 3, $\nu = 1$, Navier-Stokes, Crouzeix-Raviart	66
10. Beispiel 3, $\nu = 10^{-2}$, Navier-Stokes, Crouzeix-Raviart	66
11. Beispiel 3, $\nu = 1$, Navier-Stokes, Rekonstruktionsoperator	66
12. Beispiel 3, $\nu = 10^{-2}$, Navier-Stokes, Rekonstruktionsoperator	66
13. Beispiel 4 (Lid-driven cavity), Navier-Stokes, Crouzeix-Raviart	68
14. Beispiel 4 (Lid-driven cavity), Navier-Stokes, Rekonstruktionsoperator . . .	68
15. Beispiel 4 (Lid-driven cavity), Navier-Stokes, FENICS	68

A. Quelltexte

Listing A.1: C++-Programmcode (Auszug) des Hauptprogramms

```
1 #include <pdelib2/pdelib2.h>
2 // #include ...
3
4 // define exact solutions for testing purposes
5 double u(double x, double y, double z, int problemId);
6 double v(double x, double y, double z, int problemId);
7 double w(double x, double y, double z, int problemId);
8 double ux(double x, double y, double z, int problemId);
9 double uy(double x, double y, double z, int problemId);
10 double uz(double x, double y, double z, int problemId);
11 double vx(double x, double y, double z, int problemId);
12 double vy(double x, double y, double z, int problemId);
13 double vz(double x, double y, double z, int problemId);
14 double wx(double x, double y, double z, int problemId);
15 double wy(double x, double y, double z, int problemId);
16 double wz(double x, double y, double z, int problemId);
17 double p(double x, double y, double z, double nu, bool nonlinearMode, int problemId);
18 double f1(double x, double y, double z, double nu, bool nonlinearMode, int problemId);
19 double f2(double x, double y, double z, double nu, bool nonlinearMode, int problemId);
20 double f3(double x, double y, double z, double nu, bool nonlinearMode, int problemId);
21
22 void copyMatrix(const Ptr< VMatrix<double> > mtx, VMatrix<double >& cpmtx);
23
24 double computeResiduumLinorm(const Ptr< VMatrix <double> > a,
25                             const Ptr<DArray1> x, const Ptr<DArray1> b,
26                             const Ptr<IArray1> isPenalizedDOF, double penalty) {
27     double llnorm;
28     double bi;
29     double w;
30     int n;
31     int i;
32     int j;
33     int iaCnt;
34     llnorm = 0.0;
35     n = a->nrow();
36     for (i = 1; i <= n; ++i) {
37         if (IA1(isPenalizedDOF, i))
38             w = 1.0 / penalty;
39         else
40             w = 1.0;
41         bi = w * DA1(b, i);
```

A. Quelltexte

```

42     for (iaCnt = IA1(a->crs->IA, i); iaCnt < IA1(a->crs->IA, i + 1); ++iaCnt) {
43         j = IA1(a->crs->JA, iaCnt);
44         bi -= w * DA1(a->crs->A, iaCnt) * DA1(x, j);
45     }
46     l1norm += fabs(bi);
47 }
48 return l1norm;
49 }
50
51 void getCR_BaseFuncs(const IArray2& simplex2faces, const IArray2& face2nodes,
52                     const Ptr<DArray1> NX, const Ptr<DArray1> NY,
53                     const Ptr<DArray1> NZ, int i,
54                     DArray1& phi_as, DArray1& phi_bs,
55                     DArray1& phi_cs, DArray1& phi_ds) {
56     DArray1& nx = NX;
57     DArray1& ny = NY;
58     DArray1& nz = NZ;
59     Ptr<IArray1> IPiv = new IArray1(4);
60     Ptr<DArray2> PhiMat = new DArray2(4, 4);
61     DArray2& phi_mat = PhiMat;
62     DArray1 phi_sol(4);
63     int fa1 = simplex2faces(i, 1);
64     // [...]
65     double fmp1x = (nx(face2nodes(fa1, 1)) + nx(face2nodes(fa1, 2))
66                   + nx(face2nodes(fa1, 3))) / 3.0;
67     double fmp1y = (ny(face2nodes(fa1, 1)) + ny(face2nodes(fa1, 2))
68                   + ny(face2nodes(fa1, 3))) / 3.0;
69     double fmp1z = (nz(face2nodes(fa1, 1)) + nz(face2nodes(fa1, 2))
70                   + nz(face2nodes(fa1, 3))) / 3.0;
71     // [...]
72     phi_mat(1, 1) = 1;
73     phi_mat(2, 1) = 1;
74     phi_mat(3, 1) = 1;
75     phi_mat(4, 1) = 1;
76     phi_mat(1, 2) = fmp1x;
77     phi_mat(2, 2) = fmp2x;
78     phi_mat(3, 2) = fmp3x;
79     phi_mat(4, 2) = fmp4x;
80     // [...]
81     // Calculate lu decomposition of local matrix
82     matlu(phi_mat, IPiv);
83     matsolv(phi_mat, IPiv, phi_sol, da1(1, 0, 0, 0));
84     phi_as(1) = phi_sol(1);
85     phi_bs(1) = phi_sol(2);
86     phi_cs(1) = phi_sol(3);
87     phi_ds(1) = phi_sol(4);
88     matsolv(phi_mat, IPiv, phi_sol, da1(0, 1, 0, 0));
89     // [...]
90 }
91
92 void reconst_CR_BaseFuncs(IArray2& simplex2nodes, IArray2& simplex2faces,
93                           IArray2& face2nodes,
94                           const Ptr<DArray1> NX, const Ptr<DArray1> NY,
95                           const Ptr<DArray1> NZ,

```

A. Quelltexte

```

96         const Ptr<DArray1> TetVols, int tetIndex, int ii,
97         int direction,
98         DArray1& phi_bs, DArray1& phi_cs, DArray1& phi_ds,
99         double& aTx, double& aTy, double& aTz, double& cT) {
100     // ii is the index for the local face in the tetrahedron
101     DArray1& nx = NX;
102     DArray1& ny = NY;
103     DArray1& nz = NZ;
104     DArray1& tetvols = TetVols;
105     IArray1 le(4);
106     int fa1 = simplex2faces(tetIndex, 1);
107     // [...]
108     le(1) = fa1;
109     // [...]
110     int n1 = simplex2nodes(tetIndex, 1);
111     // [...]
112     double xbc = (nx(n1) + nx(n2) + nx(n3) + nx(n4)) / 4.0;
113     // [...]
114     double xa = nx(face2nodes(le(ii), 1));
115     double ya = ny(face2nodes(le(ii), 1));
116     double za = nz(face2nodes(le(ii), 1));
117     // [...]
118     double fmpx = (xa + xb + xc) / 3;
119     double fmpy = (ya + yb + yc) / 3;
120     double fmpz = (za + zb + zc) / 3;
121     // compute face area: 1/2 * norm(cross(p(1)-p(2), p(1)-p(3)), 2)
122     // double sigma = ...
123     // compute normal to face
124     double nsigmax = ((ya - yb) * (za - zc) - (za - zb) * (ya - yc)) / (2 * sigma);
125     double nsigmay = ((za - zb) * (xa - xc) - (xa - xb) * (za - zc)) / (2 * sigma);
126     double nsigmaz = ((xa - xb) * (ya - yc) - (ya - yb) * (xa - xc)) / (2 * sigma);
127     // get right orientation of face normal
128     if (((fmpx - xbc) * nsigmax + (fmpy - ybc) * nsigmay
129         + (fmpz - zbc) * nsigmaz) < 0.0) {
130         nsigmax = -nsigmax;
131         nsigmay = -nsigmay;
132         nsigmaz = -nsigmaz;
133     }
134     if (direction == 1) { // x-direction
135         aTx = 1.0 / tetvols(tetIndex) * sigma * nsigmax * (fmpx - xbc);
136         aTy = 1.0 / tetvols(tetIndex) * sigma * nsigmax * (fmpy - ybc);
137         aTz = 1.0 / tetvols(tetIndex) * sigma * nsigmax * (fmpz - zbc);
138         cT = phi_bs(ii) / 3; // cT is 1/3 of the div. of hat func. (0, 1)
139     }
140     else if (direction == 2) { // y-direction
141         aTx = 1.0 / tetvols(tetIndex) * sigma * nsigmay * (fmpx - xbc);
142         // [...]
143         cT = phi_cs(ii) / 3;
144     }
145     else { // z-direction
146         // [...]
147     }
148 }
149

```

A. Quelltexte

```

150 void run(double nu, Ptr<Grid> g,
151         Ptr<QuadratureFormula> QF, Ptr<QuadratureFormula> QF_bnd,
152         int isRT0mode, bool nonlinearMode, int maxIts, double damping, int problemId,
153         FILE* out, FILE* outVelo, FILE* outPress,
154         Ptr<GLViewer2 > veloviewer, Ptr<GLViewer2 > pressviewer) {
155     // extract data from quadrature formulas
156     Ptr<DArray2> QMat = QF->getPoints();
157     Ptr<DArray1> WS = QF->getWeights();
158     Ptr<DArray2> QMat_bnd = QF_bnd->getPoints();
159     Ptr<DArray1> WS_bnd = QF_bnd->getWeights();
160     // for the 3D nonlinear assembly we'll need a quadrature formula
161     // on tetrahedra with order 2
162     Ptr<QuadratureFormula> QF_nl = qfFindOrder(egTETRAHEDRON, 2, (char*)"bary");
163     Ptr<DArray2> QMat_nl = QF_nl->getPoints();
164     Ptr<DArray1> WS_nl = QF_nl->getWeights();
165     gPrepareFaces(g); // prepare face degrees of freedom
166     int facenum;
167     int tetnum;
168     facenum = gNElements(g, gFace); // #faces of the tetrahedra
169     tetnum = gNElements(g, gCell); // #tetrahedra
170     int nodenum = gNElements(g, gNode);
171     int bfacenum = gNElements(g, gBFace);
172     // value for Dirichlet penalty parameter
173     const double penalty = 1.0e30;
174     // number of dof for crouzeix-raviart
175     int dofnum = 3 * facenum + tetnum;
176     // create sparse matrix for incompressible Stokes equations
177     Ptr<VMatrix <double> > Mat = new VMatrix <double> (dofnum, dofnum);
178     VMatrix <double>& mat = Mat; // now we can write mat(i, j) instead of (*Mat)(i, j)
179     // create sparse matrix for nonlinear iteration
180     Ptr<VMatrix <double> > Nlmtx;
181     for (int i = 1; i <= dofnum; ++i)
182         mat(i, i) = 0.0; // set pivot elements
183     // create RHS vector
184     Ptr<DArray1> Rhs = new DArray1(dofnum);
185     DArray1& rhs = Rhs;
186     vset(0.0, Rhs);
187     Ptr<DArray1> Sol = new DArray1(dofnum);
188     DArray1& sol = Sol;
189     // references to further objects
190     DArray2& qmat = QMat; // now we can write qmat(i, j)
191     DArray1& ws = WS; // now we can write ws(i, j)
192     int qN = da2NRow(QMat);
193     DArray2& qmat_bnd = QMat_bnd; // now we can write qmat_bnd(i, j)
194     DArray1& ws_bnd = WS_bnd; // now we can write ws_bnd(i, j)
195     int qN_bnd = da2NRow(QMat_bnd);
196     DArray2& qmat_nl = QMat_nl;
197     DArray1& ws_nl = WS_nl;
198     int qN_nl = da2NRow(QMat_nl);
199     // extract adjacency data from grid
200     IArray2& simplex2nodes = gCellNodes(g);
201     IArray2& bface2nodes = gBFaceNodes(g);
202     IArray2& simplex2edges = gCellEdges(g);
203     IArray2& edge2nodes = gEdgeNodes(g);

```

A. Quelltexte

```

204   IArray2& bface2edges = gBFaceEdges(g);
205   IArray2& simplex2faces = gCellFaces(g);
206   IArray2& face2nodes = gFaceNodes(g);
207   IArray2& bface2faces = gBFaceFaces(g);
208   // create nx, ny, nz as columnviews of coordinate array
209   Ptr<DArray2> Coord = gCoord(g);
210   Ptr<DArray1> NX = da2Col(Coord, 1);
211   DArray1& nx = NX;
212   Ptr<DArray1> NY = da2Col(Coord, 2);
213   DArray1& ny = NY;
214   Ptr<DArray1> NZ = da2Col(Coord, 3);
215   DArray1& nz = NZ;
216   Ptr<DArray1> TetVols = new DArray1(tetnum);
217   DArray1& tetvols = TetVols;
218   // create array of tetrahedra volumes
219   for (int i = 1; i <= tetnum; i++) {
220       int n1 = simplex2nodes(i, 1);
221       int n2 = simplex2nodes(i, 2);
222       int n3 = simplex2nodes(i, 3);
223       int n4 = simplex2nodes(i, 4);
224       double dx12 = nx(n1) - nx(n2);
225       double dy12 = ny(n1) - ny(n2);
226       double dz12 = nz(n1) - nz(n2);
227       double dx13 = nx(n1) - nx(n3);
228       // [...]
229       // double det = ...
230       if (det < 0.0)
231           det = -det;
232       tetvols(i) = det;
233   } // end for(i)
234   // array for penalization information of DOFs
235   Ptr<IArray1> IsPenalizedDOF = new IArray1(dofnum);
236   IArray1& isPenalizedDOF = IsPenalizedDOF;
237   for (int i = 1; i <= dofnum; ++i) {
238       isPenalizedDOF(i) = 0;
239   } // end for(i)
240   for (int i = 1; i <= bfacenum; ++i) {
241       int e = bface2faces(i, 1);
242       isPenalizedDOF(e) = 1;
243       isPenalizedDOF(facenum + e) = 1;
244       isPenalizedDOF(2 * facenum + e) = 1;
245   } // end for(i)
246   // create arrays for local use
247   Ptr<DArray2> PhiMat = new DArray2(4, 4);
248   Ptr<IArray1> IPiv = new IArray1(4);
249   DArray1 phi_sol(4);
250   DArray1 phi_as(4);
251   DArray1 phi_bs(4);
252   DArray1 phi_cs(4);
253   DArray1 phi_ds(4);
254   IArray1 le(4);
255   // Stokes assembly loop
256   for (int i = 1; i <= tetnum; ++i) {
257       int fa1 = simplex2faces(i, 1);

```


A. Quelltexte

```

258 // [...]
259 int n1 = simplex2nodes(i, 1);
260 // [...]
261 getCR_BaseFuncs(simplex2faces, face2nodes,
262                 NX, NY, NZ, i, phi_as, phi_bs, phi_cs, phi_ds);
263 le(1) = fa1;
264 // [...]
265 // assembly of the Laplacian operator
266 for (int ii = 1; ii <= 4; ++ii)
267     for (int jj = 1; jj <= 4; ++jj) {
268         mat(le(jj), le(ii)) += nu * (tetvols(i) * (phi_bs(ii) * phi_bs(jj)
269                                     + phi_cs(ii) * phi_cs(jj)
270                                     + phi_ds(ii) * phi_ds(jj)));
271         mat(facenum + le(jj), facenum + le(ii))
272             += nu * (tetvols(i) * (phi_bs(ii) * phi_bs(jj)
273                                     + phi_cs(ii) * phi_cs(jj)
274                                     + phi_ds(ii) * phi_ds(jj)));
275         mat(2 * facenum + le(jj), 2 * facenum + le(ii))
276             += nu * (tetvols(i) * (phi_bs(ii) * phi_bs(jj)
277                                     + phi_cs(ii) * phi_cs(jj)
278                                     + phi_ds(ii) * phi_ds(jj)));
279     } // end for(jj)
280 // assembly of the divergence and the pressure gradient operators
281 for (int ii = 1; ii <= 4; ++ii) {
282     // (-u_x, q)
283     mat(3 * facenum + i, le(ii)) += tetvols(i) * (-phi_bs(ii));
284     mat(le(ii), 3 * facenum + i) += tetvols(i) * (-phi_bs(ii));
285     // (-v_y, q)
286     mat(3 * facenum + i, facenum + le(ii)) += tetvols(i) * (-phi_cs(ii));
287     mat(facenum + le(ii), 3 * facenum + i) += tetvols(i) * (-phi_cs(ii));
288     // (-w_z, q)
289     mat(3 * facenum + i, 2 * facenum + le(ii)) += tetvols(i) * (-phi_ds(ii));
290     mat(2 * facenum + le(ii), 3 * facenum + i) += tetvols(i) * (-phi_ds(ii));
291 } // end for(ii)
292 // assembly of the right hand side
293 if (!isRT0mode) {
294     for (int qi = 1; qi <= qN; qi++) {
295         double xl = qmat(qi, 1) * nx(n1) + qmat(qi, 2) * nx(n2)
296                 + qmat(qi, 3) * nx(n3) + qmat(qi, 4) * nx(n4);
297         double yl = qmat(qi, 1) * ny(n1) + qmat(qi, 2) * ny(n2)
298                 + qmat(qi, 3) * ny(n3) + qmat(qi, 4) * ny(n4);
299         double zl = qmat(qi, 1) * nz(n1) + qmat(qi, 2) * nz(n2)
300                 + qmat(qi, 3) * nz(n3) + qmat(qi, 4) * nz(n4);
301         double we = tetvols(i) * ws(qi);
302         for (int ii = 1; ii <= 4; ++ii) {
303             rhs(le(ii)) += we * f1(xl, yl, zl, nu, nonlinearMode, problemId)
304                     * (phi_as(ii) + xl * phi_bs(ii)
305                       + yl * phi_cs(ii) + zl * phi_ds(ii));
306             rhs(facenum + le(ii)) += we * f2(xl, yl, zl, nu, nonlinearMode, problemId)
307                     * (phi_as(ii) + xl * phi_bs(ii)
308                       + yl * phi_cs(ii) + zl * phi_ds(ii));
309             rhs(2 * facenum + le(ii)) += we
310                     * f3(xl, yl, zl, nu, nonlinearMode, problemId)
311                     * (phi_as(ii) + xl * phi_bs(ii)

```

A. Quelltexte

```

312                                     + yl * phi_cs(ii) + zl * phi_ds(ii));
313     } // end for(ii)
314     } // end for(qi)
315 } // end if (!isRT0mode)
316 // RTO reconstruction of the form  $v = a_T + c_T (x - x_T)$ 
317 if (isRT0mode) {
318     // barycenter for RTO reconstruction
319     double xbc = (nx(n1) + nx(n2) + nx(n3) + nx(n4)) / 4.0;
320     double ybc = (ny(n1) + ny(n2) + ny(n3) + ny(n4)) / 4.0;
321     double zbc = (nz(n1) + nz(n2) + nz(n3) + nz(n4)) / 4.0;
322     // declare aTx, aTy, aTz, cT ...
323     for (int qi = 1; qi <= qN; qi++) {
324         double xl = qmat(qi, 1) * nx(n1) + qmat(qi, 2) * nx(n2)
325             + qmat(qi, 3) * nx(n3) + qmat(qi, 4) * nx(n4);
326         // [...]
327         for (int ii = 1; ii <= 4; ++ii) {
328             // reconstruct degree of freedom in x-direction
329             reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
330                 TetVols, i, ii, 1, phi_bs, phi_cs, phi_ds,
331                 aTx, aTy, aTz, cT);
332             rhs(le(ii)) += tetvols(i) * ws(qi)
333                 * (f1(xl, yl, zl, nu, nonlinearMode, problemId)
334                     * (aTx + cT * (xl - xbc))
335                     + f2(xl, yl, zl, nu, nonlinearMode, problemId)
336                     * (aTy + cT * (yl - ybc))
337                     + f3(xl, yl, zl, nu, nonlinearMode, problemId)
338                     * (aTz + cT * (zl - zbc)));
339             // reconstruct degree of freedom in y-direction
340             reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
341                 TetVols, i, ii, 2, phi_bs, phi_cs, phi_ds,
342                 aTx, aTy, aTz, cT);
343             rhs(facenum + le(ii)) += tetvols(i) * ws(qi)
344                 * (f1(xl, yl, zl, nu, nonlinearMode, problemId)
345                     * (aTx + cT * (xl - xbc))
346                     + f2(xl, yl, zl, nu, nonlinearMode, problemId)
347                     * (aTy + cT * (yl - ybc))
348                     + f3(xl, yl, zl, nu, nonlinearMode, problemId)
349                     * (aTz + cT * (zl - zbc)));
350             // reconstruct degree of freedom in z-direction
351             reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
352                 TetVols, i, ii, 3, phi_bs, phi_cs, phi_ds,
353                 aTx, aTy, aTz, cT);
354             rhs(2 * facenum + le(ii)) += tetvols(i) * ws(qi)
355                 * (f1(xl, yl, zl, nu, nonlinearMode, problemId)
356                     * (aTx + cT * (xl - xbc))
357                     + f2(xl, yl, zl, nu, nonlinearMode, problemId)
358                     * (aTy + cT * (yl - ybc))
359                     + f3(xl, yl, zl, nu, nonlinearMode, problemId)
360                     * (aTz + cT * (zl - zbc)));
361         } // end for(ii)
362     } // end for(q)
363 } // end if (isRT0mode)
364 } // end for(int i=1; i <= tetnum; ++i)
365 // assembly of the boundary behavior using penalty

```

A. Quelltexte

```

366 for (int i = 1; i <= bfacenum; i++) {
367     int f = bface2faces(i, 1);
368     int n1 = face2nodes(f, 1);
369     // [...]
370     for (int qi = 1; qi <= qN_bnd; ++qi) {
371         double xl = qmat_bnd(qi, 1) * nx(n1) + qmat_bnd(qi, 2) * nx(n2)
372             + qmat_bnd(qi, 3) * nx(n3);
373         double yl = qmat_bnd(qi, 1) * ny(n1) + qmat_bnd(qi, 2) * ny(n2)
374             + qmat_bnd(qi, 3) * ny(n3);
375         double zl = qmat_bnd(qi, 1) * nz(n1) + qmat_bnd(qi, 2) * nz(n2)
376             + qmat_bnd(qi, 3) * nz(n3);
377         faceavgU += ws_bnd(qi) * u(xl, yl, zl, problemId);
378         faceavgV += ws_bnd(qi) * v(xl, yl, zl, problemId);
379         faceavgW += ws_bnd(qi) * w(xl, yl, zl, problemId);
380     } // end for(qi)
381     mat(f, f) += penalty;
382     mat(facenum + f, facenum + f) += penalty;
383     mat(2 * facenum + f, 2 * facenum + f) += penalty;
384     rhs(f) += faceavgU * penalty;
385     rhs(facenum + f) += faceavgV * penalty;
386     rhs(2 * facenum + f) += faceavgW * penalty;
387 } // end for(i)
388 // fix one pressure dof
389 mat(3 * facenum + 1, 3 * facenum + 1) += -penalty;
390 isPenalizedDOF(3 * facenum + 1) = 1;
391 // Stokes assembly finished
392 // Flush matrix, i.e. transform from resizeable assembly structure
393 // to actual crs format for solver
394 Mat->flush();
395 // Here begins the nonlinear iteration
396 Ptr<DArray1> BestSol = new DArray1(dofnum);
397 DArray1& bestSol = BestSol;
398 double reslnorm;
399 double oldReslnorm = 0.0;
400 double minReslnorm = -1.0;
401 double resContraction;
402 const double nonlinearItEps = 1.0e-20;
403 int its; // nonlinear iterations
404 bool isConverged = false;
405 for (its = 1; (!isConverged) && (its == 1 || nonlinearMode)
406     && (its <= maxIts); ++its) {
407     Nlmtx = 0;
408     Nlmtx = new VMatrix<double>(dofnum, dofnum);
409     VMatrix <double>& nlmtx = Nlmtx;
410     copyMatrix(Mat, Nlmtx);
411     // begin assemble nonlinearity
412     if (its > 1 && nonlinearMode) {
413         // assemble nonlinear Navier-Stokes term in rotational form
414         // for a Picard iteration
415         for (int i = 1; i <= tetnum; ++i) {
416             int fa1 = simplex2faces(i, 1);
417             // [...]
418             getCR_BaseFuncs(simplex2faces, face2nodes,
419                 NX, NY, NZ, i, phi_as, phi_bs, phi_cs, phi_ds);

```

A. Quelltexte

```

420     le(1) = fa1;
421     // [...]
422     // CR function given by: (au + bu x + cu y + du z,
423     //      av + bv x + cv y + dv z, aw + bw x + cw y + dw z) ~T
424     // declare and initialize variables ...
425     for (int ii = 1; ii <= 4; ++ii) {
426         au += sol(le(ii)) * phi_as(ii);
427         bu += sol(le(ii)) * phi_bs(ii);
428         cu += sol(le(ii)) * phi_cs(ii);
429         du += sol(le(ii)) * phi_ds(ii);
430         av += sol(facenum + le(ii)) * phi_as(ii);
431         // [...]
432     } // end for(ii)
433     if (isRT0mode == 0) {
434         int n1 = simplex2nodes(i, 1);
435         // [...]
436         for (int ii = 1; ii <= 4; ++ii) {
437             for (int jj = 1; jj <= 4; ++jj) {
438                 for (int qp = 1; qp <= qN_nl; ++qp) {
439                     double xl = nx(n1) * qmat_nl(qp, 1) + nx(n2) * qmat_nl(qp, 2)
440                     + nx(n3) * qmat_nl(qp, 3) + nx(n4) * qmat_nl(qp, 4);
441                     // [...]
442                     double uqp = au + bu * xl + cu * yl + du * zl;
443                     double vqp = av + bv * xl + cv * yl + dv * zl;
444                     double wqp = aw + bw * xl + cw * yl + dw * zl;
445                     double psi_qp = phi_as(ii) + phi_bs(ii) * xl
446                     + phi_cs(ii) * yl + phi_ds(ii) * zl;
447                     double we = tetvols(i) * ws_nl(qp);
448                     double a2 = phi_bs(jj);
449                     double a3 = phi_cs(jj);
450                     double a4 = phi_ds(jj);
451                     nlmtx(le(ii), le(jj)) += we *
452                     ( wqp * a4 * psi_qp + (-vqp) * (-a3) * psi_qp );
453                     nlmtx(facenum + le(ii), le(jj)) += we *
454                     ( uqp * (-a3) * psi_qp );
455                     nlmtx(2 * facenum + le(ii), le(jj)) += we *
456                     ( (-uqp) * a4 * psi_qp );
457                     nlmtx(le(ii), facenum + le(jj)) += we *
458                     ( (-vqp) * a2 * psi_qp );
459                     nlmtx(facenum + le(ii), facenum + le(jj)) += we *
460                     ( (-wqp) * (-a4) * psi_qp + uqp * a2 * psi_qp );
461                     nlmtx(2 * facenum + le(ii), facenum + le(jj)) += we *
462                     ( vqp * (-a4) * psi_qp );
463                     nlmtx(le(ii), 2 * facenum + le(jj)) += we *
464                     ( wqp * (-a2) * psi_qp );
465                     nlmtx(facenum + le(ii), 2 * facenum + le(jj)) += we *
466                     ( (-wqp) * a3 * psi_qp );
467                     nlmtx(2 * facenum + le(ii), 2 * facenum + le(jj)) += we
468                     ( vqp * a3 * psi_qp + (-uqp) * (-a2) * psi_qp );
469                 } // end for(qp)
470             } // end for(jj)
471         } // end for(ii)
472     }
473     else { // isRT0mode

```

A. Quelltexte

```

474 // declare variables ...
475 int n1 = simplex2nodes(i, 1);
476 // [...]
477 // barycenter for RTO reconstruction
478 double xbc = (nx(n1) + nx(n2) + nx(n3) + nx(n4)) / 4.0;
479 double ybc = (ny(n1) + ny(n2) + ny(n3) + ny(n4)) / 4.0;
480 double zbc = (nz(n1) + nz(n2) + nz(n3) + nz(n4)) / 4.0;
481 // assemble Picard part of the nonlinearity
482 for (int ii = 1; ii <= 4; ++ii) {
483     reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
484                          TetVols, i, ii, 1, phi_bs, phi_cs, phi_ds,
485                          aII_1_Tx, aII_1_Ty, aII_1_Tz, cII_1_T);
486     reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
487                          TetVols, i, ii, 2, phi_bs, phi_cs, phi_ds,
488                          aII_2_Tx, aII_2_Ty, aII_2_Tz, cII_2_T);
489     reconst_CR_BaseFuncs(simplex2nodes, simplex2faces, face2nodes, NX, NY, NZ,
490                          TetVols, i, ii, 3, phi_bs, phi_cs, phi_ds,
491                          aII_3_Tx, aII_3_Ty, aII_3_Tz, cII_3_T);
492     for (int jj = 1; jj <= 4; ++jj) {
493         for (int qp = 1; qp <= qN_nl; ++qp) {
494             double xl = nx(n1) * qmat_nl(qp, 1) + nx(n2) * qmat_nl(qp, 2)
495                    + nx(n3) * qmat_nl(qp, 3) + nx(n4) * qmat_nl(qp, 4);
496             // [...]
497             double uqp = au + bu * xl + cu * yl + du * zl;
498             double vqp = av + bv * xl + cv * yl + dv * zl;
499             double wqp = aw + bw * xl + cw * yl + dw * zl;
500             double psi_1_x_qp
501                 = aII_1_Tx + cII_1_T * (xl - xbc);
502             double psi_1_y_qp
503                 = aII_1_Ty + cII_1_T * (yl - ybc);
504             double psi_1_z_qp
505                 = aII_1_Tz + cII_1_T * (zl - zbc);
506             double psi_2_x_qp
507                 = aII_2_Tx + cII_2_T * (xl - xbc);
508             // [...]
509             double we = tetvols(i) * ws_nl(qp);
510             double a2 = phi_bs(jj);
511             double a3 = phi_cs(jj);
512             double a4 = phi_ds(jj);
513             nlmix(le(ii), le(jj)) += we * ( wqp * a4 * psi_1_x_qp +
514                (-vqp) * (-a3) * psi_1_x_qp + uqp * (-a3) * psi_1_y_qp +
515                (-uqp) * a4 * psi_1_z_qp );
516             nlmix(facenum + le(ii), le(jj)) += we * ( wqp * a4 * psi_2_x_qp +
517                (-vqp) * (-a3) * psi_2_x_qp + uqp * (-a3) * psi_2_y_qp +
518                (-uqp) * a4 * psi_2_z_qp );
519             nlmix(2 * facenum + le(ii), le(jj)) += we * ( wqp * a4 * psi_3_x_qp +
520                (-vqp) * (-a3) * psi_3_x_qp + uqp * (-a3) * psi_3_y_qp +
521                (-uqp) * a4 * psi_3_z_qp );
522             nlmix(le(ii), facenum + le(jj)) += we * ( (-vqp) * a2 * psi_1_x_qp +
523                (-wqp) * (-a4) * psi_1_y_qp + uqp * a2 * psi_1_y_qp +
524                vqp * (-a4) * psi_1_z_qp );
525             nlmix(facenum + le(ii), facenum + le(jj))
526                 += we * ( (-vqp) * a2 * psi_2_x_qp + (-wqp) * (-a4) * psi_2_y_qp +
527                    uqp * a2 * psi_2_y_qp + vqp * (-a4) * psi_2_z_qp );

```

A. Quelltexte

```

528         nlmtx(2 * facenum + le(ii), facenum + le(jj))
529         += we * ( (-vqp) * a2 * psi_3_x_qp + (-wqp) * (-a4) * psi_3_y_qp +
530                 uqp * a2 * psi_3_y_qp + vqp * (-a4) * psi_3_z_qp );
531         nlmtx(le(ii), 2 * facenum + le(jj))
532         += we * ( wqp * (-a2) * psi_1_x_qp + (-wqp) * a3 * psi_1_y_qp +
533                 vqp * a3 * psi_1_z_qp + (-uqp) * (-a2) * psi_1_z_qp );
534         nlmtx(facenum + le(ii), 2 * facenum + le(jj))
535         += we * ( wqp * (-a2) * psi_2_x_qp + (-wqp) * a3 * psi_2_y_qp +
536                 vqp * a3 * psi_2_z_qp + (-uqp) * (-a2) * psi_2_z_qp );
537         nlmtx(2 * facenum + le(ii), 2 * facenum + le(jj))
538         += we * ( wqp * (-a2) * psi_3_x_qp + (-wqp) * a3 * psi_3_y_qp +
539                 vqp * a3 * psi_3_z_qp + (-uqp) * (-a2) * psi_3_z_qp );
540     } // end for (qp)
541 } // end for (jj)
542 } // end for (ii)
543 } // if (isRT0mode == 0)
544 } // end for (i)
545 } // end if (its > 1 && nonlinearMode)
546 // end assemble nonlinearity
547 Nlmtx->flush();
548 if (nonlinearMode && its > 1) {
549     oldReslnorm = reslnorm;
550     reslnorm = computeResiduumLinorm(Nlmtx, Sol, Rhs, IsPenalizedDOF, penalty);
551     if (its == 2 || reslnorm < minReslnorm) {
552         minReslnorm = reslnorm;
553         for (int i = 1; i <= dofnum; ++i) {
554             bestSol(i) = sol(i);
555         } // end for(i)
556     } // end if
557     if (reslnorm < nonlinearItEps) {
558         isConverged = true;
559     } // end if
560 } // end if
561 Ptr< VSolverPardiso <double> > Solver = new VSolverPardiso <double> (Nlmtx, 11);
562 // set solver parameters ...
563 Solver->update(); // update solver (analysis + LU factorization)
564 Solver->solve(Sol, Rhs);
565 if (its >= 2) {
566     for (int i = 1; i <= dofnum; ++i) {
567         sol(i) = (1.0 - damping) * bestSol(i) + damping * sol(i);
568     } // end for(i)
569 } // end if
570 if (!isConverged && its > 2) {
571     resContraction = reslnorm / oldReslnorm;
572     if (resContraction >= 1.0) {
573         isConverged = true;
574     } // end if
575 } // end if
576 } // end for(its)
577 if (minReslnorm >= -1.0e-16) {
578     for (int i = 1; i <= dofnum; ++i) {
579         sol(i) = bestSol(i);
580     } // end for(i)
581 } // end if

```

A. Quelltexte

```

582 // Enforce that discrete pressure has a zero average
583 double pAverage = 0.0;
584 double area = 0.0;
585 for (int i = 1; i <= tetnum; ++i) {
586     area += tetvols(i);
587     pAverage += tetvols(i) * sol(3 * facenum + i);
588 } // end for(int i)
589 pAverage = pAverage / area;
590 for (int i = 1; i <= tetnum; ++i) {
591     sol(3 * facenum + i) = sol(3 * facenum + i) - pAverage;
592 } // end for(int i)
593 // evaluate norms and error norms w.r.t.
594 // the discrete and continuous solution
595 // declare and initialize variables ...
596 for (int i = 1; i <= tetnum; ++i) {
597     int n1 = simplex2nodes(i, 1);
598     // [...]
599     double fa1 = simplex2faces(i, 1);
600     // [...]
601     getCR_BaseFuncs(simplex2faces, face2nodes,
602                     NX, NY, NZ, i, phi_as, phi_bs, phi_cs, phi_ds);
603     le(1) = fa1;
604     // [...]
605     // evaluate function u, v, w at quadrature points
606     for (int qi = 1; qi <= qN; ++qi) {
607         double xl
608             = qmat(qi, 1) * nx(n1) + qmat(qi, 2) * nx(n2)
609             + qmat(qi, 3) * nx(n3) + qmat(qi, 4) * nx(n4);
610         // [...]
611         double we = tetvols(i) * ws(qi);
612         l2normExact += we *
613             (u(xl, yl, zl, problemId) * u(xl, yl, zl, problemId)
614              + v(xl, yl, zl, problemId) * v(xl, yl, zl, problemId)
615              + w(xl, yl, zl, problemId) * w(xl, yl, zl, problemId));
616         h1normExact += we *
617             (ux(xl, yl, zl, problemId) * ux(xl, yl, zl, problemId)
618              + uy(xl, yl, zl, problemId) * uy(xl, yl, zl, problemId)
619              + uz(xl, yl, zl, problemId) * uz(xl, yl, zl, problemId)
620              + vx(xl, yl, zl, problemId) * vx(xl, yl, zl, problemId)
621              + vy(xl, yl, zl, problemId) * vy(xl, yl, zl, problemId)
622              + vz(xl, yl, zl, problemId) * vz(xl, yl, zl, problemId)
623              + wx(xl, yl, zl, problemId) * wx(xl, yl, zl, problemId)
624              + wy(xl, yl, zl, problemId) * wy(xl, yl, zl, problemId)
625              + wz(xl, yl, zl, problemId) * wz(xl, yl, zl, problemId));
626         pnormExact += we * p(xl, yl, zl, nu, nonlinearMode, problemId)
627             * p(xl, yl, zl, nu, nonlinearMode, problemId);
628     // declare and initialize variables ...
629     for (int ii = 1; ii <= 4; ++ii) {
630         uu += sol(le(ii)) * (phi_as(ii) + xl * phi_bs(ii)
631                             + yl * phi_cs(ii) + zl * phi_ds(ii));
632         vv += sol(facenum + le(ii)) * (phi_as(ii) + xl * phi_bs(ii) +
633                                       yl * phi_cs(ii) + zl * phi_ds(ii));
634         ww += sol(2 * facenum + le(ii)) * (phi_as(ii) + xl * phi_bs(ii) +
635                                             yl * phi_cs(ii) + zl * phi_ds(ii));

```

A. Quelltexte

```

636     gux += sol(le(ii)) * phi_bs(ii);
637     guy += sol(le(ii)) * phi_cs(ii);
638     guz += sol(le(ii)) * phi_ds(ii);
639     gvz += sol(facenum + le(ii)) * phi_bs(ii);
640     // [...]
641 } // end for(ii)
642 l2norm += we * (uu * uu + vv * vv + ww * ww);
643 l2errnorm += we *
644     ((u(xl, yl, zl, problemId) - uu) * (u(xl, yl, zl, problemId) - uu)
645     + (v(xl, yl, zl, problemId) - vv) * (v(xl, yl, zl, problemId) - vv)
646     + (w(xl, yl, zl, problemId) - ww) * (w(xl, yl, zl, problemId) - ww));
647 hinorm += we * (gux * gux + guy * guy + guz * guz
648     + gvz * gvz + gvy * gvy + gvz * gvz
649     + gwx * gwx + gwy * gwy + gwz * gwz);
650 hierrnorm += we * ((ux(xl, yl, zl, problemId) - gux) *
651     (ux(xl, yl, zl, problemId) - gux)
652     + (uy(xl, yl, zl, problemId) - guy) *
653     (uy(xl, yl, zl, problemId) - guy)
654     + (uz(xl, yl, zl, problemId) - guz) *
655     (uz(xl, yl, zl, problemId) - guz)
656     + (vx(xl, yl, zl, problemId) - gvz) *
657     (vx(xl, yl, zl, problemId) - gvz)
658     // [...]
659     + (wz(xl, yl, zl, problemId) - gwz) *
660     (wz(xl, yl, zl, problemId) - gwz));
661 divnorm += we * (gux + gvy + gwz) * (gux + gvy + gwz);
662 pnorm += we * sol(3 * facenum + i) * sol(3 * facenum + i);
663 perrnorm += we * (p(xl, yl, zl, nu, nonlinearMode, problemId)
664     - sol(3 * facenum + i))
665     * (p(xl, yl, zl, nu, nonlinearMode, problemId)
666     - sol(3 * facenum + i));
667 } // end for(qi)
668 } // end for(int i)
669 l2errnorm = sqrt(l2errnorm);
670 hierrnorm = sqrt(hierrnorm);
671 divnorm = sqrt(divnorm);
672 perrnorm = sqrt(perrnorm);
673 // [...]
674 // print norms...
675 // write norms to data file...
676 // interpolate values on faces to values on nodes...
677 // start graphical viewer...
678 // dump data into files...
679 }
680
681 int main (const int argc, const char* argv[]) {
682     pdelib2init(argc, argv); // initialize pdelib2
683     // process input arguments ...
684     Ptr<Grid> g = gRead(argFile(1)); // read grid object from file
685     // initialize output files and graphical viewers ...
686     // find appropriate quadrature rules
687     Ptr<QuadratureFormula> QF = qfFindOrder(egTETRAHEDRON, order,
688     (char*)"bary");
689     Ptr<QuadratureFormula> QF_bnd = qfFindOrder(egTRIANGLE, 15,

```


A. Quelltexte

```
690         (char*)"bary");
691     // run code
692     run(nu, g, QF, QF_bnd, rt, nonlinearMode, maxIts, damping,
693         problemId, out, outVelo, outPress, veloviewer, pressviewer);
694     // close output files ...
695     return 0;
696 }
```

Listing A.2: Python/FENICS-Programmcode zu Beispiel 4 (Lid-driven cavity)

```
1 from dolfin import *
2 import numpy as np
3 # Variables
4 nu = 1.0
5 h = 8
6 tol = 1e-11
7 maxiter = 25
8 # Domain
9 mesh = UnitCubeMesh(h, h, h)
10 # Define function spaces
11 V = VectorFunctionSpace(mesh, "CR", 1)
12 Q = FunctionSpace(mesh, "DG", 0)
13 W = V * Q
14 # Input Data
15 f = Constant((0.0, 0.0, 0.0))
16 sol_u_k = interpolate(Constant((0.0, 0.0, 0.0)), V)
17 # Define boundary behaviour
18 class Top(SubDomain):
19     def inside(self, x, on_boundary):
20         return on_boundary and x[2] > 1.0 - DOLFIN_EPS
21 top_domain = Top()
22 top_val = Expression(("1.0", "0.0", "0.0"))
23 class NoSlipDomain(SubDomain):
24     def inside(self, x, on_boundary):
25         return on_boundary and x[2] < 1.0 - DOLFIN_EPS
26 noslip_val = Constant((0.0, 0.0, 0.0))
27 noslip_domain = NoSlipDomain()
28 class PinPoint(SubDomain):
29     def inside(self, x, on_boundary):
30         return x[0] < DOLFIN_EPS and x[1] < DOLFIN_EPS and x[2] < DOLFIN_EPS
31 pin_domain = PinPoint()
32 pin_val = Constant(0.0)
33 bc_top = DirichletBC(W.sub(0), top_val, top_domain)
34 bc_noslip = DirichletBC(W.sub(0), noslip_val, noslip_domain)
35 bc_pin = DirichletBC(W.sub(1), pin_val, pin_domain)
36 bc = [bc_top, bc_noslip, bc_pin]
37 # Define trial and test functions
38 (u, p) = TrialFunctions(W)
39 (v, q) = TestFunctions(W)
40 # Solution Function
41 w = Function(W)
42 # Define the variational problems
43 a = nu * inner(grad(u), grad(v)) * dx - p * div(v) * dx + div(u) * q * dx \
44     + inner(dot(grad(u), sol_u_k), v) * dx
```

A. Quelltexte

```
45 L = inner(f, v) * dx
46 # Start Picard iteration
47 eps = 1.0
48 iters = 0
49 while eps > tol and iters < maxiter:
50     iters += 1
51     solve(a == L, w, bc)
52     sol_u, sol_p = w.split(deepcopy = True)
53     diff = sol_u_k.vector().array() - sol_u.vector().array()
54     eps = np.linalg.norm(diff, ord = np.Inf)
55     print "iter={iters}: norm={eps:5.2e}".format(iters = iters, eps = eps)
56     sol_u_k.assign(sol_u)
57 norm = np.sqrt(assemble(inner(sol_u, sol_u) * dx))
58 print "n_dof = ", W.dim()
59 print "nu = {nu}".format(nu = nu)
60 print "L2 Norm = {norm:8.7e}".format(norm = norm)
```

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

