Fakultät 5
Institut für Strömungsmechanik und Technische Akustik,
Fachgebiet Numerische Fluiddynamik

Numerische Modellierung der Fluid-Struktur Interaktion fraktaler Bäume

Masterarbeit

eingereicht von: Büchner, Elias
geboren am: 19.02.1988 in: Köthen
Studiengang: Physikalische Ingenieurwissenschaft
Erstgutachter: Prof. Dr. sc. techn. habil. Jörn Sesterhenn
Zweitgutachter: Dr.-Ing. Thomas Engels
Abgabedatum 29. September 2016

Eidesstattliche Versicherung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den September 29, 2016

Unterschrift

Inhaltsverzeichnis

Eidesstattliche Versicherung i

Inhaltsverzeichnis ii

Abbildungsverzeichnis iv

Tabellenverzeichnis vii

1 Kurzzusammenfassung 1

2 Einleitung 2

3 Fluid 5
3.1 Volume-Penalisation-Methode 5
3.2 Diskretisierung 6
3.3 Biot-Savart 7

4 Festkörper 9
4.1 Fraktaler Baum 13
4.2 Bewegungsgleichung des Baumes: Drehimpulssatz . . 16
4.3 Geschwindigkeit des Baumes 17

5 Kopplung von der Strömung und dem Festkörper 20
5.1 Masken 20
5.1.1 Maske für den Boden 21
5.1.2 Maske für den Baum 21
5.2 Sponge 29
5.3 Kraft und Momentberechnung 31
5.4 Simulation 32

6 Ergebnisse 34
6.1 Kraft und Moment für den starren Baum (rigid) . . 34
6.2 Kraft und Moment für den bewegten Baum für die FSI 39
6.3 Winkel und Winkelgeschwindigkeit den bewegten Baum (FSI) 45

6.4 Zeitlich gemittelte Geschwindigkeitsfelder 48
6.5 Turbulente kinetische Energie 62
6.6 Wirbelstärke 67

7 Fazit **72**

8 Ausblick **72**

Literatur **75**

Abbildungsverzeichnis

1 Gebiete vom Fulid Ω_f, vom Festkörper Boden und Baum Ω_s, der Rand vom Festkörper $\partial\Omega_s$ 6
2 Varianten von fraktalen Bäumen in 2D und in 3D . . 9
3 Mögliche Unterteilung eines Systems in Hauptgruppen und Untergruppen für die Rigid Finite Element Methode aus [24] 11
4 Beispiel für die Rigid Finit Element Methode. Variante der Zerteilung eines Systems in Untersysteme aus [21] Seite 59 12
5 Beispiele für Variationen von fraktalen Bäumen, hierbei ist $\#g$ die Anzahl der Generationen, α der Winkel der nächsten Astgeneration, und $r_{h,0}$ eine Längenvariable (siehe unten) 13
6 Darstellung der Baumeigenschaften für Fraktale Bäume 15
7 Schematische Darstellung der Baumherstellungsfunktion 15
8 Freischnitt des Baumes und Darstellung der Momente 17
9 Winkel α bezogen auf die Symmetrieachse des Baumes 18
10 Darstellung des Radius $r(\boldsymbol{x})$, Abstand zum ersten Punkt $\boldsymbol{x}_0$ zum jeweiligen Punkt des Baumes 18
11 Kombination von zwei Masken zu einer Maske 22
12 Gebiet innen, oben und unten 23
13 Gebiete der drei Fälle und das lokale Koordinatensystem in Richtung des Astes 24
14 Kreisfunktion 28
15 Enden des Baumes ohne Kreis, mit Kreis und der ganze Baum 29
16 Sponge mask, χ_{sponge}, wie sie gesetzt wird um die Auslassbedingung zu realisieren 30
17 Anfahrfunktion 33
18 Die Kraft in x-Richtung, F_x, für den starren Baum bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen . 35
19 Die Kraft in y-Richtung, F_x, für den starren Baum bei $Re = 100$. 35

20 Der Betrag der Kraft, $|\boldsymbol{F}|$, für den starren Baum bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen . . . 36
21 Die z-Komponente des Moments M_z für den starren Baum bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen 36
22 Die Kraft in x-Richtung, F_x, für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen . 37
23 Die Kraft in y-Richtung, F_y, für den starren Baum bei $Re = 600$. 37
24 Der Betrag der Kraft, $|\boldsymbol{F}|$, für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen . . . 38
25 Die z-Komponente des Moments M_z für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen 38
26 Kraft in x-Richtung F_x für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen . . . 40
27 Kraft in y-Richtung F_y für die FSI-Simulation bei $Re = 100$ 41
28 Betrag der Kraft $|\boldsymbol{F}|$ für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen 41
29 Die z Komponente des Moments M_z FSI für die FSI-Simulation, zubeachten y-Achse 3 lineare Skalen 42
30 Kraft in x-Richtung F_x für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen . . . 42
31 Kraft in y-Richtung F_y für die FSI-Simulation bei $Re = 100$ 43
32 Betrag der Kraft $|\boldsymbol{F}|$ für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen 43
33 Die z Komponente des Moments M_z FSI für die FSI-Simulation, zubeachten y-Achse 2 lineare Skalen 44
34 Der Winkel des Baumes φ für die FSI-Simulation bei $Re = 100$ 46
35 Der Winkel des Baumes φ für die FSI-Simulation bei $Re = 600$ 46
36 Der Winkelgeschwindigkeit des Baumes $\dot{\varphi}$ für die FSI-Simulation bei $Re = 100$ 47

37 Der Winkelgeschwindigkeit des Baumes $\dot{\varphi}$ für die FSI-Simulation bei $Re = 600$ 47
38 Durch die periodischen Randbedingungen ist hinterm Rechengebiet (links) ein Baum, auch Ghost-Baum genannt und oben vom Rechengebiet eine Wand. . . 49
39 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1536 für $Re = 100$ 51
40 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1563 für $Re = 600$ 52
41 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 3072x3072 für $Re = 100$ 53
42 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 3072x3072 für $Re = 600$ 54
43 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1536 für $Re = 100$ 55
44 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 6144x6144 für $Re = 600$ 56
45 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 1024x1024 für $Re = 600$ 57
46 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 512x512 für $Re = 600$ 58
47 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 256x256 für $Re = 600$ 59
48 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 512x512 für $Re = 100$ 60
49 Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 256x256 für $Re = 100$ 61
50 Turbulente kinetische Energie k, für die starre Baumsimulation und für verschiedene Auflösungen bei $Re = 100$ 63

51 Turbulente kinetische Energie k, für die starre Baumsimulation und für verschiedene Auflösungen bei $Re = 600$ 64
52 Turbulente kinetische Energie k für die FSI-Simulation, bei $Re = 600$ 65
53 Turbulente kinetische Energie k für die FSI-Simulation, bei $Re = 100$ 66
54 Wirbelstärke ω_z für die rigid Simulation für die Zeit $t = 6$ 68
55 Wirbelstärke ω_z für die FSI Simulation für die Zeit $t = 6$ 69
56 Darstellung der Wirbelstärke ω_z für die FSI-Simulation in der Auflösung 1024x1024 bei $Re = 600$ für verschiedene Zeiten $t \in [0, 6{,}2]$ 70
57 Darstellung der Wirbelstärke ω_z für die FSI-Simulation bei $Re = 600$ in der Auflösung 1024x1024 für verschiedene Zeiten $t \in [6{,}8, 13]$ 71
58 Varianten für das mechanische Ersatzmodell: einzelne bewegliche Äste 74

Tabellenverzeichnis

1 Bedingung für das Gesamtgebiet des Astes für den Fall 1 und Fall 2, hierbei ist $B_\mathrm{g} = B_\mathrm{i}/2 + B_\mathrm{s}$, $\max\_y = \max(P_{1y},P_{2y})$, $\min\_y = \min(P_{1y},P_{2y})$, $\max\_x = \max(P_{1x},P_{2x})$, $\min\_x = \min(P_{1x},P_{2x})$ und $A = Ast\_gesamt$ 24

1 Kurzzusammenfassung

In der Natur werden Bäume umströmt. Ziel dieser Arbeit ist es, zuerst den Baum zu modellieren. Danach soll die Kopplung zwischen der Strömung und dem Baum beschrieben werden. Hieraus gewinnen wir erste Erkenntnisse zum Strömungsfeld.

In dieser Arbeit wird mit der Volume-Penalisation-Methode, mit der FOURIER-pseudospectral-Diskretisierung, mit einem strukturierten äquidistanten Gitter zum einen die Umströmung eines starren Baumes berechnet. Zum anderen wird der Baum mit einer Fluid-Struktur-Interaktion (FSI) berechnet. Bei der FSI-Simulation wird der Baum als starrer Körper angenommen, der durch eine Drehfeder am Boden befestigt ist. Hierdurch hat der Baum einen Freiheitsgrad, die Drehung um diesen Punkt. Zur Beschreibung der Bewegung des Baumes wird der Drehimpulssatz verwendet. Der Baum ist ein fraktaler Baum und entspricht von der Dimension nicht einem realen in der Natur vorkommenden Baum, liefert aber erste Einblicke in die Turbulenzgenerierung fraktaler Bäume.

Keywords: Volume-Penalisation-Methode, Fourier-pseudospectral-Diskretisierung, fraktaler Baum, Fluid-Struktur-Interaktion (FSI)

2 Einleitung

Warum FSI, fraktale Bäume und Turbulenzgenerierung? Warum sollte man sich für Bäume interessieren? Turbulenz ist überall in der Natur zu finden und ist insbesondere für fliegende Tiere wichtig.

In dieser Arbeit wird zum einen die Umströmung eines starren Baumes und zum anderen die Umströmung eines Baumes, der sich um seinen Verbindungspunkt zum Boden drehen kann und mit einer Drehfeder versehen ist, simuliert.

Baummodell Der Baum wurde als Fraktaler Baum angenommen, da er somit leicht reproduzierbar ist und nur durch wenige Parameter eindeutig beschrieben wird. Das mechanische Ersatzmodell beschränkt sich darauf, dass der Baum als starrer Körper angenommen wird, der sich um seinen ersten Punkt drehen kann. An diesem Punkt befinden sich ein Lager, ein Dämpfer und eine Drehfeder. Die Dynamik des Baumes wird unter Zuhilfenahme des Drehimpulssatzes berechnet [16].

Numerik Um die Fluid-Struktur-Interaktion zu simulieren, gibt es zwei Hauptgruppen bzgl. der Gitterwahl. In der ersten Gruppe wird bei der Verformung bzw. Bewegung des Festkörpers das Gitter mitbewegt (siehe z.B. [2]). Hier besteht die Schwierigkeit zum einen darin, die Gitterqualität zu erhalten, zum Beispiel Innenwinkel und Seitenverhältnisse, zum anderen darin, die Werte zwischen den verschiedenen, zeitabhängigen Gittern zu extrapolieren. In der zweiten Gruppe wird die Verformung bzw. Bewegung des Festkörpers durch einen Zusatzterm in den Bilanzgleichungen mit berücksichtigt (siehe [4], [9], [8], [20] und [13]). Vorteile hierbei sind, dass das Gitter nicht aufwändig gebaut werden muss und das strukturierte äquidistante Gitter verwendet werden können. Die Simulation kann somit parallelisiert werden und auf einem Cluster bzw. Super PC berechnet werden.

Die Strömung des Fluids wird in dieser Arbeit mit der Volume-Penalisation-Methode, mit einer pseudospektral-Diskretisierung für äquidistante Gitter berechnet. Wenn inkompressibel gerechnet wird, muss die POISSON-Gleichung gelöst werden. Bei Finite-Differenzen-Verfahren nimmt dies große Rechenkapazitäten ein. Ein großer Vorteil der pseudo-spektralen-Diskretisierung für äquidistante Gitter ist, dass die Lösung der POISSON-Gleichung durch eine einfach Division gewonnen werden kann, weil der LAPLACE-Operator im FOURIERraum diagonalisiert. Die Lösung ist somit schnell zu berechnen ist [8].

Vorher Nachher Abgrenzung In dem bestehenden Fluid-Solver-Code von Thomas Engels wird die Fluid-Struktur-Interaktion (FSI) implementiert. Der Code zur Generierung des Fraktalbaums von Thomas Engels wird von mir erweitert.

Für die FSI-Simulationen wird die Software Matlab [14] verwendet.

Gliederung und Aufbau Im ersten Abschnitt werden die Grundlagen zur Modellierung des Fluids näher erläutert. Die Unterpunkte sind: dimensionslose Impulsbilanzgleichung mit dem NAVIER-STOKESschem Materialgesetz, die Volume-Penalisation-Methode (siehe [4] und [6]), die verwendete Diskretisierung der partiellen Differenzialgleichungen und die Umrechnung der Wirbelstärke zur Schwankungsgeschwindigkeit.

Im zweiten Abschnitt wird der Baum näher beschrieben. Hierzu gehören: die Findung des mechanischen Ersatzmodells für einen Baum, die Findung der Form des Baumes und die Berechnung der Dynamik des Baumes mit Zuhilfenahme des Drehimpulssatzes [16].

Im dritten Abschnitt wird die Kopplung zwischen dem Baum und dem Fluid beschrieben. Hierzu gehört zum einen das Erstellen der Masken und des Sponges und zum anderen die Berechnung der Kräfte und der Momente. Zudem wird die Frage geklärt, wie die Geschwindigkeit in jedem Punkt des Baumes bestimmt wird.

Im letzten Abschnitt wird die Kraft, das Moment, die Auslenkung und die Auslenkungsgeschwindigkeit des Baumes, die zeitlich gemittelte Geschwindigkeit, die turbulente kinetische Energie und die Wirbelstärke ausgewertet. Anschließend wird ein Fazit und ein Ausblick gegeben.

3 Fluid

Im folgenden Kapitel werden die Bedingungen der Simulation, die Volume-Penalisation-Methode, die Diskretisierung in Raum und Zeit und das Gesetz von Biot-Savart näher erläutert.

In dieser Arbeit sollen die Bedingungen:

$$Ma = \frac{U}{c}, \qquad Ma^2 \ll 1, \qquad \frac{f^2 L^2}{c^2} \ll 1, \tag{3.1}$$

gelten, hierbei sind U, L und f die charakteristische Geschwindigkeit, Länge und Frequenz und c die Schallgeschwindigkeit. Die charakteristische Geschwindigkeit und Länge in dieser Arbeit ist 1. Weiterhin soll gelten, dass die Machzahl Ma sehr klein ist und damit die Annahme der Inkompressibilität gerechtfertigt ist.

3.1 Volume-Penalisation-Methode

Die Grundidee der Volume-Penalisation-Methode besteht darin, statt für das Gebiet des Festkörpers, im Folgenden Solid genannt, eine komplizierte Berandung zu bauen, einen zusätzlichen Term für das Solid-Gebiet einzufügen. Die dimensionslose Impulsbilanzgleichung für das Randbedingungsproblem mit Anfangswert aus [8] mit dem Navier-Stokesschen Materialgesetz lautet:

$$\partial_t \boldsymbol{u} + \boldsymbol{\omega} \times \boldsymbol{u} = -\nabla \boldsymbol{\Pi} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}_p \tag{3.2}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \boldsymbol{\omega} = \nabla \times \boldsymbol{u} \qquad \boldsymbol{\Pi} = p + \frac{1}{2}\boldsymbol{u} \cdot \boldsymbol{u} \tag{3.3}$$

$$\boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{u}_0(\boldsymbol{x}) \qquad \boldsymbol{u}|_{\partial\Omega_\mathrm{s}(\mathrm{s},t)}(\mathrm{s}, t = 0) = \boldsymbol{u}_\mathrm{s}(\mathrm{s},t) \tag{3.4}$$

hierbei ist $\boldsymbol{u}(\boldsymbol{x}, t)$ die Geschwindigkeit des Fluids, $\boldsymbol{u}_\mathrm{s}$ die Geschwindigkeit des Solids, $\boldsymbol{\omega}$ die Wirbelstärke, $\boldsymbol{\Pi}$ der totale Druck, p der statische Druck, ν die kinematische Viskosität und $\boldsymbol{F}_p$ die externen Kräfte. In Abb. 1 ist das gesamte Gebiet Ω, das Gebiet des Fluids Ω_f, das Gebiet des Solids Ω_s und der Rand des Gebiets des Solids $\partial\Omega_\mathrm{s}$ dargestellt. Es gilt $\Omega = \Omega_\mathrm{f} \cup \Omega_\mathrm{s}$. Die dimensionslose Impulsbi-

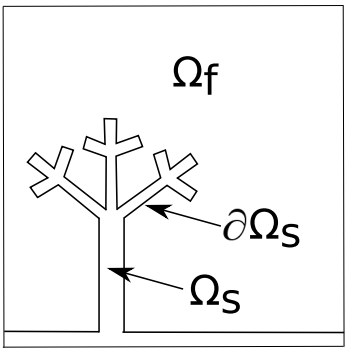


Abb. 1: Gebiete vom Fulid Ω_f, vom Festkörper Boden und Baum Ω_s, der Rand vom Festkörper $\partial\Omega_\mathrm{s}$

lanzgleichung für periodische Randbedingung (siehe z.B. [4], [20], [9] und [13]) mit dem Penalisation-Term lautet:

$$\partial_t \boldsymbol{u} + \boldsymbol{\omega} \times \boldsymbol{u} = -\nabla \boldsymbol{\Pi} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}_p - \frac{\chi}{C_\eta} (\boldsymbol{u} - \boldsymbol{u}_\mathrm{s}). \tag{3.5}$$

Hierbei ist C_η eine Konstante. In dieser Arbeit ergibt sich dieser Wert aus:

$$C_\eta = \frac{(\Delta x K_\eta)^2}{\nu} \qquad K_\eta = 2 \cdot 10^{-1}. \tag{3.6}$$

Der äquidistante Gitterabstand ist Δx und K_η eine Konstante. Laut [8] sollte diese Konstante so klein wie möglich und so groß wie nötig sein. Diese Erkenntnis kommt aus zahlreichen Validierungstests [8].

3.2 Diskretisierung

Im Folgenden wird die Diskretisierung im Zeit- und Ortsbereich näher erläutert. Da die Poisson-Gleichung im Fourierraum für äquidistante Gitter sich durch eine einfache Division lösen lässt, wird die partielle Differentialgleichung (3.5) mit Hilfe der Fourier-pseudospectral-Diskretisierung gelöst. Eine Größe $q = q(\boldsymbol{x},t)$ lässt

sich mit der diskreten FOURIERtransformation aus [5] wie folgt darstellen:

$$q(\boldsymbol{x},t) = \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \hat{q}(\boldsymbol{k},t) \exp(i\boldsymbol{k} \cdot \boldsymbol{x}). \tag{3.7}$$

Hierbei sind N_x, N_y die Anzahl der Gitterpunkte in der jeweiligen Richtung, $\boldsymbol{k} = (k_x, k_y)^T$ die Wellenvektor, $\hat{q}$ ist im FOURIERraum und ist der FOUIERkoeffizient und $i = \sqrt{-1}$. Die Variable q ist im physikalischem Raum und repräsentiert z.B. die Geschwindigkeit, den Druck.

Als Zeitintegrationsverfahren wird das RUNGE-KUTTA-Verfahren zweiter Ordnung verwendet und für jeden Zeitschritt wird $\hat{\boldsymbol{u}}$ gelöst. In jedem Zeitschritt werden nicht auflösbare Wellenzahlen weggefiltert (dealiasing).

3.3 Biot-Savart

Im folgenden Abschnitt wird erläutert, wie sich aus der Wirbelstärke im FOURIERraum $\hat{\boldsymbol{\omega}}$ die Geschwindigkeit im physikalischem Raum $\boldsymbol{u}$ errechnen lässt. Mit Hilfe der HELMHOLTZ-Zerlegung wird die Geschwindigkeit in einen rotationsfreien und einen rotationsbehafteten Anteil aufgeteilt zu:

$$\boldsymbol{u} = \nabla\Phi + \nabla \times \boldsymbol{\Psi}. \tag{3.8}$$

Der rotationsfreie Anteil ist die Potentialströmung $\nabla\Phi$. Der Rotationsanteil ist $\nabla \times \boldsymbol{\Psi}$ und divergenzfrei. Die Gl. (3.8) wird mit $\nabla\cdot$ durchmultipliziert und wir erhalten:

$$\nabla \cdot \boldsymbol{u} \overset{!}{=} 0 = \nabla^2\Phi + \underbrace{\nabla \cdot \nabla \times \boldsymbol{\Psi}}_{=0} = \nabla^2\Phi. \tag{3.9}$$

Aus der FOURIER-pseudospectral-Diskretisierung folgt, dass die Randbedingungen periodisch sind. Unter dieser Voraussetzung erhal-

ten wir

$$0 = |k^2|\hat{\Phi}_k \qquad \hat{\Phi} = 0 \qquad \Rightarrow \qquad \Phi = \text{const.} \tag{3.10}$$

Daher lässt sich die Geschwindigkeit wie folgt darstellen:

$$\boldsymbol{u} = \boldsymbol{u}_\infty + \boldsymbol{u}'. \tag{3.11}$$

Wenn wir die Gl. (3.8) mit $\nabla\times$ multiplizieren, erhalten wir:

$$\nabla \times \boldsymbol{u} = \boldsymbol{\omega} = \underbrace{\nabla \times \nabla\Phi}_{=0} + \nabla \times \nabla \times \boldsymbol{\Psi} \tag{3.12}$$

$$\boldsymbol{\omega} = \underbrace{\nabla(\nabla \cdot \boldsymbol{\Psi})}_{=0} - \nabla^2\boldsymbol{\Psi} \qquad \hat{\boldsymbol{\omega}} = |k|^2\hat{\boldsymbol{\Psi}}_k. \tag{3.13}$$

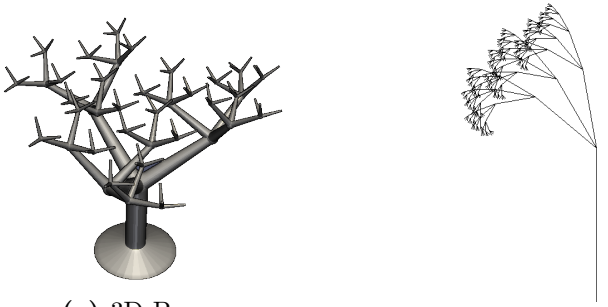
Der Term $\nabla(\nabla \cdot \boldsymbol{\Psi})$ ist Null, da $\boldsymbol{\Psi}$ divergenzfrei ist. Die FOURIERtransfomierte von $\boldsymbol{\omega}$ ist $\hat{\boldsymbol{\omega}}$.

$$\boldsymbol{u}' = -\nabla \times \frac{\boldsymbol{\omega}^2}{\nabla^2} \tag{3.14}$$

Hiermit wurde ein Weg gefunden, aus der Wirbelstärke die Geschwindigkeit zu berechnen.

4 Festkörper

In diesem Kapitel wird ein Überblick gegeben, wie der Baum approximiert wird. Des Weiteren werden die Bewegungsgleichungen für den Baum und deren Lösungen näher erläutert. Der Baum besteht aus einem Stamm und Ästen. Der Stamm und jeder Ast haben die Form eines Zylinders im Dreidimensionalen und die Form eines Rechteckes im Zweidimensionalen (siehe Abb. 2). Die Übergänge vom Baum zum Fluid sollen fließend sein, sodass keine Lücken entstehen, (siehe Abb. 1 auf S. 6).



(a) 3D Baum

(b) 2D Baum

Abb. 2: Varianten von fraktalen Bäumen in 2D und in 3D

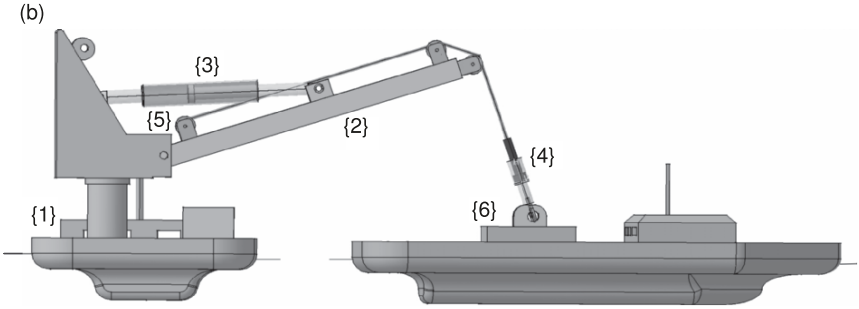
Die erste Überlegung war, die Bewegung des Baumes mithilfe der *Rigid-Finite-Element*-Methode zu beschreiben. Dabei werden die Äste als starre Körper angenommen und zwischen den Ästen werden Drehfedern und Dämpfer eingebaut. Hieraus würde bei linearen Feder- und Dämpferkonstanten, ein nichtlineares partielles Differenzialgleichungssystem zweiter Ordnung entstehen, in dem die Zustandsgrößen (Ort, Geschwindigkeit der Äste) miteinander gekoppelt sind. Die Feder- bzw. Dämpferkonstanten können auch nichtlinear sein. Das Modell könnte erweitert werden, indem der Stamm und die einzelnen Äste in Teilsysteme unterteilt werden. Hierdurch würde der Ast in starre Astabschnitte unterteilt werden. Für den Grenzfall, dass die Anzahl der Teilsysteme gegen Unendlich geht, konvergiert die

Lösung gegen die kontinuierlichen Lösung. Ein Beispiel für die Unterteilung ist dem Buch [24] auf Seite 186 entnommen (siehe Abb. 3). Die interagierenden Systeme in Abb. 3a sind mit geschweifter Klammer {Systemnummer} gekennzeichnet. Zum Beispiel ist der Kranarm mit der Nummer {2} und das Schiff mit der Nummer {6} gekennzeichnet. Diese Systeme werden in Untersysteme unterteilt (siehe Abb. 3b). Ein weites Beispiel für die Unterteilung interagierender Systeme in Untersysteme ist in dem Buch *Dynamics of Tree-Type Robotic Systems* [21] auf Seite 59 zu finden und in Abb. 4 dargestellt. In den Büchern [24], [3], [10] wird die Methode erläutert. Diese Methode wird in der Robotertechnik verwendet, (siehe *Springer Handbook of Robotics* [22]). Die *Rigid-Finite-Element*-Methode wurde aus Zeitmangel nicht weiter verfolgt.

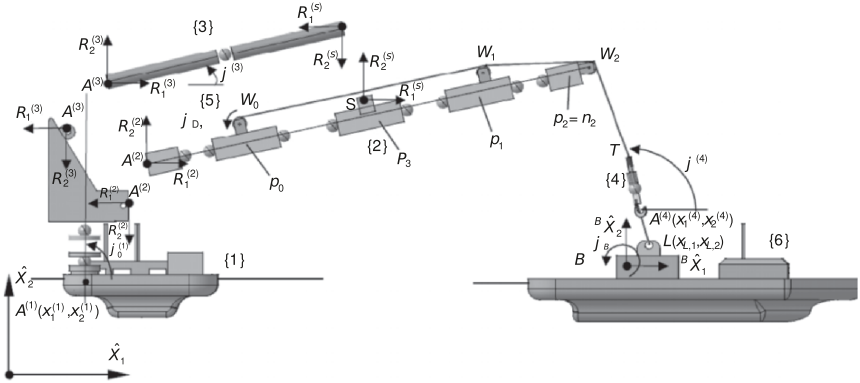
Ein anderer Ansatz wäre, den Baum über Punktmassen, die mit Federn verbunden sind, zu approximieren.

Mit der *Finite-Elemente-Methode* (FEM) könnte der gesamte Baum in einem Stück berechnet werden. Als Basisfunktionen könnten LAGRANGEelemente benutzt werden. Das Materialgesetz könnte linear oder nichtlinear angenommen werden. Da der Baum eine große Verschiebung bezogen auf die Referenzlage haben könnte, bei starkem Wind, ist es wichtig geometrische Nichtlinearität anzunehmen. In der Arbeit [1] ist ein Beispiel mit FEM und der Annahme der geometrischen Nichtlinearität mit der Software *Fenics* für die Verformung eines Balkens berechnet worden. Der Elastizitätsmodul des Baumes (von Holz) ist anisotrop bzw. transversal isotrop.

Das reduzierte Modell beschränkt sich darauf, dass der Baum als starrer Körper angenommen wird, der sich um seinen ersten Punkt $\boldsymbol{x}_0$ drehen kann. An diesem Punkt befinden sich ein Lager, ein Dämpfer und eine Drehfeder. Der Freischnitt des Systems ist in Abb. 8 auf S. 17 dargestellt.



(a) Die interagieren Hauptgruppen (Karnarm, Schiff) werden jeweils mit {} gekennzeichnet. [24] Seite 186



(b) Ein Hauptgruppe wir in Untergruppen unterteilt [24] Seite 186

Abb. 3: Mögliche Unterteilung eines Systems in Hauptgruppen und Untergruppen für die Rigid Finite Element Methode aus [24]

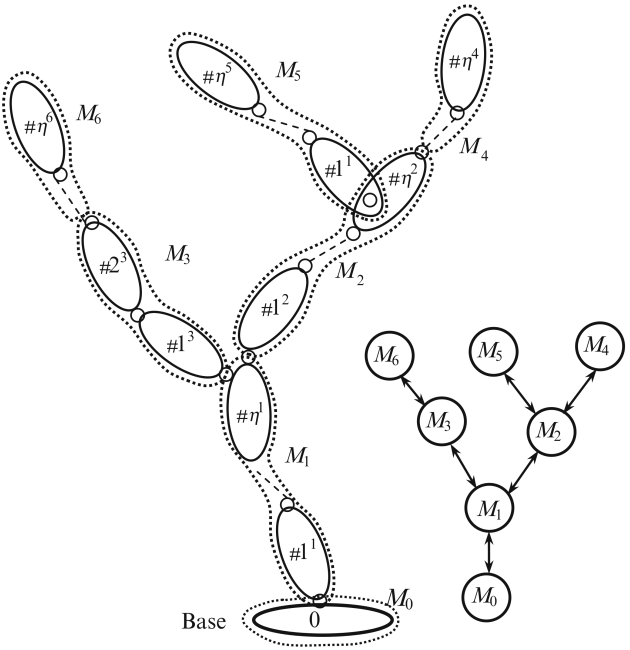
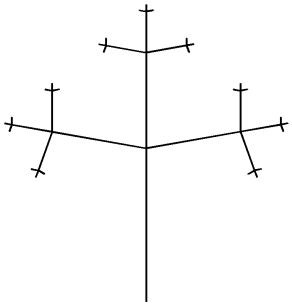


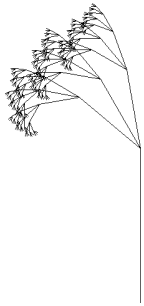
Abb. 4: Beispiel für die Rigid Finit Element Methode. Variante der Zerteilung eines Systems in Untersysteme aus [21] Seite 59

4.1 Fraktaler Baum

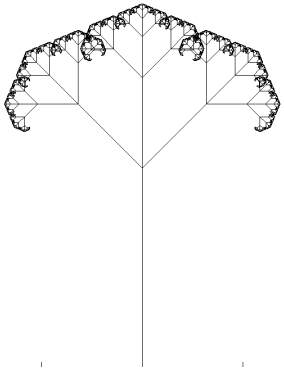
In dieser Arbeit wird der Baum durch einen fraktalen Baum approximiert. In der Abb. 5 sind einige Beispiele für fraktale Bäume



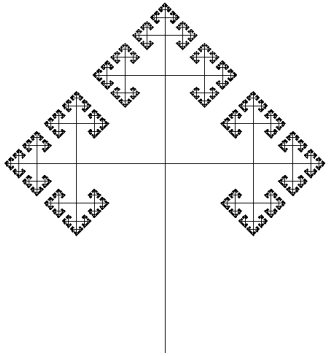
(a) $\#g = 3$, $\alpha = (-45°,0°,45°)$, $r_{\mathrm{h},0} = 0.45$



(b) $\#g = 5$, $\alpha = (50°,30°,10°)$, $r_{\mathrm{h},0} = 0.5$



(c) $\#g = 8$, $\alpha = (-45°,0°,45°)$, $r_{\mathrm{h},0} = 0.45$



(d) $\#g = 8$, $\alpha = (-90°,0°,90°)$, $r_{\mathrm{h},0} = 0.45$

Abb. 5: Beispiele für Variationen von fraktalen Bäumen, hierbei ist $\#g$ die Anzahl der Generationen, α der Winkel der nächsten Astgeneration, und $r_{\mathrm{h},0}$ eine Längenvariable (siehe unten)

gegeben. Das Wort *fractus* stammt aus dem lateinischem und bedeutet gebrochen. Der Vorteil ist, dass der fraktale Baum eine hohe Selbstähnlichkeit aufweist, einem realen Baum ähnlich sieht und leicht zu programmieren ist. Zudem ist der fraktale Baum klar de-

finiert und somit reproduzierbar. Es folgt eine Beschreibung der Eigenschaften und die Herstellung des Baumes.

Die Parameter des Baumes sind: Lage des ersten Punktes des Stammes $\boldsymbol{x}_0$ die Länge und Richtung zum zweiten Punkt, Anzahl der Äste der jeweiligen Generation, die Länge, und Winkel α des jeweiligen Astes.

Die Länge des jeweiligen Astes lässt sich durch eine rekursive Funktion zur vorherigen Generation festlegen. In Abb. 5a sind die Funktionen folgendermaßen gewählt:

$$l_0 = 1 \qquad l_1 = 1 r_{\mathrm{h},0} \tag{4.1}$$

$$l_{j+1} = \frac{l_j r_{\mathrm{h},j}}{(0{,}4j+1)j} \qquad r_{\mathrm{h},j} = \frac{r_{\mathrm{h},j-1}}{j+1} \text{ für } j > 1. \tag{4.2}$$

Hierbei ist l_0 die Länge des Stammes, l_1 die Länge der ersten Generation, l_j die Länge der j-ten Generation und $r_{\mathrm{h},j}$ eine Variable der j-ten Generation. Der Baum in Abb. 5a wurde für die Simulation verwendet, zum einen da er wie ein Baum aussieht und zum anderen, weil die Äste viel Bewegungsfreiraum haben und er sich daher für spätere Arbeiten eignet, in denen die Äste sich bewegen können. Dies wurde in dieser Arbeit nicht realisiert.

Für Abb. 5b bis Abb. 5d sind die Funktionen wie folgt gewählt:

$$l_0 = 1 \qquad l_{j+1} = l_j r_{\mathrm{h},0}. \tag{4.3}$$

Hierbei ist j die Generation und $\hbar$ eine Variable, welche vorher festgelegt wird und in der Regel kleiner als 1 ist.

Der Baum besteht aus einem Hauptstamm von Punkt $\boldsymbol{x}_0$ beginnend bis zum zweiten Punkt $\boldsymbol{P}_2$. Der Hauptstamm hat die charakteristische Länge 1. Der zweite Punkt ist gegeben durch:

$$\boldsymbol{P}_2 = (x_{0x}, x_{0y} + 1)^T. \tag{4.4}$$

Hierbei stehen der Index x und y für die x- und y- Komponente des Vektors. Von diesem Hauptstamm ragen die Äste der ersten Genera-

tion mit einem bestimmten Winkel, bezogen auf die Längsachse des Hauptstammes mit einer bestimmten Länge heraus.

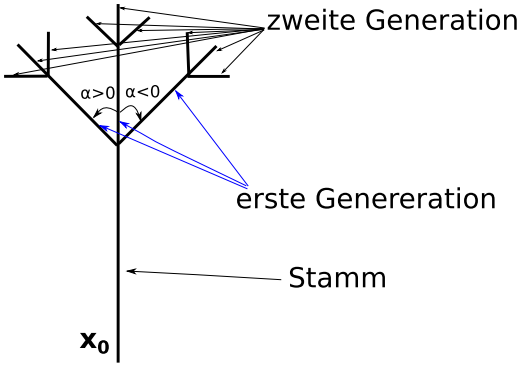


Abb. 6: Darstellung der Baumeigenschaften für Fraktale Bäume

Die in Abb. 6 dargestellten Eigenschaften (Lage vom ersten Punkt $\boldsymbol{x}_0$, Anzahl der Generationen, Länge und Winkel) werden für die Baumherstellungsfunktion benötigt.

In der Baumherstellungs-Funktion wird die Funktion `draw branch` rekursiv aufgerufen. In der Abb. 7 ist eine schematische Darstellung des Programmablaufes zu finden.

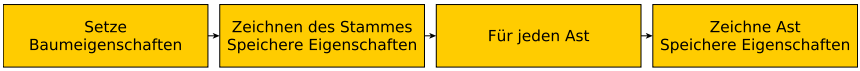


Abb. 7: Schematische Darstellung der Baumherstellungsfunktion

Jeder Ast besteht aus zwei Punkten. Der erste Punkt, ist der Ursprung des Baumes. Der zweite Punkt ist vom ersten Punkt die Länge 1 in y-Richtung entfernt. Die Punkte werden in einem Punktarray abgespeichert. Der Ast, der aus zwei Punkten besteht, wird im Astarray, mit seiner Generations-, und Längeneigenschaft abgespeichert. Von dem Stamm geht die erste Astgeneration los. Für jeden Ast wird ein Winkel definiert, sowie eine Länge. Der für die Simulation verwendete Baum hat die Winkel $\boldsymbol{\alpha} = -80°, 0°, 80°$. Hieraus werden die nächsten Punkte berechnet. Wieder werden die neuen Punkte im

Punktarray abgespeichert und die Äste werden im Astarray abgespeichert, mit der Generations- und Längeneigenschaft. Von jedem Ast der i-ten Generation entstehen neue Äste der $i+1$-ten Generation. Am Ende der Berechnung sind alle Punkte des Baumes und alle Äste des Baumes mit der Generations- und Längeneigenschaft berechnet. Diese Eigenschaften werden benötigt, um im Abschn. 5.1 auf S. 20 die Maske zu bauen und den Linienschwerpunkt $\boldsymbol{x}_\mathrm{s}$ im folgenden Unterabschnitt zu berechnen.

4.2 Bewegungsgleichung des Baumes: Drehimpulssatz

Die Dynamik des Baumes wird mit Hilfe des Drehimpulssatzes (siehe [16]):

$$J_{x_0}\ddot{\varphi} = \sum_i M^i = M_\mathrm{spring} + M_\mathrm{damp} + M_g + M_\mathrm{aero} \tag{4.5}$$

$$M_\mathrm{spring} = -c_\mathrm{spring}\varphi \quad M_\mathrm{damp} = -c_\mathrm{damp}\dot{\varphi} \quad M_g = r\sin(\varphi)mg \tag{4.6}$$

im Zweidimensionalen um $\boldsymbol{x}_0$ bestimmt. Hierbei sind J_{x_0} das Massenträgheitsmoment um den Schwerpunkt $\boldsymbol{x}_0$, φ die Auslenkung des Hauptstammes bezogen auf die Ruhelage, $\dot{\varphi}$ die Winkelgeschwindigkeit des Baumes, m die dimensionslose charakteristische Masse, c_damp Dämpfungskonstante, c_spring Federkonstante, M_spring das Moment, welches durch die Drehfeder verursacht wird, M_damp, welches durch den Dämpfer verursacht wird, M_g, das Moment welches durch die Gewichtskraft verursacht wird und M_aero, das Moment welches durch die Strömung (Fluid) verursacht wird. Die Momente sind in Abb. 8 dargestellt. Der Abstand zum Linienschwerpunkt $\boldsymbol{x}_\mathrm{s}$ ist r. Die Berechnung des Linienschwerpunktes lautet:

$$\boldsymbol{x}_\mathrm{s}(t=0) = (x_\mathrm{s}, y_\mathrm{s})^T, \quad r = \sqrt{x_\mathrm{s}^2 + y_\mathrm{s}^2} \tag{4.7}$$

$$x_\mathrm{s} = \frac{\sum_i x_{s,i} l_{x,i}}{\sum_i l_{x,i}}, \quad y_\mathrm{s} = \frac{\sum_i y_{s,i} l_{y,i}}{\sum_i l_{y,i}} \tag{4.8}$$

Für das Massenträgheitsmoment wird $J = mr^2/3$ angenommen.

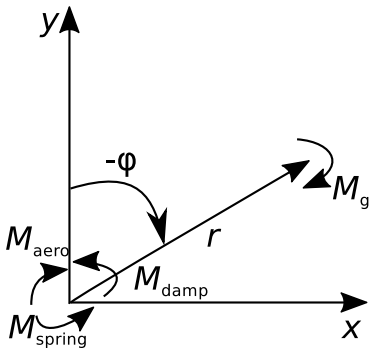


Abb. 8: Freischnitt des Baumes und Darstellung der Momente

In späteren Arbeiten könnte das Massenträgheitsmoment mit Hilfe des Satzes von STEINER genau berechnet werden. Wir nehmen es hier als gegeben Parameter an. Für das Lösen von Gl. (4.5) wird die gewöhnliche Differenzialgleichung zweiter Ordnung auf ein System erster Ordnung reduziert und für die Zeitintegration wird ein explizites EULER-Verfahren erster Ordnung verwendet.

4.3 Geschwindigkeit des Baumes

Die Baumgeschwindigkeitsvariable $\boldsymbol{u}_\mathrm{s}$ beschreibt für jeden Punkt des Baumes die Geschwindigkeit in x- und in y-Richtung. Um die Baumgeschwindigkeit zu berechnen wird wie folgt vorgegangen: Zuerst wird die Winkelgeschwindigkeit $\dot{\varphi}$ aus dem Drehimpulssatz berechnet. Die Winkelgeschwindigkeit ist für jeden Punkt des Baumes gleich, jedoch von der Zeit abhängig:

$$\dot{\varphi} = \dot{\varphi}(t). \tag{4.9}$$

Einmalig muss für jeden Punkt zum einen der Winkel α zwischen der Symmetrielinie des Baumes zum jeweiligen Punkt berechnet werden (siehe Abb. 10). Und zum anderen muss der Abstand r für jeden Punkt des Baumes zwischen $\boldsymbol{x}_0$ und dem jeweiligen Punkt berechnet werden (siehe Abb. 10). Hierfür wird der Baum verschoben, sodass

der erste Punkt im Koordinatenursprung liegt:

$$\boldsymbol{P}_0 = \boldsymbol{P} - \boldsymbol{x}_0. \tag{4.10}$$

Der Winkel wird mit:

$$\alpha_0 = \alpha(\boldsymbol{x}, t = 0) = -\tan\left(\frac{P_{0,x}}{P_{0,y}}\right) \tag{4.11}$$

berechnet. Hierbei ist $P_{0,x}$ die x-Komponente und $P_{0,y}$ die y-Komponente vom Punkt $\boldsymbol{P}_0$.

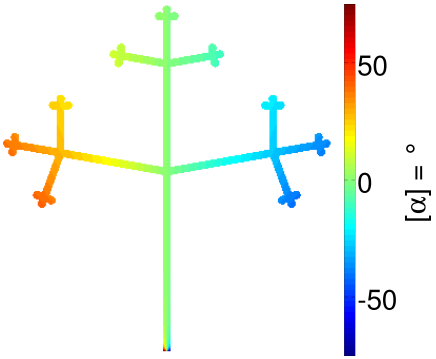


Abb. 9: Winkel α bezogen auf die Symmetrieachse des Baumes

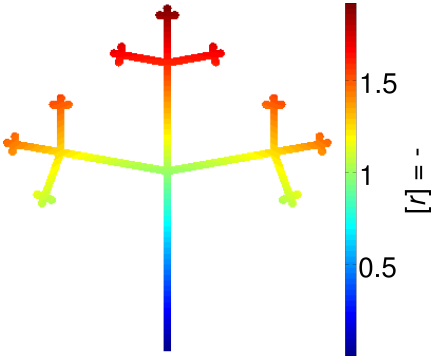


Abb. 10: Darstellung des Radius $r(\boldsymbol{x})$, Abstand zum ersten Punkt $\boldsymbol{x}_0$ zum jeweiligen Punkt des Baumes

Der Abstand berechnet sich durch:

$$r(\boldsymbol{x}) = \sqrt{(P_x - x_{0x})^2 + (P_y - x_{0y})^2}\,. \tag{4.12}$$

Hierbei ist r der Abstand vom Punkt $\boldsymbol{P}$ zum Punkt $\boldsymbol{x}_0$. Die Indizes x und y stehen für die jeweiligen Komponenten des Vektors. Bei einer Rotation des Baumes um den Winkel φ ist der Winkel zwischen der Symmetrieachse zum Punkt gegeben durch:

$$\alpha(\boldsymbol{x}, t) = \alpha_0(\boldsymbol{x}) + \phi(t). \tag{4.13}$$

Danach wird für jeden Punkt die Geschwindigkeit mit:

$$u_{\mathrm{s},x}(\boldsymbol{x},t) = -r(\boldsymbol{x})\dot{\varphi}(t)\cos(\alpha(\boldsymbol{x},t)) \tag{4.14}$$

$$u_{\mathrm{s},y}(\boldsymbol{x},t) = -r(\boldsymbol{x})\dot{\varphi}(t)\sin(\alpha(\boldsymbol{x},t)) \tag{4.15}$$

ausgerechnet.

5 Kopplung von der Strömung und dem Festkörper

In diesem Kapitel wird auf die Kopplung zwischen der Strömung und dem Solid eingegangen. Im ersten Abschnitt werden die Masken und der Sponge näher erläutert. Im zweiten Abschnitt werden die Interaktion der Kräfte und der Momente erklärt. Im letzten Abschnitt wird die Zeitintegrationsmethode näher beleuchtet.

5.1 Masken

Die Maske wird als *mask* oder als χ bezeichnet und ist ein Feld, das für jeden Gitterpunkt einen skalaren Wert hat:

$$\chi(\boldsymbol{x},t) = \begin{cases} 0, & \text{wenn } \boldsymbol{x} \in \Omega_\mathrm{f} \\ 1, & \text{wenn } \boldsymbol{x} \in \Omega_\mathrm{s} \\ ]0,1[& \text{wenn } \boldsymbol{x} \in \Omega_\mathrm{smooth} \end{cases} \tag{5.1}$$

Ziel ist es, die Solidmaske *mask* zu erstellen. Die Gebiete $\Omega_\mathrm{s}, \Omega_\mathrm{f}$ sind in Abb. 1 auf S. 6 dargestellt. Damit der Übergang vom Baum zum Fluid glatter ist, wird eine Smoothing-Layer gesetzt. Dies ist numerisch stabiler. Ohne Smoothing-Layer würde der Baum über die Gitterpunkte sich ruckartig bewegen (siehe aus [13]), da der Baum nur diskret dargestellt ist. Die Motivation ist, die Interaktion zwischen dem Fluid und dem Solid durch den Penalisation-Term in Gl. (3.5) auf S. 6 zu realisieren. Um die Solidmaske *mask* zu erhalten, wird aus der Maske des Bodens *mask_ground* und des Baumes *mask_tree* der jeweilige Maximalwert übernommen. Um die einzelnen Schritte verständlich zu machen, wird im Folgenden Schritt für Schritt die Berechnung der Maske erklärt.

5.1.1 Maske für den Boden

Bei der Bodenmaske sind alle Werte, die vertikal bis zum untersten Punkt des Baumes gehen 1, der Rest ist 0. Hier ist ein Code-Beispiel gegeben:

Listing 1: Code für die Erstellung der Boden Maske

```
groundhigh                          = x0(2);
mask_ground                         = zeros(nx,ny);
mask_ground(Y<=groundhigh)          = 1;
```

5.1.2 Maske für den Baum

Als erstes wird ein Überblick gegeben, wie die Baummaske erzeugt wird. Danach wird die Funktion *twoPointsToMask* genauer erläutert. Und am Ende wird die Handhabung der letzten Astgeneration erklärt.

Zum Überblick: Von dem fraktalen Baum sind alle Punkte, Äste, Längen und die Generation gegeben. Im ersten Schritt wird für einen Ast, der aus zwei Punkten besteht, eine Maske *mask_tree* mit der Funktion *twoPointsToMask* erzeugt. Im zweiten Schritt wird für den nächsten Ast die Maske *mask_i* mit der Funktion *twoPointsToMask* erzeugt. Auf dem Rechengebiet haben wir zwei skalare Felder. Im dritten Schritt werden beide Masken zu einer Maske, indem für jeden Punkt der jeweilige maximale Wert übernommen wird. Der gesamte Baum entsteht, indem die Schritte zwei bis drei für alle weiteren Äste vollzogen werden. Die Schritte eins bis drei sind für die ersten beiden Äste in Abb. 11 von links nach rechts dargestellt.

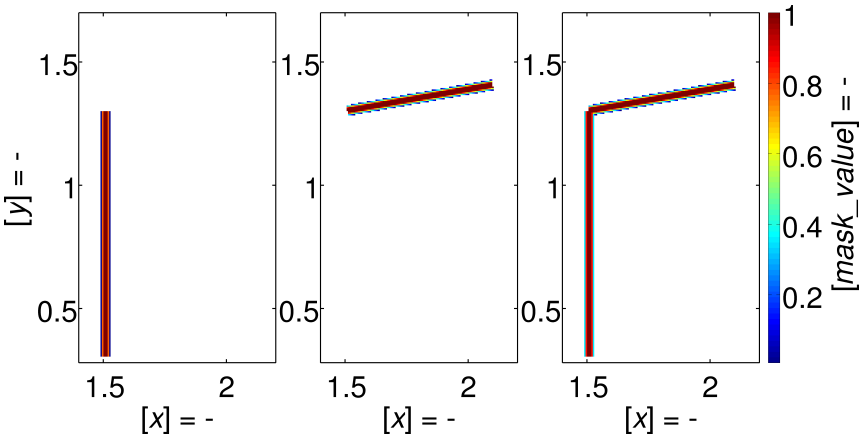


Abb. 11: Kombination von zwei Masken zu einer Maske

Funktion twoPointsToMask Ziel der Funktion *twoPointToMask* ist es, mit dem Input von zwei Punkten ($\boldsymbol{P}_1$, $\boldsymbol{P}_2$), der Länge L_b und der Generation vom Ast als Output die Maske des Astes zu erstellen. Zum Aufbau: Als erstes wird die Idee der Gebietseinteilung näher erläutert. Im Anschluss werden die Bedingungen für das gesamte Gebiet und das lokale Koordinatensystem beschrieben. Am Ende wird auf die Bedingung und auf die Wertzuweisung der Gitterpunkte für die Gebiete innen, oben und unten eingegangen.

Zur Idee der Gebietseinteilung
Jeder Ast hat eine rechteckige Form mit einer konstanten Innenbreite B_i und einer konstanten Smooting-Layer-Breite B_s. Die Innenbreite beträgt 2% der charakteristischen Stammhöhe 1 und die Smoothing-Layer-Breite beträgt $2\Delta x$. Durch die Größen Länge L_b und Breite, innen und außen, lassen sich vier Gebiete festlegen. Das erste Gebiet *Ast_gesamt* ist die gesamte Umrandung des Astes inklusive des Smoothing-Layer. Das zweite Gebiet ist nur innen und wird daher *inside* genannt, das dritte Gebiet ist nur oben und wird daher *above* genannt und das vierte Gebiet ist unten und wird daher *below* genannt. In Abb. 12 sind die Gebiete dargestellt, die Punkte sind mit einem $*$ gekennzeichnet. Ziel ist es, jedem Git-

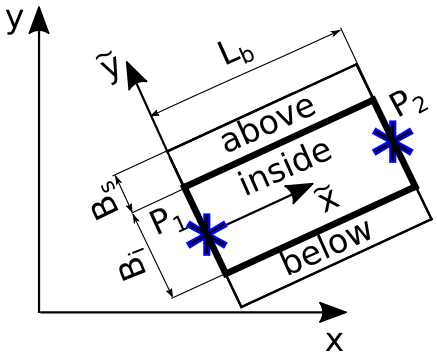


Abb. 12: Gebiet innen, oben und unten

terpunkt im gesamten Maskengebiet einen skalaren Wert zuzuordnen.

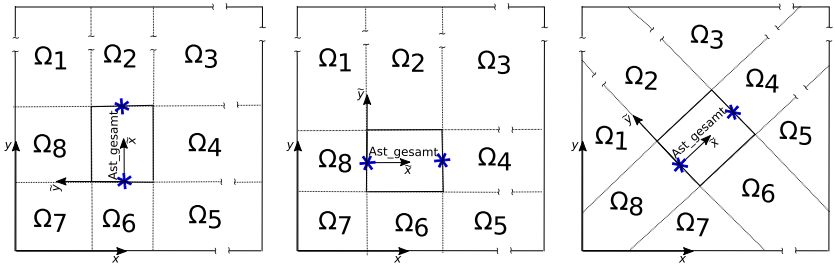
Finden der Bedingung für das gesamte Gebiet *Ast_gesamt*
Im folgenden Schritt wird aus dem gesamten Gebiet Ω der Maske mit $N_x \times N_y$ Punkten die Menge an Punkten $N_{\text{b,gesamt}}$ bestimmt, welche in der gesamten Umrandung des Astes *Ast_gesamt* liegt. Um die Bedingung für das Rechteckgebiet des Astes zu bestimmen, werden vier Grenzen festgelegt: oben, unten, links und rechts. Das Rechteckgebiet muss nicht kollinear mit den Achsen sein. In Lst. 2 ist die Zuweisung für die Punkte des Astes zu finden.

Listing 2: allgemeine Bedingung

```
con_insideAoutside = con_up & con_down & con_right & con_left;
```

Hierfür unterscheiden wir drei Fälle. Im ersten Fall haben beide Punkte die gleiche x-Komponente $P_{1x} = P_{2x}$. Im zweiten Fall haben beide Punkte die gleiche y-Komponente $P_{1y} = P_{2y}$. In beiden dieser Fälle liegen die Rechteckkanten parallel zum Koordinatensystem $x - y$. Der dritte Fall beinhaltet den Rest. In Abb. 13 sind alle Fälle dargestellt.

Im ersten wie im zweiten Fall (siehe Tab. 1) sind die Grenzen Konstanten im dritten Fall sind die Grenzen lineare Funktionen (siehe Lst. 3).



(a) Fall 1 **(b)** Fall 2 **(c)** Fall 3

Abb. 13: Gebiete der drei Fälle und das lokale Koordinatensystem in Richtung des Astes

	Fall 1	Fall 2	Ω
con_up	$Y \leq \max\_y$	$Y \leq P_{1y} + B_g$	4,5,6, 7,8, A
con_down	$Y \geq \min\_y$	$Y \geq P_{1y} - B_g$	1,2,3, 4,8, A
con_right	$X \leq P_{1x} + B_g$	$X \leq \min\_x$	1,2,8, A,6,7
con_left	$X \geq P_{1x} - B_g$	$X \geq \max\_x$	2,3,A, 4,5,6

Tab. 1: Bedingung für das Gesamtgebiet des Astes für den Fall 1 und Fall 2, hierbei ist $B_g = B_i/2+B_s$, $\max\_y = \max(P_{1y},P_{2y})$, $\min\_y = \min(P_{1y},P_{2y})$, $\max\_x = \max(P_{1x},P_{2x})$, $\min\_x = \min(P_{1x},P_{2x})$ und $A = Ast\_gesamt$

Listing 3: Drittes Gebiet

```
if  (P1(1) > P2(1) & P1(2) > P2(2))
        mrem    = P1;
        P1      = P2;
        P2      = mrem;
end
dP = P2-P1;
angle = atan(dP(2)./dP(1));      % in rad
mpp = tan(angle);                % gradient P1 to P2
mspp = -1./mpp;                  % perpendicular on mpp
if P1(2) < P2(2)
        Pymax   = P2;
        Pymin   = P1;
else
        Pymax   = P1;
        Pymin   = P2;
end
% intersection with y-axis
yx0PP           = P1(2) - mpp.*P1(1);
```

```
yx0P_above      = Pymax(2) - mspp.*Pymax(1);
yx0P_below      = Pymin(2) - mspp.*Pymin(1);
% distanz  y-axis yx0PP
hy_inAout       = (B_i./2 + B_s)./cos(angle);
hy_in           = (B_i./2 )./cos(angle);

con_up          = (Y <= mpp.*Xc + yx0PP + hy_inAout);
con_above       = (Y >= mpp.*Xc + yx0PP - hy_inAout);
con_right       = (Y <= mspp.*Xc + yx0P_above);
con_left        = (Y >= mspp.*Xc + yx0P_below);
```

Zur Einführung und Umsetzung des lokalen Koordinatensystems

Jetzt wird für die Punktmenge $N_{\text{b,gesamt}}$ ein lokales Koordinatensystem $\tilde{\boldsymbol{x}} = (\tilde{x},\tilde{y})^T$ eingeführt. Mit dem lokalen Koordinatensystem werden aus der gesamten Punktmenge $N_{\text{b,gesamt}}$ vom *Ast_gesamt* die Punktmenge der anderen drei Gebiete (inside, above, below) bestimmt. Dies hat zum einen den Vorteil, dass die Gebiete und die Smoothing-Layer-Funktion besser bestimmt werden können, und zum anderen, dass die Menge gesuchter Punkte für die anderen drei Gebiete (inside, above, below) deutlich kleiner ist als das gesamte Rechengebiet Ω mit $N_x \times N_y$ Punkten. Somit wird Rechenaufwand eingespart. In allen Fällen muss ein Koordinatenursprung *Proof_org* festgelegt werden. Im ersten Fall wird als Koordinatenursprung für das lokale Koordinatensystem derjenige Punkt von $\boldsymbol{P}_1$ und $\boldsymbol{P}_2$ genommen, der die kleinste y-Komponente hat. In dem zweiten und dem dritten Fall wird hingegen der Punkt genommen, der die kleinste x-Komponente hat. Im zweiten Fall brauchen wir für die Koordinatentransformation nur eine Translation. In Abb. 13 ist das lokale Koordinatensystem dargestellt.

In den anderen Fällen brauchen wir eine Translation und eine Rotation für die Koordinatentransformation. Im ersten Fall ist der Rotationswinkel 90°.

Listing 4: Koordinatentransformation Fall 1

```
Xroof = + Yc(con_insideAoutside) - Proof_org(2);
Yroof = - Xc(con_insideAoutside) - Proof_org(1);
```

Listing 5: Koordinatentransformation Fall 2

```
Xroof = + Xc(con_insideAoutside) - Proof_org(1);
```

```
        Yroof = + Yc(con_insideAoutside) - Proof_org(2);
```

Listing 6: Koordinatentransformation Fall 3

```
Xroof = cos(angle).*(Xc(con_insideAoutside) - Proof_org(1)) + sin(
    angle).*(Yc(con_insideAoutside) - Proof_org(2));
Yroof = -sin(angle).*(Xc(con_insideAoutside) - Proof_org(1)) + cos
    (angle).*(Yc(con_insideAoutside) - Proof_org(2));
```

Untergebiete und Funktionen definieren und Werte schreiben

Vom Baum aus zu seiner Umgebung fällt der Wert χ linear von 1 auf 0 ab. Wie stark der Wert abfällt, hängt von der Smoothing-Layer-Breite ab. In der Simulation ist die Smoothing-Layer-Breite $B_\mathrm{s} = 2\Delta x$ (siehe Abb. 12). In Lst. 7 unter `mfct_below_outside` und `mfct_above_outside` ist die Smoothing-Layer-Funktion definiert.

Listing 7: Gebietseinteilung und Funktionsbestimmung

```
Id2c    = Id1c(con_insideAoutside);
% left to right
con_lr          =       Xroof >= 0 & Xroof <= L_B;
mrange_lr       = (Id2c(con_lr));

% below to above inside
con_abb = Yroof >= - B_i./2 & Yroof <= B_i./2;
mrange_abb      = (Id2c(con_abb));

% area inside
con_inside      = con_lr & con_abb;
mrange_inside   = Id2c(con_inside);
mfct_inside     = 1;

% area outside above
con_above_outside = con_lr &\ldots
                        Yroof <= B_i./2 + B_s & \ldots
                        Yroof >= B_i./2;
mrange_above_outside = Id2c(con_above_outside);
mfct_above_outside = 1 - (Yroof(con_above_outside) - B_i./2)./B_s;

% area below outside
con_below_outside = con_lr &\ldots
                        Yroof <= -B_i./2  & \ldots
                        Yroof >= -B_i./2 - B_s;
mrange_below_outside = Id2c(con_below_outside);
mfct_below_outside = 1 + (Yroof(con_below_outside) + B_i./2)./B_s;
```

Sind die Gebiete und die Funktion mit Zuhilfenahme des lokalen Koordinatensystems zugeordnet worden (siehe Lst. 7), werden die Werte in die Maske geschrieben (siehe Lst. 8).

Listing 8: Werte ins Feld schreiben

```
H_fieldc                        = zeros(length(Xroof),1);
H_fieldc(con_inside)            = mfct_inside;
H_fieldc(con_above_outside)     = mfct_above_outside;
H_fieldc(con_below_outside)     = mfct_below_outside;
F_fieldc                        = H_fieldc;
F_field_c_global                = zeros(nx*ny,1);
F_field_c_global(Id2c)          = F_fieldc;
probeam.F_field_global          = reshape(F_field_c_global,nx,ny);
```

Letzte Ast-Generation und der Kreis Eine Besonderheit gibt es für die Äste der letzten Generation. Da kein weiter Ast folgt, ist der Smoothing-Layer auch in Längenrichtung des Astes erforderlich. Zur Umsetzung: Die Idee ist, am letzten Punkt $\boldsymbol{P}$ des Astes, zur nächsthöheren Generation, die es nicht mehr gibt, einen Kreis zu realisieren. Der Kreis hat drei Gebiete: innen $\Omega_{\text{circle, inside}}$, außen $\Omega_{\text{circle, smooth}}$ und das gesamte Kreisgebiet $\Omega_{\text{circle, all}}$. Das Außengebiet geht vom Innengebiet radial die Smoothing-Layer-Breite nach außen und fällt linear auf 0. Dies ist in Abb. 14 veranschaulicht und lässt sich in Gleichungen aufschreiben als:

$$\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{P})^T(\boldsymbol{x}-\boldsymbol{P})} \tag{5.2}$$

$$\Omega_{\text{circle, all}} = \boldsymbol{r} \leq B_\text{s} + B_\text{i}/2 \qquad \forall \boldsymbol{x} \in \Omega \tag{5.3}$$

$$\Omega_{\text{circle, inside}} = \boldsymbol{r} \leq B_\text{i}/2 \qquad \forall \boldsymbol{x} \in \Omega_{\text{circle, all}} \tag{5.4}$$

$$\Omega_{\text{circle, smooth}} = \boldsymbol{r} > B_\text{i}/2 \qquad \forall \boldsymbol{x} \in \Omega_{\text{circle, all}}. \tag{5.5}$$

Wie bei der Funktion *twoPointsToMask* ist der Vorteil, dass für die Definition des inneren sowie des äußeren Gebietes auf einem kleineren Gebiet gesucht wird. Dies spart Rechenzeit. Die Werte werden für

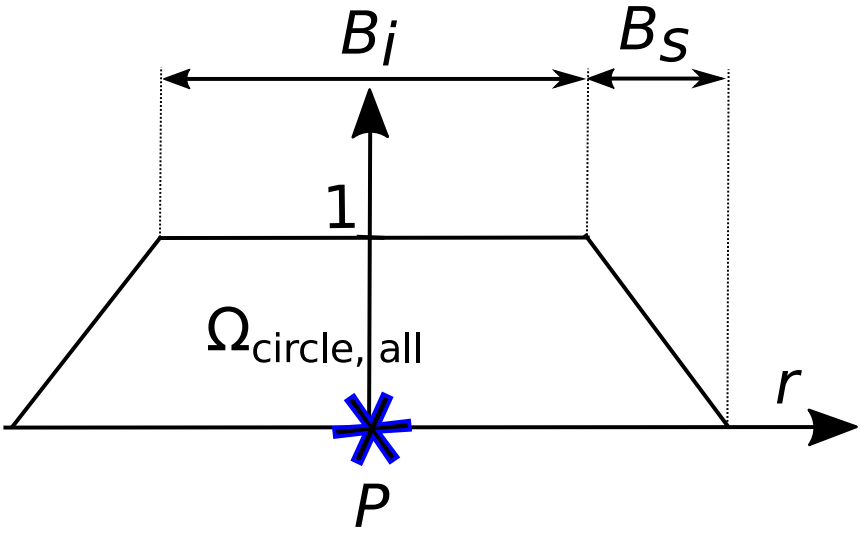


Abb. 14: Kreisfunktion

die *mask_circle* geschrieben:

$$mask\_circle(\boldsymbol{x},t) = \begin{cases} 0, & \text{wenn } \boldsymbol{x} \in \Omega \backslash \Omega_{\text{circle,all}} \\ 1, & \text{wenn } \boldsymbol{x} \in \Omega_{\text{circle, inside}} \\ 1 - \frac{\sqrt{r^2 - B_\text{i}/2}}{B_\text{s}}, & \text{wenn } \boldsymbol{x} \in \Omega_{\text{circle, smooth}}. \end{cases} \tag{5.6}$$

Die Maske *mask* wird gebildet aus der vorherigen Maske *mask* und der Kreismaske, indem der Maximalwert der jeweiligen Masken übernommen wird. In Abb. 15 sind die Masken ohne Kreis, mit Kreis am Ende des Astes und die gesamte Baummaske veranschaulicht.

Anmerkung Es könnte gleich auf einem Untergebiet gesucht werden. Denkbar wäre hier zum Beispiel die Punkte zu nehmen, die in Baumnähe sind. Weiterhin könnten andere Speicherformen für die Maske zum Beispiel *sparse* verwendet werden. Außerdem könnten die schon mit 1 besetzten Punkte in der Maske *mask* aus der noch zu suchenden einzelnen Maske (fehlende Äste und Masken) und

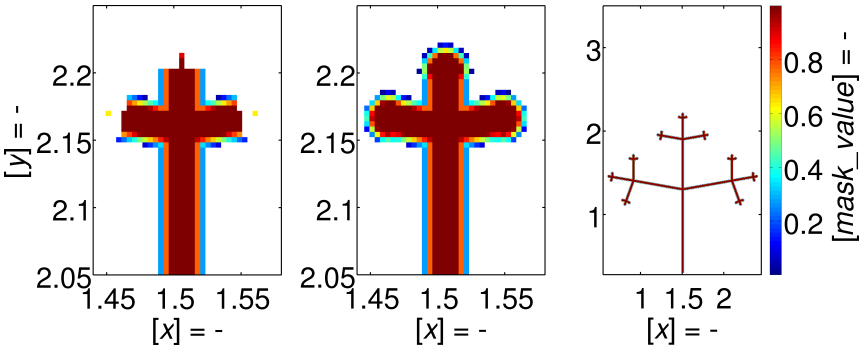


Abb. 15: Enden des Baumes ohne Kreis, mit Kreis und der ganze Baum

der Maximalwertbestimmung herausgenommen werden, da 1 der mögliche Maximalwert ist.

5.2 Sponge

Der Sponge wird verwendet, um den Einfluss der periodischen Randbedingungen zu minimieren. Die periodischen Randbedingungen sind Konsequenz der verwendeten diskreten FOURIER-Transformation. Im physikalischen und FOURIERraum ist das Signal diskret und endlich. Für die diskrete FOURIER-Transformation wird das Signal im physikalischen und FOURIERraum periodisch fortgesetzt [12].

Ohne Sponge würden die Wirbel bzw. die Störung, wenn sie aus dem Rechengebiet hinausschwimmen, in die gegenüberliegende Seite oben und unten bzw. links und rechts hineinschwimmen. Somit ist die Anströmung turbulenzfrei.

Ein idealer Sponge ist unendlich lang und sehr schwach. Die Wirbel dissipieren durch diesen Sponge. Das Rechengebiet ist so kurz wie möglich und so lang wie nötig, da die Rechenleistung und der Speicherplatz begrenzt sind. In der Simulation ist der Sponge-Layer links, rechts und oben im Rechengebiet (siehe Abb. 16). Links und rechts beträgt die Breite `frame_thicknessx` $= 32\Delta x$ und oben ist die Breite `frame_thicknessy` $= 32\Delta y$. Die Sponge-Konstante

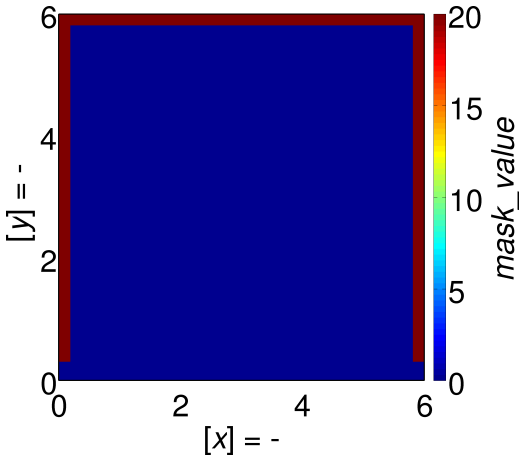


Abb. 16: Sponge mask, χ_{sponge}, wie sie gesetzt wird um die Auslassbedingung zu realisieren

ist $C_{\text{sponge}} = 1/5e - 2$. In Lst. 9 ist das Erstellen der Spongemaske gelistet.

Listing 9: Code Sponge bauen

```
groundhigh = x0(2);% position of trunk is x0
frame_condition =       Y >= Ly - frame_thicknessy |...
                        X <= frame_thicknessx & Y > x0(2) |...
                        X >= Lx-frame_thicknessx  & Y > x0(2) ;
masksponge(frame_condition ) = C_sponge;
```

Die Spongemaske, mit der Variablen χ_{sponge} ist in der Auflösung 1024x1024 in Abb. 16 zu sehen.

Bei dem Sponge handelt sich um einen Penalization Term für die Wirbelstärke, die mit dem Biot-Savart Operator zur Gl. (3.5) auf S. 6 hinzugefügt wird:

$$\partial_t \boldsymbol{u} + \boldsymbol{\omega} \times \boldsymbol{u} = -\nabla \boldsymbol{\Pi} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{F}_p - \frac{\chi}{C\,\eta}(\boldsymbol{u} - \boldsymbol{u}_{\text{s}}) - \nabla \times \frac{\left(\frac{\chi_{\text{sponge}}}{C_{sponge}}(\boldsymbol{\omega} - \boldsymbol{\omega}_0)\right)}{\nabla^2}. \tag{5.7}$$

In der Simulation wird der Sponge beim Lösen der rechten Seite im Runge-Kutta-Verfahren zweiter Ordnung (RK2) verwendet, bevor die Geschwindigkeit im Fourierraum zurückgegeben wird. Hierfür wird die Spongemaske mit der Wirbelstärke multipliziert. Das entstandene Feld wird in den Fourierraum transformiert und daraus die Geschwindigkeit berechnet. Diese Geschwindigkeit wird von der zurückgegebenen Geschwindigkeit vom RK2 subtrahiert (siehe Lst. 10).

Listing 10: Code Sponge

```
% sponge term, if active
if strcmp(vorticity_sponge,'yes')
    vor = vor.*masksponge;
    vortk = fft2(vor);
    nlk = nlk - vor2u(vortk);
end
```

5.3 Kraft und Momentberechnung

Die auf den Baum wirkende Kraft $\boldsymbol{F}$ sei wie folgt definiert (aus [13]):

$$\boldsymbol{F} = \int_{\Omega} \frac{\chi}{C_\eta}(\boldsymbol{u} - \boldsymbol{u}_\mathrm{s})\,\mathrm{d}V + \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_s} \boldsymbol{u}_\mathrm{s}\,\mathrm{d}V. \tag{5.8}$$

Der letzte Term wird vernachlässigt, da der Integrand klein ist. Bei der Simulation mit dem starren Baum (rigid) verschwindet er, da $\boldsymbol{u}_\mathrm{s} \neq \boldsymbol{u}_\mathrm{s}(t)$. Das Moment $\boldsymbol{M}$ ist definiert als

$$\boldsymbol{M} = \boldsymbol{r} \times \boldsymbol{F} \tag{5.9}$$

und der Abstand $\boldsymbol{r}$:

$$\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_\mathrm{c} \tag{5.10}$$

ist der Vektor zwischen dem Ort $\boldsymbol{x}$ der wirkenden Kraft und dem Referenzpunkt $\boldsymbol{x}_\mathrm{c}$. Das Moment sei definiert als:

$$\boldsymbol{M} = \int_{\partial V} \boldsymbol{r} \times (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \,\mathrm{d}\gamma \tag{5.11}$$

$$= \int_V \boldsymbol{r} \times (\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) \,\mathrm{d}V + \int_V \boldsymbol{r} \times \frac{\chi}{C_\eta} (\boldsymbol{u} - \boldsymbol{u}_\mathrm{s}) \,\mathrm{d}V. \tag{5.12}$$

Hierbei ist $\boldsymbol{\sigma}$ die Spannung, ∂V die Umrandung des Gebietes von V. Der erste Term wird vernachlässigt, da der Integrand klein ist. So erhalten wir die Gleichung für das Moment:

$$\boldsymbol{M} = \int_V \boldsymbol{r} \times \frac{\chi}{C_\eta} (\boldsymbol{u} - \boldsymbol{u}_\mathrm{s}) \,\mathrm{d}V. \tag{5.13}$$

5.4 Simulation

In diesem Abschnitt wird auf die Zeitintegration und auf die FSI-Simulation eingegangen.

Der Zeitschritt Δt wird für die Zeitintegrationsmethode benötigt. Bei der Simulation des starren Baumes, auch rigid genannt, ist der Zeitschritt das Minimum aus:

$$\Delta t = \min(t_\mathrm{CFL}, 0{,}95\eta), \qquad (\eta = \Delta x K_\eta)^2 / \nu \tag{5.14}$$

$$t_\mathrm{CFL} = K_\mathrm{CFL} \Delta x / u_{max}, \qquad K_\mathrm{CFL} = 0{,}25 \tag{5.15}$$

$$u_{max} = \max_{\boldsymbol{x} \in \Omega, t = t_i} \left(\sqrt{(\boldsymbol{u}(\boldsymbol{x},t))^T \boldsymbol{u}(\boldsymbol{x},t)} \right). \tag{5.16}$$

Für die FSI Simulation wird noch die Eigenfrequenz des Baumes mitberücksichtigt und wir erhalten:

$$\Delta t = \min(t_\mathrm{CFL}, 0{,}95\eta, t_\mathrm{mech}) \tag{5.17}$$

$$t_\mathrm{mech} < \sqrt{\frac{m}{c_\mathrm{spring}}} \tag{5.18}$$

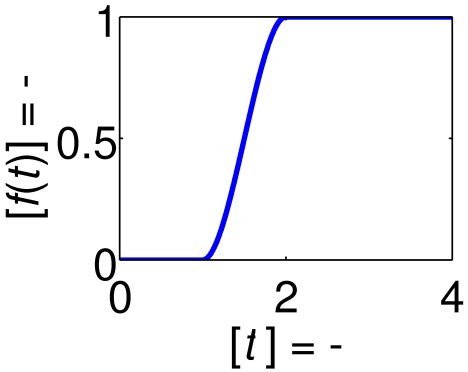


Abb. 17: Anfahrfunktion

Phasen der FSI-Simulation: Die FSI-Simulation unterteilt sich in drei Phasen. In der ersten Phase bis $t = 1$ ist der Baum starr (rigid). In der zweiten Phase und dritten Phase $t > 1$ findet eine Fluid-Interaktion statt, wobei in der zweiten Phase das Moment vom Fluid M_{aero} mit einer Anfahrfunktion multipliziert wird. Die Anfahrfunktion ist definiert als:

$$f(t) = \begin{cases} 0, & \text{wenn } 0 < t < 1 \\ -2(t-1)^3 + 3(t-1)^2, & \text{wenn } 1 < t < 2 \\ 1, & \text{wenn } 2 < t. \end{cases} \tag{5.19}$$

und in Abb. 17 für $t \in [0,4]$ dargestellt.

6 Ergebnisse

In diesem Kapitel wird auf die Ergebnisse der Simulation eingegangen. Betrachte werden die Kraft und das Moment, die zeitlich gemittelte Geschwindigkeit, die turbulente kinetische Energie und die Wirbelstärke.

6.1 Kraft und Moment für den starren Baum (rigid)

In den folgenden Abbildungen von Abb. 18 bis Abb. 25 sind die dimensionslose Kraft (in $x-$, $y-$Richtung), der Betrag der Kraft, und die z-Komponente des Moments für $Re = 100$ und $Re = 600$ und für drei verschiedene Auflösungen Level 1 mit 1536x1536, Level 2 mit 3072x3072 und Level 3 mit 6144x6144 über die dimensionslose Zeit t dargestellt. Die Graphen der jeweiligen Reynoldszahlen verlaufen am Anfang dicht beieinander. Die $Re = 600$ Graphen laufen bei $t = 2$ auseinander, wohingegen die $Re = 100$ Graphen erst bei $t = 4$ auseinander laufen. Dies lässt sich durch die hohe Nichtlinearität und der Steifheit der partiellen Differenzialgleichung (Gl. (3.5) auf S. 6) erklären. Die Phasentrajektorien laufen bei $Re = 600$ auseinander. In der Theorie chaotischer Systeme ist der Begriff des Lyaponov-Exponenten ein Maßstab hierfür.

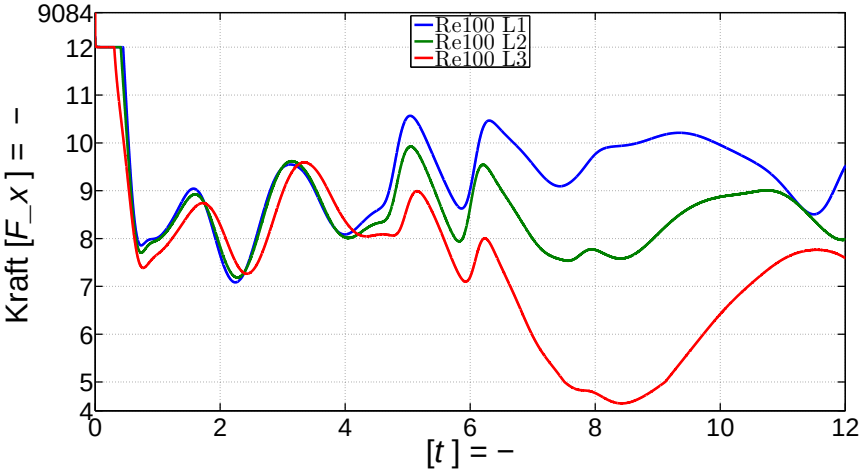


Abb. 18: Die Kraft in x-Richtung, F_x, für den starren Baum bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen

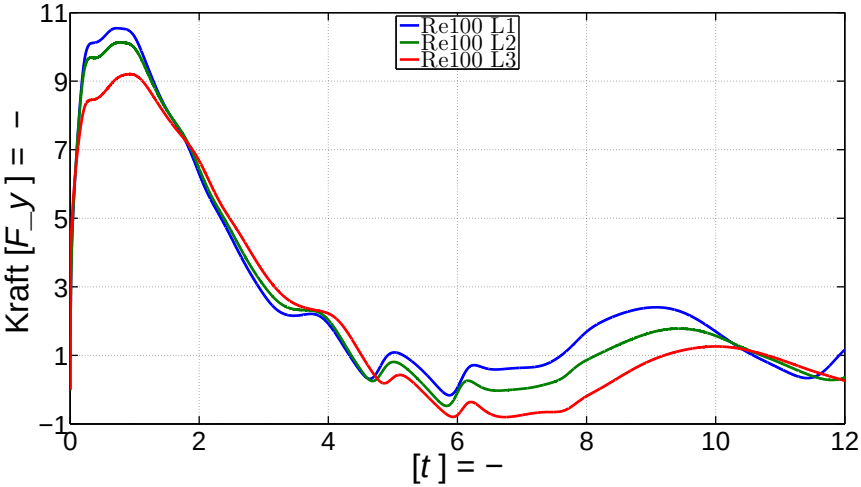


Abb. 19: Die Kraft in y-Richtung, F_x, für den starren Baum bei $Re = 100$.

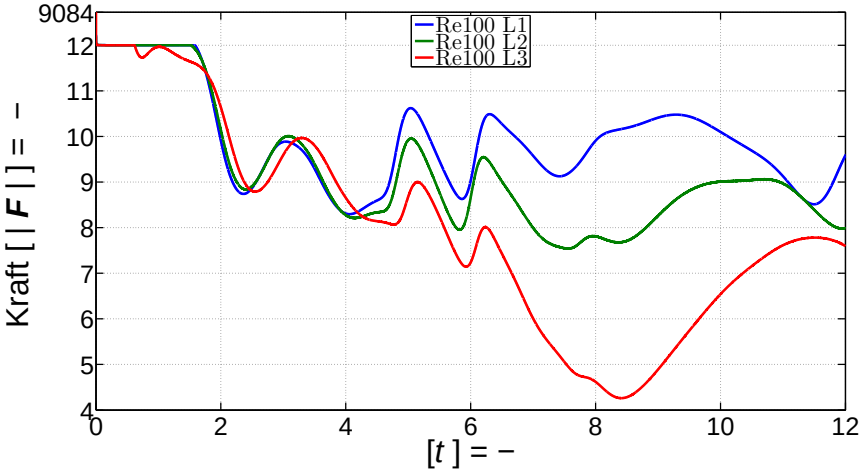


Abb. 20: Der Betrag der Kraft, $|\boldsymbol{F}|$, für den starren Baum bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen

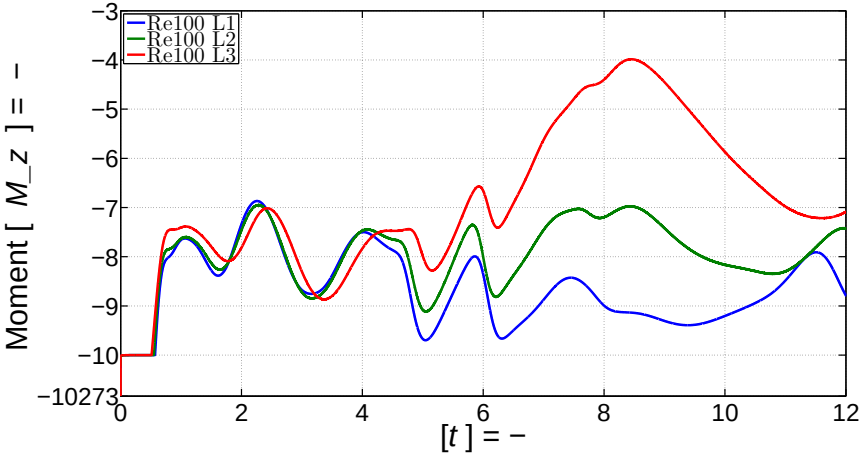


Abb. 21: Die z-Komponente des Moments M_z für den starren Baum bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen

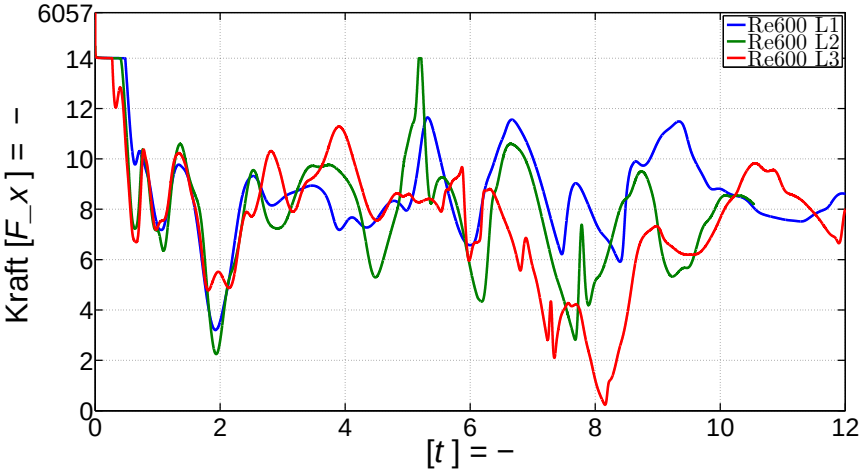


Abb. 22: Die Kraft in x-Richtung, F_x, für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen

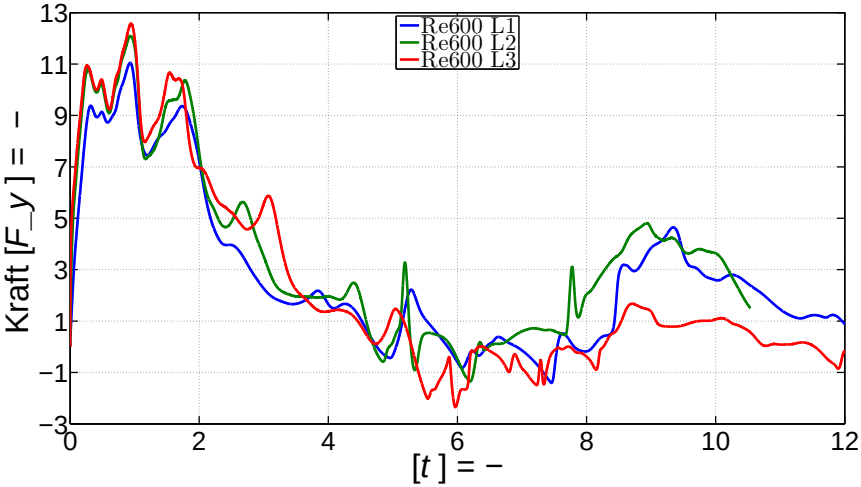


Abb. 23: Die Kraft in y-Richtung, F_y, für den starren Baum bei $Re = 600$.

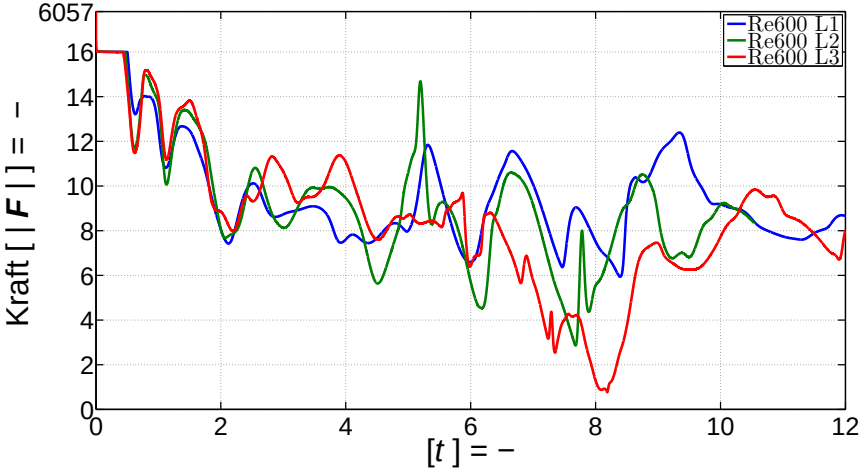


Abb. 24: Der Betrag der Kraft, $|\boldsymbol{F}|$, für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen

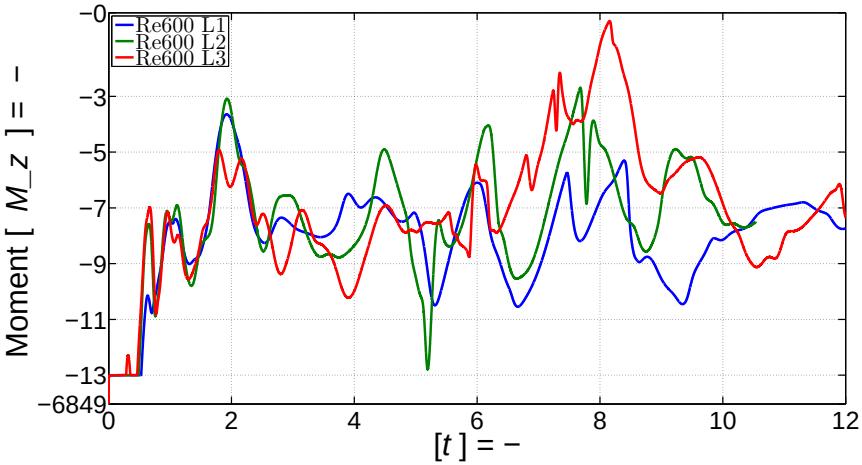


Abb. 25: Die z-Komponente des Moments M_z für den starren Baum bei $Re = 600$, zubeachten y-Achse 2 lineare Skalen

6.2 Kraft und Moment für den bewegten Baum für die FSI

Es folgt die Auswertung der Kräfte und Momente für die FSI-Simulation. In den Grafiken von Abb. 26 bis Abb. 33 sind die jeweiligen Größen über die dimensionslose Zeit t, für die REYNOLDS-Zahl 600 die Auflösung 1024x1024, 512x512, 256x256 und für die REYNOLDS-Zahl 100 die Auflösung 512x512, 256x256, aufgetragen. In den Graphiken ist für die REYNOLDS-Zahle 600 die Auflösung 1024x1024, 512x512, 256x256 und für die REYNOLDS-Zahl 100 die Auflösung 512x512, 256x256 dargestellt.

Drei Bereiche Die FSI-Simulation lässt sich in drei Bereiche einteilen. Im ersten Bereich ($t < 1$) ist der Baum starr (rigid). In der Grafik ist der Bereich mit dunkelgrau gekennzeichnet. Im zweiten Bereich ($1 < t < 2$) findet die Fluid-Struktur-Interaktion statt, hierbei wird das Moment M_{aero} mit einer Funktion beeinflusst (siehe Unterabschn. 5.4 auf S. 32). In der Grafik ist dieser Bereich hellgrau gekennzeichnet. Im dritten Bereich ($t > 2$) findet die Fluid-Struktur-Interaktion statt, ohne Einfluss auf das Moment zu nehmen. Die y-Achse wurde in verschiedene Teilintervalle, die jeweils linear skaliert sind, unterteilt. Die Motivation ist, die Werte für $t > 4$ sichtbar zu machen, da der Bereich vorher unphysikalisch ist.

In allen Graphiken ist keine Konvergenz vorhanden. Da das System chaotisch, sensitiv und steif ist, ist die Konvergenz für $t > 4$ unwahrscheinlich. Gut zu erkennen ist, dass im ersten Bereich (Anfahrbereich) sehr große Werte auftreten können. In den Bereichen zwei bis drei pendeln sich die Werte in einem bestimmten Intervall ein. Die x-Komponente der Kraft, die für den Windwiderstand Drag steht, ist am größten. Das war zu erwarten, da der Baum horizontal angeströmt wird.
Der integrale Wert $\boldsymbol{F}$ hat für große t bei der rigid Simulation für $Re = 600$ nicht konvergiert, daher ist eine Konvergenz bei der FSI-Simulation nicht zu erwarten.
Der Wert schwankt stark bei $Re = 100$ mit der Auflösung 512x512 im zweiten Bereich und pendelt sich später auf einen anderen Wert ein. Hierfür muss noch eine Erklärung gefunden werden.

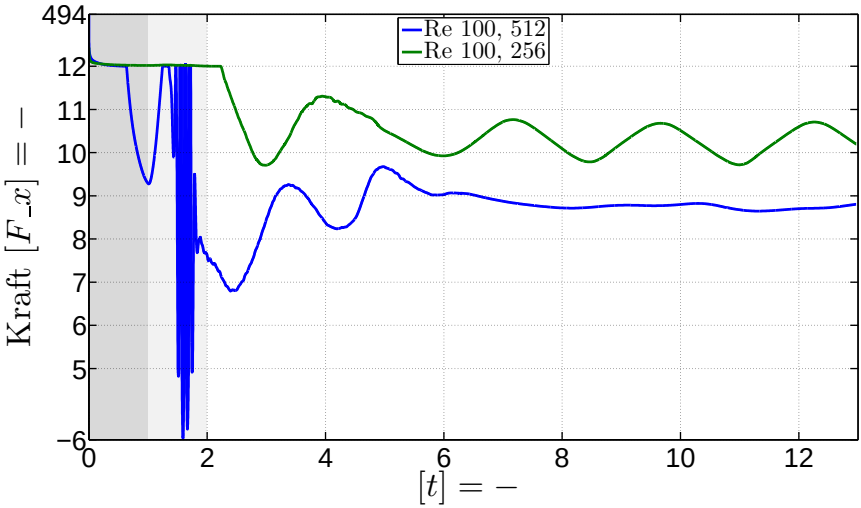


Abb. 26: Kraft in x-Richtung F_x für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen

Die y-Komponente der Kraft (Lift) ist positiv, der Baum hat Auftrieb. Die Geschwindigkeit ist oberhalb des Baumes am Größten, daher ist nach der Gleichung von Bernoulli der Druck niedriger als in der Umgebung. Der Baum wird hochgesaugt.

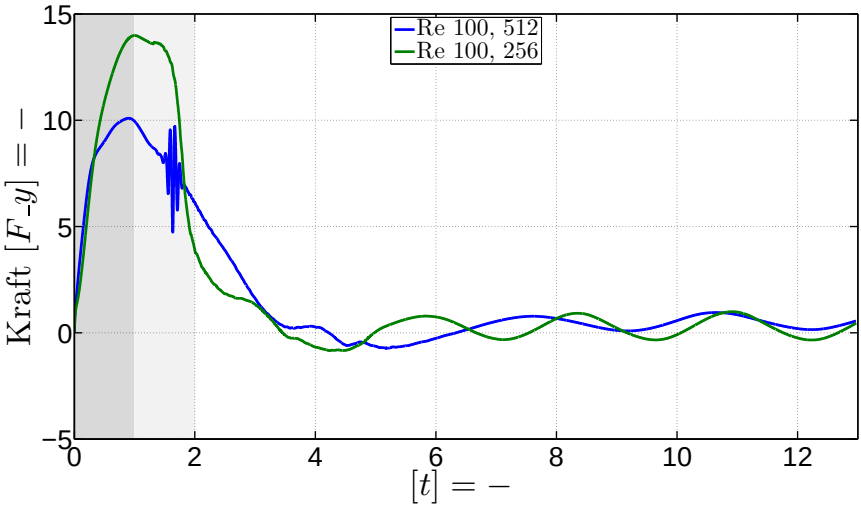


Abb. 27: Kraft in y-Richtung F_y für die FSI-Simulation bei $Re = 100$

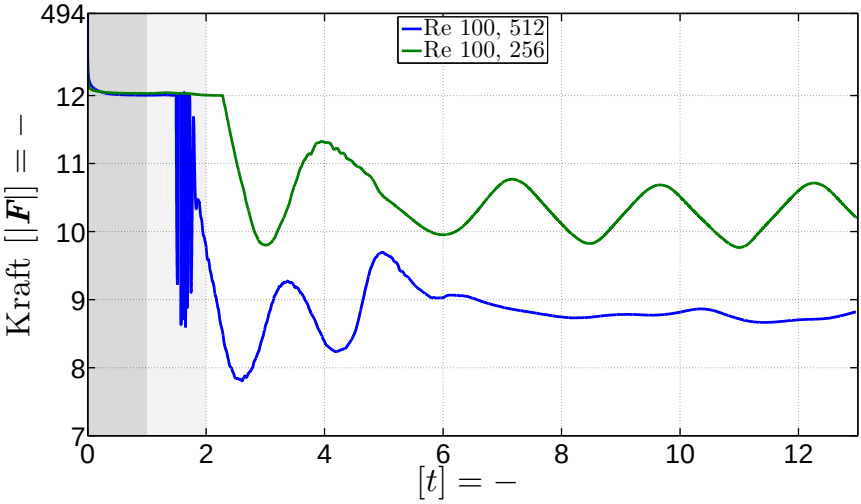


Abb. 28: Betrag der Kraft $|\boldsymbol{F}|$ für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen

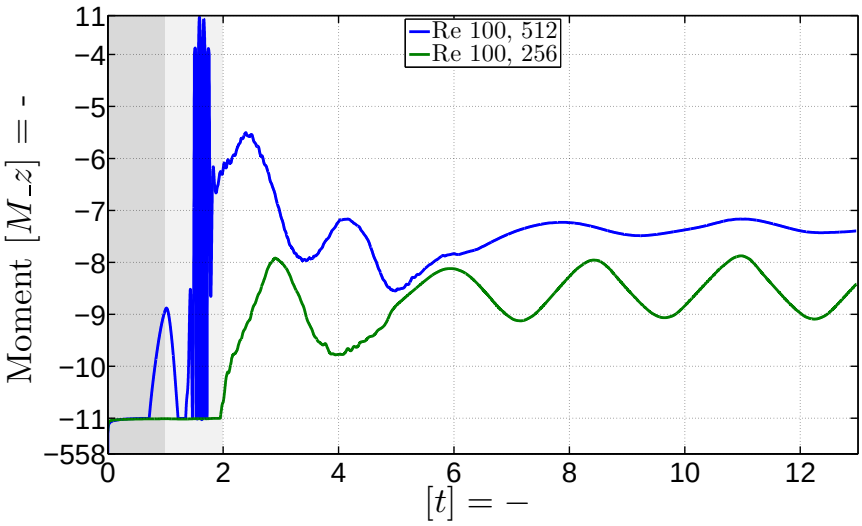


Abb. 29: Die z Komponente des Moments M_z FSI für die FSI-Simulation, zubeachten y-Achse 3 lineare Skalen

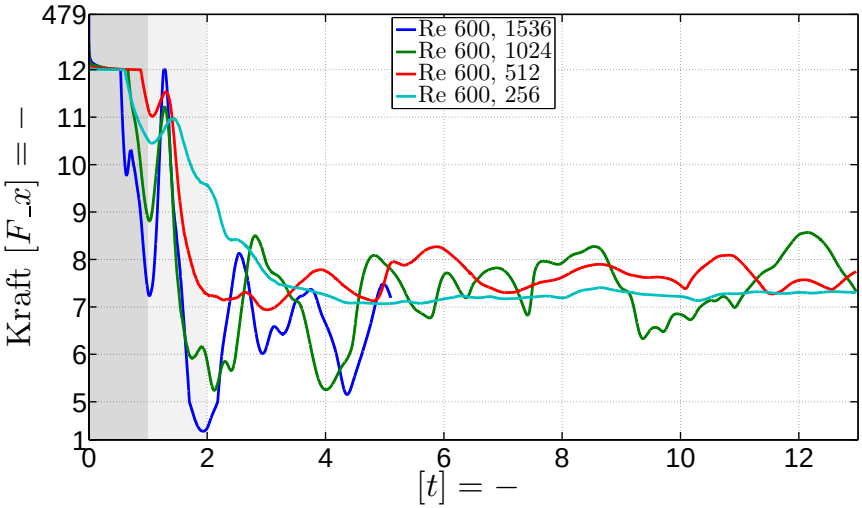


Abb. 30: Kraft in x-Richtung F_x für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 3 lineare Skalen

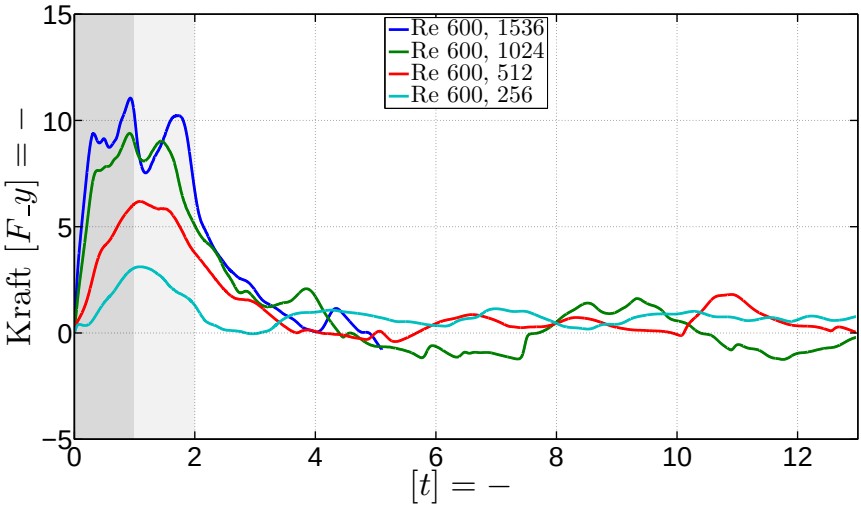


Abb. 31: Kraft in y-Richtung F_y für die FSI-Simulation bei $Re = 100$

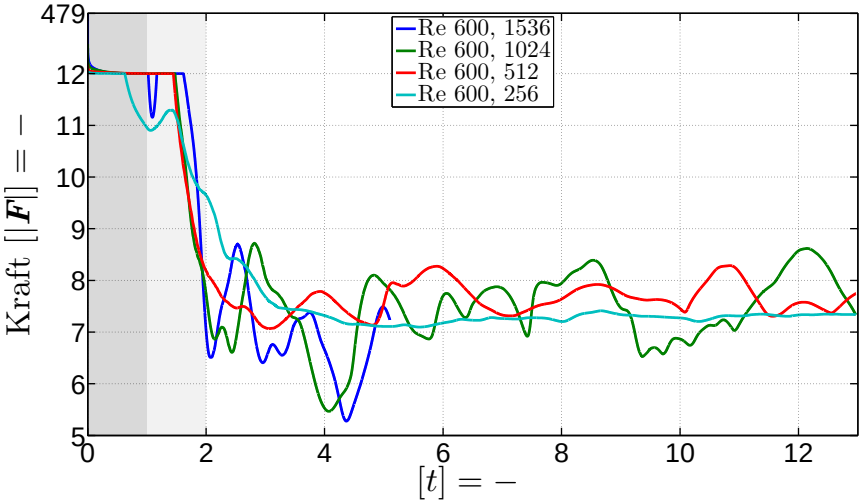


Abb. 32: Betrag der Kraft $|\boldsymbol{F}|$ für die FSI-Simulation bei $Re = 100$, zubeachten y-Achse 2 lineare Skalen

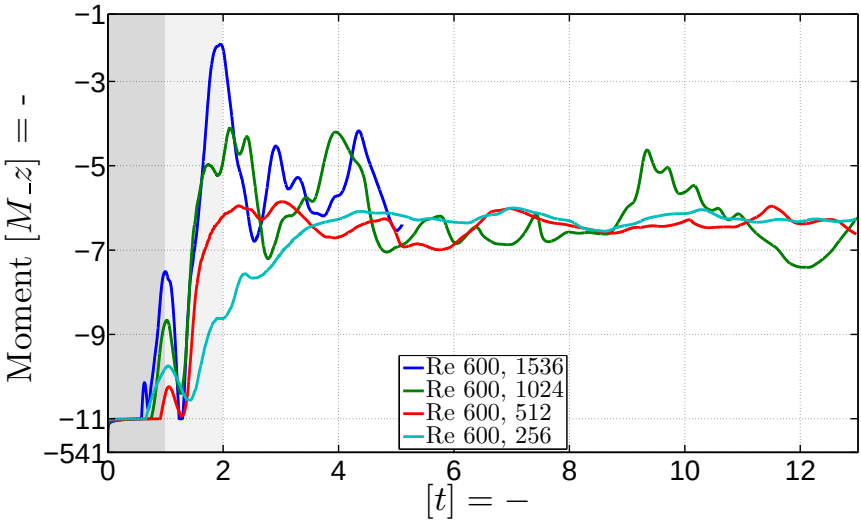


Abb. 33: Die z Komponente des Moments M_z FSI für die FSI-Simulation, zubeachten y-Achse 2 lineare Skalen

6.3 Winkel und Winkelgeschwindigkeit den bewegten Baum (FSI)

Für $0 < t < 1$ ist die Auslenkung des Baumes und die Auslenkungsgeschwindigkeit 0 und wird daher nicht dargestellt. Zur Einteilung in Bereiche und zur Skalierung siehe den vorherigen Abschnitt.

In den Abb. 34 und Abb. 35 ist der Auslenkungswinkel des Baumes φ über der Zeit t dargestellt. Der Baum kippt in Anströmrichtung. Die Auslenkung φ fällt bis $t \approx 6$ und pendelt um ein Wert.
In den Abb. 36 und Abb. 37 ist die Auslenkungswinkelgeschwindikeit des Baumes $\dot{\varphi}$ über die Zeit t dargestellt. Bis $t \approx 1{,}9$ fallen die Graphen und danach wachsen sie. Für $t > 6$ pendeln die Graphen um einen Wert. Der Anstieg der Winkelgeschwindigkeit $\frac{\Delta\dot{\varphi}}{\Delta t}$ im Graphen Abb. 37 ist größer als der Anstieg des Winkels $\frac{\Delta\varphi}{\Delta t}$ im Graphen Abb. 35. Auffallend ist das Schwanken bei dem $\dot{\varphi}$ Graphen $Re = 100$ und der Auflösung $512x512$. Hierfür muss noch eine Erklärung gefunden werden.

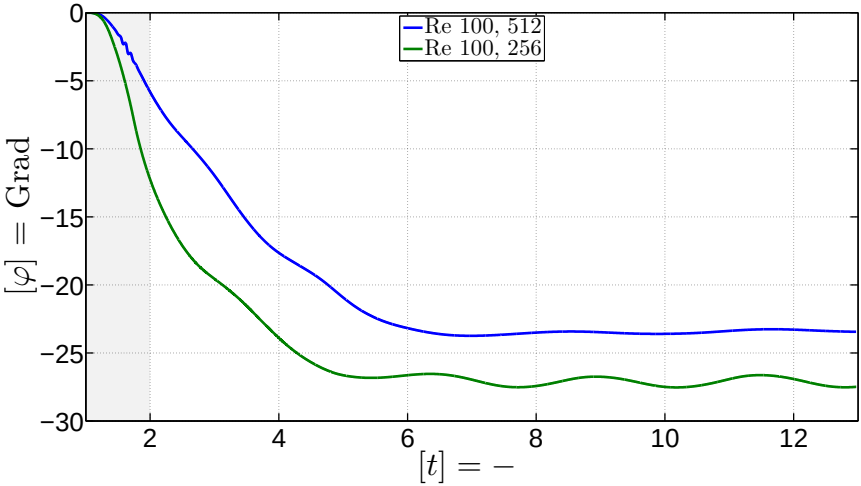


Abb. 34: Der Winkel des Baumes φ für die FSI-Simulation bei $Re = 100$

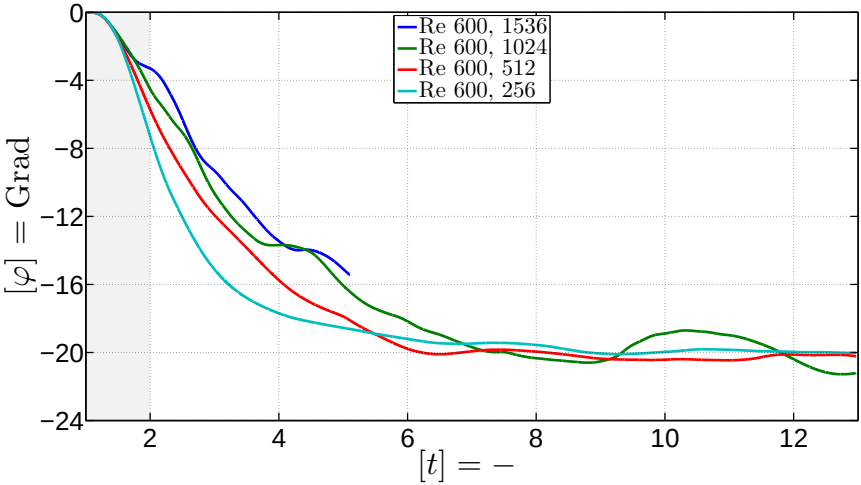


Abb. 35: Der Winkel des Baumes φ für die FSI-Simulation bei $Re = 600$

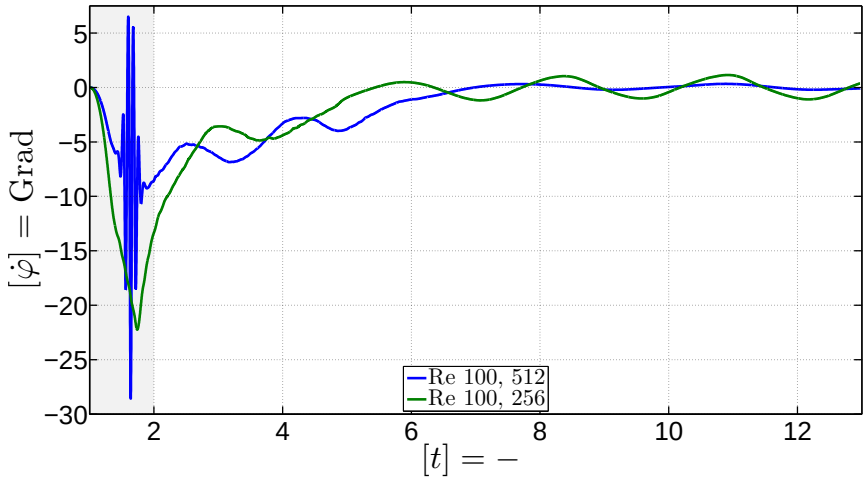


Abb. 36: Der Winkelgeschwindigkeit des Baumes $\dot{\varphi}$ für die FSI-Simulation bei $Re = 100$

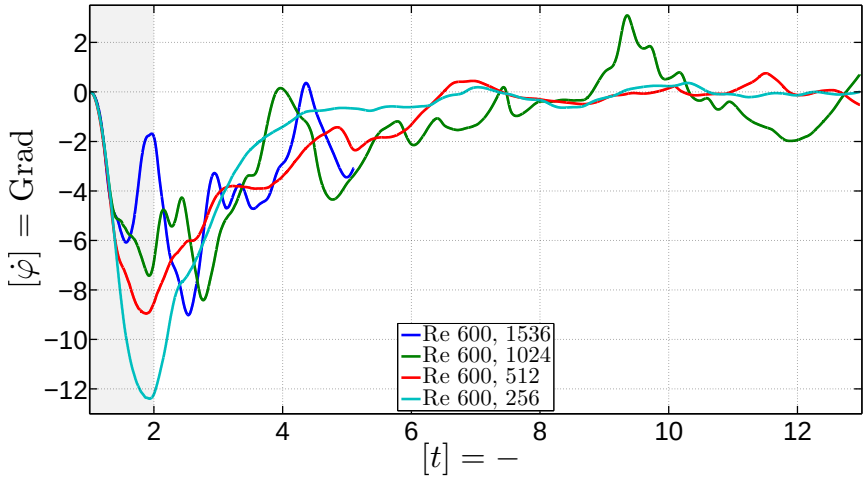


Abb. 37: Der Winkelgeschwindigkeit des Baumes $\dot{\varphi}$ für die FSI-Simulation bei $Re = 600$

6.4 Zeitlich gemittelte Geschwindigkeitsfelder

In diesem Kapitel wird zum einen die Berechnung der zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ erläutert und zum anderen wird die Simulationsauswertung für diese Größe diskutiert.

Die Geschwindigkeit $\boldsymbol{u}$ lässt sich aufteilen in eine mittlere Geschwindigkeit $\boldsymbol{U}$ und ein eine Schwankungsgeschwindigkeit. $\boldsymbol{u}'$:

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{U}(t) + \boldsymbol{u}'(\boldsymbol{x},t). \tag{6.1}$$

Die mittlere Geschwindigkeit berechnet sich aus der zeitlichen Mittlung der Geschwindigkeit:

$$\boldsymbol{U}(t) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \boldsymbol{u}(\boldsymbol{x},t)\,\mathrm{d}t \tag{6.2}$$

$$\approx \frac{1}{T} \sum_i \boldsymbol{u}_i(\boldsymbol{x},t_i)\Delta t_i \qquad \Delta t_i = t_i - t_{i-1} \tag{6.3}$$

Die Geschwindigkeit wurde über das Zeitintervall $t \in [6, 12]$ gemittelt, da vorher der Anfahrprozess ist.

In den Abbildungen von Abb. 39 bis Abb. 49 sind von dem zeitlich gemittelten Strömungsfeld der Betrag $|\boldsymbol{U}|$, die x-Komponente U_x und die y-Komponente U_y dargestellt. Beim starren Baum, der als rigid bezeichnet wird, sind die Auflösungen 1536x1536, 3072x3072 und 6144x6144 dargestellt. Für den beweglichen Baum (FSI) sind für die Reynoldszahl $Re = 600$ die Auflösungen 1024x1024, 512x512, 256x256 und für die Reynoldszahl $Re = 100$ die Auflösungen 512x512, 256x256 dargestellt. Es folgt eine qualitative Beschreibung für dieser Abbildungen.

Qualitativ ist die zeitlich gemittelte Geschwindigkeit in allen Abbildungen gleich. Der Betrag der zeitlich gemittelten Geschwindigkeit ist oberhalb des Baumes am größten. Durch die periodischen Randbedingungen, haben wir oben auch einen Boden. Der Baum steht in einem Kanal (siehe Abb. 38). Somit ist der Querschnitt an der Position des Baumes kleinsten. Die Geschwindigkeit ist hier am

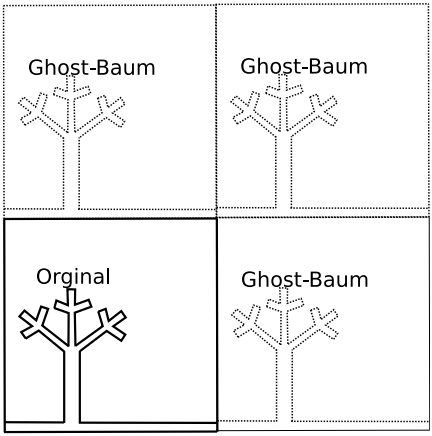
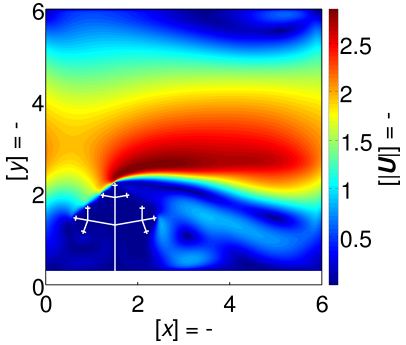


Abb. 38: Durch die periodischen Randbedingungen ist hinterm Rechengebiet (links) ein Baum, auch Ghost-Baum genannt und oben vom Rechengebiet eine Wand.

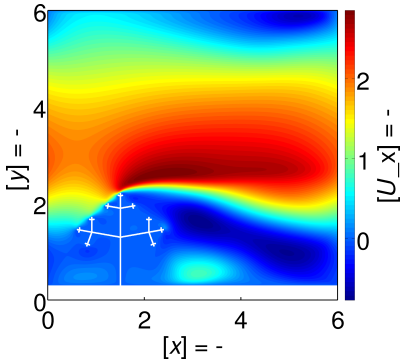
größten, da die Kontinuitätsgleichung gilt und der Massenstrom konstant ist. Der Betrag der zeitlich gemittelten Geschwindigkeit ist im unteren Rechengebiet, außer bei $Re = 600$ 256x256, klein. Unten ist der Boden und wir haben eine noslip-Randbedingung, dadurch bildet sich eine Grenzschicht aus. Oben im Rechengebiet ist die Geschwindigkeit auch klein, da wir oben wegen der periodischen Randbedingungen auch einen Boden haben (siehe Abb. 38). Bei der Ausnahme $Re = 600$ 256x256 könnte es sein, dass die Maske eine Lücke zwischen dem Boden und dem Baum hat und daher der Baum unterströmt werden kann.

Links oberhalb des Baumes strömt es nach oben wegen des *Ghost*-Baumes. Ursache hierfür sind die periodischen Randbedingungen (siehe Abb. 38). Vor und hinter dem Baum ist der Betrag der zeitlich gemittelten Geschwindigkeit klein. Da das Fluid vor dem Baum aufgestaut wird und sich hinterm Baum der Nachlauf befindet. Die negative x-Komponente des zeitlich gemittelten Geschwindigkeitsfeldes U_x deutet auf den Nachlauf und auf das Aufstauen der Strömung vor dem Baum hin. Die y-Komponente des zeitlich gemittelten Geschwindigkeitsfeldes U_y verdeutlicht zum einen die Umströmung in

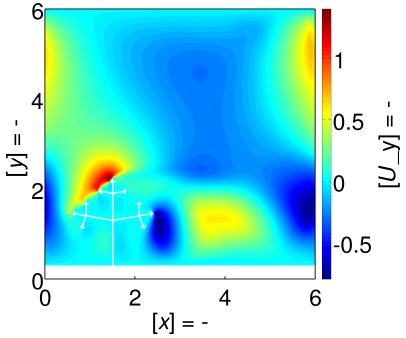
Baumnähe und zum anderen den Einfluss der periodischen Randbedingungen (*Ghost*-Baum, siehe Abb. 38). Daher ist am unteren Ende des Rechengebietes eine negative vertikale Strömungsbewegung zu verzeichnen.



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

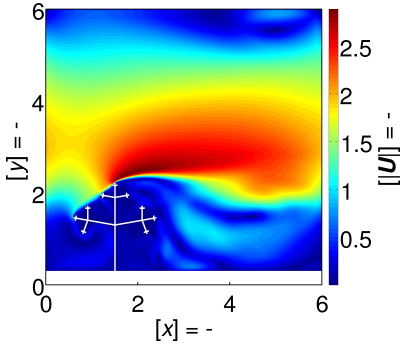


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

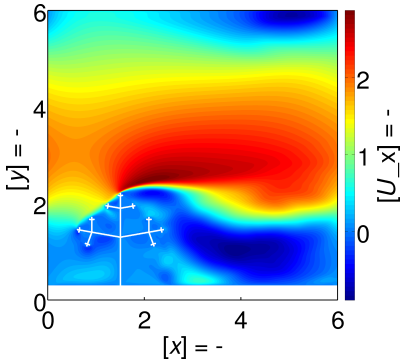


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

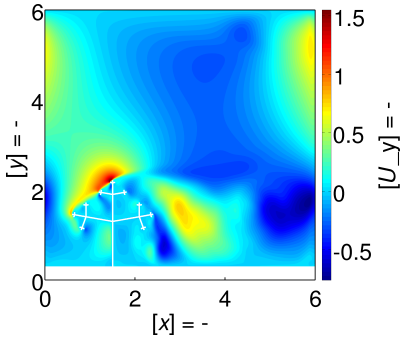
Abb. 39: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1536 für $Re = 100$



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

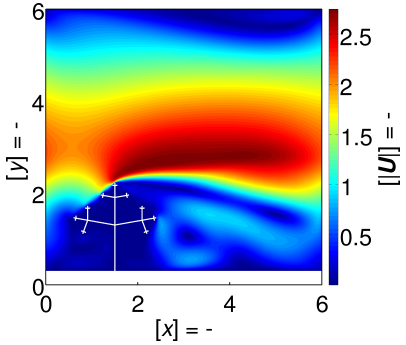


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

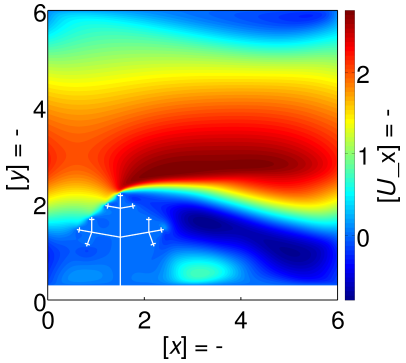


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

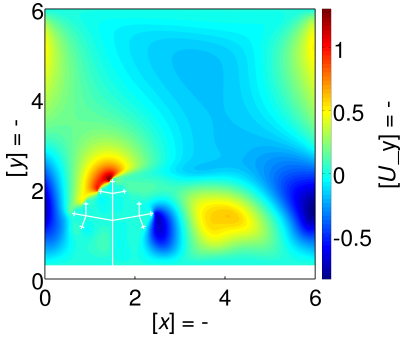
Abb. 40: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1563 für $Re = 600$



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

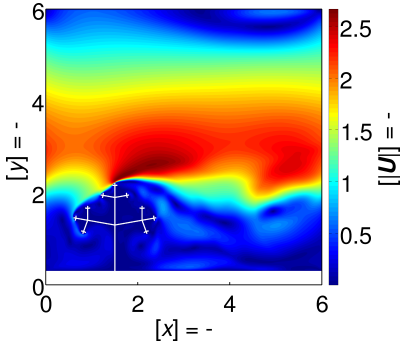


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

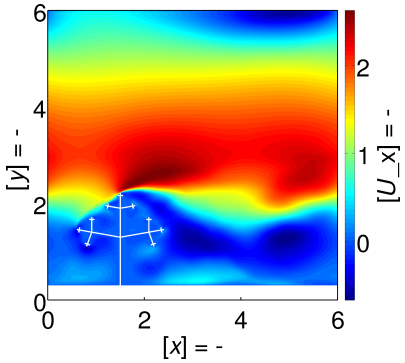


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

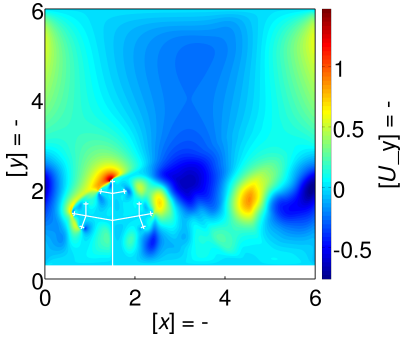
Abb. 41: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 3072x3072 für $Re = 100$



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

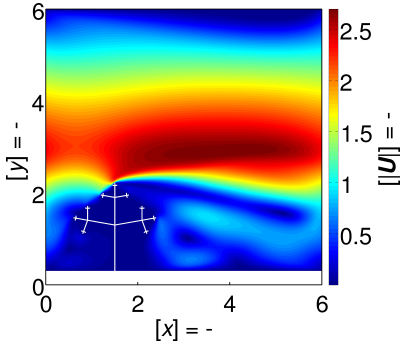


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

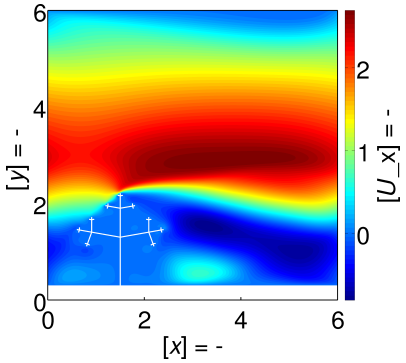


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

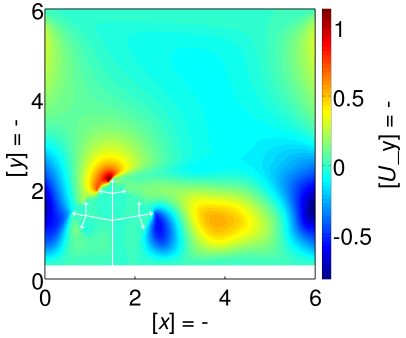
Abb. 42: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 3072x3072 für $Re = 600$



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

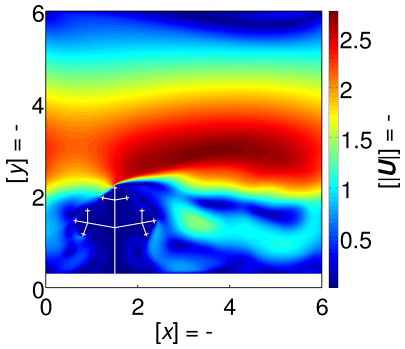


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

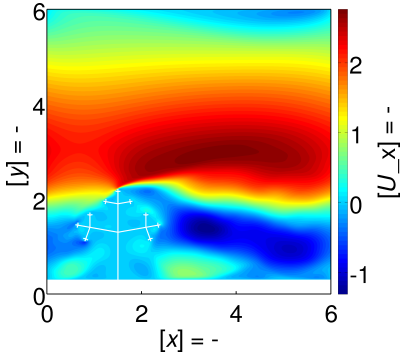


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

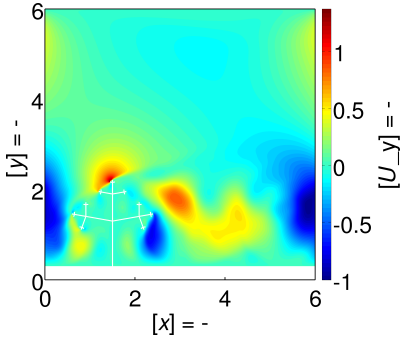
Abb. 43: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 1536x1536 für $Re = 100$



(a) Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

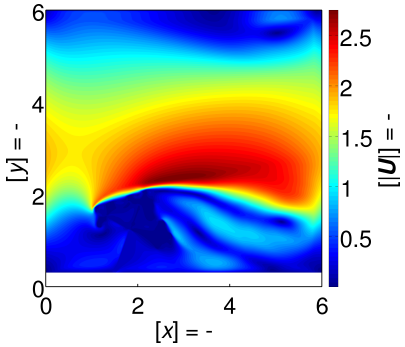


(b) x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

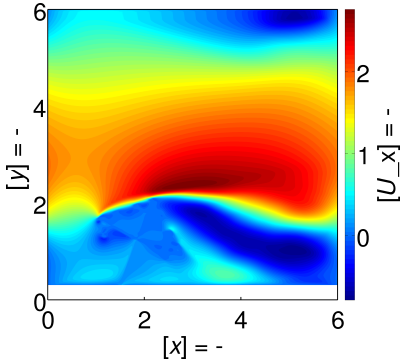


(c) y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

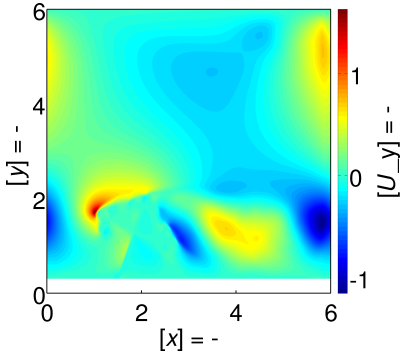
Abb. 44: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ für die starre Baumsimulation und der Auflösung 6144x6144 für $Re = 600$



(a) Der Betrag der zeitlich gemittelte Geschwindigkeit $|\boldsymbol{U}|$

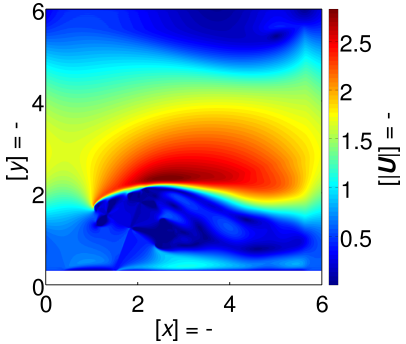


(b) Die x-Komponente der zeitlich gemittelte Geschwindigkeit U_x

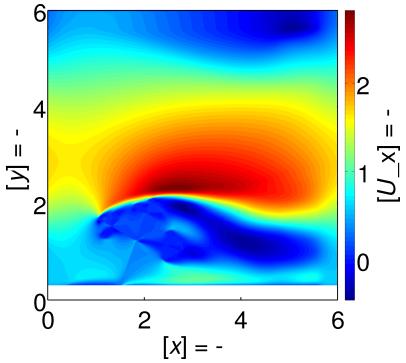


(c) Die y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

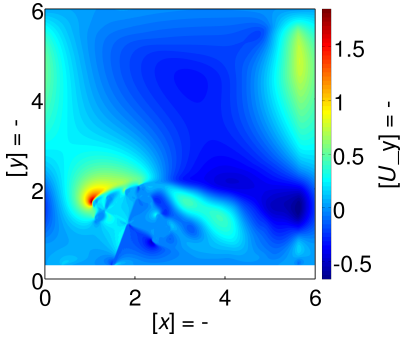
Abb. 45: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 1024x1024 für $Re = 600$



(a) Der Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

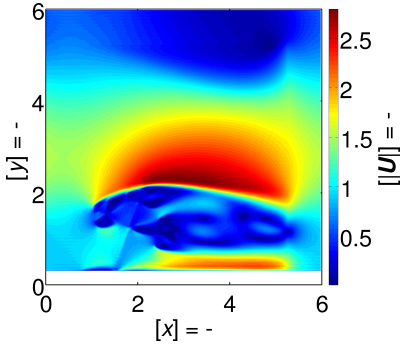


(b) Die x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

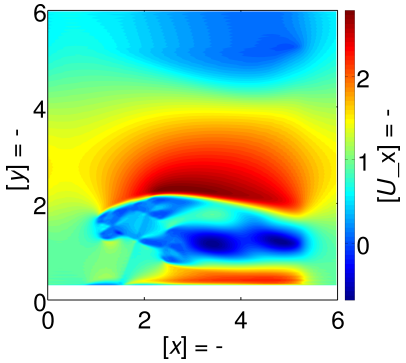


(c) Die y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

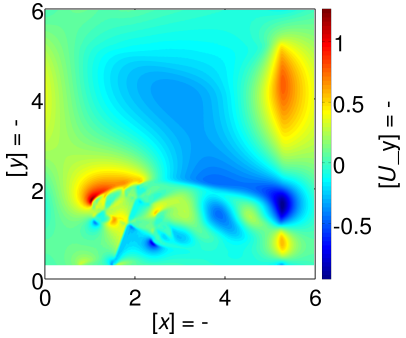
Abb. 46: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 512x512 für $Re = 600$



(a) Der Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

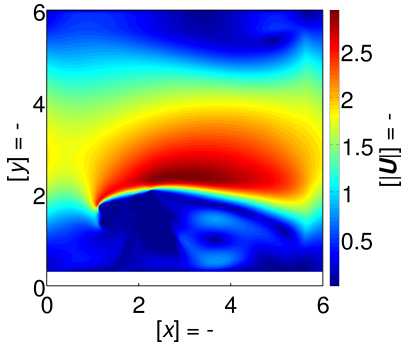


(b) Die x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

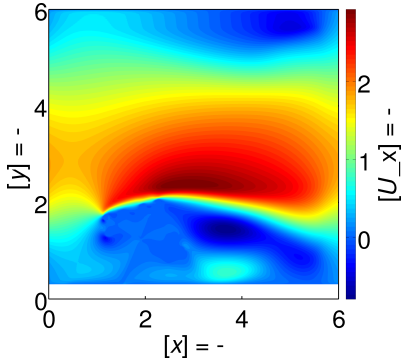


(c) Die y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

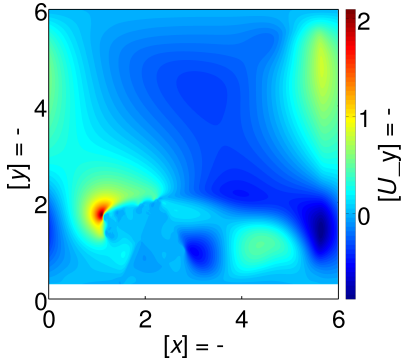
Abb. 47: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 256x256 für $Re = 600$



(a) Der Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$

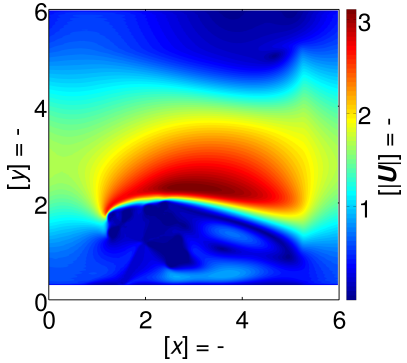


(b) Die x-Komponente der zeitlich gemittelten Geschwindigkeit U_x

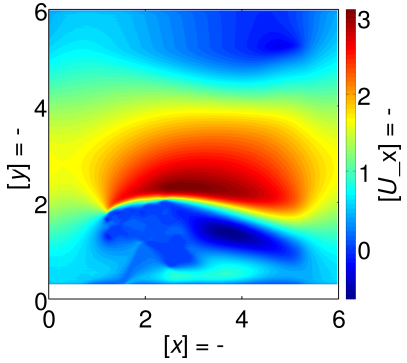


(c) Die y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

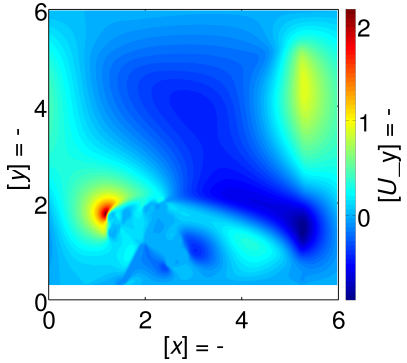
Abb. 48: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 512x512 für $Re = 100$



(a) Der Betrag der zeitlich gemittelten Geschwindigkeit $|\boldsymbol{U}|$



(b) Die x-Komponente der zeitlich gemittelten Geschwindigkeit U_x



(c) Die y-Komponente der zeitlich gemittelten Geschwindigkeit U_y

Abb. 49: Die zeitlich gemittelten Geschwindigkeit $\boldsymbol{U}$ bei der FSI-Simulation und der Auflösung 256x256 für $Re = 100$

6.5 Turbulente kinetische Energie

Zuerst wird die Berechnung der turbulenten kinetischen Energie k erklärt und danach die turbulente kinetische Energie aus den Simulationen beschrieben. Die turbulente kinetische Energie k berechnet sich aus:

$$k(\boldsymbol{x}) = \frac{1}{2}\overline{u_i' u_i'} = \frac{1}{2}\overline{(\boldsymbol{u}'(\boldsymbol{x},t))^T \boldsymbol{u}'(\boldsymbol{x},t)} \tag{6.4}$$

$$= \frac{1}{2}\overline{(\boldsymbol{u}(\boldsymbol{x},t))^T \boldsymbol{u}(\boldsymbol{x},t)} - \frac{1}{2}\left((\boldsymbol{U}(\boldsymbol{x}))^T \boldsymbol{U}(\boldsymbol{x})\right) \tag{6.5}$$

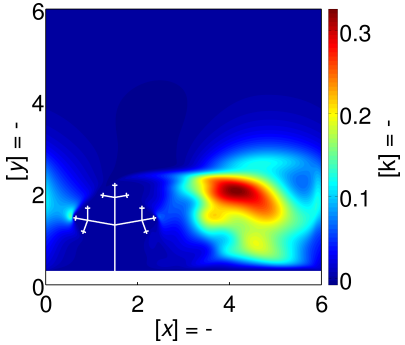
$$\overline{e}_{\text{kin}}(\boldsymbol{x}) = \frac{1}{2}(\overline{(\boldsymbol{u}(\boldsymbol{x},t))^T \boldsymbol{u}(\boldsymbol{x},t)}) \tag{6.6}$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_0^T e_{\text{kin}}(\boldsymbol{x},t)\,\mathrm{d}t \approx \frac{1}{T}\sum_i e_{\text{kin}}(\boldsymbol{x},t_i)\Delta t_i \tag{6.7}$$

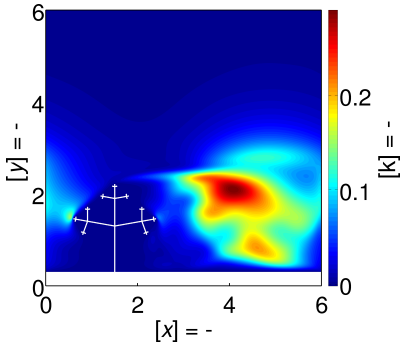
$$e_{\text{kin}}(\boldsymbol{x},t) = \frac{1}{2}\,(\boldsymbol{u}(\boldsymbol{x},t))^T \boldsymbol{u}(\boldsymbol{x},t). \tag{6.8}$$

Hierbei ist $e_{\text{kin}}(\boldsymbol{x},t)$ die kinetische Energie und $\overline{e}_{\text{kin}}$ die zeitliche gemittelte kinetische Energie. Es wurde über das Zeitintervall $t \in [6, 12]$ gemittelt. In Abb. 50 bis Abb. 53 ist die turbulente kinetische Energie dargestellt. Beim starren Baum sind die Auflösungen 1536x1536, 3072x3072 und 6144x6144 dargestellt. Für den beweglichen Baum bei der $Re = 600$ sind die Auflösungen 1024x1024, 512x512, 256x256 und der $Re = 100$ die Auflösungen 512x512, 256x256 dargestellt.

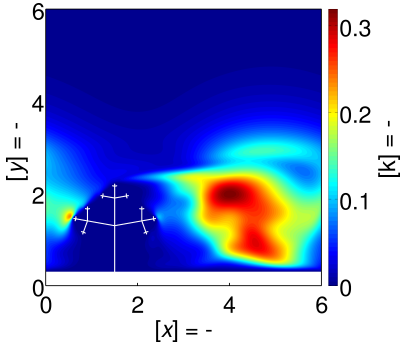
Qualitativ ist die Verteilung der turbulenten kinetischen Energie in den Abbildungen gleich. Quantitativ sind Unterschiede zu verzeichnen, gerade für die FSI bei $Re = 600$. In allen Fällen befindet sich die maximale turbulente kinetische Energie im Nachlauf des Baumes. Die maximale turbulente kinetische Energie ist bei $Re = 600$ größer als bei $Re = 100$. Beim starren Baum (rigid) Level 3 und $Re = 600$ ist auch ein großer Wert vor dem Baum zu verzeichnen. Dies liegt womöglich an den periodischen Randbedingungen.



(a) Turbulente kinetische Energie k, bei der Auflösung 1536x1536

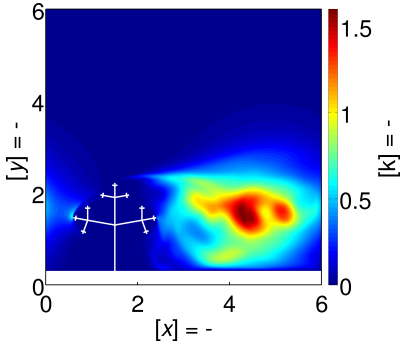


(b) Turbulente kinetische Energie k, bei der Auflösung 3072x3072

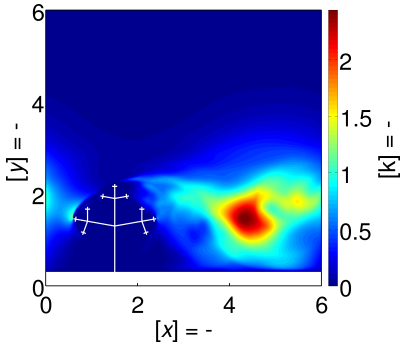


(c) Turbulente kinetische Energie k, bei der Auflösung 6144x6144

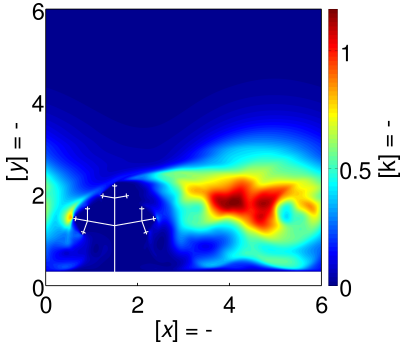
Abb. 50: Turbulente kinetische Energie k, für die starre Baumsimulation und für verschiedene Auflösungen bei $Re = 100$



(a) Turbulente kinetische Energie k, bei der Auflösung 1536x1536

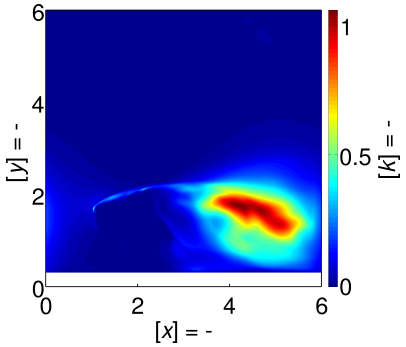


(b) Turbulente kinetische Energie k, bei der Auflösung 3072x3072

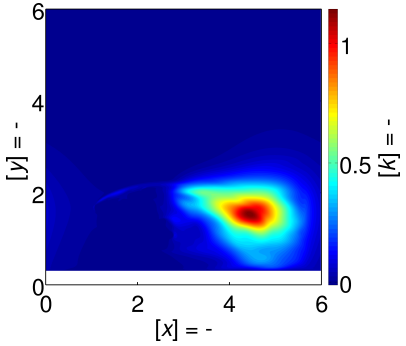


(c) Turbulente kinetische Energie k, bei der Auflösung 6144x6144

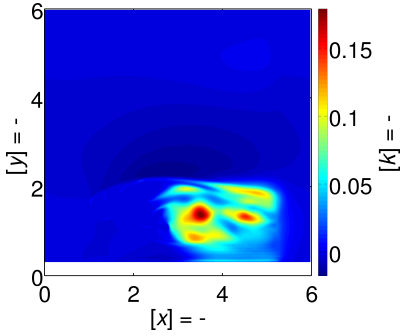
Abb. 51: Turbulente kinetische Energie k, für die starre Baumsimulation und für verschiedene Auflösungen bei $Re = 600$



(a) Turbulente kinetische Energie k bei der Auflösung 1024x1024

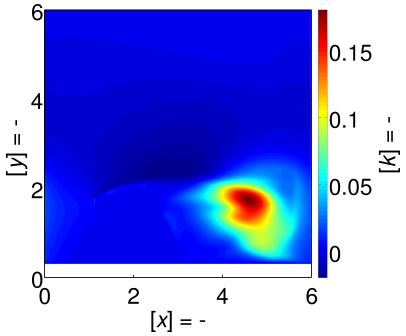


(b) Turbulente kinetische Energie k bei der Auflösung 512x512

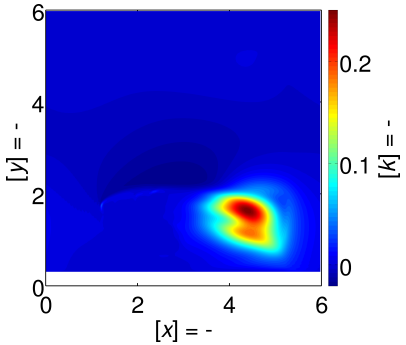


(c) Turbulente kinetische Energie k bei der Auflösung 256x256

Abb. 52: Turbulente kinetische Energie k für die FSI-Simulation, bei $Re = 600$



(a) Turbulente kinetische Energie k bei der Auflösung 512x512



(b) Turbulente kinetische Energie k bei der Auflösung 256x256

Abb. 53: Turbulente kinetische Energie k für die FSI-Simulation, bei $Re = 100$

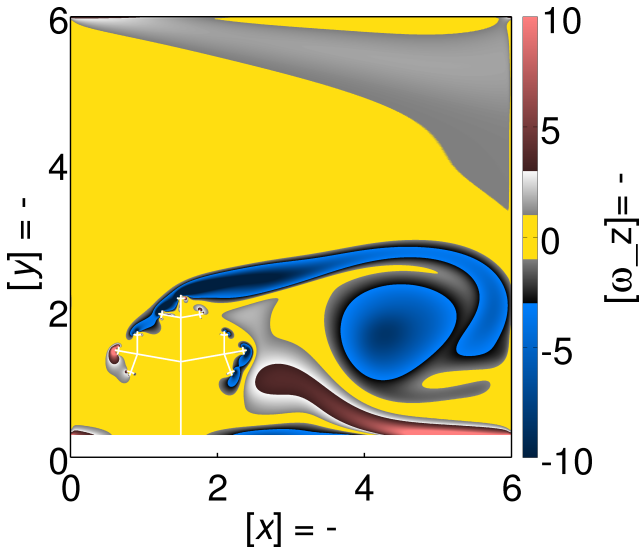
6.6 Wirbelstärke

Für die Zeit $t = 6$ ist die z-Komponente der Wirbelstärke ω_z im Feld in Abb. 54 bis Abb. 55 dargestellt. In diesen Abbildungen ist der Betrag der Wirbelstärke hinter dem Baum (Nachlauf) am größten. Dies deckt sich mit der Auswertung der turbulenten kinetischen Energie, welche im Nachlauf am größten war. Die Wirbel schwimmen vom Baum ab und durch den Nachlauf durch.

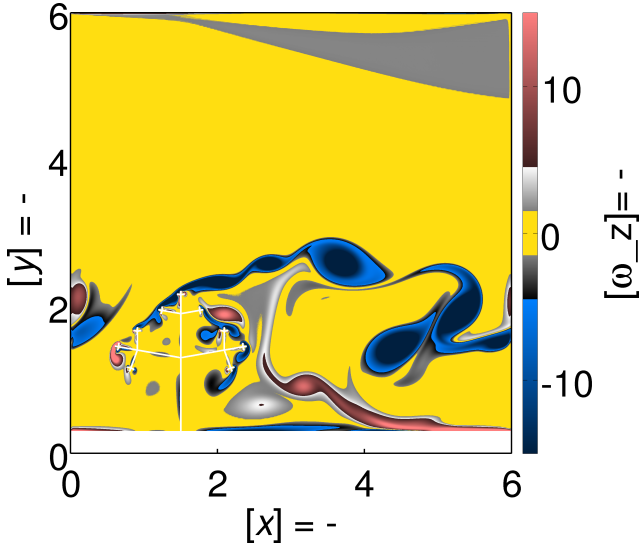
Für die Wirbelidentifikation gibt es kein eindeutiges Kriterium. Unter der Annahme, dass ein Wirbel bzw. eine kohärente Struktur durch eine große Wirbelstärke gekennzeichnet ist, befinden sich im Nachlauf Wirbel. Diese Wirbel drehen im Uhrzeigersinn, wenn sie blau sind und gegen den Uhrzeigersinn, wenn sie rot sind. Der Vollständigkeit halber werden noch die Wirbelkriterien Druckminimum, Geschwindigkeitsbasierendekriterien: Q-Kriterium, λ_2-Kriterium, und Δ-Kriterium, erwähnt [12].

Für die rigid-Simulation bei einer Reynolds-Zahl von 600 und einer Auflösung 6144x6144 ist in der Abb. 54b erkennbar, dass Störungen ins Rechengebiet hineinschwimmen.

In der Abb. 56 und Abb. 57 ist die Wirbelstärke für die Simulation $Re = 600$ für die Auflösung 1024x1024 für verschiedene Zeiten dargestellt, um das Abschwimmen der Wirbel nachvollziehbar zu machen. Um die Achsen und die Farbskala zuzuordnen siehe Abb. 55b.

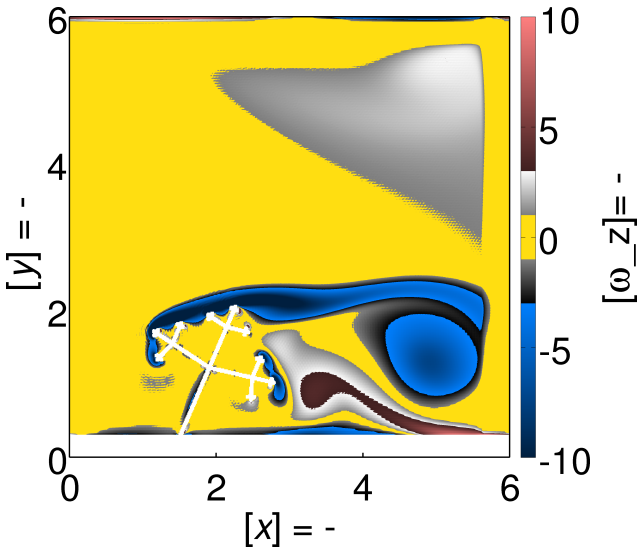


(a) Wirbelstärke ω_z bei der Auflösung 6144x6144 und bei $Re = 100$

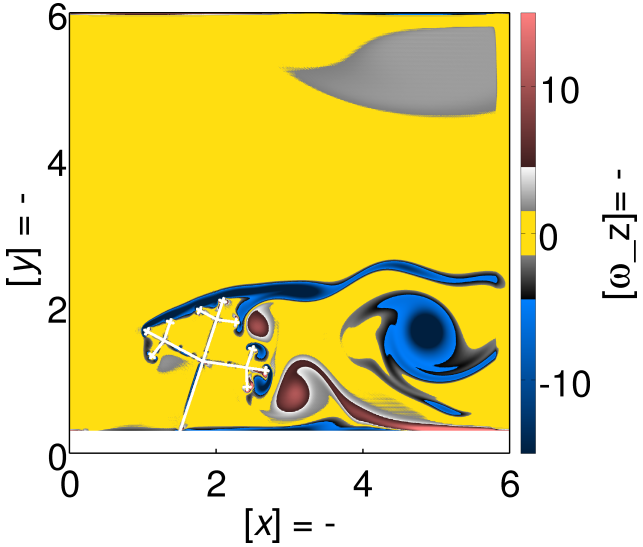


(b) Wirbelstärke ω_z bei der Auflösung 6144x6144 und bei $Re = 600$

Abb. 54: Wirbelstärke ω_z für die rigid Simulation für die Zeit $t = 6$

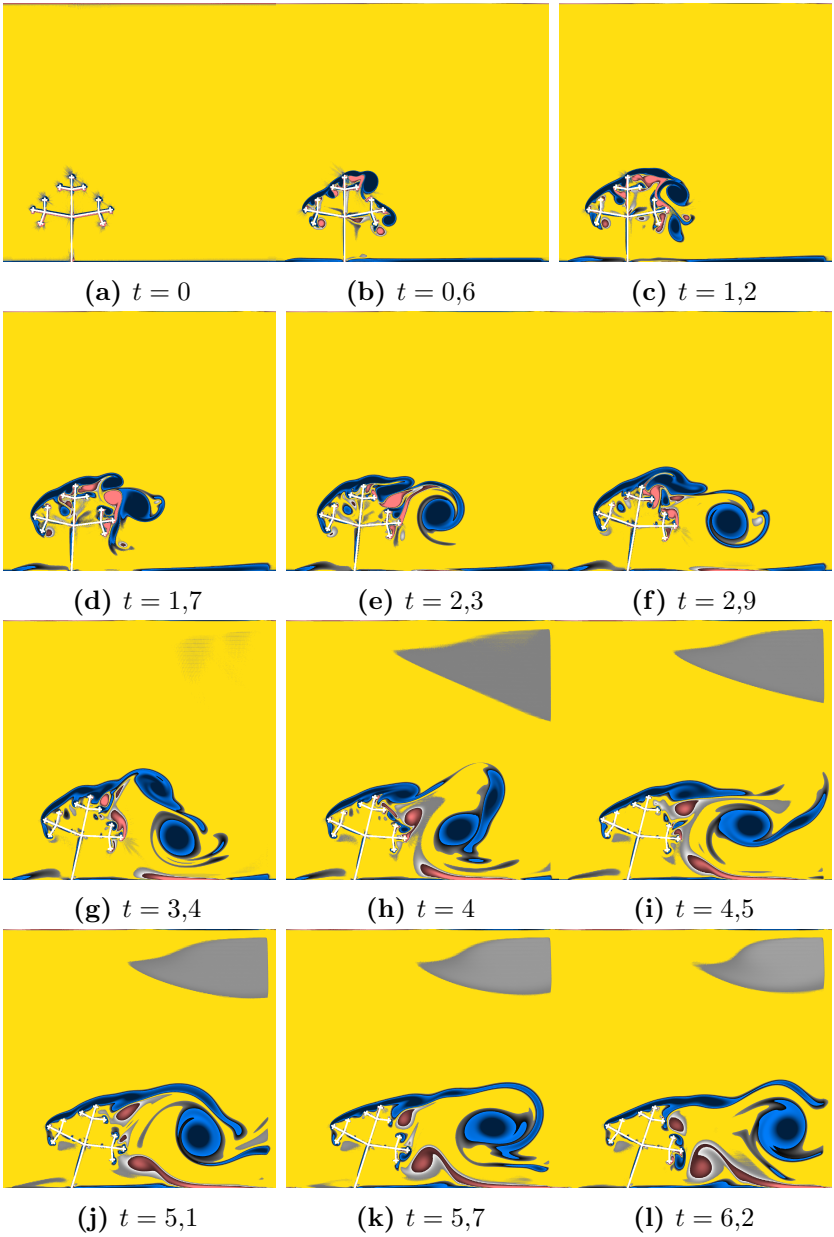


(a) Wirbelstärke ω_z bei der Auflösung 512x512 und bei $Re = 100$



(b) Wirbelstärke ω_z bei der Auflösung 1024x1024 und bei $Re = 600$

Abb. 55: Wirbelstärke ω_z für die FSI Simulation für die Zeit $t = 6$



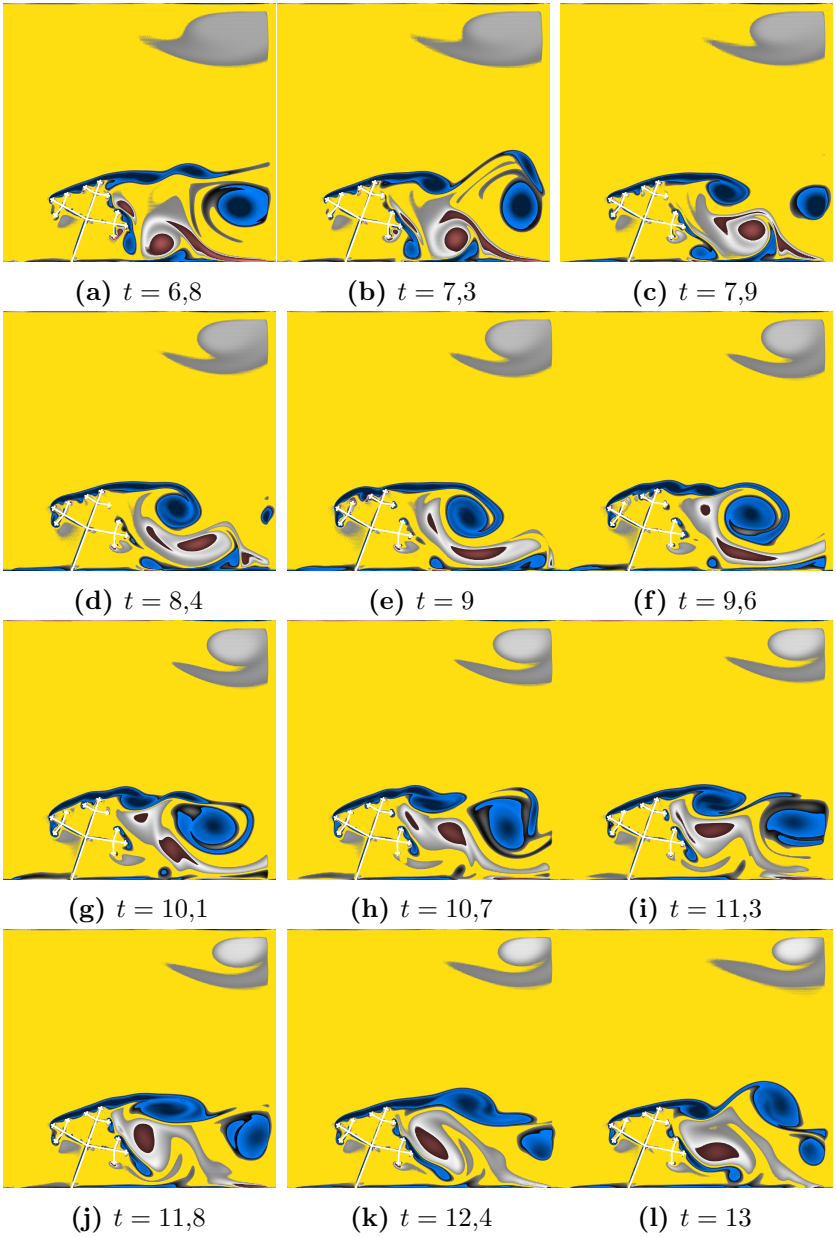
(a) $t = 0$ **(b)** $t = 0{,}6$ **(c)** $t = 1{,}2$

(d) $t = 1{,}7$ **(e)** $t = 2{,}3$ **(f)** $t = 2{,}9$

(g) $t = 3{,}4$ **(h)** $t = 4$ **(i)** $t = 4{,}5$

(j) $t = 5{,}1$ **(k)** $t = 5{,}7$ **(l)** $t = 6{,}2$

Abb. 56: Darstellung der Wirbelstärke ω_z für die FSI-Simulation in der Auflösung 1024x1024 bei $Re = 600$ für verschiedene Zeiten $t \in [0,\ 6{,}2]$



(a) $t = 6{,}8$ **(b)** $t = 7{,}3$ **(c)** $t = 7{,}9$

(d) $t = 8{,}4$ **(e)** $t = 9$ **(f)** $t = 9{,}6$

(g) $t = 10{,}1$ **(h)** $t = 10{,}7$ **(i)** $t = 11{,}3$

(j) $t = 11{,}8$ **(k)** $t = 12{,}4$ **(l)** $t = 13$

Abb. 57: Darstellung der Wirbelstärke ω_z für die FSI-Simulation bei $Re = 600$ in der Auflösung 1024x1024 für verschiedene Zeiten $t \in [6{,}8,\ 13]$

7 Fazit

Für die Simulationen konnte gezeigt werden, dass die Umströmung des Baumes plausibel ist. Ebenso wurde gezeigt, dass die zeitlich gemittelten Geschwindigkeitsfelder und die turbulente kinetische Energie qualitativ gleich aussehen. Das Vorzeichen der Kräfte und der Momente ist bei der Anströmrichtung plausibel. Die Baumauslenkung bei der FSI-Simulation ist in Anströmungsrichtung.

Quantitativ konnte keine Konvergenz gezeigt werden.

8 Ausblick

Konvergenz Ein Vorschlag wäre, in weiteren Arbeiten den Hauptfokus auf die Realisierung der Konvergenz in Raum und Zeit zu legen. Bei der Simulation hat sich gezeigt, dass die Anströmung störungsbehaftet ist. Diese Störungen kommen durch die periodischen Randbedingungen zustande. Mit einem anderen Sponge (Variation der Sponge-Konstante oder -Dicke) können diese Störungen verhindert werden.

Um den Einfluss der periodischen Randbedingungen (*Ghost*-Baum siehe Abb. 38) zu minimieren, könnte beispielsweise das Rechengebiet vergrößert werden. Auch könnten andere Diskretisierungen und Auflösungen für Raum und Zeit gewählt werden, um eine Konvergenz in Raum und Zeit zu erreichen. Wenn Oszillationen auftreten würden (siehe [13]), könnten diese durch die Variation des Smoothing-Layer (Funktion, Breite) verhindern werden.

Analyse Wenn die Konvergenz in Raum und Zeit realisiert ist, bietet es sich an, die Simulation mit anderen Reynolds-Zahlen, anderen mechanischen Ersatzmodellen (siehe Unterabschn. 4 auf S. 9), anderen fraktalen Bäumen oder in 3D durchzuführen. Auch könnten sich einzelne Äste bewegen, siehe Abb. 58. Es bietet sich an im Nachlauf die kohärenten Anteile mit der POD- oder der

DMD-Methode zu analysieren [19] [18] [17] [15] [11] [7] [23]. Für die turbulente Längenskalenanalyse könnten noch die TAYLOR-Micro- und -Macro-Scale untersucht werden.

Simulation Die Erstellung der Baummaske könnte noch optimiert werden. Siehe Unterabschn. 5.1.2 auf S. 21 unter Anmerkung. Die Simulation könnte in 3D realisiert werden. Bei dem fraktalen Baum könnte die Astdicke je nach Generation verändert werden. Der Baum könnte noch Blätter bekommen. Der Anfahrprozess der FSI-Simulation könnte noch näher untersucht bzw. verändert werden, um das Überschwingen von der FSI-Simulation mit der $Re = 100$ und der Auflösung 512x512 zu glätten.

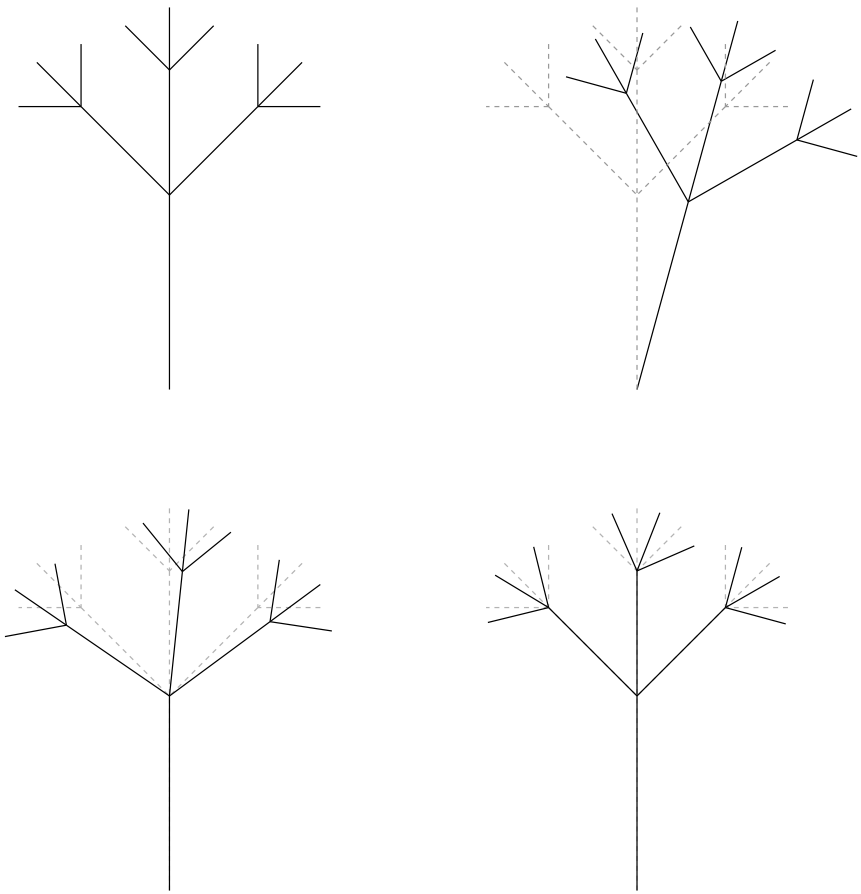


Abb. 58: Varianten für das mechanische Ersatzmodell: einzelne bewegliche Äste

Literatur

[1] B. Emek Abali. *Computational Reality*. 2015 (Entwurf).

[2] B. Emek Abali, Wolfgang H. Müller, and Dimitri V. Georgievskii. A discrete-mechanical approach for computation of three-dimensional flows. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 93(12):868–881, 2013.

[3] Farid Amirouche. *Fundamentals of multibody dynamics: theory and applications*. Springer Science & Business Media, 2007.

[4] P. Angot, C. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.*, 81:497–520, 1999.

[5] M. W. Wong (auth.). *Discrete Fourier Analysis*. Pseudo-Differential Operators 5. Birkhäuser Basel, 1 edition, 2011.

[6] G. Carbou and P. Fabrie. Boundary layer for a penalization method for viscous incompressible flow. *Adv. Diff. Equ.*, 8:1453–2480, 2003.

[7] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.

[8] T. Engels. *Numerical Modeling of Fluid–Structure Interaction in Bio–Inspired Propulsion*. PhD thesis, Aix-Marseille Université / Technische Universität Berlin, 2015.

[9] Thomas Engels, Jörn Sesterhenn, Dmitry Kolomenskiy, Kai Schneider, and Fritz-Olaf Lehmann. Der hummelflug in turbulenz. *Physik in unserer Zeit*, 47(3):111–112, 2016.

[10] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[11] J. B. Freund and T. Colonius. Turbulence and sound-field POD analysis of a turbulent jet. *aeroacoustics*, 8(4):337–354, 2009.

[12] G. A. Mensah J.P. Moeck. Skriptum zur Vorlesung Methoden der Datenanalyse in der Thermodfluiddynamik 2015. Technical report, TU Berlin Fakultät V, Fachgebiet: Experimentelle Strömungsmechanik, Oktober 2015.

[13] D. Kolomenskiy and K. Schneider. A Fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles. *J. Comput. Phys.*, 228:5687–5709, 2009.

[14] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[15] Tomas W Muld, Gunilla Efraimsson, and Dan S Henningson. Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. *Computers & Fluids*, 57:87–97, 2012.

[16] Wolfgang H Müller and Ferdinand Ferber. *Technische Mechanik für Ingenieure*. Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2008.

[17] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*, San Antonio, Texas, November 2008. American Physical Society.

[18] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[19] PJ Schmid, L Li, MP Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[20] K. Schneider. Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method. *Computers & Fluids*, 34:1223–1238, 2005.

[21] Suril Vijaykumar Shah, Subir Kumar Saha, and Jayanta Kumar Dutt. Dynamics of robotic systems. In *Dynamics of Tree-Type Robotic Systems*, pages 9–25. Springer, 2013.

[22] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[23] Jonathan Tu. *Dynamic Mode Decomposition: Theory and Applications*. Dissertation, Princeton, 2013.

[24] Edmund Wittbrodt, Iwona Adamiec-Wójcik, and Stanislaw Wojciech. *Dynamics of flexible multibody systems: rigid finite element method*. Springer Science & Business Media, 2007.