MASTER OF SCIENCE THESIS

Finite element discretization of the Spalart-Allmaras turbulence model

Pepijn Carlo Johannes Kessels

12-08-2016

Faculty of Aerospace Engineering · Delft University of Technology



Challenge the future

Finite element discretization of the Spalart-Allmaras turbulence model

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

Pepijn Carlo Johannes Kessels

12-08-2016

Faculty of Aerospace Engineering · Delft University of Technology





This thesis was carried out at San Diego State University. Copyright © Pepijn Carlo Johannes Kessels All rights reserved.

Delft University Of Technology Department Of Aerodynamics, Wind Energy, Flight Performance and Propulsion

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled "Finite element discretization of the Spalart-Allmaras turbulence model" by Pepijn Carlo Johannes Kessels in partial fulfillment of the requirements for the degree of Master of Science.

Dated: <u>12-08-2016</u>

Head of department:

Dr. ir. M.I. Gerritsma

Supervisor:

Dr. S.J. Hulshoff

Daily supervisor:

Reader:

Dr. R. Cavallaro

Dr. ir. A.C. Viré

Summary

The Galerkin formulation of the finite element method (FEM) is known to perform poorly when the advection or reaction terms of a partial differential equation (PDE) dominate the problem [27]. Unstable behavior arises for such situations, .e.g, in the form of oscillatory behavior of the solution. Many stabilized FEM formulations have been proposed over the years in an attempt to correct this unstable behavior. The streamline upwind/Petrov-Galerkin [12] (SUPG) method is an older popular approach for the stabilization of advection dominated problems. The variational multiscale [47] (VMS) stabilization method is based on a more firm mathematical foundation, and employs a decomposition of the flow field into large, resolvable scales, and small, unresolvable scales. In this thesis the objective is to improve the robustness of the Reynolds-averaged Navier-Stokes (RANS) based Spalart-Allmaras (S-A) turbulence model [61], by implementing several stabilized formulations and comparing their performance. From the literature review, Hauke's multiscale stabilization parameter [38] τ was found most promising and its definition is therefore used for the multiscale stabilized formulation.

This thesis is carried at San Diego State University. We use the C++ FEM library **libMesh** [53] for the numerical simulation of various problems. Hence, the implementation in the code is started from an unstabilized, unsteady Navier-Stokes (N-S) routine. In order to carry out interesting test cases on coarse and fine meshes the N-S equations are first stabilized using the SUPG, pressure stabilizing/Petrov-Galerkin (PSPG) and least-squares on incompressibility constraint (LSIC) stabilization terms. Several test cases are performed in order to assess their proper implementation.

A new subroutine is then added to the code for solving the S-A model in a segregated manner. Since the advection-diffusion-reaction (A-D-R) equation is a model problem for RANS based turbulence models, we first implement its multiscale stabilized form and test its correct implementation for several test cases. The multiscale stabilized A-D-R equation using Hauke's stabilization parameter shows to have superior accuracy and robustness on a coarse mesh compared to the Galerkin and SUPG method.

By adding the turbulent viscosity to the kinematic viscosity in the N-S equation the transition to RANS is made. The A-D-R coefficients are set such that the equation represents the S-A turbulence model. A test case using the method of manufactured solutions shows its correct implementation.

Three stabilized formulations are proposed for the S-A turbulence model. One is based on the work of Khurram et al. [52] in which the A-D-R equation is employed as a linearized form of the S-A model. The multiscale method is then applied to the A-D-R equation to obtain the stabilized formulation. The other proposed multiscale formulation is based on a more traditional multiscale approach in which the splitting of scales is applied directly to the S-A equation. The two formulations are compared and several similarities and differences are pointed out, i.e., for both formulations the higher order fine scale terms are ignored. The third and final proposed formulation is an SUPG type stabilization of the S-A model. The SUPG stabilized S-A equation is obtained by only using the advective stabilizing weighting term of the traditional multiscale formulation.

Two final test cases are run to determine the robustness of the Galerkin as well as each of the proposed stabilized formulations. A turbulent channel flow problem is run at a Reynolds number based on the friction velocity of $\text{Re}_{\tau} = 550$. The test case is run on a fine mesh and validated using direction numerical simulation (DNS) results. The fine mesh results are compared to coarse mesh results for each of the S-A model formulations. The unstabilized S-A formulation shows to be very robust for this case and not require any form of stabilization. The SUPG stabilized formulation does not affect the solution much, but some differences are noted using both multiscale formulations. Using Khurram's S-A model formulation, worse results are obtained in terms of accuracy on all meshes. The traditional multiscale method shows a deterioration of accuracy on medium coarse meshes but improves in accuracy compared to the Galerkin formulation as the mesh is refined.

A backward facing step case is run at a Reynolds number based on the step height H of $\text{Re}_H = 36,000$. A fine and coarse mesh is used in order to compare the coarse mesh Galerkin and stabilized S-A formulation results to the fine mesh results. Again, good results are obtained on the coarse mesh using the unstabilized S-A model, such as a similar determination of the reattachment point of the flow compared to the fine mesh. The SUPG and traditional multiscale method show very similar results, though the latter is determined to behave slightly overly diffusive. Khurram's multiscale model shows to be very under diffusive, and its results are much worse than the other models.

A more careful examination of the results for this test case reveals oscillatory behavior of the S-A model solution for the Galerkin and SUPG method in regions for which the diffusion term equals zero. This is found to mainly be the case mid-channel upstream and far downstream of the step, where the solution is constant or behaves linearly, and the mesh is coarse. Both multiscale formulations suppress the wiggles and provide a smooth solution. Therefore the multiscale formulations provide a more robust formulation in the absence of diffusion. Only the traditional multiscale formulation for the S-A model provides reasonably accurate results compared to the Galerkin and SUPG formulations. In the presence of diffusion the latter two methods are robust and produce good results, also on coarser meshes.

For the derived traditional multiscale formulation for the S-A model it is noted improvements can still be obtained as follows:

- Inclusion of higher order fine scale terms to improve coarse mesh accuracy.
- Inclusion of the fine scale velocity term in the S-A equation.
- Use of a nondiagonal τ to include the residual of each equation for each unknown for the coupled system of equations.

Acknowledgements

I would like to thank Luciano Demasi for granting me the opportunity to carry out my thesis in his research group at San Diego State University in California. For the completion of my thesis I want to thank Rauno Cavallaro for his ongoing support, and Steve Hulshoff for the assistance in the final stages of the thesis.

Delft, The Netherlands 12-08-2016

Pepijn Carlo Johannes Kessels

Contents

Sυ	Summary v			
A	Acknowledgements vii			
\mathbf{Li}	st of Figures	xii		
N	omenclature	xiii		
1	Introduction1.1Current state of research1.2Objective of the thesis1.3Programming language, libraries1.4Carried out work1.5Structure of the thesis	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		
2	Governing equations 2.1 Incompressible Navier-Stokes equations 2.2 Reynolds-averaged Navier-Stokes equations 2.3 Spalart-Allmaras turbulence model 2.3.1 Boussinesq approximation 2.3.2 The baseline model 2.3.3 Modifications to the baseline model	9 . 9 . 11 . 12 . 12 . 12 . 13 . 15		
3	Finite Element Method for the Navier-Stokes equations3.1Weak form of the problem3.2Temporal discretization and iterative solution procedure3.2.1Theta method3.2.2Newton's method3.3Galerkin and matrix formulation3.4Stabilization of the Galerkin formulation3.4.1SUPG and LSIC stabilization3.4.3Choice of stabilization parameters	$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
4	Finite Element Method for RANS and the S-A turbulence model4.1Finite element method for the RANS equations4.1.1Weak form4.1.2Discretization of the turbulent diffusion terms4.2The variational multiscale method4.3Multiscale finite element method applied to the S-A turbulence model4.3.1Weak form4.3.2Temporal discretization and iterative solution procedure4.3.3Analogy of the S-A model with the A-D-R equation	$\begin{array}{cccc} {\bf 37} \\ {\bf .} & {\bf 37} \\ {\bf .} & {\bf 38} \\ {\bf .} & {\bf 39} \\ {\bf .} & {\bf 40} \\ {\bf .} & {\bf 44} \\ {\bf .} & {\bf 45} \\ {\bf .} & {\bf 46} \end{array}$		

	4.4	4.3.4 4.3.5 4.3.6 Tradit	Variational multiscale method applied to the A-D-R equation Choice of stabilization parameter	$47 \\ 48 \\ 49 \\ 50$
	4.5	model 4.4.1 4.4.2 4.4.3 4.4.4 Compa	Fine scale problem Coarse scale problem Coarse scale problem Discretization and iterative solution procedure SUPG type stabilization for the S-A turbulence model Support the second	$50 \\ 50 \\ 51 \\ 53 \\ 55 \\ 55 \\ 55$
-	N T	· · · · · · · · · · · · · · · · · · ·		
9	5.1	Navier 5.1.1 5.1.2	-Stokes test cases Manufactured solution	57 57 59
	5.2	5.1.3 Advect 5.2.1	Laminar channel flow	$ \begin{array}{r} 61 \\ 62 \\ 62 \\ 64 \end{array} $
	5.3	5.2.2 Reyno 5.3.1	Skew advection case	$ \begin{array}{r} 64 \\ 67 \\ 67 \\ 67 \\ \end{array} $
		5.3.2	Turbulent channel flow \dots	
		5.3.3	 5.3.2.3 Wagnitude of S-A equation terms for the channel now problem	73 76 76 79 81
6	Con	clusio	ns & Recommendations	89
Re	efere	nces		93
\mathbf{A}	Law	of the	e wall	99
в	Pen	alty m	ethod	101
С	FEN	A disci	retized Navier-Stokes equations	103
D	FEN	A disci	retized Navier-Stokes stabilization terms	105
E	FEN	A disci	retized RANS diffusion terms	109
F	FEM discretized S-A turbulence model 1			113
G	Multiscale method applied to Spalart-Allmaras reaction terms			115
н	MM	[S sour	rce terms for RANS and Spalart-Allmaras	121

List of Figures

1.1	Comparison between the Galerkin method and Franca and Valentin's sta- bilized method for a skew advection problem with positive reaction term (destruction case). From [37]
2.1	Domain with Dirichlet and Neumann boundary condition, and the outward
2.2	unit normal n
$2.3 \\ 2.4$	Modified magnitude of vorticity [2]
$3.1 \\ 3.2$	Piecewise linear basis functions in 1-dimensional space Element transformation from the global domain to the normalized domain for a four-node quadrilateral
$4.1 \\ 4.2 \\ 4.3$	Bounded domain discretized into element subdomains. $\dots \dots \dots \dots$ Resolvable scales (left) and unresolvable, sub-grid scales (right). $\dots \dots \dots$ Computation of characteristic mesh size $h. \dots \dots \dots \dots \dots \dots \dots$
5.1	Convergence of L_2 error for the N-S equation unknowns with refining mesh
5.2	size. Computed pressure distribution for $p_{exact} = x(1-x)$ on a 32x32 uniform
5.3	element mesh
5.4	Lid-driven cavity flow test case
5.5	Streamlines and velocity magnitude for lid-driven cavity flow on an 80x80 4-node element mesh
5.6	Channel flow test case
5.7	Laminar channel flow solution for $\nu = \frac{1}{25}$ compared to exact solution at
58	$\frac{1}{L} = 0.5.$
0.0	mesh size.
5.9	Problem description for the skew advection case
5.10	Various solutions for production case: $\alpha = 1.5, s = -0.5, \ldots, \ldots$
5.11	Diagonal line plots of fully resolved solution versus multiscale, SUPG and
F 10	Galerkin method for various production cases ($s < 0$)
5.12	Fully resolved solution versus multiscale, SUPG and Galerkin method for destruction a_{22} ($a > 0$); $a_{22} = 10.0$
	destruction case ($s > 0$): $\alpha = 1.0, \sigma = 10.0.$

5.13	Fully resolved solution versus multiscale, SUPG and Galerkin method for	
5.14	destruction case $(s > 0)$: $\alpha = 10.0$, $\sigma = 0.1$	67
0.11	S-A turbulence model unknown with refining mesh size.	68
5.15	Turbulent channel flow test case.	69
5.16	Comparison of velocity profile using DNS (dashed line) and RANS with	
	S-A model (solid line) for turbulent channel flow simulated at $Re_{\tau} = 550$.	70
5.17	Comparison of velocity profile on several meshes for turbulent channel flow	70
	simulated at $Re_{\tau} = 550^{\circ}$ using the unstabilized S-A formulation	71
5.18	Comparison of modified turbulent viscosity profile on several meshes for	
	turbulent channel flow simulated at $Re_{\tau} = 550$ using the unstabilized S-A	70
5.19	Magnitude of the S-A equation terms that are nonzero for the turbulent	72
0.10	channel flow problem using 128 elements in y -direction.	74
5.20	Error distribution of ν_e on a mesh of 128 elements using the unstabilized	
	S-A formulation, compared to the fine mesh solution	75
5.21	Péclet number distribution throughout the interior of the channel for 8, 64	75
5 99	and 2048 elements used in <i>y</i> -direction	() 77
5.22 5.23	Coarsest mesh used for the backward facing step problem.	77
5.24	Close-up of the coarsest backward facing step mesh at the region of interest	••
	(x = -4H to x = 8H).	78
5.25	Different regions to be distinguished for calculating the wall distance d (not	
	drawn to scale).	79
5.26	Streamlines and magnitude of velocity contours throughout the domain for	70
5.27	the mesh case. <i>u</i> -velocity profile normalized by u_{ref} at several locations for fine mesh com-	79
	putation and experiment.	80
5.28	Lower wall skin friction (left) and pressure coefficient (right), computed	
	fine mesh solution versus experiment.	81
5.29	u-velocity profile normalized by u_{ref} upstream of the step at $x/H = -4$	00
5 30	on fine and coarse mesh for several S-A formulations	82
0.00	on fine and coarse mesh for several S-A formulations	82
5.31	<i>u</i> -velocity profile normalized by u_{ref} downstream of the step at $x/H = 4$	-
F 00	on fine and coarse mesh for several S-A formulations	83
5.32	u-velocity profile normalized by u_{ref} downstream of the step at $x/H = b$	09
5.33	Lower wall skin friction coefficient on fine and coarse mesh for several S-A	83
E 94	formulations.	84
0.54	formulations	8/
5.35	Modified turbulent viscosity field normalized by the kinematic viscosity for	01
	the fine grid backward facing step case.	85
5.36	Modified turbulent viscosity field normalized by the kinematic viscosity for	05
5 37	Line plot of the modified turbulent vicesity normalized by the kinematic	80
0.01	viscosity midway the channel $(y/H = 5)$ upstream of the step. The results	
	are given on fine and coarse mesh for several S-A formulations	86
5.38	Line plot of the magnitude of the diffusion term for the backward facing	-
	step case at several locations upstream of the step for the fine mesh solution	<u> </u>
	(values have been interpolated)	87
Δ 1	Nomenclature for wall-bounded turbulent flows [49]	100
11.1		100

Nomenclature

Latin Symbols

a	Advection vector	[-]
$a(\cdot,\cdot)$	Bilinear form, see equation (3.6)	[-]
$a_t(\cdot,\cdot)$	Bilinear form, see equation (4.4)	[-]
b	Right-hand-side vector for N-S equations	[-]
$b(\cdot,\cdot)$	Bilinear form, see equation (3.7)	[-]
С	Convection matrix	[-]
$c(\cdot;\cdot,\cdot)$	Trilinear form, see equation (3.8)	[-]
c_{xx}	S-A model constants, see section 2.3	[-]
d	Wall distance	[m]
f, \mathbf{f}	Source term (vector)	[-]
f_{xx}	S-A model parameter, see section 2.3	[-]
${\cal G}$	Covariant metric tensor	[-]
\mathbf{G}, \mathbf{G}^T	Gradient and divergence operator matrices, respectively	[-]
g	S-A model parameter, see section $2.3/$ Green's function	[-]
H	Sobolev space	[-]
Η	Step height	[m]
h	Characteristic mesh size	[m]
I	Identity matrix	[-]
J	Jacobian matrix	[-]
J	Jacobian	[-]
К	Viscosity matrix	[-]
k	Turbulent kinetic energy	$[m^2/s^2]$

\mathbf{M}	Mass matrix	[—]
M	Mach number	[—]
n	Unit outward normal vector	[—]
n_{el}	Number of elements	[—]
n_{qp}	Number of quadrature points	[—]
n_{sd}	Number of spatial dimensions	[—]
p	Vector containing nodal unknowns for p	[—]
\mathcal{P}	Operator applied to weighting function	[—]
p	Pressure	$[kg/(m\cdot s^2)]$
\mathcal{Q}	Trial and weighting scalar function space	[—]
q	Scalar weighting function	[—]
\mathcal{R}, \mathcal{R}	Residual (vector)	[—]
r	S-A model parameter, see section 2.3	[—]
R	Radius	[m]
Re	Reynolds number	[—]
S	Vector space of trial solutions	[—]
S	Magnitude of vorticity, see section 2.3	[1/s]
S_e	Modified magnitude of vorticity, see section 2.3	[1/s]
s	Reaction coefficient	[1/s]
T	Temperature	[K]
\mathbf{t},\mathbf{t}'	Traction or pseudo-traction boundary condition	[—]
t	Time	[s]
U	Navier-Stokes solution vector $\begin{bmatrix} u & v & p \end{bmatrix}^T$	[-]
\boldsymbol{u}	Vector containing nodal unknowns for u	[-]
u	Velocity vector $\begin{bmatrix} u & v \end{bmatrix}^T$	[-]
u	Velocity in x -direction	[m/s]
\mathcal{V}	Vector space of weighting functions	[—]
v	Vector containing nodal unknowns for v	[—]
\mathbf{v}_h	Auxiliary velocity field (velocity field minus boundary)	[—]
v_{iA}	Nodal velocity unknown with respect to spatial component \boldsymbol{i}	and node $A[-]$
v	Velocity in y -direction	[m/s]
w, \mathbf{w}	Weighting function (vector)	[-]

Greek Symbols

α	Element Péclet number	[-]
Г	Boundary	[-]

ϵ	Strain rate tensor	[-]
η	Set of global velocity nodes	[-]
θ	Theta method constant	[-]
κ	Diffusion coefficient	$[m^2/s]$
κ	Von Kármán constant, equals 0.41	[-]
ν	Kinematic fluid viscosity	$[m^2/s]$
$ u_e$	Modified turbulent viscosity	$[m^2/s]$
$ u_t$	Turbulent viscosity	$[m^2/s]$
ρ	Density of the fluid	$[kg/m^3]$
σ	Stress tensor	[-]
σ	Element Damköhler number	[-]
σ	S-A model constant, equals $\frac{2}{3}$	[-]
au	Shear stress tensor/Stabilization parameter matrix	[-]
au	Stabilization parameter	[s]
$ au_w$	Wall shear stress	$[kg/(m\cdot s^2)]$
μ	Dynamic fluid viscosity	$[kg/(m\cdot s)]$
ϕ_a,ϕ_A	Basis function for local node a or global node A	[-]
$arphi_{\hat{a}},arphi_{\hat{A}}$	Basis function for local or global pressure node \hat{a} or \hat{A} , respect	ively [m]
χ	S-A model parameter defined as $\frac{\nu_e}{\nu}$	[-]
Ω	Rotation rate tensor, see section 2.3	[1/s]
Ω	Domain	[-]

Subscripts

A	Global node A
a	Local element node a
В	Global node B
b	Local element node b
с	Continuity
D	Dirichlet/Diameter
h	Finite dimensional approximation of the variable
i	Spatial component index
m	Momentum
Ν	Neumann
t	Turbulent
,x	Partial derivative with respect to x
,y	Partial derivative with respect to y
$\cup \Omega^e$	Union of element interiors

au	Friction
qp	Quadrature point
ref	Reference

Superscripts

\bar{x}	Variable ensemble average
\hat{x}	Variable at old or previous time step
\overline{x}	Variable coarse scales
$ ilde{x}$	Variable fluctuations
+	Non-dimensionalized wall unit
e	Referring to a specific element e
k	Newton iteration number
T	Transpose
x'	Variable fine scales

Abbreviations

A-D-R	Advection-diffusion-reaction
ASM	Additive Schwarz method
\mathbf{CFD}	Computational fluid dynamics
DNS	Direct Numerical Simulation
FEM	Finite element method
GMRES	Generalized minimal residual method
LBB	Ladyzhenskaya-Babuška-Brezzi
LES	Large eddy simulation
LSIC	Least-squares on incompressibility constraint
MMS	Method of manufactured solutions
MUMPS	Multifrontal massively parallel sparse direct solver
N-S	Navier-Stokes
PDE	Partial differential equation
PETSc	Portable, Extensible Toolkit for Scientific Computation
\mathbf{PG}	Petrov-Galerkin
PSPG	Pressure stabilizing/Petrov-Galerkin
RANS	Reynolds-averaged Navier-Stokes
S-A	Spalart-Allmaras
SUPG	Streamline upwind/Petrov-Galerkin

VMS Variational multiscale

Other Symbols

<=>	Left- and right-hand-side are additions to the Jacobian matrix and right-hand-side vector, respectively
∇	Del operator
∂	Partial derivative
\mathbb{R}^{n}	The n -dimensional real space
$\langle \bullet \rangle$	Reynolds average operator
(\mathbf{x}, \mathbf{x})	$\int_{\Omega} \mathbf{x} \cdot \mathbf{x} d\Omega$
$(\mathbf{x},\mathbf{x})_{\Gamma}$	$\int_{\Gamma} \mathbf{x} \cdot \mathbf{x} d\Gamma$

Chapter 1

Introduction

In the study of aerodynamics, one of the unsolved problems is a complete description of turbulence. It is believed that the Navier-Stokes (N-S) equations can be used to describe turbulence properly [54]. Computational fluid dynamics (CFD) play a great in role in numerically approximating the solution to these equations. This has led to a reduction in cost of product and process development and optimization activities, improved reliability and reduced the need for physical experimentation. CFD is applied in many different industries, including aerospace, automotive and medical research to name a few. Numerical solutions to the Navier-Stokes equations using CFD, however, are difficult to obtain. A stable solution requires an extremely fine mesh to capture the smallest scale of eddies. This results in an infeasible amount of computational work to be performed to solve them using direct numerical simulation (DNS). Several methods have been proposed in order to approximate the solution of the Navier-Stokes equations at a much lower computational cost, the most popular among them are large eddy simulation (LES) and the Reynoldsaveraged Navier-Stokes (RANS) equations. While in LES turbulence is still partly solved for, using RANS, all turbulence is modeled using complex empirical models. It is therefore the cheaper and more feasible option with regard to computational cost, especially for industrial cases.

The RANS equations contain the nonlinear Reynolds stress term that needs additional modeling for closing the equations. Using the Boussinesq hypothesis for this term leads to an eddy viscosity model. For this purpose, many turbulence models, in which the unknown is the eddy viscosity, have been developed. Some popular turbulence models are the Spalart-Allmaras [61] (S-A), $k - \epsilon$ [25], $k - \omega$ [65] and Menter's Shear Stress Transport [57] (SST) model. Each method has its advantages and disadvantages in terms of accuracy, stability and computational cost. Throughout this work, use will be made of the one-equation S-A turbulence model. The transport equation for the eddy viscosity used by the S-A model is known to have competitive accuracy and a relatively low cost [14].

Using the finite element method (FEM) as discretization approach to find a solution to the RANS equations and turbulence models can be quite challenging. The standard Galerkin formulation of the problem deals with potential sources of numerical instabilities. This is, for example, the case when convection or reaction terms dominate the flow, see e.g. [23, 37, 38, 52]. It is a challenge to find such a stabilization that makes solving the equations as robust as possible. A stabilized FEM formulation can be created by adding mesh-dependent, consistent and numerically stabilizing terms to the standard Galerkin method. Various stabilized formulations have been proposed, a lot of which are based on the Petrov-Galerkin (PG) approach which uses a modification of the standard Galerkin weighting term. Examples of popular PG methods are the streamline upwind/Petrov-Galerkin [12] (SUPG) and pressure stabilizing/Petrov-Galerkin [44] (PSPG) scheme.

The SUPG method has successfully been applied to partial differential equations (PDEs) containing convection and diffusion terms such as the Navier-Stokes equations. This artificial diffusion based method prevents oscillatory behavior in the solution for convection dominated flows. The idea stems from the finite difference method, where it was already observed that adding artificial diffusion can prevent oscillatory behavior of the solution. The added diffusion using the SUPG method is only added in the streamline direction, otherwise overly diffusive behavior of the method ensues. To accomplish this the weighting function of the standard Galerkin method is modified by adding a streamline upwind perturbation. This is done such that the advective part of the PDE is multiplied by the weight and added to the weighting function. The added term is weighted by the stabilization parameter τ . Later it was found that this form of stabilization stems from the variational multiscale concept, introduced below.

Turbulence models such as the S-A model also contain a reactive term, i.e., a zeroth order spatial derivative term. This term plays an important role in boundary layer prediction, where reaction-like effects dominate the flow behavior. Next to turbulence models, reaction terms also arise in mechanical engineering problems related to turbomachinery, e.g., in the modeling of Coriolis forces. Other examples are found in chemistry, heat transfer with radiation, acoustics and overland flow applications. For reaction (production or destruction) dominated flows artificial diffusion based methods are inadequate [52]. Hence the interest arises for a robust stabilization scheme that can cope with reaction dominated flow conditions. It is noted that the advection-diffusion-reaction (A-D-R) equation can be used as a model problem for RANS-based turbulence models. Therefore, work that has been performed on the stabilization of this equation can be used for the stabilization of the S-A turbulence model as well.

An approach proposed by Hughes [47] in 1995 called the variational multiscale (VMS) method, is based on a decomposition of the scalar field into coarse (resolved) and fine (unresolved) scales. While applying VMS it is not desired to describe the unresolvable scales in detail; instead, it is desired to compute their effect on the resolvable scales. This stabilization method can deal much better with the reaction term in turbulence models, though it was pointed out by Hauke [38] that VMS also needs some improvements in the presence of high reaction rates. In order to see what the current state of research is on this topic, the more recent papers on multiscale stabilization of turbulence models and the A-D-R equation are discussed in section 1.1. Section 1.2 will then present the objective of this thesis. The programming language and libraries used are introduced in section 1.3. In the final section of the introduction a brief chapter by chapter overview will be given of the topics discussed throughout this thesis.

1.1 Current state of research

The variational multiscale method was introduced by Hughes [47] in 1995. The multiscale interpretation amounts to assuming that unresolvable, small scale behavior exists within each element of the domain, but not on element boundaries. The stabilization of equations such as the advection-diffusion-reaction equation relies on the derivation of a stability parameter τ , also called intrinsic time scale. In the paper by Hughes [47] the small scales are solved for using a corresponding Green's function problem, where it is shown that the subgrid scales are driven by the residual of the resolved scales. It can be shown that τ can be calculated using the element Green's function g(x, y) as

$$\tau = \frac{1}{\text{meas}(\Omega^e)} \int_{\Omega^e} \int_{\Omega^e} g(x, y) \mathrm{d}\Omega_x \mathrm{d}\Omega_y, \qquad (1.1)$$

with Ω^e the domain elements. This shows that in this way τ is defined as the element mean value of the Green's function.

By representing the small scales using bubble functions, which vanish on element boundaries, an approximate element Green's function can also be generated. Although it is admitted that the quality of the approximation is usually poor due to the incapability of the bubbles to represent the subgrid scale phenomena. The idea of using bubble functions was already used before with success, see e.g. [36]. However, Hughes showed in his paper [47] that this stabilization could be derived from a firm theoretical foundation. A definition of the stabilization parameter τ emerged, as shown in equation (1.1). A year later Brezzi et al. [11] showed the equivalence between the residual-free bubbles method and the variational multiscale method.

Franca and Valentin [37] derived a mesh dependent stabilization parameter for the advection-diffusion-reaction equation from convergence and stability theory. Bubble functions are used for the fine scales. The equation for τ is then derived by static condensation of the bubbles and subtracted from the standard Galerkin method. It includes two switches for the asymptotic behavior of the advection, diffusion and reaction parameters. These switches improve the accuracy of the solution compared to the initial stabilization parameter presented in [36]. To give an idea of the improved stability achieved by stabilized methods, figure 1.1 shows an improved stability using the Franca and Valentin stability parameter. Here the standard Galerkin method is compared to the stabilized method for a skew advection case.

Codina [17, 18, 19] also proposed a stabilization parameter derived from the maximum principle. Both this and Franca and Valentin's [37] parameter were originally valid for a positive reaction term only. Hauke [38] extended both of these parameters by including the negative reaction term. In this paper Hauke compares several stability parameters including both of the extended stability parameters for the exponential (distinguishing between positive and negative reaction term) and propagation regime. Using the results of Hauke et al. [40] and references therein shows that the extension of the Franca and Valentin [37] parameter is the most accurate.

Hulshoff et al. [50] used the research by Hauke [38] by using the stability parameter that was deemed to perform best, the extension of the Franca and Valentin [37] parameter, for multiple RANS based turbulence models. The VMS method was applied to both the S-A [61] and Menter SST [57] turbulence model using this stability parameter. The work however lacked testing for more difficult cases to make conclusions about the robustness



Figure 1.1: Comparison between the Galerkin method and Franca and Valentin's stabilized method for a skew advection problem with positive reaction term (destruction case). From [37].

of this approach for the S-A model.

Hauke et al. [40, 39, 41] presented several alternative stabilized subgrid scale (SGS) methods. In [40] the exact VMS and SGS methods were developed for the advectiondiffusion-reaction equation. It is shown that the derived stabilization parameter τ is very close to those proposed by Codina [17, 18, 19], and Franca and Valentin [37]. In [39] a combined method is proposed which comprises the SGS and Gradient SGS (GSGS) methods yielding the SGS-GSGS method. It is noted that in the regime of strong advection combined with a negative source term, the stability parameter proposed by Franca and Valentin [37, 38] performs better. In [41] the SGS-GSGS is further investigated. The two stability parameters are chosen to obtain nodally exact one-dimensional solutions for zero forcing term problems. This method is extended to a multi-dimensional case and improved results are obtained in comparison to the Galerkin and SGS methods.

Corsini et al. [23] introduced the variable-subgrid scale (V-SGS) method using the variational multiscale approach. It is designed for quadratic elements with a variable stabilization parameter dependent on space. The use of higher order finite element spaces guarantees the best compromise between solution stability and accuracy, as shown in Borello et al. [10]. The equation for the designed stability parameter is computed by exact integration over each element of $g_e(x, y)$:

$$\tau^{V-SGS}(y) = \int_{\Omega_e} g_e(x, y) \mathrm{d}\Omega_x.$$
(1.2)

A criticism that is noted on the use of an element-wise constant τ is that it is only capable of controlling element-wise constant residuals, which are obtained on advective-diffusive problems with linear elements [48]. The residual will be variable when reactive terms and/or higher order elements are used, thus the need for a space-dependent τ such as the proposed τ^{V-SGS} . It can be shown analytically that for a one-dimensional advectiondiffusion problem this parameter can be split up into exactly the SUPG stabilizing parameter (which is not space dependent) and a space dependent zero mean function. For a one-dimensional advection-diffusion-reaction problem it is shown that the V-SGS method performs better than the SUPG method when using quadratic elements. The method is also applied to the RANS equations with a $k - \varepsilon - v^2 - f$ turbulence model [29]. A test is performed on a NACA airfoil and results show a better performance using the V-SGS stabilization compared to a standard SUPG stabilization.

The V-SGS method was further investigated by Corsini et al. in [24, 22]. In [24], an additional stabilization term is added to the V-SGS method. The additional term is called diffusion for reaction-dominated with local variation jump (DRDJ) term. The diffusion for reaction-dominated (DRD) term was introduced by Tezduyar and Park [63], and adapted in [24] to account for local jumps. The DRDJ term is used as discontinuitycapturing tool, such that diffusion is added only when reaction rates and solution gradients are both high. Tests with this method shows how adding the DRDJ term improves the solution compared to the SUPG method, and also compared to adding just the DRD term. In [22] the V-SGS method with DRDJ term is applied to the RANS equations with a $k - \varepsilon$ turbulence model, for which three-dimensional flow computations are performed. The robustness of the method is shown and improved results are obtained compared to the SUPG method with DRDJ term.

Masud and Khurram [55] developed a multiscale stabilized finite element method for the advection-diffusion equation. In this method, bubble shape functions are used to represent the fine scale trial solutions and weighting functions. This method was then extended to the A-D-R equation in [51] by Khurram and Habashi. The S-A turbulence model is then applied to this method as the A-D-R equation is a model problem for it. Extending this line of work, Khurram et al. [52] then used the multiscale stabilized Spalart-Allmaras turbulence model for a three-dimensional detached-eddy simulation (DES) test case. The multiscale stabilized A-D-R equation is shown to have superior stabilization to SUPG for reaction dominated flows. With the backward-facing step as model problem the Spalart-Allmaras model for DES provides good results and is able to predict the main features of turbulent flow. In a follow-up paper by Zhang et al. [66] the developed model is used for test cases in computational wind engineering. The results show to be superior to using unsteady RANS and compare well to wind tunnel experiment values. This shows the robustness of the developed model.

1.2 Objective of the thesis

From the literature study it is noted that most research puts focus on comparing the developed multiscale stabilization method to SUPG stabilization. Hence it is not immediately straightforward which method produces the best results. Hauke [38] did compare several multiscale stabilization parameters for the A-D-R equation, and Hulshoff et al. [50] used this work in order to try to improve the robustness of the S-A turbulence model. Since this work lacked proper testing it is interesting to follow up on this line of work.

To accomplish this we will make use of an SUPG type and two different multiscale approaches for the S-A turbulence model. One multiscale approach will be based on the work of Khurram et al. [52], for which their paper is used as a guideline. In this method VMS is not applied to the S-A equation itself but the A-D-R equation, which is a model problem for RANS based turbulence models. The advection, diffusion and reaction coefficient are then defined such that they represent the S-A turbulence model. The second

multiscale approach is based on a more traditional way of applying the multiscale method as it is applied directly to the S-A equation. For both methods we will use the extended stabilization parameter of Franca and Valentin [37] derived by Hauke [38]. From the latter multiscale approach the SUPG formulation is derived by only using the advection part of the weighting function in the stabilization term. The research objective is then formulated as follows:

The objective of the research project is to improve the robustness and stability, compared to other stabilization methods, of the RANS-based Spalart-Allmaras turbulence model in the context of finite elements by deriving, testing and comparing several stabilized formulations of the turbulence model for a challenging test case.

1.3 Programming language, libraries

The C++ language with the libMesh [53] library will be used throughout this thesis. This library provides a finite element framework that can be used for the numerical simulation of partial differential equations on serial and parallel platforms. The libMesh library is an excellent tool for programming the finite element method and can be used for one-, two-, and three-dimensional steady and transient simulations. We use the Portable, Extensible Toolkit for Scientific Computation [7] (PETSc) library for the solution of linear systems, both in serial and parallel. As some solvers and/or preconditioners are not supported in parallel by PETSc, use is also made of the multifrontal massively parallel sparse direct solver [3, 4] (MUMPS) library.

1.4 Carried out work

The libMesh library contains a series of examples, including a simple non-stabilized unsteady nonlinear Navier-Stokes code which solves a lid-driven cavity flow problem. This was the starting point of this thesis.

In order to properly test the stabilized S-A model, stabilization terms were first added to the N-S code. A new subroutine for the S-A turbulence model was then added to solve for the turbulent viscosity in a segregated manner. Segregated solving of the flow and turbulence model equations will result in a high degree of modularity of the code. Other turbulence models may be added to the code and be chosen to be used instead, depending on the problem being solved. The multiscale stabilized A-D-R equation was first added to this subroutine, using the extended stabilization parameter of Franca and Valentin [38]. Several stabilized formulations of the S-A turbulence model were added by using different definitions of the A-D-R coefficients. Finally, the change of Navier-Stokes to RANS was made in the code. Each step has been accompanied by code verification using the method of manufactured solutions (MMS). Next to this, test cases were performed to check the stability of the stabilized formulation. For the final testing, a turbulent channel flow problem as well as a backward facing step problem were analyzed.

1.5 Structure of the thesis

The thesis is structured as follows. Chapter 2 will introduce the governing equations used throughout this thesis. The Navier-Stokes and RANS equations are addressed as well as the S-A turbulence model. Chapter 3 gives a step-by-step explanation of the finite element method, applied to the N-S equations. Besides this discretization approach, we also discuss Newton's method as iterative solution procedure and the theta method for the temporal discretization. The final section of this chapter introduces the stabilized N-S formulation, including the discretization of these additional terms. In chapter 4 the weak form of the RANS equations is given. Only the diffusion term is then discretized as it is the only term different from the N-S equations. We then review the theory behind the variational multiscale method, after which it is applied to the advection-diffusion-reaction equation. This equation is put in such a form such that it represents the S-A turbulence model. The definition of the stabilization parameter used is also provided. Two separate multiscale stabilized formulations as well as an SUPG type stabilized formulation are derived. The numerical results of the code verification and validation are discussed in chapter 5. First the N-S and A-D-R equation results are presented, after which we show and discuss the final test case results for the RANS equations with S-A model. These are the turbulent channel flow problem and the backward facing step case. Finally, in chapter 6 the conclusions of the test results are summarized and some recommendations for future work are given.

Chapter 2

Governing equations

In this chapter the partial differential equations used for the simulation of turbulent fluid flow are presented. Since the simulation performed is two-dimensional, the equations presented here are in two-dimensional form. First the well-known Navier-Stokes equations are introduced. Next, the Reynolds-averaged Navier-Stokes equations are shown to emerge after Reynolds averaging of the Navier-Stokes equations. Finally, the Spalart-Allmaras turbulence model is presented for the modeling of the Reynolds stress tensor.

2.1Incompressible Navier-Stokes equations

Using the principle of mass conservation inside a fluid element the continuity equation can be derived. For an incompressible fluid the density of the fluid is constant throughout the flow. The continuity equation is then given by the divergence of the (unknown) velocity vector $\mathbf{u}(\mathbf{x}, t)$,

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times]0, T[\tag{2.1}$$

 $\mathbf{v} \cdot \mathbf{u} = \mathbf{0}$ in $\mathfrak{U} \times [\mathbf{0}, T]$ (2.1) with $\mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^T$ and Ω is the flow domain. Note that u and v are the velocities in xand y-direction, respectively.

From the conservation of momentum, using Newton's second law, we can find that for an incompressible Newtonian viscous fluid the momentum equation reads as,

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \,\mathbf{u}\right) - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f} \quad \text{in } \Omega \times]0, T[\tag{2.2}$$

with the first left-hand-side term being the velocity time derivative, $\nabla \mathbf{u}$ the velocity gradient, ρ the fluid density, σ the stress tensor and f the external source term. The stress tensor consists of normal and shear stresses,

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$$

= $-p\mathbf{I} + 2\mu\boldsymbol{\epsilon}$
= $-p\mathbf{I} + \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)$ (2.3)

with $p(\mathbf{x},t)$ the (unknown) pressure, I the identity matrix, $\boldsymbol{\tau}$ the shear stress tensor, μ the dynamic viscosity and $(\nabla \mathbf{u})^T$ indicates the transpose of the velocity gradient. The strain rate tensor $\boldsymbol{\epsilon}$ is defined as,

$$\boldsymbol{\epsilon} = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \tag{2.4}$$

and equals the symmetric part of the velocity gradient. Taking the divergence of the stress tensor, the expression for ϵ can be simplified as,

$$\nabla \cdot \boldsymbol{\sigma} = \nabla \cdot \left(-p\mathbf{I} + \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right)$$
$$= -\nabla p + \mu \nabla^2 \mathbf{u}$$
(2.5)

with ∇^2 the Laplace operator. Here we used the vector identity $\nabla \cdot (\nabla \mathbf{u})^T = \nabla (\nabla \cdot \mathbf{u})$. Using the continuity equation (2.1) we can see this term equals zero. Implementing (2.5) into (2.2) yields the convenient form of the momentum equation,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \,\mathbf{u} - \nu \nabla^2 \mathbf{u} + \frac{1}{\rho} \nabla p = \mathbf{f}$$
(2.6)

where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity. Furthermore, for simplicity, let us assume uniform density $\rho = 1$. Finally we have,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \,\mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f}$$
(2.7)

From left to right in equation (2.7) we can identify the temporal, convection, diffusion, pressure and external source term, respectively. Equation (2.7) and (2.1) constitute the form of the Navier-Stokes equations for incompressible viscous fluid flow that will be used throughout this work.

For a complete problem description a domain $\Omega \in \mathbb{R}^2$ with proper boundary conditions on the boundary $\Gamma = \partial \Omega$ must be prescribed. We introduce Dirichlet, Γ_D , and Neumann, Γ_N , type boundary conditions, and assume $\partial \Omega = \Gamma_D \cup \Gamma_N$. This domain is portrayed in figure 2.1. Typical boundary conditions consist of prescribing the value of the velocity on Γ_D , while prescribing the pseudo-traction boundary condition σ' on Γ_N , i.e.,

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_D(\mathbf{x},t), \quad \mathbf{x} \in \Gamma_D, \quad t \in]0,T[$$

$$\mathbf{\tau}'.\mathbf{n}(\mathbf{x},t) = \mathbf{t}'(\mathbf{x},t), \quad \mathbf{x} \in \Gamma_N, \quad t \in]0,T[$$
(2.8)

with the matrix-vector multiplication defined as $(\mathbf{A}.\mathbf{a})_i = A_{ij}a_j$, **n** the unit outward normal and,

$$\boldsymbol{\sigma}'.\mathbf{n} = -p\mathbf{n} + \nu(\mathbf{n}\cdot\nabla)\mathbf{u} \tag{2.9}$$

In some situations instead we may prescribe a force acting on Γ_N . This is called the traction boundary condition σ and is given as,

$$\boldsymbol{\sigma}.\mathbf{n}(\mathbf{x},t) = \mathbf{t}(\mathbf{x},t), \quad \mathbf{x} \in \Gamma_N, \quad t \in]0,T[$$
(2.10)

with σ the shear stress tensor given by equation (2.3).

Next to the boundary condition, an initial condition for the velocity field must be specified as well:

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0, \quad \mathbf{x} \in \Omega.$$
(2.11)

The initial velocity field \mathbf{u}_0 must be divergence free in order to satisfy the continuity equation. Next to this, no initial conditions are imposed on p since no time derivative of p appears in the governing equations. It is noted that for purely Dirichlet boundary conditions the pressure can only be determined up to an arbitrary constant, since only its gradient is determined from the governing equations.



Figure 2.1: Domain with Dirichlet and Neumann boundary condition, and the outward unit normal n.

2.2 Reynolds-averaged Navier-Stokes equations

By decomposing the flow into a mean and fluctuating part, a time-average of the Navier-Stokes equations can be derived. With this Reynolds decomposition the turbulent fluctuations of the flow can be defined in a single tensor, called the Reynolds stress tensor. The Reynolds decomposition for the velocity and pressure are defined as,

$$\mathbf{u}(\mathbf{x},t) = \bar{\mathbf{u}}(\mathbf{x},t) + \tilde{\mathbf{u}}(\mathbf{x},t), \qquad p(\mathbf{x},t) = \bar{p}(\mathbf{x},t) + \tilde{p}(\mathbf{x},t)$$

where $\bar{\mathbf{u}}$ and \bar{p} indicate the ensemble average (i.e., the mean flow is the result of the average of many instances of the flow) at (\mathbf{x}, t) . $\tilde{\mathbf{u}}$ and \tilde{p} indicate the turbulent fluctuations at (\mathbf{x}, t) .

We now introduce the Reynolds average operator as $\langle \cdot \rangle$. Taking the Reynolds average we can compute the ensemble average of a variable at (\mathbf{x}, t) . The following identities hold for the Reynolds average of a variable ϕ or φ :

•	$\left\langle \frac{\partial \phi}{\partial \mathbf{x}} \right\rangle = \frac{\partial \bar{\phi}}{\partial \mathbf{x}}$	•	$\left\langle \frac{\partial \phi}{\partial t} \right\rangle = \frac{\partial \bar{\phi}}{\partial t}$
•	$egin{array}{lll} \langle ar{\phi} angle = ar{\phi} \ \langle ar{\phi} angle = 0 \end{array}$	•	$\langle \bar{\phi} \tilde{\varphi}\rangle = 0$
•	$\langle \bar{\phi} \bar{\varphi} \rangle = \bar{\phi} \bar{\varphi}$	•	$\langle \phi + \varphi \rangle = \bar{\phi} +$

It can be noted that the ensemble average of the turbulent fluctuating term equals zero, since it is a zero average term around the mean.

 $\bar{\varphi}$

By substituting the decomposed variables into the incompressible Navier-Stokes equations, (2.1) and (2.6), and computing the Reynolds average of this equation we can derive

the Reynolds-averaged Navier-Stokes (RANS) equations. For the continuity equation we have,

$$\nabla \cdot \bar{\mathbf{u}} = 0 \quad \text{in } \Omega \times]0, T[\tag{2.12}$$

and we thus also have that,

$$\nabla \cdot \tilde{\mathbf{u}} = 0. \tag{2.13}$$

It can be seen that a linear term results in an averaged term, since the fluctuating term equals zero when applying the Reynolds average operator. The momentum equation contains the nonlinear convection term and hence will result in an additional term. When applying the above operations, the Reynolds-averaged momentum equation can be written as,

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \,\bar{\mathbf{u}} + \overline{\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}} - \nu \nabla^2 \bar{\mathbf{u}} + \frac{1}{\rho} \nabla \bar{p} = \bar{\mathbf{f}}$$
(2.14)

where the additional term $\overline{\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}}$ can be observed. Using equation (2.13) we can rewrite this expression as,

$$\overline{\tilde{\mathbf{u}}\cdot\nabla\tilde{\mathbf{u}}}=\nabla\cdot\overline{\tilde{\mathbf{u}}}\,\overline{\tilde{\mathbf{u}}}\,$$

Equation (2.14) can upon manipulation be written as,

$$\rho \frac{\partial \bar{\mathbf{u}}}{\partial t} + \rho \left(\bar{\mathbf{u}} \cdot \nabla \right) \bar{\mathbf{u}} = -\nabla \bar{p} + \nabla \cdot \left(\bar{\boldsymbol{\tau}} - \rho \overline{\tilde{\mathbf{u}}} \bar{\mathbf{u}} \right) + \rho \bar{\mathbf{f}}$$
(2.15)

with $\bar{\boldsymbol{\tau}} = \mu \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right)$. This is the final form of the time-averaged momentum equation, and together with (2.12) make up for the Reynolds-averaged Navier-Stokes equations. The equations express the evolution of the mean solution. Note the similarity to the equations for the instantaneous variables, the Navier-Stokes equations. The only additional term, $-\rho \tilde{\mathbf{u}} \tilde{\mathbf{u}}$, can be seen to act as an apparent stress, and is thus termed the Reynolds stress tensor. The term is however related to the behavior of turbulent velocity fluctuations, and not to any viscous processes.

It can be observed that the RANS equations are not closed. The Reynolds stress tensor would have to be expressed in terms of the problem parameters and the unknown mean solution in order to solve the closure problem. The Reynolds stress tensor can however be approximated by use of empirical turbulence models such as the Spalart-Allmaras turbulence model. This model is discussed in the next section.

2.3 Spalart-Allmaras turbulence model

In this section the Spalart-Allmaras turbulence model will be discussed. First the Boussinesq approximation for the Reynolds stress tensor will be addressed. Then the turbulence model will be introduced, after which some modifications for the implementation of the model will be presented.

2.3.1 Boussinesq approximation

To deal with the closure problem of the RANS equations, Boussinesq proposed relating the Reynolds stress tensor to the mean flow. The Boussinesq approximation leads to linear eddy viscosity models and reads as,

$$-\rho \overline{\tilde{\mathbf{u}} \overline{\tilde{\mathbf{u}}}} = 2\mu_t \overline{\epsilon} - \frac{2}{3}\rho k \mathbf{I}$$
$$= \mu_t \left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T\right) - \frac{2}{3}\rho k \mathbf{I}$$

with μ_t the turbulent or eddy viscosity and k the the turbulent kinetic energy. Note that the eddy viscosity is not a property of the fluid but is dependent on the turbulence of the flow at (\mathbf{x}, t) . The closure problem is now solved if a solution for the turbulent viscosity can be obtained. The linear eddy viscosity model used here is the one-equation Spalart-Allmaras turbulence model. Since this model does not solve for k, the inclusion of this term is not required. The Boussinesq approximation then becomes,

$$-\rho \overline{\tilde{\mathbf{u}}} \overline{\tilde{\mathbf{u}}} = \mu_t \left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T \right) .$$

Note that this expression for the turbulent viscosity is directly analogous to the relation for the viscous stress in a Newtonian fluid. With this approximation for the Reynolds stress tensor equation (2.15) can be written as,

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \,\bar{\mathbf{u}} = -\frac{1}{\rho} \nabla \bar{p} + \nabla \cdot \left[(\nu + \nu_t) \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right) \right] + \bar{\mathbf{f}} \quad \text{in } \Omega \times]0, T[. \quad (2.16)$$

Again for a uniform density field $\rho = 1$ we have,

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \,\bar{\mathbf{u}} = -\nabla \bar{p} + \nabla \cdot \left[(\nu + \nu_t) \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right) \right] + \bar{\mathbf{f}} \quad \text{in } \Omega \times]0, T[. \qquad (2.17)$$

2.3.2 The baseline model

The Spalart-Allmaras (S-A) turbulence model, introduced in [61], is a popular choice for turbulence modeling due to its competitive accuracy and relatively low cost [13]. The model directly computes the evolution of ν_t and was designed from empiricism and dimensional analysis. The equation has advection, diffusion and reaction terms, and it can be noted that the advection-diffusion-reaction equation is a model problem for the Spalart-Allmaras turbulence model [51]. This characteristic will be made use of in chapter 4 for applying multiscale stabilization to this equation.

The S-A turbulence model does not solve directly for the turbulent viscosity ν_t , but rather for the modified turbulent viscosity ν_e . We introduce the von Kármán constant κ and the friction velocity u_{τ} . ν_e is scaled such that it equals $\kappa y u_{\tau}$ in the intermediate layer as well as the buffer and viscous sublayer (refer to appendix A for an explanation on boundary layer terminology). This is beneficial for numerical solutions, since ν_e varies linearly with the distance from the wall. In this way its behavior can be captured on relatively coarse meshes.

The S-A turbulence equation has different variants, among which models with and without trip terms. Including the trip terms one can set where in the flow field the flow should be tripped and become turbulent. For our application we do not need the use of a trip term since we are examining fully turbulent flows, where the flow is turbulent anywhere vorticity is present. Next to this we are not interested in the exact physical behavior of the flow. We are rather interested in the stability of the derived formulation for coarser grids while preserving accuracy. The fully turbulent model will thus be employed for all cases. This model is given by the following equation,

$$\frac{\partial \nu_e}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \nu_e - c_{b1} S_e \nu_e + c_{w1} f_w \left(\frac{\nu_e}{d}\right)^2 -\frac{1}{\sigma} \left[\nabla \cdot \left(\left(\nu + \nu_e\right) \nabla \nu_e\right) + c_{b2} \nabla \nu_e \cdot \nabla \nu_e\right] = 0 \quad \text{in } \Omega \times \left]0, T\right[.$$
(2.18)

We note the first two terms make up the material derivative of ν_e . The next two terms are reaction terms and model production and destruction, respectively. Finally, the last two terms model the conservative and non-conservative diffusion effects, respectively. The turbulent kinematic viscosity can be obtained from,

$$\nu_t = \nu_e f_{v1}$$

with

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$
 and $\chi = \frac{\nu_e}{\nu}$.

The magnitude of vorticity S is modified to S_e such that again the log-law behavior in the intermediate layer extends all the way to the wall. It is argued by Spalart and Allmaras that vorticity is used for the production term because turbulence is only present where vorticity is. Both of them emanating from solid boundaries. The modified magnitude of vorticity is defined as,

$$S_e = S + \frac{\nu_e}{\kappa^2 d^2} f_{v2}$$

with the magnitude of the vorticity,

$$S = \sqrt{2\mathbf{\Omega}:\mathbf{\Omega}}, \qquad \mathbf{\Omega} = \frac{1}{2} \left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T \right)$$

Here Ω is the rotation rate tensor, d is the distance to the nearest wall and $\Omega : \Omega = \Omega_{ij}\Omega_{ij}$ indicates the Frobenius inner product of two matrices. The function f_{v2} is defined as

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \,.$$

The function f_w is used as a calibration for the destruction term since it was found to decay too slow in the outer region of the boundary layer. It equals 1 in the intermediate or log layer and decays towards the outer part of the boundary layer, this is shown in figure 2.2. f_w is given as,

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}}$$

with,

$$g = r + c_{w2} \left(r^6 - r \right)$$
 and $r = \min \left[\frac{\nu_e}{\kappa^2 d^2 S_e}, 10 \right].$

The closure coefficients of the model are:

$$\begin{aligned} c_{b1} &= 0.1355 \,, \quad c_{b2} &= 0.622 \,, \quad c_{v1} &= 7.1 \,, \quad \sigma = \frac{2}{3} \,, \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma} \,, \quad c_{w2} &= 0.3 \,, \quad c_{w3} &= 2 \,, \quad \kappa = 0.41 \,. \end{aligned}$$



Figure 2.2: f_w plotted against r. APG: adverse pressure gradient; FPG: favorable pressure gradient [2].

2.3.3 Modifications to the baseline model

Some modifications to the baseline S-A turbulence model are applied, and are taken from [2]. These modifications are implemented in order to account for numerical problems when dealing with under-resolved grids and non-physical transient states. Using these modifications should result in negligible differences with the original model when using a well resolved grid in most cases. Two modifications to the original model are used here: the first one preventing negative values of the modified magnitude of vorticity S_e , the second one deals with negative values of the modified turbulent kinematic viscosity ν_e .

Preventing negative values of S_e

A value of S_e below 0.3S is non-physical, but due to the fact that f_{v2} is negative over a range of χ it can reach lower values and even become negative. This can cause numerical issues due to the disruption of other S-A correlation functions. The modified version of S_e is equal to the original for $S_e > 0.3S$, but remains positive when S is nonzero. We define \overline{S} as,

$$\overline{S} = \frac{\nu_e}{\kappa^2 d^2} f_{v2} \,.$$

The modified version of S_e is then,

$$S_e = \begin{cases} S + \overline{S} & \text{for } \overline{S} \ge -c_{v2}S \\ S + \frac{S(c_{v2}^2 S + c_{v3}\overline{S})}{(c_{v3} - 2c_{v2})S - \overline{S}} & \text{for } \overline{S} < -c_{v2}S \end{cases}$$



with $c_{v2} = 0.7$ and $c_{v3} = 0.9$. This new function is plotted in figure 2.3 and compared with the original definition of S_e .

Figure 2.3: Modified magnitude of vorticity [2].

Negative S-A model

The solution of the baseline S-A model may be negative for coarse grids and certain transient states. The original model is however only meant to admit non-negative solutions, given non-negative boundary and initial conditions. A negative solution is often encountered at the edge of boundary layers and wakes. For discretized domains the large gradients in the solution here can cause undershoots. If these undershoots are below zero some action is required to eliminate the negative solution.

The method proposed in [2] is a continuation of the S-A model for negative ν_e solutions to deal with these undershoots. The primary purpose again is to deal with under-resolved grids and non-physical transient states. When the solution is zero or positive, the original model is used. If the solution is negative however, a different PDE is solved and the turbulent viscosity ν_t is set to zero. ν_e thus becomes a passive scalar since it does not influence the value of ν_t directly. The new PDE stems from the requirements of energy stability and C^1 continuity for the PDE terms at $\nu_e = 0$. The negative S-A model reads as,

$$\frac{\partial \nu_e}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \nu_e - c_{b1} S \nu_e - c_{w1} \left(\frac{\nu_e}{d}\right)^2 - \frac{1}{\sigma} \left[\nabla \cdot \left(\left(\nu + \nu_e f_n\right) \nabla \nu_e\right) + c_{b2} \nabla \nu_e \cdot \nabla \nu_e\right] = 0 \quad (2.19)$$
with,

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}$$
and $c_{n1} = 16$. Note that the production term is now defined in terms of S instead of S_e . Next to this the destruction term now has the opposite sign compared to the baseline model. Finally, $f_n(\chi)$ modifies the diffusion coefficient to $(\nu + \nu_e f_n)$. Dividing by ν the non-dimensional diffusion coefficient $(1 + \chi f_n)$ can be plotted against χ . This is done in figure 2.4 on the right, and it is shown the diffusion coefficient does not turn negative anymore because of this modification. On the left side a plot of f_n versus χ is provided.



Figure 2.4: f_n and diffusion coefficient versus χ for negative S-A model [2].

Chapter 3

Finite Element Method for the Navier-Stokes equations

In the previous chapter the governing equations in their so-called strong form have been presented. In this chapter the weak or variational form of the governing equations is first derived, as part of the application of the finite element method to the problem. Some spaces of functions will be introduced here too, as well as changes in the mathematical notation. Note that for the formulation and mathematical notation of the problem we mostly follow standard finite element books such as those by Donea and Huerta [27] and Hughes [43].

Before the spatial domain is discretized first a temporal discretization is performed, as well as an iterative solution procedure to solve the semi-discretized nonlinear problem. For the discretization in time the theta method is used, while for the iterative solution procedure Newton's method is employed.

Focus will then be placed on the Galerkin method, which is used to convert the continuous domain into a discrete domain. The approximations of the continuous spaces are introduced and from there the problem can be derived in matrix form. The element-wise contributions to the global matrix in normalized element coordinates are discussed in the last part of this section.

Finally, attention is payed to the problems resulting from the Galerkin method. Spurious oscillations in the solution can occur if convection dominates the problem. Next to this, the saddle-point problem that arises from the weak formulation of the Navier-Stokes equations can cause numerical difficulties. A properly stabilized finite element formulation will reduce the likelihood of numerical instabilities. The stabilization techniques that will be addressed and used are the least-squares on incompressibility constraint (LSIC), streamline upwind/Petrov-Galerkin (SUPG) and pressure stabilization procedures stem from the variational multiscale concept, introduced in the next chapter.

3.1 Weak form of the problem

To apply the finite element method (FEM) the problem needs to be put in the weak form first. This can be done with the method of weighted residuals. We multiply the residual of the N-S momentum equation by a weighting function $\mathbf{w}(\mathbf{x})$, integrate this equation over the domain Ω , and require it to be zero for all suitable weighting functions. The same is done for the continuity equation, but instead it is multiplied by the scalar weighting function $q(\mathbf{x})$. The weighting functions are also called test functions since they provide a test for the local residual. By integrating over the domain the problem is relaxed, we are only finding a solution that satisfies the strong form in a distributional sense. For the Navier-Stokes equations (2.1) and (2.7) we have,

$$\begin{cases} \int_{\Omega} \mathbf{w} \cdot \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \, \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p - \mathbf{f} \right) d\Omega = 0 \\ \int_{\Omega} q \left(\nabla \cdot \mathbf{u} \right) d\Omega = 0 \,. \end{cases}$$
(3.1)

We can now use integration by parts in order to reduce the continuity requirement of the solution. Integration by parts can usefully be applied to the diffusion and pressure term as follows.

$$-\nu \int_{\Omega} \mathbf{w} \cdot \nabla^2 \mathbf{u} \, d\Omega = -\nu \int_{\Gamma} (\mathbf{n} \cdot \nabla) \, \mathbf{u} \cdot \mathbf{w} \, d\Gamma + \nu \int_{\Omega} \nabla \mathbf{w} : \nabla \mathbf{u} \, d\Omega \tag{3.2}$$

$$\int_{\Omega} \mathbf{w} \cdot \nabla p \, d\Omega = \int_{\Gamma} \mathbf{w} \cdot p \mathbf{n} \, d\Gamma - \int_{\Omega} \nabla \cdot \mathbf{w} \, p \, d\Omega \tag{3.3}$$

Before substituting, let us first introduce the following notation. We define the operation (\cdot, \cdot) as the L^2 -inner product over the domain Ω , i.e., for two vector functions **w** and **u** we have,

$$(\mathbf{w},\mathbf{u}) = \int_{\Omega} \mathbf{w} \cdot \mathbf{u} \, d\Omega$$
.

Notice that the arguments inside the brackets determine if the operation is either a scalar product, a dot product or a Frobenius inner product. An integral over a different domain than Ω can be indicated by a subscript.

Now substituting (3.2) and (3.3) into (3.1), and using the notation introduced yields,

$$\begin{pmatrix} \mathbf{w}, \frac{\partial \mathbf{u}}{\partial t} \end{pmatrix} + (\mathbf{w}, (\mathbf{u} \cdot \nabla) \mathbf{u}) + (\nabla \mathbf{w}, \nu \nabla \mathbf{u}) - (\nabla \cdot \mathbf{w}, p)$$

$$= (\mathbf{w}, \mathbf{f}) + (\mathbf{w}, (-p\mathbf{n} + \nu (\mathbf{n} \cdot \nabla) \mathbf{u}))_{\Gamma} .$$

$$(3.4)$$

Notice that this formulation imposes weaker conditions on the smoothness of \mathbf{u} and p compared to the strong form (3.1).

Next, we can take a look at the boundary term. On the Dirichlet portion of the boundary Γ_D the solution is prescribed, hence it is not necessary to test the residual here since the solution is known. We can thus take $\mathbf{w} = 0$ on Γ_D . The Neumann boundary condition does not specify the solution for \mathbf{u} , but its derivative. Hence we do need to test the residual here. Note how the Neumann boundary condition naturally arises in the formulation, it is therefore referred to as natural boundary condition. We can rewrite equation (3.4) using the pseudo-traction \mathbf{t}' given in equation (2.9), and introduce the

spaces of vector and scalar functions \mathcal{S} , \mathcal{V} and \mathcal{Q} . The weak form problem statement can be formulated as follows. Given \mathbf{u}_D , \mathbf{t}' and \mathbf{f} , find $\mathbf{u}(\mathbf{x},t) \in \mathcal{S} \times]0, T[$ and $p(\mathbf{x},t) \in \mathcal{Q} \times]0, T[$ such that,

$$\begin{cases} \left(\mathbf{w}, \frac{\partial \mathbf{u}}{\partial t}\right) + \left(\mathbf{w}, \left(\mathbf{u} \cdot \nabla\right) \mathbf{u}\right) + \left(\nabla \mathbf{w}, \nu \nabla \mathbf{u}\right) - \left(\nabla \cdot \mathbf{w}, p\right) = \left(\mathbf{w}, \mathbf{f}\right) + \left(\mathbf{w}, \mathbf{t}'\right)_{\Gamma_N} \\ \left(q, \nabla \cdot \mathbf{u}\right) = 0 \end{cases}$$
(3.5)

for all $(\mathbf{w}, q) \in \mathcal{V} \times \mathcal{Q}$.

Let us describe the spaces of functions. \boldsymbol{S} is the vector space of trial solutions and contains functions of the velocity \mathbf{u} . Since the Dirichlet boundary conditions are given for a problem and we do not test the residual here, these functions must satisfy a priori the boundary conditions on Γ_D . The trial solution space is given as,

$$\boldsymbol{\mathcal{S}} := \left\{ \mathbf{u} \in \boldsymbol{\mathcal{H}}^1 \, | \, \mathbf{u} = \mathbf{u}_D \; \; \mathrm{on} \; \; \Gamma_D
ight\}$$

where \mathcal{H}^1 is a Sobolev space of order 1. It consists of all functions in the domain Ω for which the functions itself, as well as its derivative are square integrable. It is important that **u** belongs to this space such that the all the integrals, especially the one over the diffusive term, in the N-S equations are well defined. The Sobolev space of order 1 for **u** is defined as,

$$\mathcal{H}^1(\Omega) = \{ \mathbf{u} \in L_2; \nabla \mathbf{u} \in L_2 \}$$

with

$$L_2(\Omega) \equiv \mathcal{H}^0 = \left\{ \int_{\Omega} \mathbf{u} \cdot \mathbf{u} \, d\Omega < \infty \right\} \,.$$

The vector space of weighting or test functions is termed by $\boldsymbol{\mathcal{V}}$. It has the same characteristics as $\boldsymbol{\mathcal{S}}$, but, as explained before, the test functions equal 0 on the Dirichlet portion of the boundary. Not that for $\mathbf{u}_D = 0$ the trial and test spaces are the same. Symbolically the vector space $\boldsymbol{\mathcal{V}}$ is defined by,

$$\boldsymbol{\mathcal{V}} := \left\{ \mathbf{w} \in \boldsymbol{\mathcal{H}}^1 \, | \, \mathbf{w} = 0 \; \; \mathrm{on} \; \; \Gamma_D
ight\}$$

The space of functions for the pressure is denoted by Q. It is only required to be square integrable, since its spatial derivative does not appear in the weak form (3.5). Moreover, since there are no boundary conditions on pressure, a single space suffices as trial and weighting function space. Q is defined as,

$$\mathcal{Q} := L_2(\Omega)$$

Now let us introduce the following bi- and trilinear forms:

$$a\left(\mathbf{w},\mathbf{u}\right) = \int_{\Omega} \nabla \mathbf{w} : \nu \nabla \mathbf{u} \, d\Omega \qquad \forall \mathbf{u}, \mathbf{w} \in \mathcal{H}^{1}(\Omega)$$
(3.6)

$$b(\mathbf{u},q) = \int_{\Omega} q \nabla \cdot \mathbf{u} \, d\Omega \qquad \forall \mathbf{u} \in \mathcal{H}^1(\Omega) \text{ and } q \in L_2(\Omega)$$
 (3.7)

$$c(\mathbf{v}; \mathbf{w}, \mathbf{u}) = \int_{\Omega} \mathbf{w} \cdot (\mathbf{v} \cdot \nabla) \mathbf{u} \, d\Omega \qquad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{H}^{1}(\Omega)$$
(3.8)

We can write equation (3.5) with these new forms as,

$$\begin{cases} \left(\mathbf{w}, \frac{\partial \mathbf{u}}{\partial t}\right) + c\left(\mathbf{u}; \mathbf{w}, \mathbf{u}\right) + a\left(\mathbf{w}, \mathbf{u}\right) - b\left(\mathbf{w}, p\right) = (\mathbf{w}, \mathbf{f}) + \left(\mathbf{w}, \mathbf{t}'\right)_{\Gamma_N} \\ b\left(\mathbf{u}, q\right) = 0. \end{cases}$$
(3.9)

The proof of existence and uniqueness of weak solutions for (3.9) can be found in theoretical books on the Navier-Stokes equations such as [62].

3.2 Temporal discretization and iterative solution procedure

Before continuing with a discretization of the spatial domain, let us apply a discretization for the temporal term first. Then the solution procedure for solving the continuous nonlinear problem is set up.

3.2.1 Theta method

For the temporal discretization of the problem the theta method is used. For a function $\frac{dy}{dt} = f(y)$ it has the general form of,

$$\frac{dy}{dt} \approx \frac{y - \hat{y}}{\Delta t} = \theta f(y) + (1 - \theta) f(\hat{y})$$

with Δt the time step and $0 \leq \theta \leq 1$ determines whether the method is explicit or implicit. The term \hat{y} indicates the value of y at the previous time step, while y indicates its value at the new time step.

It is noted that choosing $\theta \neq 0$ yields an implicit method. Implicit methods usually allow for larger time steps. There is less restriction on the magnitude of the time step compared to explicit methods in order to have a stable numerical solution. This is especially the case for stiff equations such as the N-S equations.

For our range of problems we will limit ourselves to the use of $\theta = 1$, hence obtaining the backward Euler method. This method is only first order accurate in time, while the Cranck-Nicolson method $(\theta = \frac{1}{2})$ has second-order accuracy. However, this last method does not handle problems with discontinuous initial data well, and can have oscillatory results. It is therefore preferred to sacrifice accuracy in time in favor of stability. It can thus be stated that the backward Euler method is a simple yet robust method for solving the N-S equations.

For the application of the theta method to equations (3.9) let us assume that the external source term and the boundary conditions are independent of time. This yields,

$$\begin{cases} \left(\mathbf{w}, \frac{\mathbf{u} - \hat{\mathbf{u}}}{\Delta t}\right) + \theta \left(c \left(\mathbf{u}; \mathbf{w}, \mathbf{u}\right) + a \left(\mathbf{w}, \mathbf{u}\right) - b \left(\mathbf{w}, p\right)\right) \\ + (1 - \theta) \left(c \left(\hat{\mathbf{u}}; \mathbf{w}, \hat{\mathbf{u}}\right) + a \left(\mathbf{w}, \hat{\mathbf{u}}\right) - b \left(\mathbf{w}, \hat{p}\right)\right) \\ = \left(\mathbf{w}, \mathbf{f}\right) + \left(\mathbf{w}, \mathbf{t}'\right)_{\Gamma_{N}} \\ b \left(\mathbf{u}, q\right) = 0. \end{cases}$$
(3.10)

Note that the theta method is not applied to the continuity equation since it does not contain a time derivative. This will change however when stabilization is added to this equation in section 3.4.

3.2.2 Newton's method

The nonlinear problem given by equations (3.10) can be solved using a nonlinear iteration scheme such as Newton's method. For each Newton iteration a linear system of equations is solved, until the solution has converged. A characteristic of this method is that is has quadratic convergence under the right circumstances, one of which is a close enough initial guess (\mathbf{u}^0, p^0) of the nonlinear solution. A bad initial guess can lead to the system not converging at all. An often used method to ensure a close enough initial guess for the first time step is to solve Stokes equations (i.e., the steady N-S equations without convective term) for the problem first, and use its solution as initial guess [60]. If the Reynolds number is too high the Stokes solution may not be close enough to the Navier-Stokes solution. The Navier-Stokes equations can then be solved for a smaller Reynolds number and use its solution as initial guess. Throughout this work this kind of solution procedure will be used.

Newton's method for (3.10) reads: given an initial guess $\mathbf{U}^0 \in \boldsymbol{S} \times \boldsymbol{Q}$, we seek $\delta \mathbf{U} \in \boldsymbol{S} \times \boldsymbol{Q}$ for $k = 0, 1, \ldots$ until convergence such that,

$$\mathbf{J}\left(\mathbf{U}^{k}\right).\delta\mathbf{U} = -\mathcal{R}\left(\mathbf{U}^{k}\right)$$
(3.11)

and set $\mathbf{U}^{k+1} = \mathbf{U}^k + \delta \mathbf{U}$, with $\delta \mathbf{U}$ the difference in \mathbf{U} between an iteration. Next to this we have: $\mathbf{U} = \begin{bmatrix} \mathbf{u} & p \end{bmatrix}^T$, k is the Newton iteration number, $\mathbf{J} (\mathbf{U}^k)$ is the Jacobian matrix with respect to \mathbf{U}^k , and $\mathcal{R} (\mathbf{U}^k)$ is the residual of equation (3.10) at Newton iteration k, i.e.,

$$\begin{cases} \boldsymbol{\mathcal{R}}_{\theta,m}(\mathbf{u}^{k},p^{k}) = \left(\mathbf{w},\frac{\mathbf{u}^{k}-\hat{\mathbf{u}}}{\Delta t}\right) + \theta\left(c\left(\mathbf{u}^{k};\mathbf{w},\mathbf{u}^{k}\right) + a\left(\mathbf{w},\mathbf{u}^{k}\right) - b\left(\mathbf{w},p^{k}\right)\right) \\ + (1-\theta)\left(c\left(\hat{\mathbf{u}};\mathbf{w},\hat{\mathbf{u}}\right) + a\left(\mathbf{w},\hat{\mathbf{u}}\right) - b\left(\mathbf{w},\hat{p}\right)\right) \\ - (\mathbf{w},\mathbf{f}) - \left(\mathbf{w},\mathbf{t}'\right)_{\Gamma_{N}} \\ \boldsymbol{\mathcal{R}}_{\theta,c}(\mathbf{u}^{k},p^{k}) = b\left(\mathbf{u}^{k},q\right). \end{cases}$$
(3.12)

with $\mathcal{R}_{\theta,m}$ and $\mathcal{R}_{\theta,c}$ the residual of the momentum and continuity equation after application of the theta method, respectively. If convergence is achieved for some k, set \mathbf{U}^k equal to \mathbf{U} at the new time step. A new nonlinear loop can then be started for this new time step. Note that any terms containing an unknown (i.e., a variable at iteration k + 1) will be kept on the left-hand-side, whereas all other terms will be taken to the right-hand-side vector. The system of equations that is solved each iteration is thus,

$$\mathbf{J}\left(\mathbf{U}^{k}\right).\mathbf{U}^{k+1} = -\mathcal{R}\left(\mathbf{U}^{k}\right) + \mathbf{J}\left(\mathbf{U}^{k}\right).\mathbf{U}^{k}$$
(3.13)

For a vector-valued function $\mathbf{A}(\mathbf{U}) = 0$ the Jacobian matrix is defined as,

$$J_{ij}(\mathbf{U}) = \frac{\partial A_i}{\partial U_j}(\mathbf{U}).$$
(3.14)

Note the application of the derivative to a continuous function, we thus have that the Jacobian is the Fréchet derivative. The Fréchet derivative is defined as (see, e.g., [16]),

$$\lim_{\delta \mathbf{U} \to 0} \frac{||\mathbf{A}(\mathbf{U} + \delta \mathbf{U}) - \mathbf{A}(\mathbf{U}) - \frac{\partial \mathbf{A}}{\partial \mathbf{U}} \, \delta \mathbf{U}||}{||\delta \mathbf{U}||} = 0.$$
(3.15)

Taking a look at Fréchet derivative applied to the nonlinear convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ at the current time step we get the following. Find a linear operator **B** such that,

$$\lim_{\delta \mathbf{u} \to 0} \frac{\|\left(\left(\mathbf{u}^{k} + \delta \mathbf{u}\right) \cdot \nabla\right) \left(\mathbf{u}^{k} + \delta \mathbf{u}\right) - \left(\mathbf{u}^{k} \cdot \nabla\right) \mathbf{u}^{k} - \mathbf{B} \,\delta \mathbf{u}\||}{\|\delta \mathbf{u}\|} = 0$$
$$\lim_{\delta \mathbf{u} \to 0} \frac{\|\left(\mathbf{u}^{k} \cdot \nabla\right) \delta \mathbf{u} + \left(\delta \mathbf{u} \cdot \nabla\right) \mathbf{u}^{k} + \left(\delta \mathbf{u} \cdot \nabla\right) \delta \mathbf{u} - \mathbf{B} \,\delta \mathbf{u}\||}{\|\delta \mathbf{u}\|} = 0.$$

It can be observed that the linear operator **B** must be set such that,

$$\mathbf{B}\,\delta\mathbf{u} = \left(\mathbf{u}^k\cdot\nabla\right)\delta\mathbf{u} + \left(\delta\mathbf{u}\cdot\nabla\right)\mathbf{u}^k$$

where we have ignored the term $\mathcal{O}(\delta \mathbf{u})$. We call **B** the Fréchet derivative of the convective term. Taking into account that $\delta \mathbf{u} = \mathbf{u}^{k+1} - \mathbf{u}^k$, the contribution of the current time step convective term to the left-hand-side of equation (3.11) can be seen to equal,

$$\frac{\partial c(\mathbf{u}^k; \mathbf{w}, \mathbf{u}^k)}{\partial \mathbf{u}^k} \left(\mathbf{u}^{k+1} - \mathbf{u}^k \right) = \left(c(\mathbf{u}^{k+1}; \mathbf{w}, \mathbf{u}^k) + c(\mathbf{u}^k; \mathbf{w}, \mathbf{u}^{k+1}) - 2 c(\mathbf{u}^k; \mathbf{w}, \mathbf{u}^k) \right)$$

where the last term is is known and stored in the right-hand-side vector. Since terms from the previous time step are also known and thus deemed constant, their derivative is zero and they only contribute through the residual vector.

Application of Newton's method, (3.11), to the form of the N-S equations in (3.10) yields,

$$\begin{cases} \left(\mathbf{w}, \frac{\mathbf{u}^{k+1} - \mathbf{u}^{k}}{\Delta t}\right) + \theta \left\{ c(\mathbf{u}^{k+1}; \mathbf{w}, \mathbf{u}^{k}) + c(\mathbf{u}^{k}; \mathbf{w}, \mathbf{u}^{k+1}) - 2c(\mathbf{u}^{k}; \mathbf{w}, \mathbf{u}^{k}) + a(\mathbf{w}, \mathbf{u}^{k+1} - \mathbf{u}^{k}) - b(\mathbf{w}, p^{k+1} - p^{k}) \right\} = -\mathcal{R}_{\theta,m}(\mathbf{u}^{k}, p^{k}) \\ b(\mathbf{u}^{k+1} - \mathbf{u}^{k}, q) = -\mathcal{R}_{\theta,c}(\mathbf{u}^{k}). \end{cases}$$
(3.16)

Eliminating terms we obtain,

$$\begin{cases} \left(\mathbf{w}, \mathbf{u}^{k+1}\right) + \Delta t \,\theta \left\{ c\left(\mathbf{u}^{k+1}; \mathbf{w}, \mathbf{u}^{k}\right) + c\left(\mathbf{u}^{k}; \mathbf{w}, \mathbf{u}^{k+1}\right) + a\left(\mathbf{w}, \mathbf{u}^{k+1}\right) \\ -b\left(\mathbf{w}, p^{k+1}\right) \right\} = \left(\mathbf{w}, \hat{\mathbf{u}}\right) + \Delta t \left\{ \theta \, c\left(\mathbf{u}^{k}; \mathbf{w}, \mathbf{u}^{k}\right) + (1 - \theta) \left(-c\left(\hat{\mathbf{u}}; \mathbf{w}, \hat{\mathbf{u}}\right) \\ -a\left(\mathbf{w}, \hat{\mathbf{u}}\right) + b\left(\mathbf{w}, \hat{p}\right) \right) + \left(\mathbf{w}, \mathbf{f}\right) + \left(\mathbf{w}, \mathbf{t}'\right)_{\Gamma_{N}} \right\} \end{cases}$$
(3.17)
$$b\left(\mathbf{u}^{k+1}, q\right) = 0.$$

The momentum equation has been multiplied by Δt and known terms (from the previous time step and previous Newton iteration) have been collected on the right-hand-side.

3.3 Galerkin and matrix formulation

In the finite element method the domain is discretized by approximating the infinite dimensional spaces \mathcal{S} , \mathcal{V} and \mathcal{Q} with finite dimensional subspaces denoted by \mathcal{S}_h , \mathcal{V}_h and \mathcal{Q}_h , respectively. This leads to the so called Galerkin formulation of the problem. The Galerkin formulation associated with the weak form of equation (3.17) leads to a mixed finite element method, as local approximations for the velocity, pressure and their weighing functions are introduced. The finite dimensional local approximations of the known and unknown variables are denoted by the subscript h. Note that $\mathbf{u}_h \in \mathcal{S}_h \subset \mathcal{S}$, $\mathbf{w}_h \in \mathcal{V}_h \subset \mathcal{V}$ and $(p_h, q_h) \in \mathcal{Q}_h \subset \mathcal{Q}$. It can be shown that the error due to these approximations is orthogonal to the subspace, and that a near-best fit of the exact solution is thus found. This is called Galerkin orthogonality and it is the key property of the Galerkin approach.

The Galerkin formulation of the problem, equation (3.17), can now be obtained by restricting the weak form to the finite dimensional interpolating spaces. It is given as: find the velocity field $\mathbf{u}_h \in \mathcal{S}_h \times]0, T[$ and pressure $p_h \in \mathcal{Q}_h \times]0, T[$, for all $(\mathbf{w}_h, q_h) \in \mathcal{V}_h \times \mathcal{Q}_h$, such that,

$$\begin{cases} \left(\mathbf{w}_{h}, \mathbf{u}_{h}^{k+1}\right) + \Delta t \,\theta \left\{ c\left(\mathbf{u}_{h}^{k+1}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k}\right) + c\left(\mathbf{u}_{h}^{k}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k+1}\right) + a\left(\mathbf{w}_{h}, \mathbf{u}_{h}^{k+1}\right) \\ -b\left(\mathbf{w}_{h}, p_{h}^{k+1}\right) \right\} = \left(\mathbf{w}_{h}, \hat{\mathbf{u}}_{h}\right) + \Delta t \left\{ \theta \, c\left(\mathbf{u}_{h}^{k}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k}\right) \\ + \left(1 - \theta\right) \left(- c\left(\hat{\mathbf{u}}_{h}; \mathbf{w}_{h}, \hat{\mathbf{u}}_{h}\right) - a\left(\mathbf{w}_{h}, \hat{\mathbf{u}}_{h}\right) + b\left(\mathbf{w}_{h}, \hat{p}_{h}\right) \right) \\ + \left(\mathbf{w}_{h}, \mathbf{f}_{h}\right) + \left(\mathbf{w}_{h}, \mathbf{t}_{h}'\right)_{\Gamma_{N}} \right\} \end{cases}$$
(3.18)
$$b\left(\mathbf{u}_{h}^{k+1}, q_{h}\right) = 0.$$

The domain Ω now has to be viewed as discretized into elements Ω^e , with $1 \leq e \leq n_{el}$ and n_{el} the number of elements. These elements are non-overlapping, have a piecewise smooth boundary $\Gamma^e = \partial \Omega^e$ and h is their characteristic mesh size. Each element has its own number of nodes depending on the type of element and its dimension. Let us denote by $\eta = \{1, 2, \ldots, n_{np}\}$ the set of global velocity nodes in the finite element mesh, and n_{np} the total number of velocity nodal points. Next to this, the subset of velocity nodes on the Dirchlet portion of the boundary is denoted by $\eta_{Di} \subset \eta$, with *i* the velocity component.

The finite element interpolating spaces contain piecewise polynomials which can be linear, quadratic, etc. This depends on the choice of shape or basis functions. These local functions take on the value 1 at their respective node, and are 0 for all other nodes. An example is given in figure 3.1 for a 1-dimensional case with a piecewise linear basis function. Here x_A indicates global node A, and ϕ_A is the basis function for global node A. The local velocity approximation can be expressed using basis functions as,

$$u_{hi}(\mathbf{x}) = \sum_{A \in \eta} \phi_A(\mathbf{x}) \mathbf{u}_{iA}$$
(3.19)

again with ϕ_A the basis function with respect to global velocity node number A, and u_{iA} the nodal velocity unknown with respect to spatial component *i*.



Figure 3.1: Piecewise linear basis functions in 1-dimensional space.

Let us define the canonical basis in $\mathbb{R}^{n_{sd}}$, with the number of spatial dimension $n_{sd} = 2$. The canonical basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_{n_{sd}}\}$ for two dimensions is defined as,

$$\mathbf{e}_1 = (1,0)^T, \qquad \mathbf{e}_2 = (0,1)^T.$$
 (3.20)

The array version of the velocity approximation is then,

$$\mathbf{u}_{h}(\mathbf{x},t) = \sum_{i=1}^{n_{sd}} u_{hi}(\mathbf{x}) \mathbf{e}_{i} = \sum_{i=1}^{n_{sd}} \sum_{A \in \eta} \phi_{A}(\mathbf{x}) \mathbf{u}_{iA} \mathbf{e}_{i}$$
(3.21)

The arbitrary weighting functions in the Galerkin formulation are defined such that,

$$w_{hi} \in \mathcal{V}_{hi} := \sup_{A \in \eta \setminus \eta_{Di}} \left\{ \phi_A \right\}.$$
(3.22)

The pressure field can possibly be approximated with a different set of nodes $\hat{\eta}$ as,

$$p_h(\mathbf{x}, t) = \sum_{\hat{A} \in \hat{\eta}} \varphi_{\hat{A}}(\mathbf{x}) \mathbf{p}_{\hat{A}}$$
(3.23)

where $\varphi_{\hat{A}}$ is the basis function for the pressure with respect to global pressure node number \hat{A} , and $p_{\hat{A}}$ is the value of the pressure at this node. Finally, the pressure weighting function q_h is defined as,

$$q_h \in \mathcal{Q}_h := \operatorname{span}_{\hat{A} \in \hat{\eta}} \{ \varphi_{\hat{A}} \}.$$

The matrix problem can formulated by first substituting the above expressions for the trial and weighting functions into the Navier-Stokes Galerkin form (3.18). To keep notation slightly shorter and keeping the more trivial terms out of the matrix form, let us assume $\theta = 1$ to employ backward Euler for the time stepping. Note that \mathbf{u}_h has not been split into an auxiliary velocity field and a variable containing the Dirichlet boundary conditions, as is sometimes customary, thus $\mathbf{u}_h \in \boldsymbol{S}_h$. However, application of non-homogeneous Dirichlet boundary conditions in libMesh is done through use of the penalty method. This method is explained in appendix B. The following set of nodal equations is obtained for the momentum equation: for every $A \in \eta \setminus \eta_{Di}$ and $1 \leq i \leq n_{sd}$,

$$\sum_{j=1}^{n_{sd}} \left\{ \sum_{B \in \eta \setminus \eta_{Dj}} \left[\left(\phi_A \mathbf{e}_i, \phi_B \mathbf{e}_j \right) \mathbf{u}_{jB} + \Delta t \left(c \left(\phi_B \mathbf{e}_i; \phi_A \mathbf{e}_i, \mathbf{u}_h^k \right) + c \left(\mathbf{u}_h^k; \phi_A \mathbf{e}_i, \phi_B \mathbf{e}_j \right) \right. \\ \left. + a(\phi_A \mathbf{e}_i, \phi_B \mathbf{e}_j) \right) \mathbf{u}_{jB} \right] \right\} - \Delta t \sum_{\hat{B} \in \hat{\eta}} b(\phi_A \mathbf{e}_i, \varphi_{\hat{B}}) \mathbf{p}_{\hat{B}} = (\phi_A \mathbf{e}_i, \hat{\mathbf{u}}_h) \\ \left. + \Delta t \left(c \left(\mathbf{u}_h^k; \phi_A \mathbf{e}_i, \mathbf{u}_h^k \right) + \left(\phi_A \mathbf{e}_i, \mathbf{f}_h \right) + \left(\phi_A \mathbf{e}_i, \mathbf{t}_h' \right)_{\Gamma_N} \right) \right] \right\}$$
(3.24)

Notice that the first convective term is non-zero also when the spatial components are unequal, while the other terms are only non-zero for equal spatial components.

Next the nodal equations for the continuity equation can be obtained: for each $A \in \hat{\eta}$ and $1 \leq i \leq n_{sd}$,

$$\sum_{B \in \eta \setminus \eta_{Di}} b(\phi_B \mathbf{e}_i, \varphi_{\hat{A}}) \mathbf{u}_{iB} = 0.$$
(3.25)

The layout of the resulting matrix can be found from these equations and is as follows,

$$\begin{pmatrix} (\mathbf{M} + \mathbf{K} + \mathbf{C})_{uu} & \mathbf{C}_{uv} & \mathbf{G}_{up} \\ \mathbf{C}_{vu} & (\mathbf{M} + \mathbf{K} + \mathbf{C})_{vv} & \mathbf{G}_{vp} \\ \mathbf{G}_{pu}^T & \mathbf{G}_{pv}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_u \\ \mathbf{b}_v \\ \mathbf{0} \end{pmatrix}$$
(3.26)

with \mathbf{M} , \mathbf{K} and \mathbf{C} the mass, viscosity and convection matrix, respectively. Matrices \mathbf{G} and \mathbf{G}^T represent the discrete gradient and divergence operator, respectively. Righthand-side vector \mathbf{b} contains the Neumann boundary condition, the external source term and the terms known from the previous time step or Newton iteration. The left-hand-side vector contains the nodal unknowns. Note that each matrix entry has two subscripts. The first subscript denotes whether the term is used for the x- or y-momentum equation or the continuity equation, while the second subscript denotes if it involves a u, v or p term. It can be noted that the mass and viscosity matrix in the momentum equation are zero for the off-diagonal terms, while the convection term does have a term in the uv- and vu-coupling matrix. The matrices have contributions from the following terms:

$$\begin{aligned}
\mathbf{M}_{uu}, \mathbf{M}_{vv} &\to (\mathbf{w}_{h}, \mathbf{u}_{h}^{k+1}) & \mathbf{G}_{up}, \mathbf{G}_{vp} \to -b(\mathbf{w}_{h}, p_{h}^{k+1}) \\
\mathbf{K}_{uu}, \mathbf{K}_{vv} &\to a(\mathbf{w}_{h}, \mathbf{u}_{h}^{k+1}) & \mathbf{G}_{pu}^{T}, \mathbf{G}_{pv}^{T} \to b(\mathbf{u}_{h}^{k+1}, q_{h}) \\
\mathbf{C}_{uv}, \mathbf{C}_{vu} &\to c(\mathbf{u}_{h}^{k+1}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k}) \\
\mathbf{C}_{uu}, \mathbf{C}_{vv} \to c(\mathbf{u}_{h}^{k+1}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k}) + c(\mathbf{u}_{h}^{k}; \mathbf{w}_{h}, \mathbf{u}_{h}^{k+1})
\end{aligned}$$
(3.27)

Because of the locality of the basis functions it is clear that multiplication and integration of two basis functions for different nodes will in general yield a zero result, unless the nodes are in adjacent elements. Hence the global matrix will be very sparse, and it does not make sense to evaluate the value for each position of this global matrix. It is much more sensible to evaluate the contributions for each local element on Ω^e separately. Let us denote by small letters a and b the local element nodes, then as an example we take the mass matrix term for the *uu*-coupling matrix (i.e., taking i = j = 1 in equation (3.24)) as,

$$(M^e_{ab})_{uu} = (\phi_a, \phi_b)_{\Omega^e} \tag{3.28}$$

for specific nodes a and b. As noted before, this element matrix only has entries in the matrices with equal velocity components, i.e., the uu- and vv-coupling matrices.



Figure 3.2: Element transformation from the global domain to the normalized domain for a four-node quadrilateral.

Because every element can have a different shape, the contribution of each element can easily be compared by isoparametric transformation to a master element which has normalized local coordinates. For a quadrilateral the normalized coordinates are $(\xi, \zeta) \in [-1, 1] \times [-1, 1]$. The transformation for a four-node quadrilateral element is shown in figure 3.2. The equations for the local bilinear basis functions for node 1 through 4 are then given as,

$$\phi_1 = \frac{1}{4}(1-\xi)(1-\zeta) \qquad \phi_2 = \frac{1}{4}(1+\xi)(1-\zeta) \phi_3 = \frac{1}{4}(1+\xi)(1+\zeta) \qquad \phi_4 = \frac{1}{4}(1-\xi)(1+\zeta) .$$
(3.29)

Note that such an element is called a Q1Q1 element if both the velocity and pressure are approximated by this bilinear element. The letter refers to the quadrilateral element (in 2D) and the number refers to the degree of the polynomial appearing in the basis functions.

Thus the global coordinates x and y need to be transformed to ξ and ζ , respectively. We note that x and y are dependent on both normalized coordinates and thus it follows that,

$$dx = \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \zeta} d\zeta \qquad \text{or} \qquad \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \zeta} \end{pmatrix} \begin{pmatrix} d\xi \\ d\zeta \end{pmatrix}$$
(3.30)

where the matrix is the element Jacobian matrix (not to be confused with the system Jacobian matrix). We are trying to find a factor the integrals should be multiplied with to account for the change in surface area when going from the global to the master element. The vector element area can be calculated in terms of the normalized coordinates using

the cross product rule for parallelograms as follows,

$$d\mathbf{\Omega}^{e} = d\mathbf{x} \times d\mathbf{y}$$

$$= \left(\frac{\partial x}{\partial \xi} d\xi \mathbf{i} + \frac{\partial x}{\partial \zeta} d\zeta \mathbf{j}\right) \times \left(\frac{\partial y}{\partial \xi} d\xi \mathbf{i} + \frac{\partial y}{\partial \zeta} d\zeta \mathbf{j}\right)$$

$$= \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \zeta}\right) d\xi d\zeta \mathbf{k}$$

$$= J d\xi d\zeta \mathbf{k}$$
(3.31)

where the determinant of the Jacobian matrix in equation 3.30 has been termed J. The element area $d\Omega^e$ in the global domain is thus related to the element area in the isoparametric domain as,

$$d\Omega^e = Jd\xi d\zeta \tag{3.32}$$

Applying equation (3.32) to the mass matrix term (3.28) results in,

$$(M_{ab}^e)_{uu} = \int_{-1}^1 \int_{-1}^1 J\phi_a \phi_b d\xi d\zeta \,. \tag{3.33}$$

The integrals are usually evaluated numerically using Gaussian quadrature. The integral over the domain can be approximated for the mass matrix term using,

$$(M^{e}_{ab})_{uu} = \sum_{g=1}^{n_{qp}} W_{|g} J_{|g} \phi_{a|g} \phi_{b|g}$$
(3.34)

with n_{qp} the number of quadrature points, W the weight for the current quadrature point, and the subscript g indicating the term is evaluated at the coordinates (ξ_g, ζ_g) of the current quadrature point. Finally, instead of using specific element nodes a and b, a loop is performed over all element degrees of freedom to build the mass matrix. Note that the mass matrix $(\mathbf{M}_{ab}^e)_{uu}$ has dimensions in each direction equal to the number of element nodes. The mass matrix for the uu-coupling is then given by,

$$(\mathbf{M}_{ab}^{e})_{uu} = \sum_{g=1}^{n_{qp}} \sum_{a=1}^{n_{en}} \sum_{b=1}^{n_{en}} W_{|g} J_{|g} \phi_{a|g} \phi_{b|g}$$
(3.35)

with n_{en} the number of element nodes. An overview of all terms used for the implementation of the non-stabilized Navier-Stokes equations in libMesh are given in appendix C in its final form such as equation (3.35).

The global matrix and right-hand-side vector then result from the topological assembly of element contributions. In the code libMesh takes care of this as it has a built-in assembly algorithm. In general one can say that assembly takes place such that a term in equation (3.35) will be added to the row corresponding to node a and the column corresponding to node b. More information on global matrix and vector assembly can be found in elementary books on FEM such as [43].

3.4 Stabilization of the Galerkin formulation

The Galerkin formulation derived in the previous section is known to suffer from two main sources of instability. One of these occurs for high Reynolds number flows when convection dominates the problem. An adequate stabilization for this may be achieved by using the streamline upwind/Petrov-Galerkin (SUPG) method, in combination with the the least-squares on the incompressibility constraint (LSIC) method. These techniques will be addressed in section 3.4.1. The other source of instability is due to the mixed finite element formulation that arises from the Navier-Stokes equations, and puts a restriction on the type of interpolation spaces that can be used. This will be explained in section 3.4.2, where the pressure stabilizing/Petrov-Galerkin (PSPG) method is introduced. In the final section 3.4.3 we show the choice of stabilization parameters that are used for the stabilization terms.

The stabilization schemes introduced in this section were derived in the manner shown below. Later in 1995, the introduction of the variational multiscale method by Hughes [47] made it possible to derive all these terms naturally from a firm theoretical foundation. Application of the variational multiscale method to the Navier-Stokes equations gives rise to two additional stabilizing terms, next to the SUPG, PSPG, and LSIC terms. These additional terms stem from the variational multiscale principle and will be neglected for the stabilization of the Navier-Stokes equations here. See, e.g., [8] for more details.

3.4.1 SUPG and LSIC stabilization

From a simple one-dimensional convection-diffusion problem it can be shown analytically that the Galerkin method solves a modified equation for convection dominated flow. This modified equation has a reduced diffusion coefficient, which may become negative, therefore a stable solution cannot be guaranteed. This problem is seen to be similar to finite difference methods, where a convection dominated flow will also cause spurious oscillations in the results. A solution for this method is the use of an upwind approximation of the convective term, which induces numerical diffusion. Hence it is logical to look for a similar solution in the framework of finite elements.

The key idea for such a solution is to add a weighted residual formulation in which the weighting function is selected from a different class of functions than that of the approximate solution. The most popular type of upwind schemes in the FEM framework is called the streamline upwind/Petrov-Galerkin (SUPG) method. Using this method the upwinding effect is, also in multiple dimensions, only felt along streamlines. Earlier types of methods did not account for crosswind diffusion, leading to overly diffusive results. A later model, the streamline upwind (SU) method, was termed inconsistent since the modified weighting term was only multiplied with a part of the equation. A consistent formulation is required in order for the solution of the problem to also be a solution of its weak form. Hence the modified weighting function should be used for the whole equation, such that the general form of a consistent stabilization term can be formulated as,

$$(\mathcal{P}(\mathbf{w}), \boldsymbol{\tau}\mathcal{R}(\mathbf{U}))_{\cup \mathbf{O}^e}$$
 (3.36)

with $\mathcal{P}(\mathbf{w})$ an operator applied to the weighting function, $\boldsymbol{\tau}$ is the stabilization parameter and $\mathcal{R}(\mathbf{U})$ the residual of the Navier-Stokes equations in its form in equations (2.7) and (2.1). Note that this term is already put in its weak form and that the integration is performed over the union of the element interiors, i.e., $\cup \Omega^e = \sum_{e=1}^{n_{el}} \Omega^e$. The latter is done because the modified weighting function causes the test functions to be discontinuous across element edges.

The SUPG operator is defined by taking,

$$\mathcal{P}_{SUPG}(\mathbf{w}) = (\mathbf{u} \cdot \nabla) \, \mathbf{w} \tag{3.37}$$

and $\mathcal{R}_m(\mathbf{u}, p)$ to indicate we take the residual of the momentum equation. Least-squares methods instead are defined by requiring that the stabilization term is a weighted leastsquares formulation of the original problem. The least-squares stabilization on the incompressibility constraint (LSIC) thus imposes a modified weighting function to the continuity equation with,

$$\mathcal{P}_{LSIC}(\mathbf{w}) = \nabla \cdot \mathbf{w} \tag{3.38}$$

and $\mathcal{R}_c(\mathbf{u})$ to indicate we take the residual of the continuity equation. As can be seen, it uses the same operator on the weighting function as it does on the velocity.

Adding both the SUPG and LSIC stabilization terms to the momentum equation yields additional stability for highly convective flows, i.e., flow at large Reynolds numbers. Adding them to the momentum equation in (3.9) yields,

$$\begin{pmatrix} \mathbf{w}, \frac{\partial \mathbf{u}}{\partial t} \end{pmatrix} + c (\mathbf{u}; \mathbf{w}, \mathbf{u}) + a (\mathbf{w}, \mathbf{u}) - b (\mathbf{w}, p) \\
+ ((\mathbf{u} \cdot \nabla) \mathbf{w}, \tau_m \mathcal{R}_m(\mathbf{u}, p))_{\cup \Omega^e} + (\nabla \cdot \mathbf{w}, \tau_c \mathcal{R}_c(\mathbf{u}))_{\cup \Omega^e} \\
= (\mathbf{w}, \mathbf{f}) + (\mathbf{w}, \mathbf{t}')_{\Gamma_N}.$$
(3.39)

where τ_m and τ_c are the stabilization parameters corresponding to the momentum and continuity equation, respectively. Their definition is elaborated on in section 3.4.3.

The individual stabilization terms are now fully discretized, as was done in the previous sections for the Galerkin terms of the Navier-Stokes equations. The SUPG stabilization terms can be written and temporally discretized as,

$$\begin{pmatrix} (\mathbf{u} \cdot \nabla) \mathbf{w}, \tau_m \mathcal{R}_m(\mathbf{u}, p) \end{pmatrix}_{\cup \Omega^e} \\ = \left((\mathbf{u} \cdot \nabla) \mathbf{w}, \tau_m \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p - \mathbf{f} \right] \right)_{\cup \Omega^e} \\ \xrightarrow{\theta \text{-method}} \left((\mathbf{u} \cdot \nabla) \mathbf{w}, \tau_m \left[\mathbf{u} - \hat{\mathbf{u}} + \Delta t \Big\{ \theta \big((\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p \big) + (1 - \theta) \big((\hat{\mathbf{u}} \cdot \nabla) \hat{\mathbf{u}} - \nu \nabla^2 \hat{\mathbf{u}} + \nabla \hat{p} \big) - \mathbf{f} \Big\} \right] \right)_{\cup \Omega^e}$$
(3.40)

where the multiplication by Δt is consistent with the Galerkin discretization. Note that almost every term is now nonlinear due to the the modified weighting function containing the unknown velocity.

Now Newton's method can be applied for equation (3.40), as done in section 3.2.2, by using,

$$\mathbf{J}(\mathbf{U}^k).\delta\mathbf{U} = -\boldsymbol{\mathcal{R}}_{\theta,m}(\mathbf{u}^k, p^k)$$
(3.41)

with $\mathbf{J}(\mathbf{U}^k)$ the Jacobian of equation (3.40), and $\mathcal{R}_{\theta,m}(\mathbf{u}^k, p^k)$ its residual (i.e., the equation itself). After canceling out terms we get,

$$\tau_{m} \left[\left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, \mathbf{u}^{k} \right)_{\cup \Omega^{e}} + \left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, \mathbf{u}^{k+1} \right)_{\cup \Omega^{e}} - \left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, \mathbf{\hat{u}}^{k} \right)_{\cup \Omega^{e}} \right. \\ \left. + \Delta t \left\{ \theta \left[\left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, (\mathbf{u}^{k} \cdot \nabla) \mathbf{u}^{k} \right)_{\cup \Omega^{e}} + \left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, (\mathbf{u}^{k+1} \cdot \nabla) \mathbf{u}^{k} \right)_{\cup \Omega^{e}} \right. \\ \left. + \left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, (\mathbf{u}^{k} \cdot \nabla) \mathbf{u}^{k+1} \right)_{\cup \Omega^{e}} - \left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, \nu \nabla^{2} \mathbf{u}^{k} \right)_{\cup \Omega^{e}} \right. \\ \left. - \left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, \nu \nabla^{2} \mathbf{u}^{k+1} \right)_{\cup \Omega^{e}} + \left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, \nabla p^{k} \right)_{\cup \Omega^{e}} + \left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, \nabla p^{k+1} \right)_{\cup \Omega^{e}} \right] \right. \\ \left. + \left(1 - \theta \right) \left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, (\mathbf{\hat{u}} \cdot \nabla) \mathbf{\hat{u}} - \nu \nabla^{2} \mathbf{\hat{u}} + \nabla \hat{p} \right)_{\cup \Omega^{e}} - \left((\mathbf{u}^{k+1} \cdot \nabla) \mathbf{w}, \mathbf{f} \right)_{\cup \Omega^{e}} \right\} \right] \\ \left. < = \left. \tau_{m} \left[\left((\mathbf{u}^{k} \cdot \nabla) \mathbf{w}, \mathbf{u}^{k} + \Delta t \left\{ \theta \left[2(\mathbf{u}^{k} \cdot \nabla) \mathbf{u}^{k} - \nu \nabla^{2} \mathbf{u}^{k} + \nabla p^{k} \right] \right\} \right]_{\cup \Omega^{e}} \right] \right]$$

$$(3.42)$$

where the $\langle = \rangle$ -sign indicates the left- and right-hand-side are additions to the Jacobian matrix and right-hand-side vector, respectively. The application of Newton's method to the nonlinear terms can be seen to result in many additional terms, where again the known terms have been moved to the right-hand-side. Note that second order derivatives are present for the diffusion terms, these terms will vanish when using linear interpolation spaces.

We then apply the Galerkin method through a spatial discretization of the domain. By restricting the SUPG terms to the finite dimensional subspaces S_h , \mathcal{V}_h and \mathcal{Q}_h such that $\mathbf{u}_h \in S_h \subset S$, $\mathbf{w}_h \in \mathcal{V}_h \subset \mathcal{V}$ and $p_h \in \mathcal{Q}_h \subset \mathcal{Q}$, we automatically compute each term for the element interiors. Then replacing the unknowns by their basis functions (equations (3.21) and (3.23)) for the domain-discretized SUPG stabilization terms, we can find the expressions to be added to the element matrices. This process is equal to that for the Galerkin terms, shown in section 3.3. Appendix D contains the additional element-level equations that are added to the Galerkin terms for the stabilization of the Navier-Stokes equations.

Similarly for the LSIC stabilization we perform the temporal discretization as follows.

(

$$\nabla \cdot \mathbf{w}, \tau_c \mathcal{R}_c(\mathbf{u}) \Big|_{\cup \Omega^e} = \left(\nabla \cdot \mathbf{w}, \tau_c \nabla \cdot \mathbf{u} \right)_{\cup \Omega^e}$$
$$\xrightarrow{\theta \text{-method}} \left(\nabla \cdot \mathbf{w}, \tau_c \Delta t \left[\theta \nabla \cdot \mathbf{u} + (1 - \theta) \nabla \cdot \hat{\mathbf{u}} \right] \right)_{\cup \Omega^e}$$
(3.43)

The LSIC stabilization term does not contain a temporal term, but has to be discretized in time since it is part of the time-dependent equation (3.39). Note again the consistent multiplication by Δt . For Newton's method the following equation can be used,

$$\mathbf{J}(\mathbf{u}^k).\delta\mathbf{u} = -\mathcal{R}_{\theta,c}(\mathbf{u}^k) \tag{3.44}$$

with $\mathbf{J}(\mathbf{u}^k)$ the Jacobian of equation (3.43), and $\mathcal{R}_{\theta,c}(\mathbf{u}^k)$ equal to equation (3.43). The LSIC stabilization terms can then be found to be,

$$\tau_c \Delta t \theta \left(\nabla \cdot \mathbf{w}, \nabla \cdot \mathbf{u}^{k+1} \right)_{\cup \Omega^e} <=> -\tau_c \Delta t (1-\theta) \left(\nabla \cdot \mathbf{w}, \nabla \cdot \hat{\mathbf{u}} \right)_{\cup \Omega^e}$$
(3.45)

Similarly as for the SUPG terms, the discretization in space can be performed such that $\mathbf{u}_h \in \boldsymbol{S}_h \subset \boldsymbol{S}$ and $\mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$. The LSIC stabilization terms to be added to the element matrices can be derived from this, these are presented in appendix D.

3.4.2 PSPG stabilization

Let us go back and take a look at the matrix given by equation (3.26). Note the zero matrix on the bottom right, giving rise to a zero diagonal due to a pressure term missing in the continuity equation. The constraint imposed by the continuity equation is that the velocity field should be divergence free. The zero diagonal in the matrix can give problems in terms of the solvability of the matrix, and in turn can be shown to put restrictions on the type of interpolation spaces to be used in order for the matrix to be non-singular. Noting that the matrix should have a full rank to ensure that the system has a unique solution, a condition that can be derived is,

$$\lim \mathcal{Q}_h \le \dim \mathcal{V}_h \tag{3.46}$$

which shows that there is a necessary link between spaces, and they cannot be chosen arbitrarily. The Ladyzhenskaya-Babuška-Brezzi (LBB) condition was derived for the steady Stokes problem which has a similar matrix structure, and provides a sufficient condition linking the pressure and velocity spaces. It states that: The existence of a stable finite element approximate solution $(\mathbf{u}_h, \mathbf{p}_h)$ to the steady Stokes problem depends on choosing a pair of spaces \mathcal{V}_h and \mathcal{Q}_h , such that the following inf-sup condition holds:

$$\inf_{q_h \in \mathcal{Q}_h \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h} \sup_{\||q_h||_0 \||\mathbf{w}_h\|_1} \ge \alpha > 0$$
(3.47)

with α independent of mesh size h, and the norms defined as,

$$\begin{aligned} ||q||_0 &= \sqrt{(q,q)} \\ ||\mathbf{w}||_1 &= \sqrt{(\mathbf{w},\mathbf{w}) + (\nabla \mathbf{w},\nabla \mathbf{w})} \end{aligned}$$

If this condition is not satisfied then there exist $q_h^* \in \mathcal{Q}_h$ such that,

$$b(\mathbf{w}_h, q_h^*) = 0 \quad \forall \mathbf{w}_h \in \mathcal{V}_h \tag{3.48}$$

meaning the matrix is not full rank, and if (\mathbf{u}_h, p_h) is a solution of the problem, then so will be $(\mathbf{u}_h, p_h + q_h^*)$. A unique solution is thus prevented by the functions q_h^* , they are referred to as spurious pressure modes.

The LBB-condition prevents the use of the popular equal order Q1Q1 element, and renders one to use, e.g., Q2Q1 elements, which do satisfy the LBB-condition. The use of biquadratic elements for the velocity however results in many additional degrees of freedom. In several cases a biquadratic interpolation space is not necessary, and the use of Q2Q1 elements thus results in unwanted additional computation time. An often used option is to circumvent the LBB-condition by using the pressure stabilizing/Petrov-Galerkin (PSPG) method. The stabilization term added using the PSPG method results in a nonzero diagonal in the *pp*-coupling matrix. The idea of this method is thus to add a stabilization term which enforces positive definiteness of the global matrix.

The PSPG operator is defined by taking,

$$\mathcal{P}(q) = \nabla q \tag{3.49}$$

and $\mathcal{R}_m(\mathbf{u}, p)$ the residual of the momentum equation. Adding the PSPG stabilization term to the continuity equation in (3.9) yields,

$$b(\mathbf{u},q) + \left(\nabla q, \tau_m \mathcal{R}_m(\mathbf{u},p)\right)_{\cup \Omega^e} = 0 \tag{3.50}$$

The residual of the momentum equation contains the velocity time derivative and the theta method is now thus also used for the continuity equation. Hence the Galerkin term is rederived for the stabilized case. First using the theta method we obtain,

$$\begin{split} \Delta t \Big(\theta b(\mathbf{u}, q) + (1 - \theta) b(\hat{\mathbf{u}}, q) \Big) + \Big(\nabla q, \tau_m \Big[\mathbf{u} - \hat{\mathbf{u}} + \Delta t \Big\{ \theta \big((\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p \big) \\ &+ (1 - \theta) \big((\hat{\mathbf{u}} \cdot \nabla) \hat{\mathbf{u}} - \nu \nabla^2 \hat{\mathbf{u}} + \nabla \hat{p} \big) - \mathbf{f} \Big\} \Big] \Big)_{\cup \Omega^e} = 0 \end{split}$$

where the equation has been multiplied by the time step. Then Newton's method can be applied as before,

Similarly as for the SUPG and LSIC terms, the discretization in space can be performed such that $\mathbf{u}_h \in \mathcal{S}_h \subset \mathcal{S}$, $\mathbf{w}_h \in \mathcal{V}_h \subset \mathcal{V}$ and $p_h \in \mathcal{Q}_h \subset \mathcal{Q}$. The PSPG stabilization terms to be added to the element matrices can be derived from this, these are presented in appendix D.

3.4.3 Choice of stabilization parameters

The definition of the stabilization parameters τ_m and τ_c determines how well the system is stabilized. The design of a stability parameter is a whole study by itself, and can for example be derived through linear error estimates, dimensional design, and convergence and stability theory. Only for highly restricted problems, such as a one-dimensional case, one can find an optimal definition of τ . Note that τ is in fact a stabilization coefficient matrix for a system of equations. This can be seen as follows. We would like to have optimal behavior with respect to each of the Navier-Stokes equation components. Therefore if τ is the same for each of the components we might have τ too big for one component, leading to overly diffusive results for this component. A particular component having τ too small will lead to spurious oscillations for this component. We thus require a distinct τ for each component, as mentioned by Hughes and Mallet [45].

Here we will follow the work of Bazilevs et al. [8] and use a separate τ for the momentum and continuity equation, which have been shown to function well. These are defined as,

$$\tau_m = \left(\frac{4}{(\Delta t)^2} + \mathbf{u}_h \cdot (\boldsymbol{\mathcal{G}}.\mathbf{v}_h) + c_I \nu^2 \boldsymbol{\mathcal{G}} : \boldsymbol{\mathcal{G}}\right)^{-\frac{1}{2}}$$
(3.52)

$$\tau_c = \left(\tau_m \sqrt{\mathcal{G} : \mathcal{G}}\right)^{-1} \tag{3.53}$$

with c_I a positive constant, independent of mesh size h, which can be derived from a so-called element-wise inverse estimate. Observe that τ_m is left a scalar since it is equal

for both components of the momentum equation. ${\cal G}$ is the covariant metric tensor and is defined as,

$$\mathcal{G}_{ij} = \frac{\partial \xi}{\partial x_i} \frac{\partial \xi}{\partial x_j} + \frac{\partial \zeta}{\partial x_i} \frac{\partial \zeta}{\partial x_j}$$
(3.54)

and

$$\mathcal{G}: \mathcal{G} = \sum_{i,j=1}^{2} \mathcal{G}_{ij} \mathcal{G}_{ij} .$$
(3.55)

It can be seen that the stability parameters are dependent on the time step. This has the drawback that as $\Delta t \to 0$, then $\tau_m \to 0$ and $\tau_c \to \infty$, i.e., they become degenerate. As long as the time step is of order $\Delta t = \mathcal{O}(h/|\mathbf{u}_h|)$ though, it behaves properly. Hsu et al. [42] mention that the above definition of τ_m may not be suitable for time-dependent problems which have a steady solution. An option is to omit the dependence on Δt by dropping this term out of equation (3.52), thus,

$$\tau_m = \left(\mathbf{u}_h \cdot (\boldsymbol{\mathcal{G}}.\mathbf{u}_h) + c_I \nu^2 \boldsymbol{\mathcal{G}} : \boldsymbol{\mathcal{G}}\right)^{-\frac{1}{2}}$$
(3.56)

where it is noted that mainly in problems which have a steady solution this could be favorable.

Chapter 4

Finite Element Method for RANS and the S-A turbulence model

By decomposing the flow field into a mean and fluctuating part we can obtain the Reynolds-averaged Navier-Stokes (RANS) equations from the Navier-Stokes (N-S) equations, as introduced in chapter 2. These allow us to fully model turbulence by estimating the Reynolds stress tensor using a turbulence model such as the Spalart-Allmaras (S-A) model. This chapter will deal with deriving the terms required for the implementation of the RANS equations and the S-A turbulence model into libMesh using the finite element method.

For the RANS equations the same stabilization techniques are used as for the N-S equations, section 4.1 will deal with the derivation of the Galerkin and stabilization terms. In section 4.2 the theory behind the variational multiscale (VMS) method is more thoroughly explained. The analogy between the S-A model and the advection-diffusion-reaction (A-D-R) equation is shown in section 4.3. Here the S-A model is substituted into the A-D-R equation, after which VMS is employed on the A-D-R equation. In section 4.4 a different approach is taken to derive the multiscale stabilized S-A equation, as VMS is directly applied to it. This results in a different stabilized formulation, as is shown in the final section of this chapter in which a comparison of the two methods is performed.

4.1 Finite element method for the RANS equations

When comparing the incompressible N-S and RANS equations, (2.7) and (2.17), respectively, two important differences can be noted. The first is that the variables for the RANS equations have to be interpreted as ensemble average. The second is the additional eddy viscosity in the diffusion term. Though the interpretation of the variables is different to those of the N-S equations, all terms in the equation besides the diffusion term are still equal. Hence the only term that needs to be reevaluated is the diffusion term, since the other terms have been discretized for the N-S equations in chapter 3.

In this section the stabilized weak form of the RANS equations will first be presented, accompanied by the corresponding spaces of functions. Then we will focus on the discretization of diffusion term, which leads to modified element-level equations compared to the N-S equations.

4.1.1 Weak form

Let us define the modified shear stress tensor τ_t for the RANS equations as,

$$\boldsymbol{\tau}_t = (\nu + \nu_t) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \,. \tag{4.1}$$

Note that the overlines indicating the ensemble average of the variables have been dropped for convenience. We then multiply the diffusion term in equation (2.17) by the weighting function \mathbf{w} , and apply integration by parts as follows.

$$(\mathbf{w}, \nabla \cdot \boldsymbol{\tau}_t) = (\mathbf{w}, \boldsymbol{\tau}_t \cdot \mathbf{n})_{\Gamma} - (\nabla \mathbf{w}, \boldsymbol{\tau}_t)$$
(4.2)

Hence the weak form problem statement for the stabilized RANS equations can be formulated as follows. Given \mathbf{u}_D , \mathbf{t}_t and \mathbf{f} , find $\mathbf{u}(\mathbf{x},t) \in \mathcal{S} \times]0, T[$ and $p(\mathbf{x},t) \in \mathcal{Q} \times]0, T[$ such that,

$$\begin{cases} \left(\mathbf{w}, \frac{\partial \mathbf{u}}{\partial t}\right) + c\left(\mathbf{u}; \mathbf{w}, \mathbf{u}\right) + a_t\left(\mathbf{w}, \mathbf{u}\right) - b\left(\mathbf{w}, p\right) - \left(\mathbf{w}, \mathbf{f}\right) - \left(\mathbf{w}, \mathbf{t}_t\right)_{\Gamma_N} \\ + \left(\left(\mathbf{u} \cdot \nabla\right) \mathbf{w}, \tau_m \mathcal{R}_m(\mathbf{u}, p)\right)_{\cup \Omega^e} + \left(\nabla \cdot \mathbf{w}, \tau_c \mathcal{R}_c(\mathbf{u})\right)_{\cup \Omega^e} = 0 \qquad (4.3)\\ b\left(\mathbf{u}, q\right) + \left(\nabla q, \tau_m \mathcal{R}_m(\mathbf{u}, p)\right)_{\cup \Omega^e} = 0\end{cases}$$

for all $(\mathbf{w}, q) \in \mathcal{V} \times \mathcal{Q}$. Here the bi- and trilinear forms introduced in equations (3.7) and (3.8) have been used. The bilinear form $a_t(\mathbf{w}, \mathbf{u})$ satisfies,

$$a_t \left(\mathbf{w}, \mathbf{u} \right) = \int_{\Omega} \nabla \mathbf{w} : \left(\nu + \nu_t \right) \left(\nabla \mathbf{u} + \left(\nabla \mathbf{u} \right)^T \right) d\Omega \qquad \forall \mathbf{u}, \mathbf{w} \in \mathcal{H}^1(\Omega)$$
(4.4)

and the modified traction boundary condition \mathbf{t}_t is defined as,

$$\mathbf{t}_t = -p\mathbf{n} + \boldsymbol{\tau}_t \cdot \mathbf{n} \,. \tag{4.5}$$

The definition of the trial and weighting solution spaces \mathcal{S} , \mathcal{V} and \mathcal{Q} are equal to those for the N-S equations, i.e.,

$$S := \left\{ \mathbf{u} \in \mathcal{H}^1 \, | \, \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D \right\}$$
$$\mathcal{V} := \left\{ \mathbf{w} \in \mathcal{H}^1 \, | \, \mathbf{w} = 0 \text{ on } \Gamma_D \right\}$$
$$\mathcal{Q} := L_2(\Omega) \, .$$

Finally, for the residual of the momentum equation we write for the diffusion term,

$$\nabla \cdot \left[(\nu + \nu_t) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] = \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right)$$
$$= \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla^2 \mathbf{u}$$
(4.6)

where the vector identity $\nabla \cdot (\nabla \mathbf{u})^T = \nabla (\nabla \cdot \mathbf{u})$ was used for the second term. Note that due to the incompressibility constraint this terms equals zero.

4.1.2 Discretization of the turbulent diffusion terms

Let us now focus on the diffusion terms in the stabilized RANS equations. Below we first apply the theta method to the Galerkin, SUPG stabilization and PSPG stabilization terms. Note that the terms are consistently multiplied by the time step, and equation (4.6) is used for the residual of the diffusion term.

Galerkin diffusion term

$$a_t(\mathbf{w}, \mathbf{u}) \xrightarrow{\theta \text{-method}} \Delta t \theta \, a_t(\mathbf{w}, \mathbf{u}) + \Delta t (1 - \theta) a_t(\mathbf{w}, \hat{\mathbf{u}})$$
(4.7)

SUPG stabilization diffusion term

$$\begin{aligned} &-\tau_m \Big((\mathbf{u} \cdot \nabla) \mathbf{w}, \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla^2 \mathbf{u} \Big)_{\cup \Omega^e} \\ & \xrightarrow{\theta \text{-method}} \quad -\tau_m \Delta t \theta \left((\mathbf{u} \cdot \nabla) \mathbf{w}, \nabla \nu_t \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + (\nu + \nu_t) \nabla^2 \mathbf{u} \right)_{\cup \Omega^e} \\ & -\tau_m \Delta t (1 - \theta) \left((\mathbf{u} \cdot \nabla) \mathbf{w}, \left(\nabla \hat{\mathbf{u}} + (\nabla \hat{\mathbf{u}})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla^2 \hat{\mathbf{u}} \right)_{\cup \Omega^e} \end{aligned}$$
(4.8)

PSPG stabilization diffusion term

$$\begin{aligned} -\tau_m \Big(\nabla q, \nabla \nu_t \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + (\nu + \nu_t) \nabla^2 \mathbf{u} \Big)_{\cup \Omega^e} \\ \xrightarrow{\theta \text{-method}} & -\tau_m \Delta t \theta \left(\nabla q, \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla^2 \mathbf{u} \right)_{\cup \Omega^e} \\ & -\tau_m \Delta t (1 - \theta) \left(\nabla q, \left(\nabla \mathbf{\hat{u}} + (\nabla \mathbf{\hat{u}})^T \right) \cdot \nabla \nu_t + (\nu + \nu_t) \nabla^2 \mathbf{\hat{u}} \right)_{\cup \Omega^e} . \end{aligned}$$

$$(4.9)$$

As for the N-S equations we now apply Newton's method, equation (3.11), for the RANS diffusion and stabilization diffusion terms. This results in the following contributions to the Jacobian matrix and right-hand-side vector.

Galerkin diffusion term

$$\Delta t\theta \, a_t \left(\mathbf{w}, \mathbf{u}^{k+1} \right) <=> -\Delta t (1-\theta) a_t \left(\mathbf{w}, \hat{\mathbf{u}} \right) \tag{4.10}$$

SUPG stabilization diffusion term

$$-\tau_{m}\Delta t \left[\theta \left(\left(\mathbf{u}^{k+1} \cdot \nabla \right) \mathbf{w}, \left(\nabla \mathbf{u}^{k} + \left(\nabla \mathbf{u}^{k} \right)^{T} \right) \cdot \nabla \nu_{t} + \left(\nu + \nu_{t} \right) \nabla^{2} \mathbf{u}^{k} \right)_{\cup \Omega^{e}} + \theta \left(\left(\mathbf{u}^{k} \cdot \nabla \right) \mathbf{w}, \left(\nabla \mathbf{u}^{k+1} + \left(\nabla \mathbf{u}^{k+1} \right)^{T} \right) \cdot \nabla \nu_{t} + \left(\nu + \nu_{t} \right) \nabla^{2} \mathbf{u}^{k+1} \right)_{\cup \Omega^{e}} + \left(1 - \theta \right) \left(\left(\mathbf{u}^{k+1} \cdot \nabla \right) \mathbf{w}, \left(\nabla \hat{\mathbf{u}} + \left(\nabla \hat{\mathbf{u}} \right)^{T} \right) \cdot \nabla \nu_{t} + \left(\nu + \nu_{t} \right) \nabla^{2} \hat{\mathbf{u}} \right)_{\cup \Omega^{e}} \right]$$

$$<=> -\tau_{m}\Delta t \theta \left(\left(\mathbf{u}^{k} \cdot \nabla \right) \mathbf{w}, \left(\nabla \mathbf{u}^{k} + \left(\nabla \mathbf{u}^{k} \right)^{T} \right) \cdot \nabla \nu_{t} + \left(\nu + \nu_{t} \right) \nabla^{2} \mathbf{u}^{k} \right)_{\cup \Omega^{e}}$$
EDC stabilization diffusion terms

PSPG stabilization diffusion term

$$-\tau_{m}\Delta t\theta \left(\nabla q, \left(\nabla \mathbf{u}^{k+1} + \left(\nabla \mathbf{u}^{k+1}\right)^{T}\right) \cdot \nabla \nu_{t} + (\nu + \nu_{t}) \nabla^{2} \mathbf{u}^{k+1}\right)_{\cup\Omega^{e}}$$

$$<=>\tau_{m}\Delta t(1-\theta) \left(\nabla q, \left(\nabla \mathbf{\hat{u}} + \left(\nabla \mathbf{\hat{u}}\right)^{T}\right) \cdot \nabla \nu_{t} + (\nu + \nu_{t}) \nabla^{2} \mathbf{\hat{u}}\right)_{\cup\Omega^{e}}$$

$$(4.12)$$

The $\langle = \rangle$ -sign again indicates the left- and right-hand-side are additions to the Jacobian matrix and right-hand-side vector, respectively. We then apply the Galerkin method through a spatial discretization of the domain. It is noted again that \mathbf{u}_h contains both the auxiliary velocity field as well as the Dirichlet boundary nodes, and thus belongs to S_h . In libMesh Dirichlet boundary conditions are enforced through use of the penalty method, explained in appendix B. We thus restrict the diffusion terms to the finite dimensional subspaces S_h , \mathcal{V}_h and \mathcal{Q}_h such that $\mathbf{u}_h \in S_h \subset S$, $\mathbf{w}_h \in \mathcal{V}_h \subset \mathcal{V}$ and $p_h \in \mathcal{Q}_h \subset \mathcal{Q}$. By replacing the unknown terms (appended with 'k+1') with their basis functions (equations (3.21) and (3.23)) for the domain-discretized terms, we can find the expressions to be added to the element matrices. This process was shown in section 3.3. Appendix E contains the element-level equations that replace the Navier-Stokes diffusion terms for employing the RANS equations.

Finally, it is noted that the stabilization parameter used for the momentum equation τ_m now has a changed definition. This can be observed through the change of the diffusion term, which is now multiplied by $(\nu + \nu_t)$. Hence when using the RANS equations the stabilization parameter τ_m , given by equation (3.52) for the N-S equations, is calculated using $(\nu + \nu_t)$ for the diffusion term in the stabilization parameter instead of ν .

4.2 The variational multiscale method

The theoretical basis of the S-A model stabilization relies on the variational multiscale (VMS) method framework using the finite element method. As mentioned in the literature review part in the introduction, the VMS method was derived by Hughes [47] and relies on a splitting of the unknowns into large, resolvable scales and small, unresolvable scales. The small scale behavior exists within each element of the domain, but is neglected on element boundaries. As presented in his paper the small scales can be solved for using a corresponding Green's function problem, where it is shown that the sub-grid scales are driven by the residual of the resolved scales.

Mathematically this can be written down as follows for a 2D problem. Consider a bounded domain $\Omega \subset \mathbb{R}^2$ with boundary Γ . The domain has n_{el} non-overlapping subdomains Ω^e $(e = 1, 2, ..., n_{el})$ with boundary Γ^e , as shown in figure 4.1.



Figure 4.1: Bounded domain discretized into element subdomains.

Let

$$\Omega = \bigcup_{e=1}^{n_{el}} \overline{\Omega^e} \tag{4.13}$$

where the overline indicates the union of the element interior and its boundary. Let Ω' and Γ' denote the union of element interiors and element boundaries, respectively,

$$\Omega' = \bigcup_{e=1}^{n_{el}} \Omega^e \qquad \text{(element interiors)} \tag{4.14}$$

$$\Gamma' = \bigcup_{e=1}^{n_{el}} \Gamma^e \qquad \text{(element boundaries)}. \tag{4.15}$$

Now considering a homogeneous Dirichlet problem,

$$\mathcal{L}u = f \quad \text{in } \Omega \tag{4.16}$$

$$u = 0 \quad \text{on} \quad \Gamma \tag{4.17}$$

where \mathcal{L} can be, for example, the advection-diffusion-reaction operator, u is an unknown variable (not necessarily the velocity) and f is a smooth valued function.

As in section 3.1, the space of functions is now introduced. Let

$$\mathcal{V} = \left\{ (u, w) \in \mathcal{H}^1 \,|\, (u, w) = 0 \text{ on } \Gamma \right\}$$

with \mathcal{V} the weighting function and trial solution space, and w the weighting function. We also define the bilinear form,

$$a(w,u) = (w,\mathcal{L}u) = (\mathcal{L}^*w,u) \tag{4.18}$$

where \mathcal{L}^* is the adjoint operator of \mathcal{L} . Also, recall that $(\cdot, \cdot) = (\cdot, \cdot)_{\Omega}$, the L^2 -inner product over the domain Ω .

Now u and w are split into large, resolvable scales, and small, unresolvable scales, as overlapping sum decomposition,

$$u = \overline{u} + u'$$

$$w = \overline{w} + w'$$
(4.19)

with the overlines indicating the large scales and the apostrophe indicating the small scales. The notation of the large scales \bar{u} should not be confused with the notation of the ensemble average \bar{u} used for the RANS equations. A visualization of the splitting is shown in figure 4.2. This is accompanied by a splitting of the functional space by direct



Figure 4.2: Resolvable scales (left) and unresolvable, sub-grid scales (right).

sum decomposition,

$$\mathcal{V} = \overline{\mathcal{V}} \oplus \mathcal{V}' \tag{4.20}$$

with

$$u' = 0 \text{ on } \Gamma' \quad \forall u' \in \mathcal{V}' w' = 0 \text{ on } \Gamma' \quad \forall w' \in \mathcal{V}'.$$

$$(4.21)$$

Here $\overline{\mathcal{V}}$ represents the coarse scale finite dimensional subspace, while \mathcal{V}' represents the fine scale subspace and is necessarily infinite dimensional. As mentioned before, it is assumed sub-grid scales vanish on element boundaries. Let us return to this assumption later on.

The variational form of the problem is then formulated as follows. Given f, find $u \in \mathcal{V}$ such that $\forall w \in \mathcal{V}$,

$$a(w,u) = (w,f)$$
 (4.22)

Substituting the splitting of u and w as given by equation (4.19), equation (4.22) becomes,

$$a(\overline{w} + w', \overline{u} + u') = (\overline{w} + w', f).$$

$$(4.23)$$

The objective is now to find an equation governed only by \overline{u} , such that it can be solved for numerically. To do this we need to find an analytic expression for u' in terms of \overline{u} . Equation (4.23) can be divided into a large and small scale problem by virtue of linear independence of \overline{w} and w' as follows,

$$\begin{cases} a(\overline{w},\overline{u}) + a(\overline{w},u') = (\overline{w},f) \\ a(\overline{w},\overline{u}) + (\mathcal{L}^*\overline{w},u') = (\overline{w},f) \end{cases} \quad \forall \overline{w} \in \overline{\mathcal{V}}$$

$$(4.24)$$

$$\begin{cases} a(w',\overline{u}) + a(w',u') = (w',f)\\ (w',\mathcal{L}\overline{u}) + (w',\mathcal{L}u') = (w',f) \end{cases} \quad \forall w' \in \mathcal{V}'$$

$$(4.25)$$

Note that neglecting u' in equation (4.24) leads to finding the Galerkin equation again.

The small scales u' can be written in terms of the coarse scale residual by introducing the relevant Green's function problem corresponding to the Euler-Lagrange equations of equation (4.25). The Green's function problem is not global but an element Green's function. From this it can be found that

$$u'(y) = -\int_{\Omega'} g(x, y) (\mathcal{L}\overline{u} - f)(x) \mathrm{d}\Omega_x, \qquad (4.26)$$

$$u' = M(\mathcal{L}\overline{u} - f), \qquad (4.27)$$

where $\mathcal{L}\overline{u} - f$ is the residual of the resolved scales, g(x, y) is the Green's function, M is an integral operator, and

$$\int_{\Omega'} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \,. \tag{4.28}$$

It can be observed that the small scales are driven by the residual of the coarse scales. The dependence on the small scales can now be eliminated by substituting (4.27) into (4.24):

$$a(\overline{w},\overline{u}) + (\mathcal{L}^*\overline{w}, M(\mathcal{L}\overline{u} - f)) = (\overline{w}, f)$$
(4.29)

where

$$(\mathcal{L}^*\overline{w}, M(\mathcal{L}\overline{u} - f)) = -\int_{\Omega'} \int_{\Omega'} (\mathcal{L}^*\overline{w})(y)g(x, y)(\mathcal{L}\overline{u} - f)(x)\mathrm{d}\Omega_x\mathrm{d}\Omega_y.$$
(4.30)

Applying the Galerkin method yields

$$a(\overline{w}_h, \overline{u}_h) + (\mathcal{L}^* \overline{w}_h, \tilde{M}(\mathcal{L}\overline{u}_h - f)) = (\overline{w}_h, f)$$
(4.31)

with \overline{w}_h and \overline{u}_h the finite dimensional approximations of \overline{w} and \overline{u} , respectively, and \tilde{M} the approximate integral operator M. By comparison with the standard form of stabilized methods, equation (3.36), we can see that the stabilization parameter τ approximates the exact integral operator M. Note that τ is an algebraic operator and we thus have that,

$$\tau = -\tilde{M} \approx -M \tag{4.32}$$

and

$$u' = -\tau(\mathcal{L}\overline{u}_h - f)$$

= $-\tau\overline{\mathcal{R}}$ (4.33)

with $\overline{\mathcal{R}} = \mathcal{L}\overline{u}_h - f$ the residual of the equation for the coarse scales. We can rewrite equation (4.31) as,

$$a(\overline{w}_h, \overline{u}_h) - (\mathcal{L}^* \overline{w}_h, \tau(\mathcal{L}\overline{u}_h - f)) = (\overline{w}_h, f)$$
(4.34)

with

$$-(\mathcal{L}^*\overline{w}_h, \tau(\mathcal{L}\overline{u}_h - f)) = -\int_{\Omega'} (\mathcal{L}^*\overline{w})(x)\tau(\mathcal{L}\overline{u} - f)(x)\mathrm{d}\Omega_x.$$
(4.35)

Note also that the modified weighting function for the variational multiscale stabilization can be observed to be

$$\mathcal{P}_{VMS}(w) = -\mathcal{L}^* \overline{w} \,. \tag{4.36}$$

From the definition of τ it can then be derived that it can be calculated using the element Green's function as

$$\tau = \frac{1}{\text{meas}(\Omega^e)} \int_{\Omega^e} \int_{\Omega^e} g(x, y) \mathrm{d}\Omega_x \mathrm{d}\Omega_y$$
(4.37)

which shows that in this way τ is defined as the element mean value of the Green's function.

The above discussion on VMS refers to Hughes' initial paper on this topic [47], a more thorough description of the method is however given in [48, 46]. Here the assumption of the sub-grid scales vanishing on element boundaries, equation (4.21), is not made at first. It is shown that for the discretized domain smoothness of the solution is only present on element interiors, hence the integration-by-parts equations (4.24) and (4.25) give rise to nonvanishing element boundary terms. Notably, the second derivative of a function, using linear elements, gives rise to a Dirac delta function on the element boundary while having a smooth function on the element interior. Therefore $\mathcal{L}^*\overline{w}$, $\mathcal{L}\overline{u}$ and $\mathcal{L}u'$ have to be viewed as Dirac distributions defined on the entire domain Ω . The exact equation for the fine scales u' then also contains a boundary term. Note that this term is not present in equation (4.26) because of assumption (4.21).

The global effect of the unresolved scales on the resolved scales can be stated explicitly through the nonvanishing element boundary terms. For a practical numerical method to solve the exact problem the Green's function will have to be approximated, and some form of localization is required. As proposed by Hughes in [47], the idea is to employ an element Green's function instead of the global one, which amounts to the local approximation of the Green's function. The result of applying an element Green's function is that the subgrid scales vanish on element boundaries, as assumed in equation (4.21). The non-local effect of the small scales on the large scales is then confined within individual elements. The initial paper on VMS by Hughes [47] thus gives a shorthand explanation of the method, since it immediately takes on the assumption of vanishing sub-grid scales.

4.3 Multiscale finite element method applied to the S-A turbulence model

Now that the variational multiscale method has been addressed, the multiscale stabilized formulation for the A-D-R equation can be derived. Similarly to chapter 3, first the weak form of the S-A model will be derived, after which the temporal discretization is performed and an iterative solution procedure is set up. The A-D-R equation is then presented, and its parameters are defined such that it represents the S-A turbulence model. The VMS method is then applied to the A-D-R equation. After a discretization in space the element matrices can be derived. Finally, the definition of the stabilization parameter is addressed, as well as modifications to the baseline S-A model.

4.3.1 Weak form

For the derivation of the weak form of the S-A turbulence model, let us assume purely homogeneous Dirichlet boundary conditions. It is reiterated that nonhomogeneous Dirichlet boundary conditions are enforced through use of the penalty method, which explained in appendix B. In this case all boundary terms vanish in the formulation, and we can write the weak form of equation 2.18 as,

$$\left(w, \frac{\partial\nu_e}{\partial t}\right) + (w, \mathbf{u} \cdot \nabla\nu_e) - (w, c_{b1}S_e\nu_e) + \left(w, c_{w1}f_w\left(\frac{\nu_e}{d}\right)^2\right) - \left(w, \frac{c_{b2}}{\sigma}\nabla\nu_e \cdot \nabla\nu_e\right) + \left(\nabla w, \frac{1}{\sigma}\left(\nu + \nu_e\right)\nabla\nu_e\right) = 0$$
(4.38)

with w the scalar weighting function, and integration by parts has been performed on the last term. Note also that the line above **u** is dropped for convenience. We can introduce the space of weighting functions and trial solutions as,

$$\mathcal{V} := \left\{ (w, \nu_e) \in \mathcal{H}^1 | (w, \nu_e) = 0 \text{ on } \Gamma_D \right\}.$$

$$(4.39)$$

Finally, we note that we can write the term that models the conservative diffusion effects in the original S-A equation (2.18) as,

$$-\frac{1}{\sigma}\nabla\cdot\left(\left(\nu+\nu_{e}\right)\nabla\nu_{e}\right) = -\frac{1}{\sigma}\left(\nabla\nu_{e}\cdot\nabla\nu_{e}+\left(\nu+\nu_{e}\right)\nabla^{2}\nu_{e}\right).$$
(4.40)

4.3.2 Temporal discretization and iterative solution procedure

We again apply the theta method for the temporal discretization of the weak formulation. Applied to equation (4.38) this yields,

$$(w, \nu_{e} - \hat{\nu}_{e}) + \Delta t \left\{ \theta \left[(w, \mathbf{u} \cdot \nabla \nu_{e}) - (w, c_{b1}S_{e}\nu_{e}) + \left(w, c_{w1}f_{w}\left(\frac{\nu_{e}}{d}\right)^{2}\right) - \left(w, \frac{c_{b2}}{\sigma}\nabla\nu_{e}\cdot\nabla\nu_{e}\right) + \left(\nabla w, \frac{1}{\sigma}\left(\nu + \nu_{e}\right)\nabla\nu_{e}\right) \right] + (1 - \theta) \left[(w, \mathbf{u} \cdot \nabla\hat{\nu}_{e}) - (w, c_{b1}S_{e}\hat{\nu}_{e}) + \left(w, c_{w1}f_{w}\left(\frac{\hat{\nu}_{e}}{d}\right)^{2}\right) - \left(w, \frac{c_{b2}}{\sigma}\nabla\hat{\nu}_{e}\cdot\nabla\hat{\nu}_{e}\right) + \left(\nabla w, \frac{1}{\sigma}\left(\nu + \hat{\nu}_{e}\right)\nabla\hat{\nu}_{e}\right) \right] \right\} = 0$$

$$(4.41)$$

where the left-hand-side is defined as the residual. The nonlinear problem is then solved using Newton's method, explained in section 3.2.2. The linear system of equations that is solved iteratively is given as,

$$J\left(\nu_e^k\right)\delta\nu_e = -\mathcal{R}^k\tag{4.42}$$

with $\delta\nu_e = \nu_e^{k+1} - \nu_e^k$, \mathcal{R}^k the residual at Newton iteration k, and J the Jacobian, i.e., the derivative of equation (4.41) with respect to ν_e . Note that the derivative of functions containing ν_e such as f_w are not considered for convenience, they are thus considered constant. Application of Newton's method to equation (4.41) yields,

$$(w,\delta\nu_{e}) + \Delta t\theta \left[(w,\mathbf{u}\cdot\nabla(\delta\nu_{e})) - (w,c_{b1}S_{e}\delta\nu_{e}) + 2\left(w,c_{w1}f_{w}\frac{\nu_{e}^{k}}{d^{2}}\delta\nu_{e}\right) - 2\left(w,\frac{c_{b2}}{\sigma}\nabla\nu_{e}^{k}\cdot\nabla(\delta\nu_{e})\right) + \left(\nabla w,\frac{\nu+\nu_{e}^{k}}{\sigma}\nabla(\delta\nu_{e})\right) \right] = -\Delta t\mathcal{R}^{k}$$

$$(4.43)$$

where the derivative to the first ν_e in the term $\Delta t \theta \left(\nabla w, \frac{1}{\sigma} \left(\nu + \nu_e \right) \nabla \nu_e \right)$ has been ignored since it does not fit the advection-diffusion-reaction equation form, as in Khurram et al. [52]. Bringing known terms to the right-hand-side and eliminating terms we obtain,

$$(w,\nu_{e}^{k+1}) + \Delta t\theta \left[\left(w, \mathbf{u} \cdot \nabla \nu_{e}^{k+1} \right) - \left(w, c_{b1}S_{e}\nu_{e}^{k+1} \right) + 2 \left(w, c_{w1}f_{w} \frac{\nu_{e}^{k}\nu_{e}^{k+1}}{d^{2}} \right) \right] - 2 \left(w, \frac{c_{b2}}{\sigma} \nabla \nu_{e}^{k} \cdot \nabla \nu_{e}^{k+1} \right) + \left(\nabla w, \frac{\left(\nu + \nu_{e}^{k} \right)}{\sigma} \nabla \nu_{e}^{k+1} \right) \right]$$
$$= (w, \hat{\nu}_{e}) + \Delta t \left\{ \theta \left[\left(w, c_{w1}f_{w} \left(\frac{\nu_{e}^{k}}{d} \right)^{2} \right) - \left(w, \frac{c_{b2}}{\sigma} \nabla \nu_{e}^{k} \cdot \nabla \nu_{e}^{k} \right) \right]$$
$$- (1 - \theta) \left[\left(w, \mathbf{u} \cdot \nabla \hat{\nu}_{e} \right) - \left(w, c_{b1}S_{e}\hat{\nu}_{e} \right) + \left(w, c_{w1}f_{w} \left(\frac{\hat{\nu}_{e}}{d} \right)^{2} \right) \right]$$
$$- \left(w, \frac{c_{b2}}{\sigma} \nabla \hat{\nu}_{e} \cdot \nabla \hat{\nu}_{e} \right) + \left(\nabla w, \frac{1}{\sigma} \left(\nu + \hat{\nu}_{e} \right) \nabla \hat{\nu}_{e} \right) \right] \right\}.$$
$$(4.44)$$

In order to obtain the correct form of this equation for the following section, we require integration by parts on the last term of the last line of equation (4.44). This yields,

$$-\left(\nabla w, \frac{1}{\sigma}\left(\nu+\hat{\nu}_{e}\right)\nabla\hat{\nu}_{e}\right) = \left(w, \frac{1}{\sigma}\nabla\hat{\nu}_{e}\cdot\nabla\hat{\nu}_{e}\right) + \left(w, \frac{1}{\sigma}\left(\nu+\hat{\nu}_{e}\right)\nabla^{2}\hat{\nu}_{e}\right).$$
(4.45)

Note that the last term on the right-hand-side will drop out when linear elements are employed.

4.3.3 Analogy of the S-A model with the A-D-R equation

The advection-diffusion-reaction equation for the modified turbulent viscosity is given as,

$$\mathbf{a} \cdot \nabla \nu_e - \kappa \nabla^2 \nu_e + s \nu_e = f \qquad \text{in } \Omega \tag{4.46}$$

with $\mathbf{a} \in \mathbb{R}^2$ the velocity field, $\kappa \geq 0$ the diffusion coefficient, s the reaction coefficient, and f the source term. For the reaction coefficient we have s > 0 for destruction and s < 0 for production. For s = 0 the equation can be seen to take the form of the advection-diffusion equation.

Taking into account the homogeneous Dirichlet boundary conditions we can put the A-D-R equation in its weak form by multiplying with the weighting function w, integrating over Ω , and performing integration by parts on the diffusion term as follows.

$$(w, \mathbf{a} \cdot \nabla \nu_e) + (\nabla w, \kappa \nabla \nu_e) + (w, s\nu_e) = (w, f)$$

$$(4.47)$$

We can adjust this equation to a suitable form for equation (4.44) by noting that the equation is linear:

$$(w, \mathbf{a} \cdot \nabla \nu_e^{k+1}) + (\nabla w, \kappa \nabla \nu_e^{k+1}) + (w, s \nu_e^{k+1}) = (w, f).$$
(4.48)

The equivalence of equations (4.44) and (4.48), using equation (4.45), can be observed by substituting the following terms for \mathbf{a} , κ , s and f:

$$\mathbf{a} = \Delta t \theta \left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \tag{4.49}$$

$$\kappa = \Delta t \theta \frac{(\nu + \nu_e^{\kappa})}{\sigma} \tag{4.50}$$

$$s = 1 + \Delta t \theta \left(-c_{b1}S_e + 2c_{w1}f_w \frac{\nu_e^k}{d^2} \right)$$

$$(4.51)$$

$$f = \hat{\nu}_{e} + \Delta t \left\{ \theta \left[c_{w1} f_{w} \left(\frac{\nu_{e}^{k}}{d} \right)^{2} - \frac{c_{b2}}{\sigma} \nabla \nu_{e}^{k} \cdot \nabla \nu_{e}^{k} \right] - (1 - \theta) \left[\mathbf{u} \cdot \nabla \hat{\nu}_{e} - c_{b1} S_{e} \hat{\nu}_{e} + c_{w1} f_{w} \left(\frac{\hat{\nu}_{e}}{d} \right)^{2} - \frac{1}{\sigma} \left((1 + c_{b2}) \nabla \hat{\nu}_{e} \cdot \nabla \hat{\nu}_{e} + (\nu + \hat{\nu}_{e}) \nabla^{2} \hat{\nu}_{e} \right) \right] \right\}.$$

$$(4.52)$$

4.3.4 Variational multiscale method applied to the A-D-R equation

The variational multiscale method was addressed in section 4.2. We follow the approach used as in this section, using the domain Ω represented by equations (4.13)-(4.15). We can split w and ν_e by overlapping sum decomposition into coarse and fine scales,

$$w = \overline{w} + w' \tag{4.53}$$

$$\nu_e = \overline{\nu_e} + \nu'_e \,. \tag{4.54}$$

A direct sum decomposition is used for the space of functions \mathcal{V} ,

$$\mathcal{V} = \overline{\mathcal{V}} \oplus \mathcal{V}' \tag{4.55}$$

with

$$\nu'_{e} = 0 \text{ on } \Gamma' \quad \forall \nu'_{e} \in \mathcal{V}' \\ w' = 0 \text{ on } \Gamma' \quad \forall w' \in \mathcal{V}'$$

$$(4.56)$$

and

$$\overline{\mathcal{V}} := \left\{ (\overline{\nu}_e, \overline{w}) \in \mathcal{H}^1 | (\overline{\nu}_e, \overline{w}) = 0 \text{ on } \Gamma \right\}.$$
(4.57)

Substituting the splitting of the trial solutions (4.54) and weighting functions (4.53) in the variational form (4.48) yields,

$$(\overline{w} + w', \mathbf{a} \cdot \nabla(\overline{\nu}_e + \nu'_e) + (\nabla(\overline{w} + w'), \kappa \nabla(\overline{\nu}_e + \nu'_e)) + (\overline{w} + w', s(\overline{\nu}_e + \nu'_e)) = (\overline{w} + w', f)$$
(4.58)

where the superscript k + 1 has been dropped for convenience. Because of linear independence of the weighting function the problem can be split into a coarse and fine scale problem. The coarse scale problem reads as,

$$(\overline{w}, \mathbf{a} \cdot \nabla(\overline{\nu}_e + \nu'_e)) + (\nabla\overline{w}, \kappa\nabla(\overline{\nu}_e + \nu'_e)) + (\overline{w}, s(\overline{\nu}_e + \nu'_e)) = (\overline{w}, f).$$
(4.59)

By employing linear independence of the solution slot we can write,

$$(\overline{w}, \mathbf{a} \cdot \nabla \overline{\nu}_e) + (\overline{w}, \mathbf{a} \cdot \nabla \nu'_e) + (\nabla \overline{w}, \kappa \nabla \overline{\nu}_e) + (\nabla \overline{w}, \kappa \nabla \nu'_e) + (\overline{w}, s \overline{\nu}_e) + (\overline{w}, s \nu'_e) = (\overline{w}, f).$$
(4.60)

The fine scale solution can be isolated by performing integration by parts on the second and fourth term on the left-hand-side, and using the assumption stated in equation (4.56). The problem then reads as,

$$(\overline{w}, \mathbf{a} \cdot \nabla \overline{\nu}_e) + (\nabla \overline{w}, \kappa \nabla \overline{\nu}_e) + (\overline{w}, s \overline{\nu}_e) - (\mathbf{a} \cdot \nabla \overline{w} + \kappa \nabla^2 \overline{w} - s \overline{w}, \nu'_e) = (\overline{w}, f).$$
(4.61)

Using the multiscale method we can approximate the fine scales ν'_e in terms of the coarse scale residual and the stability parameter τ as shown in equation (4.33). Note that the coarse scale residual in this definition is defined over the element interiors only. Discretizing the domain into n_{el} elements with characteristic mesh size h, we have the finite dimensional subspace $\overline{\nu}_h \subset \overline{\nu}$. Substituting (4.33) into equation (4.61) we obtain the following formulation: find $\overline{\nu}_{eh} \in \overline{\nu}_h$, such that,

$$(\overline{w}_{h}, \mathbf{a} \cdot \nabla \overline{\nu}_{eh}) + (\nabla \overline{w}_{h}, \kappa \nabla \overline{\nu}_{eh}) + (\overline{w}_{h}, s \overline{\nu}_{eh}) + (\mathbf{a} \cdot \nabla \overline{w}_{h} + \kappa \nabla^{2} \overline{w}_{h} - s \overline{w}_{h}, \tau (\mathbf{a} \cdot \nabla \overline{\nu}_{eh} - \kappa \nabla^{2} \overline{\nu}_{eh} + s \overline{\nu}_{eh}))_{\cup \Omega^{e}}$$
$$= (\overline{w}_{h}, f) + (\mathbf{a} \cdot \nabla \overline{w}_{h} + \kappa \nabla^{2} \overline{w}_{h} - s \overline{w}_{h}, \tau f)_{\cup \Omega^{e}}$$
(4.62)

for all $\overline{w}_h \in \overline{\mathcal{V}}_h$. The union over the element interiors is again defined as $\cup \Omega^e = \sum_{e=1}^{n_{el}} \Omega^e$. Note that this stabilized formulation could have also been found by using equation (4.34), with the linear advection-diffusion-reaction operator \mathcal{L} defined as,

$$\mathcal{L}\nu_e = \mathbf{a} \cdot \nabla \nu_e - \kappa \nabla^2 \nu_e + s\nu_e \tag{4.63}$$

and its adjoint for the weighting function,

$$\mathcal{L}^* w = -\mathbf{a} \cdot \nabla w - \kappa \nabla^2 w + sw.$$
(4.64)

The multiscale stabilized form of the Spalart-Allmaras turbulence model is obtained by substituting equations (4.49)-(4.52) in equation (4.62). As introduced in chapter 3, the coarse scale weighting functions can be expressed in terms of their basis functions. The coarse scale trial solutions can be expressed in terms of their basis functions and nodal unknowns as,

$$\overline{\nu}_{eh}(\mathbf{x}) = \sum_{A \in \eta_{\nu}} \phi_A(\mathbf{x}) \nu_{eA} \tag{4.65}$$

with η_{ν} the set of global modified turbulent viscosity nodes, ϕ_A the basis function for the nodal unknown ν_{eA} with respect to the global node number A. As this is a scalar equation the resulting element matrix contributions are straightforward, these are presented in appendix F.

4.3.5 Choice of stabilization parameter

The definition of the stability parameter is taken from Hauke [38], who extended the stability parameter derived by Franca and Valentin [37] such that it is valid for negative as well as positive reaction terms. The original stability parameter was derived from convergence and stability theory, this was elaborated on in the introduction chapter 1. This parameter is originally ment for the stabilization of the linear advection-diffusion-reaction equation. Here we will test its usefulness in the case of the nonlinear Spalart-Allmaras turbulence model equation.

The modified Franca and Valentin stability parameter for the advection-diffusionreaction equation is defined as,

$$\tau_{mfv} = \frac{1}{\frac{2\kappa}{m_k h^2} \zeta(Pe_2) + |s|\zeta(Pe_1)}$$

$$(4.66)$$

with

$$\zeta(Pe_x) = \begin{cases} 1 & \text{if } 0 \le Pe_x \le 1\\ Pe_x & \text{if } Pe_x > 1 \end{cases}$$
(4.67)

$$Pe_1 = \frac{2\kappa}{m_k |s| h^2} \tag{4.68}$$

$$Pe_2 = \frac{m_k |\mathbf{a}|h}{\kappa} \,. \tag{4.69}$$

Note that for **a** and *s* the absolute values are taken. The parameter m_k equals $\frac{1}{3}$ in practice for bilinear elements. It is noted by Franca and Valentin [37] that under the presence of advection, $\mathbf{a} \neq 0$, the characteristic mesh size *h* that yields the best numerical results can be determined using the element largest streamline distance. An illustration of this is provided in figure 4.3.



Figure 4.3: Computation of characteristic mesh size *h*.

4.3.6 Negative Spalart-Allmaras turbulence model

As introduced in section 2.3.3, some modifications to the baseline model are introduced in order to account for numerical problems arising from the original model. If the solution turns negative, the negative S-A turbulence model is solved instead. This PDE is given in equation (2.19). The multiscale stabilized advection-diffusion-reaction equation terms remain equal for this equation. However, changes in the definition of \mathbf{u} , κ , s and f have to be accounted for when solving the negative S-A turbulene model. The changes made in the negative model are small. By comparison with the S-A model it can therefore easily be observed that the definition of the parameters of the A-D-R equation for the negative S-A model are defined as,

$$\mathbf{a} = \Delta t \theta \left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \tag{4.70}$$

$$\kappa = \Delta t \theta \frac{(\nu + \nu_e^{\kappa} f_n)}{\sigma} \tag{4.71}$$

$$s = 1 + \Delta t \theta \left(-c_{b1}S - 2c_{w1}\frac{\nu_e^k}{d^2} \right)$$

$$(4.72)$$

$$f = \hat{\nu}_{e} + \Delta t \left\{ \theta \left[-c_{w1} \left(\frac{\nu_{e}^{k}}{d} \right)^{2} - \frac{c_{b2}}{\sigma} \nabla \nu_{e}^{k} \cdot \nabla \nu_{e}^{k} \right] - (1 - \theta) \left[\mathbf{u} \cdot \nabla \hat{\nu}_{e} - c_{b1} S \hat{\nu}_{e} - c_{w1} \left(\frac{\hat{\nu}_{e}}{d} \right)^{2} - \frac{1}{\sigma} \left((1 + c_{b2}) \nabla \hat{\nu}_{e} \cdot \nabla \hat{\nu}_{e} + (\nu + f_{n} \hat{\nu}_{e}) \nabla^{2} \hat{\nu}_{e} \right) \right] \right\}.$$

$$(4.73)$$

4.4 Traditional multiscale finite element formulation for the S-A turbulence model

In this section the multiscale method is directly applied to the S-A turbulence model. This more traditional way of applying the multiscale method results in a different formulation of the stabilization term. Below we first apply the multiscale method and address the fine scale problem. From this a definition of the advection, diffusion and reaction coefficient arises for the calculation of the stabilization parameter τ . Then the coarse scale problem is tackled by introducing an approximation for the fine scale unknown based on the coarse scale residual. The equations are then discretized in time and space and the iterative solution procedure in the form of Newton's method is applied. Finally, we sort the terms based on their advection, diffusion or reaction character such that the A-D-R equation is applicable to it.

4.4.1 Fine scale problem

As introduced in section 4.2, applying the multiscale method the weighting term and unknown can be split into a fine and coarse scale term. The problem can then be decomposed into a coarse and fine scale problem by virtue of linear independence of the coarse and fine scale weighting term. Let us address the fine scale problem for the Spalart-Allmaras equation in the strong form. We apply the overlapping sum decomposition to the modified turbulent viscosity $\nu_e = \overline{\nu_e} + \nu'_e$, and use the same space of functions splitting and boundary conditions as described in section 4.3.4. The fine scale problem becomes,

$$\frac{\partial(\overline{\nu_e} + \nu'_e)}{\partial t} + \mathbf{u} \cdot \nabla(\overline{\nu_e} + \nu'_e) - c_{b1} \left(\overline{S}_e + j\nu'_e\right) \left(\overline{\nu_e} + \nu'_e\right) + c_{w1} \left(\overline{f}_w + \eta\nu'_e\right) \left(\frac{\overline{\nu_e} + \nu'_e}{d}\right)^2 - \frac{c_{b2}}{\sigma} \nabla(\overline{\nu_e} + \nu'_e) \cdot \nabla(\overline{\nu_e} + \nu'_e) - \frac{1}{\sigma} \nabla \cdot \left(\left(\nu + (\overline{\nu_e} + \nu'_e)\right) \nabla(\overline{\nu_e} + \nu'_e)\right) = 0.$$

$$(4.74)$$

Here the multiscale method has also been applied to the terms f_w and S_e , as they contain the unknown as well. This process, along with the definition of j and η , is presented in appendix G.

The assumption of quasi-static subscales is made, i.e., $\frac{\partial \nu'_e}{\partial t} = 0$. This assumption is thus understood as the variation of the subscales in time being negligible [20]. Taking the coarse scale terms to right-hand-side and ignoring higher-order fine scale terms for simplicity we obtain,

$$\mathbf{u} \cdot \nabla \nu'_{e} - c_{b1} \left(\overline{S}_{e} + j \overline{\nu_{e}} \right) \nu'_{e} + \frac{c_{w1}}{d^{2}} \left(2 \overline{f}_{w} \overline{\nu_{e}} + \eta \overline{\nu_{e}}^{2} \right) \nu'_{e} - \frac{2c_{b2}}{\sigma} \left(\nabla \overline{\nu_{e}} \cdot \nabla \nu'_{e} \right) - \frac{1}{\sigma} \nabla \cdot \left((\nu + \overline{\nu_{e}}) \nabla \nu'_{e} + \nu'_{e} \nabla \overline{\nu_{e}} \right) = - \underbrace{\left(\frac{\partial \overline{\nu_{e}}}{\partial t} + \mathbf{u} \cdot \nabla \overline{\nu_{e}} - c_{b1} \overline{S}_{e} \overline{\nu_{e}} + c_{w1} \overline{f}_{w} \left(\frac{\overline{\nu_{e}}}{d} \right)^{2} - \frac{1}{\sigma} \left[\nabla \cdot \left((\nu + \overline{\nu_{e}}) \nabla \overline{\nu_{e}} \right) + c_{b2} \nabla \overline{\nu_{e}} \cdot \nabla \overline{\nu_{e}} \right] \right)}_{= \mathcal{R}(\overline{\nu_{e}})}$$

(4.75)

Observe the right-hand-side can be identified as minus the coarse scale residual. Assuming the coarse scale terms are known, the fine scale problem can be written as,

$$\mathcal{L}\nu_e' = -\mathcal{R}(\overline{\nu_e}) \tag{4.76}$$

4.4 Traditional multiscale finite element formulation for the S-A turbulence model

with \mathcal{L} a linear operator. As described in section 4.2, the operator \mathcal{L} can be inverted in order to obtain the fine scale unknown in terms of the coarse scale residual and an element Green's function. Hence the problem is localized. We approximate the Green's function by the stabilization parameter τ , which is defined by equations (4.66)-(4.69), such that $\nu'_e = -\tau \mathcal{R}(\overline{\nu_e}).$

In order to identify the advection, diffusion and reaction terms used for the calculation of τ , let us return to equation (4.75). After temporal discretization of the coarse scale time derivative term using the backward Euler method and ignoring nonlinear fine scale terms equation (4.75) becomes,

$$\Delta t \left(\mathbf{u} \cdot \nabla \nu'_{e} - c_{b1} \left(\overline{S}_{e} + j \overline{\nu_{e}} \right) \nu'_{e} + \frac{c_{w1}}{d^{2}} \left(2 \overline{f}_{w} \overline{\nu_{e}} + \eta \overline{\nu_{e}}^{2} \right) \nu'_{e} - \frac{2c_{b2}}{\sigma} \left(\nabla \overline{\nu_{e}} \cdot \nabla \nu'_{e} \right) - \frac{1}{\sigma} \nabla \cdot \left(\left(\nu + \overline{\nu_{e}} \right) \nabla \nu'_{e} + \nu'_{e} \nabla \overline{\nu_{e}} \right) \right) = - \left(\overline{\nu_{e}} - \overline{\hat{\nu}_{e}} + \Delta t \left(\mathbf{u} \cdot \nabla \overline{\nu_{e}} - c_{b1} \overline{S}_{e} \overline{\nu_{e}} + c_{w1} \overline{f}_{w} \left(\frac{\overline{\nu_{e}}}{d} \right)^{2} - \frac{1}{\sigma} \left[\nabla \cdot \left(\left(\nu + \overline{\nu_{e}} \right) \nabla \overline{\nu_{e}} \right) + c_{b2} \nabla \overline{\nu_{e}} \cdot \nabla \overline{\nu_{e}} \right] \right) \right).$$

$$(4.77)$$

The linear operator that has been inverted can be observed to have the following advection, diffusion and reaction terms with respect to the A-D-R equation (4.46):

$$\mathbf{a}_{\tau} = \Delta t \left(\mathbf{u} - \frac{2(c_{b2} + 1)}{\sigma} \overline{\nu_e} \right) \tag{4.78}$$

$$\kappa_{\tau} = \Delta t \frac{\nu + \overline{\nu_e}}{\sigma} \tag{4.79}$$

$$s_{\tau} = \Delta t \left(-c_{b1}(\overline{S}_e + j) + \frac{c_{w1}}{d^2} \left(2\overline{f}_w \overline{\nu_e} + \eta \overline{\nu_e}^2 \right) - \frac{1}{\sigma} \nabla^2 \overline{\nu_e} \right) \,. \tag{4.80}$$

Here the subscript τ has been added to indicate the use of this parameter for calculation of the stabilization parameter τ .

4.4.2Coarse scale problem

Let us now put our focus towards the coarse scale problem. The weak form of the coarse scale problem reads as,

$$\left(\overline{w}, \frac{\partial(\overline{\nu_e} + \nu'_e)}{\partial t}\right) + \left(\overline{w}, \mathbf{u} \cdot \nabla(\overline{\nu_e} + \nu'_e)\right) - \left(\overline{w}, c_{b1}\left(\overline{S}_e + j\nu'_e\right)\left(\overline{\nu_e} + \nu'_e\right)\right) + \left(\overline{w}, c_{w1}\left(\overline{f}_w + \eta\nu'_e\right)\left(\frac{\overline{\nu_e} + \nu'_e}{d}\right)^2\right) - \left(\overline{w}, \frac{c_{b2}}{\sigma}\nabla(\overline{\nu_e} + \nu'_e) \cdot \nabla(\overline{\nu_e} + \nu'_e)\right) + \left(\nabla\overline{w}, \frac{1}{\sigma}\left(\nu + \overline{\nu_e} + \nu'_e\right)\nabla(\overline{\nu_e} + \nu'_e)\right) = 0$$
(4.81)

where integration by parts has been performed on the last term assuming purely Dirichlet boundary conditions. The equation can then be expanded as follows splitting coarse and fine scale terms and ignoring higher-order fine scale terms,

$$\left(\overline{w}, \frac{\partial \overline{\nu_e}}{\partial t}\right) + \left(\overline{w}, \mathbf{u} \cdot \nabla \overline{\nu_e}\right) - \left(\overline{w}, c_{b1} \overline{S}_e \overline{\nu_e}\right) + \left(\overline{w}, c_{w1} \overline{f}_w \left(\frac{\overline{\nu_e}}{d}\right)^2\right) \\
- \left(\overline{w}, \frac{c_{b2}}{\sigma} \nabla \overline{\nu_e} \cdot \nabla \overline{\nu_e}\right) + \left(\nabla \overline{w}, \frac{1}{\sigma} \left(\nu + \overline{\nu_e}\right) \nabla \overline{\nu_e}\right) \\
+ \left(\overline{w}, \mathbf{u} \cdot \nabla \nu'_e\right) - \left(\overline{w}, c_{b1} \left(\overline{S}_e + j \overline{\nu_e}\right) \nu'_e\right) + \left(\overline{w}, \frac{c_{w1}}{d^2} \left(2\overline{f}_w \overline{\nu_e} + \eta \overline{\nu_e}^2\right) \nu'_e\right) \\
- \left(\overline{w}, \frac{2c_{b2}}{\sigma} \nabla \overline{\nu_e} \cdot \nabla \nu'_e\right) + \left(\nabla \overline{w}, \frac{1}{\sigma} \left(\left(\nu + \overline{\nu_e}\right) \nabla \nu'_e + \nu'_e \nabla \overline{\nu_e}\right)\right) = 0$$
(4.82)

where the quasi-static subscales assumption has again been made. The top two lines contain the coarse scale terms, while the bottom two lines of the equation represent the fine scale (cross-)terms. Higher-order fine scale terms are assumed small and have been neglected. Note that for coarser grids the coarse scale residual becomes larger and this assumption then does not completely hold anymore.

We would now like to have the fine scale terms only contain ν'_e in the solution slot, while bringing the other terms to the weighting slot. For example, for the advection term this can be done noting that,

$$\left(\overline{w}, \mathbf{u} \cdot \nabla \nu_e'\right) = \left(\mathbf{u}\overline{w}, \nabla \nu_e'\right) \,. \tag{4.83}$$

Then integration by parts yields,

where the integral over the boundary equals zero for the assumed Dirichlet boundary conditions. Similarly for the first term on the last row of equation (4.82) we have,

$$-\left(\overline{w}, \frac{c_{b2}}{\sigma} 2\nabla \overline{\nu_e} \cdot \nabla \nu'_e\right) = -\left(2\frac{c_{b2}}{\sigma} \nabla \overline{\nu_e} \,\overline{w}, \nabla \nu'_e\right) \,. \tag{4.85}$$

Application of integration by parts yields,

$$-\left(2\frac{c_{b2}}{\sigma}\nabla\overline{\nu_e}\,\overline{w},\nabla\nu'_e\right) = \left(2\frac{c_{b2}}{\sigma}\left(\nabla^2\overline{\nu_e}\,\overline{w} + \nabla\overline{w}\cdot\nabla\overline{\nu_e}\right),\nu'_e\right) \tag{4.86}$$

for Dirichlet boundary conditions. Treating each fine scale term in this way the following equation results for the coarse scale problem:

$$\left(\overline{w}, \frac{\partial \overline{\nu_e}}{\partial t}\right) + \left(\overline{w}, \mathbf{u} \cdot \nabla \overline{\nu_e}\right) - \left(\overline{w}, c_{b1} \overline{S}_e \overline{\nu_e}\right) + \left(\overline{w}, c_{w1} \overline{f}_w \left(\frac{\overline{\nu_e}}{d}\right)^2\right) \\
- \left(\overline{w}, \frac{c_{b2}}{\sigma} \nabla \overline{\nu_e} \cdot \nabla \overline{\nu_e}\right) + \left(\nabla \overline{w}, \frac{1}{\sigma} \left(\nu + \overline{\nu_e}\right) \nabla \overline{\nu_e}\right) \\
+ \left(-\overline{w} \nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla \overline{w} - c_{b1} \left(\overline{S}_e + j \overline{\nu_e}\right) \overline{w} + \frac{c_{w1}}{d^2} \left(2\overline{f}_w \overline{\nu_e} + \eta \overline{\nu_e}^2\right) \overline{w} \\
+ 2\frac{c_{b2}}{\sigma} \left(\nabla^2 \overline{\nu_e} \overline{w} + \nabla \overline{w} \cdot \nabla \overline{\nu_e}\right) + \frac{1}{\sigma} \nabla \overline{w} \cdot \nabla \overline{\nu_e} - \frac{1}{\sigma} \nabla \cdot \left(\left(\nu + \overline{\nu_e}\right) \nabla \overline{w}\right), \nu'_e\right) = 0.$$
(4.87)
The first two lines present the coarse scale terms. On the third and fourth line the fine scale terms have been collected in one term. Note that not taking the fine scales into account results in obtaining the unstabilized Galerkin equation.

As mentioned in the previous section, we now use an approximation for the fine scales $\nu'_e = -\tau \mathcal{R}(\overline{\nu_e})$, with $\mathcal{R}(\overline{\nu_e})$ defined in equation 4.75. The multiscale stabilized S-A equation then becomes,

$$\left(\overline{w}, \frac{\partial \overline{\nu_e}}{\partial t}\right) + \left(\overline{w}, \mathbf{u} \cdot \nabla \overline{\nu_e}\right) - \left(\overline{w}, c_{b1} \overline{S}_e \overline{\nu_e}\right) + \left(\overline{w}, c_{w1} \overline{f}_w \left(\frac{\overline{\nu_e}}{d}\right)^2\right) - \left(\overline{w}, \frac{c_{b2}}{\sigma} \nabla \overline{\nu_e} \cdot \nabla \overline{\nu_e}\right) + \left(\nabla \overline{w}, \frac{(\nu + \overline{\nu_e})}{\sigma} \nabla \overline{\nu_e}\right) + \left(\overline{w} \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \overline{w} + c_{b1} \left(\overline{S}_e + j\overline{\nu_e}\right) \overline{w} - \frac{c_{w1}}{d^2} \left(2\overline{f}_w \overline{\nu_e} + \eta \overline{\nu_e}^2\right) \overline{w} - 2\frac{c_{b2}}{\sigma} \left(\nabla^2 \overline{\nu_e} \overline{w} + \nabla \overline{\nu_e} \cdot \nabla \overline{w}\right) + \frac{(\nu + \overline{\nu_e})}{\sigma} \nabla^2 \overline{w}, \\ \tau \left(\frac{\partial \overline{\nu_e}}{\partial t} + \mathbf{u} \cdot \nabla \overline{\nu_e} - c_{b1} \overline{S}_e \overline{\nu_e} + c_{w1} \overline{f}_w \left(\frac{\overline{\nu_e}}{d}\right)^2 - \frac{1}{\sigma} \left[\nabla \cdot \left((\nu + \overline{\nu_e}) \nabla \overline{\nu_e}\right) + c_{b2} \nabla \overline{\nu_e} \cdot \nabla \overline{\nu_e}\right] \right) \right)_{\cup \Omega^e} = 0.$$

$$(4.88)$$

Discretization and iterative solution procedure 4.4.3

First the temporal discretization is applied. To keep things more clear, let us apply the backward Euler method instead of the theta method. Reordering terms and dropping the overlines for convenience this results in equation (4.88) to become,

$$\begin{pmatrix} w, \left(\mathbf{u} - \frac{c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla \nu_{e} \end{pmatrix} + \left(\nabla w, \frac{\nu + \nu_{e}}{\sigma} \nabla \nu_{e} \right) + \left(w, \left(\frac{1}{\Delta t} - c_{b1}S_{e} + c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)\nu_{e} \right) \\ + \left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\nu + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e} \right)w, \\ \tau \left(\left(\mathbf{u} - \frac{(c_{b2} + 1)}{\sigma} \nabla \nu_{e}\right) \cdot \nabla \nu_{e} - \frac{\nu + \nu_{e}}{\sigma} \nabla^{2}\nu_{e} + \left(\frac{1}{\Delta t} - c_{b1}S_{e} + c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)\nu_{e} \right)\right)_{\cup\Omega^{e}} = \left(w, \frac{\hat{\nu}_{e}}{\Delta t}\right) \\ + \left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\nu + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e}\right)w, \\ \tau \left(\frac{1}{\Delta t} - \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\nu + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e}\right)w, \\ \tau \left(\frac{1}{\Delta t} + \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\omega + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e}\right)w, \\ \tau \left(\frac{1}{\Delta t} + \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\omega + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e}\right)w, \\ \tau \left(\frac{1}{\Delta t} + \frac{1}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{\omega + \nu_{e}}{\sigma} \nabla^{2}w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_{e} + j\nu_{e}) + \frac{c_{w1}\nu_{e}}{d^{2}}\left(2f_{w} + \eta\nu_{e}\right) + \frac{2c_{b2}}{\sigma} \nabla^{2}\nu_{e}\right)w, \\ \tau \left(\frac{1}{\Delta t} + \frac{1}{\sigma} \nabla^{2}w + \frac{1}{\sigma} \nabla^{2}w$$

The first row and first term on the right-hand-side present the terms that would be obtained if the fine scales were ignored. We note that these terms are equal to those obtained by the multiscale method for the S-A model that was previously discussed in section 4.3. Hence the terms obtained after applying Newton's method will be the same for these Galerkin terms. After multiplication by Δt , and for the case of $\theta = 1$, the Galerkin terms are thus given by the A-D-R coefficients \mathbf{a}_g , κ_g , s_g and f_g as,

$$\mathbf{a}_g = \Delta t \left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \tag{4.90}$$

$$\kappa_g = \Delta t \frac{(\nu + \nu_e^{\kappa})}{\sigma} \tag{4.91}$$

$$s_g = 1 + \Delta t \left(-c_{b1}S_e + 2c_{w1}f_w \frac{\nu_e^k}{d^2} \right)$$

$$\tag{4.92}$$

$$f_g = \hat{\nu}_e + \Delta t \left(c_{w1} f_w \left(\frac{\nu_e^k}{d} \right)^2 - \frac{c_{b2}}{\sigma} \nabla \nu_e^k \cdot \nabla \nu_e^k \right) .$$
(4.93)

Application of Newton's method to the multiscale stabilization terms is done through use of equation (4.42), where the residual is made up of the stabilization term itself at Newton iteration k. To keep things clear and allow for a more easy implementation of terms the derivative with respect to ν_e is only applied to the solution slot of the stabilization term. The weighting function slot is then only considered at Newton iteration k. Using this strategy the following equation is obtained for the multiscale stabilizing terms of equation (4.89):

$$\begin{split} & \left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \cdot \nabla w + \frac{\nu + \nu_e^k}{\sigma} \nabla^2 w - \left(-\nabla \cdot \mathbf{u} - c_{b1} (S_e + j\nu_e^k) + \frac{c_{w1}\nu_e^k}{d^2} \left(2f_w + \eta\nu_e^k \right) + \frac{2c_{b2}}{\sigma} \nabla^2 \nu_e^k \right) w, \\ & \tau \left(\left(\left(\mathbf{u} - 2\frac{(c_{b2} + 1)}{\sigma} \nabla \nu_e^k \right) \cdot \nabla \nu_e^{k+1} - \frac{\nu + \nu_e^k}{\sigma} \nabla^2 \nu_e^{k+1} + \left(\frac{1}{\Delta t} - c_{b1}S_e + 2c_{w1}f_w \frac{\nu_e^k}{d^2} \right) \nu_e^{k+1} \right) \right)_{\cup \Omega^e} \\ & < = > \left(\left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \cdot \nabla w + \frac{\nu + \nu_e^k}{\sigma} \nabla^2 w - \left(-\nabla \cdot \mathbf{u} - c_{b1}(S_e + j\nu_e^k) \right) + \frac{c_{w1}\nu_e^k}{d^2} \left(2f_w + \eta\nu_e^k \right) + \frac{2c_{b2}}{\sigma} \nabla^2 \nu_e^k \right) w, \tau \left(\frac{\hat{\nu}_e}{\Delta t} - \frac{(c_{b2} + 1)}{\sigma} \nabla \nu_e^k \cdot \nabla \nu_e^k + c_{w1}f_w \left(\frac{\nu_e^k}{d} \right)^2 \right) \right)_{\cup \Omega^e} \end{split}$$

where the $\langle = \rangle$ again indicates the addition of the right- and left-hand-side to the Jacobian and right-hand-side vector, respectively. Multiplication by Δt results in the following A-D-R coefficients \mathbf{a}_w , κ_w and s_w for the weighting function slot of the multiscale stabilization term:

$$\mathbf{a}_w = \Delta t \left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_e^k \right) \tag{4.95}$$

$$\kappa_w = \Delta t \frac{\left(\nu + \nu_e^k\right)}{\sigma} \tag{4.96}$$

$$s_w = \Delta t \left(-\nabla \cdot \mathbf{u} - c_{b1} (S_e + j\nu_e^k) + \frac{c_{w1}\nu_e^k}{d^2} \left(2f_w + \eta\nu_e^k \right) + \frac{2c_{b2}}{\sigma} \nabla^2 \nu_e^k \right) \,. \tag{4.97}$$

Similarly, for the solution slot the coefficients \mathbf{a}_s , κ_s , s_s and f_s can be defined as,

$$\mathbf{a}_{s} = \Delta t \left(\mathbf{u} - 2 \frac{(c_{b2} + 1)}{\sigma} \nabla \nu_{e}^{k} \right)$$

$$(4.98)$$

$$\kappa_s = \Delta t \frac{\left(\nu + \nu_e^k\right)}{\sigma} \tag{4.99}$$

$$s_s = 1 + \Delta t \left(-c_{b1}S_e + 2c_{w1}f_w \frac{\nu_e^k}{d^2} \right)$$
(4.100)

$$f_s = \hat{\nu}_e + \Delta t \left(c_{w1} f_w \left(\frac{\nu_e^k}{d} \right)^2 - \frac{(c_{b2} + 1)}{\sigma} \nabla \nu_e^k \cdot \nabla \nu_e^k \right).$$
(4.101)

By comparison with equation (4.62) it can be observed the multiscale stabilized formulation written in the form of the A-D-R equation is given as,

$$(w, \mathbf{a}_{g} \cdot \nabla \nu_{e}) + (\nabla w, \kappa_{g} \nabla \nu_{e}) + (w, s_{g} \nu_{e}) + (\mathbf{a}_{w} \cdot \nabla w + \kappa_{w} \nabla^{2} w - s_{w} w, \tau (\mathbf{a}_{s} \cdot \nabla \nu_{e} - \kappa_{s} \nabla^{2} \nu_{e} + s_{s} \nu_{e}))_{\cup \Omega^{e}}$$
(4.102)
$$= (w, f_{g}) + (\mathbf{a}_{w} \cdot \nabla w + \kappa_{w} \nabla^{2} w - s_{w} w, \tau f_{s})_{\cup \Omega^{e}}$$

with \mathbf{a}_g , κ_g , s_g and f_g given by equations (4.90)-(4.93), \mathbf{a}_w , κ_w and s_w given by equations (4.95)-(4.97) and \mathbf{a}_s , κ_s , s_s and f_s given by equations (4.98)-(4.101). The stabilization parameter τ is given by equations (4.66)-(4.69), in which \mathbf{a} , κ and s are defined by equations (4.78)-(4.80).

4.4.4 SUPG type stabilization for the S-A turbulence model

It is interesting to obtain an SUPG stabilized formulation for the S-A model in order to compare its performance to the multiscale stabilized formulation. Characteristic of an SUPG type stabilization is the weighting function containing only the advection term. Hence from the multiscale formulation defined in the previous subsection we can obtain the SUPG stabilized formulation by simply neglecting the weighting function terms other than the advection term. Therefore the SUPG stabilized S-A model formulation can be defined as,

$$(w, \mathbf{a}_g \cdot \nabla \nu_e) + (\nabla w, \kappa_g \nabla \nu_e) + (w, s_g \nu_e) + (\mathbf{a}_w \cdot \nabla w, \tau (\mathbf{a}_s \cdot \nabla \nu_e - \kappa_s \nabla^2 \nu_e + s_s \nu_e))_{\cup \Omega^e} = (w, f_g) + (\mathbf{a}_w \cdot \nabla w, \tau f_s)_{\cup \Omega^e}$$

$$(4.103)$$

with \mathbf{a}_g , κ_g , s_g and f_g given by equations (4.90)–(4.93), \mathbf{a}_w given by equation (4.95), and \mathbf{a}_s , κ_s , s_s and f_s given by equations (4.98)–(4.101). For the stabilization parameter the same definition is used as for the SUPG stabilizing term in the RANS equations, since using Franca and Valentin's multiscale stabilization parameter would be inappropriate as it contains a reaction term. The definition of the SUPG stabilization parameter is repeated here for convenience:

$$\tau_m = \left(\frac{4}{(\Delta t)^2} + \mathbf{v}_h \cdot (\boldsymbol{\mathcal{G}} \cdot \mathbf{v}_h) + c_I (\nu + \nu_t)^2 \boldsymbol{\mathcal{G}} : \boldsymbol{\mathcal{G}}\right)^{-\frac{1}{2}}.$$
(4.104)

4.5 Comparison of two different multiscale approaches

Sections 4.3 and 4.4 describe two different approaches to obtain a multiscale stabilized formulation for the S-A turbulence model. The first one was inspired by the paper of Khurram et al. [52], while the second approach is the more traditional approach of applying the variational multiscale method. In this section the two methods are compared. Note that Khurram's approach results in one definition for \mathbf{a} , κ , s and f, while for the traditional multiscale approach we find separate definitions for the Galerkin, weighting, solution and stabilizing terms.

Let us first focus on the definition of the stabilization parameter coefficients, resulting from the fine scale problem. Khurram's approach uses application of Newton's method to the weak form of the S-A equation to determine these coefficients, which is a type of linearization. For the application of the traditional multiscale method a linearization is used as well for the fine scales. Hence both approaches lead to similar results, as can be observed by comparing equations (4.49)-(4.51) with equations (4.78)-(4.80). The change in definition of **a** results from the application of integration by parts in the weak form using Khurram's approach. The different definition of *s* however results from the assumption of the terms f_w and S_e to be constants, while these are actually highly nonlinear functions of ν_e . In the traditional multiscale approach these terms are treated as such, resulting in extra terms of which the derivation is shown in appendix G. Next to this, a temporal term can be noted in the reaction term for the Khurram method, which is not present in the traditional multiscale formulation. The reason it is not present in the latter is the assumption of quasi-static subscales. When this assumption is not made, indeed the temporal term $\frac{\partial \nu'_e}{\partial t}$ will be present. However, when applying temporal discretization to the time derivative of the fine scale unknown the fine scale term $\hat{\nu}'_e$ arises that needs to be evaluated at the previous time step. In the fine scale problem this term needs to be brought to the right-hand-side and then becomes part of the coarse scale residual. As Khurram's approach does not consider this, the presence of the temporal term in the reaction term for the stabilization parameter is questionable.

The comparison of the coarse scale equations is performed on the level before Newton's method is applied to see what equation is actually solved. The first method is described by equation (4.62) with its parameters given by equations (4.49)–(4.52). Let us revert the Newton's method used by setting $\delta\nu_e$ equal to zero, i.e., $\nu_e^{k+1} = \nu_e^k$. In this way the equation that is converged towards is obtained. Equally setting $\theta = 1$ to obtain backward Euler and setting $\delta\nu_e = 0$ results in the following equation:

$$\left(w, \left(\mathbf{u} - \frac{c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla \nu_{e}\right) + \left(\nabla w, \frac{(\nu + \nu_{e})}{\sigma} \nabla \nu_{e}\right) + \left(w, \left(\frac{1}{\Delta t} - c_{b1}S_{e} + c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)\nu_{e}\right) \\
+ \left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{(\nu + \nu_{e})}{\sigma} \nabla^{2}w - \left(\frac{1}{\Delta t} - c_{b1}S_{e} + 2c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)w, \\
\tau \left(\left(\mathbf{u} - \frac{c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla \nu_{e} - \frac{(\nu + \nu_{e})}{\sigma} \nabla^{2}\nu_{e} + \left(\frac{1}{\Delta t} - c_{b1}S_{e} + c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)\nu_{e}\right)\right)_{\cup\Omega^{e}} = \left(w, \frac{\hat{\nu}_{e}}{\Delta t}\right) \\
+ \left(\left(\mathbf{u} - \frac{2c_{b2}}{\sigma} \nabla \nu_{e}\right) \cdot \nabla w + \frac{(\nu + \nu_{e})}{\sigma} \nabla^{2}w - \left(\frac{1}{\Delta t} - c_{b1}S_{e} + 2c_{w1}f_{w}\frac{\nu_{e}}{d^{2}}\right)w, \tau \frac{\hat{\nu}_{e}}{\Delta t}\right).$$
(4.105)

Again the overlines indicating the coarse scales have been dropped for convenience. For the traditional multiscale approach equation (4.89) is written in the proper form, before application of Newton's method. As noted in the previous section, the Galerkin terms in equations (4.105) and (4.89) can be observed to be equal, which is not surprising.

Focusing on the differences in the multiscale stabilization term, the largest difference can be noted in the reaction term of the weighting function. The application of the traditional multiscale method gives rise to additional reaction terms, compared to Khurram's approach, in the form of the divergence of the velocity field and the Laplacian of ν_e . Similar to the fine scale problem, we again find the extra reaction terms arising from application of the multiscale method to the nonlinear functions f_w and S_e , which are not considered in Khurram's approach. Finally, the temporal reaction term in the weighting function of the method by Khurram can be noted, which is not present in the formulation of the traditional multiscale approach. The same difference was found for the reaction term arising from the fine scale problem. The reason it is not present in the traditional multiscale formulation is again the assumption of quasi-static subscales. As noted previously, since Khurram's approach does not consider the fine scales at the previous time step the use of this term in the weighting function reaction term is questionable.

Chapter 5

Numerical Results

In this chapter the results will be presented of the equations and code that were derived in the previous chapters and implemented into libMesh [53]. We first show code verification results for the SUPG, PSPG and LSIC stabilized Navier-Stokes equations in section 5.1. In the following section 5.2 the verification results for the multiscale stabilized advection-diffusion-reaction equation are presented. In the final section verification as well as validation is performed for the Reynolds-averaged Navier-Stokes equations with the Spalart-Allmaras turbulence model. It is noted that the validation part is not of high importance, as it is well known the Spalart-Allmaras turbulence model comes with certain modeling errors. Here we are interested in the robustness of the SUPG and multiscale finite element formulations for the Spalart-Allmaras turbulence model compared to the standard Galerkin method. This will be addressed in the final section 5.3.

5.1 Navier-Stokes test cases

In this section the correct implementation of the stabilized Navier-Stokes equations will be evaluated by means of verification methods. First the implementation of the Galerkin terms will be checked by means of the method of manufactured solutions (MMS). The stabilization of the formulation is then tested using a lid-driven cavity flow problem at a Reynolds number of 5200. Finally a simple laminar channel flow problem for which an exact analytic solution is known is analyzed.

5.1.1 Manufactured solution

A useful method for code verification is the method of manufactured solutions [58, 59]. The idea is that one can assume an exact distribution of the unknowns for a PDE on any domain, this is the manufactured solution. This solution can be substituted into the PDE to be solved, and the resulting equation can be used as source term for the original problem. Solving this new problem results in finding the original manufactured solution. One can see that this problem is solved in a reversed order compared to the normal order of operations. Instead of finding the solution to a problem which results from the choice boundary conditions and source terms, one assumes the solution and finds the corresponding problem. Of course the manufactured solution needs to satisfy

certain conditions so that it is able to test the code correctly. Some of these conditions the manufactured solution should comply with are: it should be composed of smooth functions, it should be general enough to exercise every term in the equation, and it should have a sufficient amount of non-trivial derivatives.

For verifying the libMesh code for the incompressible N-S equations, the following manufactured solution is taken from Donea and Huerta [27] on a square domain $\Omega = [0, 1[\mathbf{x}]0, 1[$:

$$u(x,y) = x^{2}(1-x)^{2} (2y - 6y^{2} + 4y^{3})$$

$$v(x,y) = -y^{2}(1-y)^{2} (2x - 6x^{2} + 4x^{3})$$

$$p(x,y) = x(1-x).$$
(5.1)

For this manufactured solution the conservation of mass is satisfied, i.e., the velocity field is divergence free [34] and hence the contribution to the continuity equation will equal zero. The magnitude of the velocity equals zero on the boundary of the domain $\Omega =]0, 1[x]0, 1[$. The Dirichlet boundary conditions for the velocity and pressure are imposed by use of the penalty method (see appendix B).

Substituting the above exact solutions into the N-S equations (2.7) and (2.1) we can get the expression for the source term with help of a symbolic manipulation program. The following source terms are obtained setting $\nu = 1$:

$$\begin{split} f_x(x,y) &= 1 - 2x - 2(1-x)^2(2y - 6y^2 + 4y^3) + 8x(1-x)(2y - 6y^2 + 4y^3) - 2x^2(2y - 6y^2 + 4y^3) \\ &\quad - x^2(1-x)^2(-12 + 24y) + x^2(1-x)^2(2y - 6y^2 + 4y^3)(2x(1-x)^2(2y - 6y^2 + 4y^3) \\ &\quad - 2x^2(1-x)(2y - 6y^2 + 4y^3)) - y^2(1-y)^2(2x - 6x^2 + 4x^3)x^2(1-x)^2(2-12y + 12y^2) \\ f_y(x,y) &= y^2(1-y)^2(-12 + 24x) + 2(1-y)^2(2x - 6x^2 + 4x^3) - 8y(1-y)(2x - 6x^2 + 4x^3) \\ &\quad + 2y^2(2x - 6x^2 + 4x^3) - x^2(1-x)^2(2y - 6y^2 + 4y^3)y^2(1-y)^2(2-12x + 12x^2) \\ &\quad - y^2(1-y)^2(2x - 6x^2 + 4x^3)(-2y(1-y)^2(2x - 6x^2 + 4x^3) + 2y^2(1-y)(2x - 6x^2 + 4x^3)) \\ f_p(x,y) &= 0 \,. \end{split}$$

Here, f_x , f_y and f_p are the source terms corresponding to the x-momentum, y-momentum and continuity equation, respectively. The source terms are substituted in the N-S equations. These equations are solved on several uniform meshes ranging from coarse to fine with Q1Q1 elements. A 9-point quadrature rule is used for numerical integration. The solution is compared to the exact solution by taking the L_2 norm of the error with respect to the mesh size. For example, for the velocity in x-direction the L_2 norm can be written as,

$$||\boldsymbol{u}|| = \left(\sum_{i=1}^{n_{el}} h^2 \left(u_{i,exact} - u_i\right)^2\right)^{\frac{1}{2}}.$$
(5.2)

with h the mesh size (in one direction) and n_{el} the number of elements. Here the velocity is taken at the corresponding cell center.

While refining the mesh, the error of the manufactured solution should converge, since the discretization error becomes smaller. In figure 5.1 the convergence of this error is shown. It can be observed the error of the pressure converges with order 1, while the error of the velocity in both directions converges with order 2. Figure 5.2 presents the computed pressure distribution on a 32x32 uniform element mesh. In figure 5.3 the error distribution for this case is shown.



Figure 5.1: Convergence of L_2 error for the N-S equation unknowns with refining mesh size.



5.1.2 Lid-driven cavity flow

The lid-driven cavity flow problem is a standard benchmark problem for the incompressible Navier-Stokes equations. Figure 5.4 shows the problem description. Here it will be used to test the code at a Reynolds number which requires some form of stabilization. A main vortex will develop in the middle of the cavity, while secondary vortices can develop in the corners of the cavity, depending on the Reynolds number used. Overly diffusive behavior caused by a stabilization method may result in the secondary vortices not being captured. Hence this problem is a good test case for making sure the stabilized formulation does not cause numerical diffusion to deteriorate the results. We will use the form of the Navier-Stokes equations with the SUPG, LSIC and PSPG stabilization terms, for which the finite element formulation was derived in chapter 3.



Figure 5.4: Lid-driven cavity flow test case

Here we will test a lid-driven cavity flow problem at Re = 5200. This test case has also been performed for two-level and multiscale finite element methods [35, 56]. In figure 5.5 the streamlines and velocity magnitude are presented for this case. A uniform 80x80 element mesh was used with first order Lagrange elements. In total we can identify four seperate vortices. Equal results are observed compared to those found for the multiscale finite element method [56].



Figure 5.5: Streamlines and velocity magnitude for lid-driven cavity flow on an 80x80 4-node element mesh.

5.1.3 Laminar channel flow

The two-dimensional channel flow test case is shown in figure 5.6. For this case periodic boundary conditions are used. Therefore this case serves well for the purpose of testing the usage of periodic boundary conditions in libMesh, since this test case will also be used for the turbulent case with the RANS equations and turbulence model employed.

The Reynolds number for channel flow based on the diameter D of the channel is defined as,

$$\operatorname{Re}_{D} = \frac{u_{max}D}{\nu} \tag{5.3}$$

with ν the kinematic viscosity of the fluid. For laminar flow conditions (Re_D < ~2040[5]) this problem results in a parabolic velocity profile for u. This can be shown through a force equilibrium of the channel. The resulting analytic equation for the velocity profile in the flow direction can be shown to be,

$$u(y) = \frac{1}{2\mu} \frac{\partial p}{\partial x} \left(R^2 - r^2 \right) \tag{5.4}$$

with μ the dynamic viscosity of the fluid, $\frac{\partial p}{\partial x}$ the pressure gradient in the flow direction, R the pipe radius and r = y - R the distance in y-direction from the center-line of the pipe.



Figure 5.6: Channel flow test case.

Taking $\rho = R = \frac{\partial p}{\partial x} = 1$ the velocity profile and maximum velocity become,

$$u(y) = \frac{2y - y^2}{2\nu}, \qquad u_{max} = \frac{1}{2\nu},$$
 (5.5)

and in return the Reynolds number becomes,

$$\operatorname{Re}_D = \frac{1}{\nu^2} \,. \tag{5.6}$$

Note that the pressure gradient is added through use of a unit source term for the xmomentum equation. Taking the pressure term to the right-hand-side in the N-S equations we obtain $-\frac{1}{\rho}\frac{\partial p}{\partial x} = f_x = 1$. The remainder of the pressure gradient term then only describes the fluctuations. Figure 5.7 shows the computed velocity profile compared to the exact solution. Here we have set the kinematic viscosity $\nu = \frac{1}{25}$, i.e., $\operatorname{Re}_D = 625$. Note from the analytic equation that this results in a maximum velocity in x-direction u of 12.5. It can be observed that the two velocity profiles agree well. This solution is computed on a Q1Q1 uniform mesh with 16 elements in y-direction and 2 elements in the direction of flow.



Figure 5.7: Laminar channel flow solution for $\nu = \frac{1}{25}$ compared to exact solution at $\frac{x}{L} = 0.5$.

5.2 Advection-diffusion-reaction test cases

Before continuing with the RANS equations and S-A turbulence model, the correctness of implementation of the stabilized advection-diffusion-reaction equation has to be tested. We will again use the method of manufactured solutions, as well as a skew advection case which checks the correct working of the stabilization parameter. These test cases are taken from Khurram et al. [52], though the skew advection case is a common benchmark problem for the stabilized A-D-R equation.

5.2.1 Manufactured solution

We use the following manufactured solution for the A-D-R equation (4.46) on the square domain $\Omega =]0, 1[x]0, 1[:$

$$\nu_e(x, y) = \sin(2\pi x) \sin(2\pi y)$$
(5.7)

with

$$\nu_e = 0 \quad \text{on } \Gamma. \tag{5.8}$$

Substitution of the manufactured solution into the A-D-R equation results in the following source term:

1

$$f = 2\pi \{ a_x \cos(2\pi x) \sin(2\pi y) + a_y \sin(2\pi x) \cos(2\pi y) \} + (8\pi^2 \kappa + s) \sin(2\pi x) \sin(2\pi y)$$

with a_x and a_y the advection in corresponding directions.

A study is performed on the convergence of the L_2 norm of the error while refining the mesh. The convergence rate for uniform meshes consisting of 4-noded quadrilaterals is shown in figure 5.8. For these cases the coefficients κ and s are taken 1 and 10, respectively, while the advection **a** is taken 10 in each spatial direction. The L_2 norm of the error can be seen to converge with an order of two.



Figure 5.8: Convergence of L_2 error for the A-D-R equation unknown with refining mesh size.



Figure 5.9: Problem description for the skew advection case.

5.2.2 Skew advection case

The skew advection case is useful for testing the proper working of the multiscale stabilization terms with the Franca and Valentin [37] stabilization parameter. The problem description is given in figure 5.9. This problem is solved for several element Péclet (α) and Damköhler (σ) numbers on a 10x10 uniform 4-node quadrilateral element mesh. The dimensionless numbers α and σ are defined for the A-D-R equations as,

$$\alpha = \frac{||\mathbf{a}||h}{2\kappa} \tag{5.9}$$

$$\sigma = \frac{sh}{||\mathbf{a}||} \,. \tag{5.10}$$

On the 10x10 mesh we solve the problem with multiscale stabilization, SUPG stabilization, as well as the Galerkin method. SUPG stabilization can be achieved by only including the advective weighting term, as shown in section 4.4.4. The solutions are compared to a fully resolved solution on an 80x80 mesh using the unstabilized Galerkin formulation. This is shown in figure 5.10, where the results are presented for a production case (s < 0) with $\alpha = 1.5$ and $\sigma = -0.5$. Indeed, the multiscale stabilized method gives improved results compared to the SUPG stabilized and Galerkin method.



(a) Fully resolved solution on an 80x80 mesh.

(b) Multiscale stabilized method on a 10x10 mesh.



(c) SUPG stabilized method on a 10x10 mesh.

(d) Galerkin method on a 10x10 mesh.

Figure 5.10: Various solutions for production case: $\alpha = 1.5$, s = -0.5.

A better view of this is obtained by creating a line plot across the diagonal of the domain. Figure 5.11 shows the solution in the form of such a line plot for three different production cases, including the one in figure 5.10. The multiscale method can be seen to cope well with the production cases as it is closest to the fully resolved solution. The Galerkin method has no form of stabilization and therefore lacks some form of stabilization for the advection and reaction terms. As mentioned in [52], the SUPG method serves well for providing stabilization to convection dominated flow, but can be seen to be inadequate for reaction dominated flow. This results in a phase error, as shown in figure 5.11d. Compared to the findings in [52], equal results are achieved for the multiscale case using the Franca and Valentin [37] stabilization parameter.

In figures 5.12 and 5.13 the solution for two destruction cases are shown. Again the fully resolved solution on an 80x80 mesh is presented, as well as the multiscale, SUPG



Figure 5.11: Diagonal line plots of fully resolved solution versus multiscale, SUPG and Galerkin method for various production cases (s < 0).

and Galerkin method on a 10x10 mesh. It can be observed the multiscale solution shows the most correct behavior, and is in accordance with the results presented in [52]. The observed wiggles in the solution in figure 5.13 for the Galerkin method can be attributed to the high element Péclet number for this case.



Figure 5.12: Fully resolved solution versus multiscale, SUPG and Galerkin method for destruction case (s > 0): $\alpha = 1.0$, $\sigma = 10.0$.



Figure 5.13: Fully resolved solution versus multiscale, SUPG and Galerkin method for destruction case (s > 0): $\alpha = 10.0$, $\sigma = 0.1$.

5.3 Reynolds-averaged Navier-Stokes test cases

In the final section of this chapter the implementation of the Reynolds-averaged Navier-Stokes equations as well as the Spalart-Allmaras turbulence model will be verified through the method of manufactured solutions. Then two more test cases are presented, one for which DNS data is available and one for which experimental data is available. These are a turbulent channel flow problem and a backward facing step problem, respectively. The goal for these two cases is to determine the robustness and accuracy of the multiscale stabilized formulations compared to the Galerkin method.

5.3.1 Manufactured solution

The manufactured solution for the RANS equations will be taken equal to those used for the N-S equations, i.e., equations (5.1). However, the variables are now interpreted as their ensemble average \bar{u} , \bar{v} and \bar{p} . Note that we drop the bars in the notation from this point for convenience. The variables are substituted into the RANS equations (2.12) and (2.17). For the kinematic viscosity ν a constant such as $\nu = \frac{1}{5}$ can be taken, while the turbulent viscosity ν_t is a variable that is computed through the S-A model. The diffusion term can be computed using the form given in equation (4.6). This way we obtain a separate source term for the x- and y-momentum equation, while the source term for the continuity equation again yields 0. The generated source terms are provided in appendix H.

For the S-A model a manufactured solution for the modified kinematic viscosity ν_e is used which is positive for the entire domain. Note that a negative ν_e would be nonphysical. A manufactured solution need not be physical, but it does offer some advantages. For example, as mentioned by Eça et al.[31], similar difficulties in the solution and error estimation processes will arise by exercising the terms in the PDE in a comparable way to a real problem. We employ the following manufactured solution for ν_e on the same domain $\Omega =]0, 1[x]0, 1[:$

$$\nu_e = 0.05 \sin(\pi x) \sin(\pi y) \,. \tag{5.11}$$

The solution is substituted into the S-A model equation (2.18) and all of its corresponding parameters. The generated source term is presented in appendix H. Note that the definition of the wall distance for the manufactured solution seems not to matter as long as they are defined in the same manner for both the S-A model and the manufactured solution.

A segregated setup is used for solving the problem. An initial solution is used for ν_e which satisfies the boundary conditions and has a small initial value for non-boundary nodes. The RANS equations are then first solved, after which the S-A equation is solved using the updated solution for the velocity. This completes one pseudo time step. Solving the manufactured solution problem in this manner the solution converges in a few pseudo time steps.

In figure 5.14 the L_2 error is presented for each of the unknowns on several meshes ranging from coarse to fine. For the RANS equations we note the velocity unknowns converge with optimal second order convergence, while the pressure converges with order close to 2. For the S-A model unknown second order convergence is also observed.



Figure 5.14: Convergence of L_2 error for the RANS equation unknowns as well as the S-A turbulence model unknown with refining mesh size.

5.3.2 Turbulent channel flow

In this section a turbulent channel flow test case will be examined using the RANS equations with the S-A turbulence model, which are solved in a segregated manner. In section 5.3.2.1 the test case setup is presented, along with the validation of the case using DNS results. Then in section 5.3.2.2 a refinement study is performed, in which we test the performance of several S-A formulations for the turbulent channel flow case on meshes ranging from coarse to fine. Finally, in section 5.3.2.3 the magnitude of the different terms of the S-A equation are analyzed for the turbulent channel flow problem. This is done in order to see which terms govern the problem, and clarifies the results found in section 5.3.2.2.

5.3.2.1 Turbulent channel flow setup and validation

The turbulent flow channel test case has the same setup as the laminar case, as shown in figure 5.15. Due to the turbulent nature of the flow however, the velocity profile is not parabolic anymore. The mixing of the fluid in turbulent flow causes the velocity distribution to be more flattened and uniform, as portrayed in the setup figure.

The wall distance for the turbulence model for this case can simply be computed using the following equations:

$$d = y_{qp} \qquad \text{if } y_{qp} \le R$$
$$d = 2R - y_{qp} \qquad \text{if } y_{qp} > R$$

Here y_{qp} represents the *y*-coordinate of the current quadrature point under consideration. For the multiscale stabilization parameter the characteristic mesh size *h* has to be computed. A simple algorithm is formulated to ensure a correct calculation of this parameter according to figure 4.3. Note that the direction of the advection velocity **a** is not equal to the direction of fluid velocity, as it is calculated using equation (4.49) or (4.78), depending on the multiscale stabilization method used. Hence *h* is in general not simply equal to the mesh width.



Figure 5.15: Turbulent channel flow test case.

Similar to the laminar channel flow case, we take $\rho = R = \frac{\partial \bar{p}}{\partial x} = 1$. Through a force equilibrium it can be shown that this leads to the friction velocity u_{τ} (see appendix A

for the nomenclature of wall-bounded flows) being equal to 1. Hence for the Reynolds number based on the friction velocity we have,

$$Re_{\tau} = \frac{u_{\tau}R}{\nu} = \frac{1}{\nu}.$$
 (5.12)

Next to this the wall units now equal $u^+ = \bar{u}$ and $y^+ = \frac{y}{\nu}$. The pressure gradient is again obtained by setting the source term on the right-hand-side of the RANS equations equal to 1.

The turbulent channel flow is simulated at $Re_{\tau} = 550$. A direct numerical simulation for this case has been performed by del Álamo and Jiménez [26]. Of course the S-A model only models turbulence using, among other things, empiricism and arguments of dimensional analysis. The use of a very fine mesh will therefore still result in modeling error being present in the solution. This is shown in figure 5.16 where the solution is presented on a semilog plot for a simulation run using 2048 uniform elements in *y*direction, while using only 2 elements in the direction of flow. We employ backward Euler $(\theta = 1)$, linear elements, periodic boundary conditions, and use a time step of $\Delta t = 0.01$. Next to this the generalized minimal residual method (GMRES) linear system solver and the incomplete LU (ILU) preconditioner from the PETSc library are used. The solution is compared with the DNS results obtained in [26]. It can be observed the S-A model has good correspondence with the DNS results, especially in the viscous sublayer.



Figure 5.16: Comparison of velocity profile using DNS (dashed line) and RANS with S-A model (solid line) for turbulent channel flow simulated at $Re_{\tau} = 550$. For RANS 2048 elements are used normal to the direction of flow.

5.3.2.2 Refinement study for the turbulence channel flow test case

In order to see how well the multiscale finite element methods for the S-A model perform, we compare the fine solution used for the validation to the solution obtained on several coarser meshes using the Galerkin method and three stabilized formulations. These are the multiscale method by Khurram et al. [52] derived in section 4.3, and the SUPG and traditional multiscale method derived in section 4.4. The RANS equations remain untouched and will retain its SUPG, PSPG and LSIC stabilization terms.

In figure 5.17 the computed solution for several coarser meshes is presented from 8 to 256 elements in the direction normal to the flow, and compared to the fine solution using 2048 elements. The solution shown here is obtained using the unstabilized S-A formulation. It can be observed the solution slowly converges to the fine mesh solution as the mesh is refined. The error for the coarser meshes is however quite large. Before looking at the stabilization of the S-A equation we note that the velocity solution on coarser meshes could also still be improved by running a full multiscale model for the flow equations.



Figure 5.17: Comparison of velocity profile on several meshes for turbulent channel flow simulated at $Re_{\tau} = 550$ using the unstabilized S-A formulation.

From the ν_e distribution in figure 5.18 it can be observed the S-A model is quite robust using only the Galerkin terms for the turbulent channel flow problem, as the solution on the coarsest mesh using 8 elements does not show any sign of oscillatory or other improper behavior. The solution is however slightly larger than the fine solution, leading to overly diffusive results in the flow equations. In the same figure it can be observed that for 64 elements very good correspondence is reached already with the fine mesh solution. The reason for the good performance of the unstabilized S-A model will be discussed in the following subsection.



Figure 5.18: Comparison of modified turbulent viscosity profile on several meshes for turbulent channel flow simulated at $Re_{\tau} = 550$ using the unstabilized S-A formulation.

The comparison between the different stabilization methods can be made most clear by giving an overview of the norm of the velocity error for the different formulations on several meshes. The norm of the error can again be calculated using equation 5.2, and the difference is taken with fine mesh solution. The numbers are presented in table 5.1, along with the maximum velocity u_{max} obtained in the channel.

	Galerkin		Multiscale		SUPG		Khurram	
# of elem.	Norm of error	u _{max}	Norm of error	u_{max}	Norm of error	u _{max}	Norm of error	u_{max}
8	7.958	8.57	7.900	8.66	7.960	8.56	7.929	8.54
16	5.110	10.10	5.103	10.12	5.112	10.10	5.191	10.07
32	2.811	12.61	2.964	12.17	2.815	12.60	3.074	11.85
64	0.9117	17.00	1.117	16.18	0.9147	16.99	1.261	15.59
128	0.1401	19.88	0.1399	19.88	0.1397	19.88	0.1647	19.74
256	1.136e-02	20.59	0.9705e-02	20.60	1.130e-02	20.59	4.139e-02	20.36
512	9.255e-04	20.67	2.378e-04	20.68	8.817e-04	20.67	6.974e-03	20.61
2048	-	20.68	-	_	-	_	-	_

Table 5.1: Comparison of coarse mesh solution to fine mesh solution using Galerkin, multiscale, SUPG and Khurram's method. Results indicated for turbulent channel flow simulated at $Re_{\tau} = 550$.

It can immediately be observed that Khurram's method produces bad results compared to the unstabilized S-A formulation. The traditional multiscale method shows to not improve the solution by much and deteriorates for 32 and 64 elements used. As could be observed from figure 5.18, at 64 elements the Galerkin method for the S-A model already shows accurate results for the ν_e distribution and cannot be improved by much. We do note however that as the mesh gets refined the multiscale formulation converges towards the fine solution faster than the unstabilized formulation. Finally, the SUPG formulation shows to not change the solution by much, compared to the unstabilized S-A model.

5.3.2.3 Magnitude of S-A equation terms for the channel flow problem

From the previous subsection it can be noted that the unstabilized S-A equation is quite robust by itself and does not seem to require any form of stabilization. To see why this is it is interesting to look at the individual terms of the equation and their magnitude. It is especially interesting to look at the magnitude of the advection terms, in comparison to the magnitude of the diffusion terms, as the Galerkin method works well for diffusiondominated problems. However, it is well known that the Galerkin method is not ideally suited for advection-dominated problems. See, e.g., [27].

Let us repeat the strong form of the S-A model here for convenience. It is given as,

$$\frac{\partial \nu_e}{\partial t} + \mathbf{u} \cdot \nabla \nu_e - c_{b1} S_e \nu_e + c_{w1} f_w \left(\frac{\nu_e}{d}\right)^2 - \frac{1}{\sigma} \left[\left((\nu + \nu_e) \,\nabla^2 \nu_e \right) + (c_{b2} + 1) \nabla \nu_e \cdot \nabla \nu_e \right] = 0 \quad \text{in } \Omega \times]0, T[.$$
(5.13)

First off, for the converged solution the temporal term $\frac{\partial \nu_e}{\partial t}$ equals zero. Then it can be noted that the advective term $\mathbf{u} \cdot \nabla \nu_e$ is identically zero as well for channel flow. This can be seen as follows, we have that,

$$\mathbf{u} \cdot \nabla \nu_e = u \frac{\partial \nu_e}{\partial x} + v \frac{\partial \nu_e}{\partial y} \,. \tag{5.14}$$

In the direction of flow there is no gradient in ν_e , and the direction of flow is in x-direction. Hence $\frac{\partial \nu_e}{\partial x}$ and v equal zero for channel flow and this term thus equals zero.

The other term which has an advective character, $-\frac{(c_{b2}+1)}{\sigma}\nabla\nu_e \cdot \nabla\nu_e$, does not equal zero as the product of the gradients in *y*-direction produces a nonzero value. The magnitude of this term for the case of 128 elements used is given in figure 5.19a. The advection term becomes larger in magnitude near the wall. From figure 5.18 it can be noted that indeed the gradient $\frac{\partial\nu_e}{\partial y}$, and thus the advection term, becomes larger near the wall.

Figures 5.19b, 5.19c and 5.19d contain the magnitude of the diffusion, production and destruction term, respectively. The magnitude of the diffusion term can be noted to be smaller compared to the other terms in the equation. The reaction terms show to be growing in magnitude towards the wall, as expected, since reaction phenomena play an important role in near-wall behavior of the flow.

Khurram's multiscale method assumes, when applying their form of the multiscale method, for the reaction terms the coefficients S_e and f_w constant. However, the highly nonlinear behavior and large magnitude of the reaction terms near the wall are good arguments to apply the multiscale method to these coefficients as well. Hence this application was chosen for in the derivation of the traditional multiscale method. For the channel flow case we note that the inclusion of the extra terms arising from this linearization of S_e and f_w does not improve results for coarser meshes by much. For finer meshes of 128 elements and up though a slight improvement is observed, since the norm of the error of the velocity distribution converges to the fine solution faster.



Figure 5.19: Magnitude of the S-A equation terms that are nonzero for the turbulent channel flow problem using 128 elements in *y*-direction.

From the above discussion it can be argued that for coarser meshes of 64 elements and below the S-A equation is not in a range appropriate for linearization of the fine scales. Especially in the near-wall region the sharp behavior of the separate terms make it difficult to capture the solution here. Hence an inclusion of the nonlinear fine scale terms could provide an improved ν_e distribution for coarser meshes, especially near the wall. Another form of improvement could be sought in the splitting of **u** in the S-A equation into coarse and fine scales, since currently only the coarse scales are computed. However, for the turbulent channel flow case this will not provide any improved results as, as explained before, the advection term containing the velocity equals zero for this case. The same holds true for the term $\mathbf{u}' \cdot \nabla \nu_e$ in the case of turbulent channel flow. Finally, by specifying off-diagonal terms in the stabilization parameter matrix improvements could be found as well. By doing this the residual of each equation can be accounted for in the small scale unknown of each equation. This makes sense since we are dealing with a coupled system of equations.

The difficulty of exactly capturing the near wall behavior of ν_e is portrayed in figure

5.20. Here the fine mesh solution for ν_e is subtracted from the solution on a 128 element mesh using the unstabilized S-A formulation. This results in an error distribution for ν_e . It can be observed that for the 128 element mesh the error near the wall is relatively up to 10 times as large as mid-channel.



Figure 5.20: Error distribution of ν_e on a mesh of 128 elements using the unstabilized S-A formulation, compared to the fine mesh solution.

To conclude the turbulent channel flow case a plot is presented in figure 5.21 of the element Péclet number magnitude α throughout the interior of the channel. The element Péclet number can again be calculated using equation (5.9). Here $||\mathbf{a}||$ can be calculated using the active advection term for the channel flow problem $-\frac{(c_{b2}+1)}{\sigma}\nabla\nu_e$. The diffusion coefficient κ instead is given by $\frac{\nu+\nu_e}{\sigma}$. As noted before, the Galekin method is well known to work well for diffusion dominated cases where $\alpha < 1$. Indeed, it can be noted from the figure that in the interior of the channel α is smaller than 1 for the coarse as well as finer meshes. Hence the difficulty of improving results for the channel flow case can be argued to stem from the diffusion dominating the advection in the S-A equation.



Figure 5.21: Péclet number distribution throughout the interior of the channel for 8, 64 and 2048 elements used in *y*-direction.

5.3.3 Backward facing step

The backward facing step problem provides more challenging test case as it deals with interesting flow phenomena such as separation and reattachment near the step. Many experiments have been performed for this case with different measurements of the step size and channel height. Here we will simulate the backward facing step problem for which an experiment has been performed by Driver and Seegmiller [28]. In the section below the test case setup is first discussed, including specifics on how the test case was run and the mesh used. Then in section 5.3.3.2 the fine mesh solution is compared to experimental results to validate the test case. Finally, in section 5.3.3.3 the fine mesh solution is compared to the solution on a coarse mesh for the unstabilized, SUPG and two multiscale stabilized S-A formulations.

5.3.3.1 Setup of the backward facing step test case

The layout of the test case including the boundary conditions are shown in figure 5.22. The step height H equals 0.0127m. The experiment is performed at atmospheric conditions with the temperature at the reference point (midway the channel at x = -4H) equal to $T_{ref} = 300K$. Hence the kinematic viscosity of the fluid used for the simulation will be $\nu = 1.57 \cdot 10^{-5} m^2/s$, while the fluid density is taken to be $\rho = 1.18 kg/m^3$. To obtain roughly the correct velocity of $u_{ref} = 44.2m/s$ at the reference point, a uniform velocity profile of $u_{in} = 41.5 m/s$ is imposed at the inlet. After a short region of symmetry the wall starts at x = -110H. The length of the channel prior to the step has been adjusted to ensure the correct thickness of the boundary layer at the step (approximately equal to the experiment). The inflow and symmetry boundary condition on the modified turbulent viscosity ν_e ensures the flow conditions to be fully turbulent. On the wall the modified turbulent viscosity equals zero. For the velocity we impose a no-slip condition on the wall. Finally, we set the pressure equal to zero in the outflow region. Note that the Reynolds number based on the step height Re_H and reference velocity roughly equals 36,000 for the above stated conditions. The Mach number using the reference velocity equals $M_{ref} = 0.128$. Hence the simulation is at essentially incompressible conditions, though compressibility may have a very small influence on the results.

The coarsest mesh used for this case is taken from the excellent NASA Langley Research Center website [15], which contains a wealth of knowledge and information on RANS based turbulence models. The mesh is converted from plot3d to Exodus II format using ParaView software [1], such that it is readable by libMesh. As the mesh is nondimensionalized by *H* the mesh is scaled by the step height within libMesh (while retaining connectivity) in order to perform the simulation at the correct Reynolds number. The coarse mesh is visualized in figure 5.23. It is a two-dimensional structured mesh comprised of four blocks. As the four blocks have overlapping nodes on their adjacent boundaries, the mesh blocks need to be stitched together in order for the mesh to be usable. This is also achieved within libMesh in such a way that the overlapping nodes are removed and replaced by single nodes. After removal of the excess nodes this coarse grid consists of 5, 157 nodes. A close-up of the mesh near the region of interest is provided in figure 5.24. It can be observed that the mesh becomes slightly skewed after the back step, such that a more optimal placement of the nodes is achieved.

A refinement study is performed in the subsections below in order to assess how well the Galerkin method and the other derived formulations for the S-A model perform on a



Figure 5.22: Backward facing step geometry and boundary conditions with H = 0.0127m.



Figure 5.23: Coarsest mesh used for the backward facing step problem.

coarse mesh compared to a refined mesh. The idea is to solve the problem on the coarsest mesh first. The problem can then be solved on a refined mesh by loading the coarse solution onto the mesh and then uniformly refining the mesh within libMesh. This will interpolate the coarse mesh solution and project it onto the refined mesh. Therefore the refined mesh is started from an initial solution that should be close to the final solution, requiring less pseudo time steps for the problem to converge. This process is performed three times such that the problem is solved on four different meshes. Note that by performing one uniform refinement the number of nodes increases by a factor of about



Figure 5.24: Close-up of the coarsest backward facing step mesh at the region of interest (x = -4H to x = 8H).

4 for the two-dimensional case. Hence the finest mesh we perform our calculations on consists of roughly $3.2 \cdot 10^5$ nodes.

To solve the initial problem on the coarsest mesh we use a continuation method in which the Reynolds number of the flow is gradually increased. This will ensure the initial guess for the Newton iteration is not too far off, such that the iteration procedure does not diverge. We again employ backward Euler ($\theta = 1$) and use linear elements. After some experimenting it is observed that a time step of $\Delta t = 1.0 \cdot 10^{-4}$ is necessary in order for the S-A model Newton iteration not to diverge.

The calculation of the wall distance for this case is a bit more involved than the simple channel flow case. An illustration of the different regions to be distinguished is given in figure 5.25. With regard to the figure, we note that regions 1, 2, 7 and 8 are indeed as simple as the channel flow case. For regions 3 and 4 the closest wall is where the wall starts, at x = -110H. Region 5 is closest to the vertical wall of the step, while region 6 is closest to the edge of the step. After recognizing this a simple algorithm can be written in order to ensure the proper calculation of the wall distance. In a similar way an algorithm is written to ensure proper calculation of the characteristic mesh size h, with regard to figure 4.3. As the mesh is now slightly skewed after the step this is a bit more involved compared to a non-skew mesh.

For the initial coarse mesh case using the GMRES linear system solver with ILU



Figure 5.25: Different regions to be distinguished for calculating the wall distance d (not drawn to scale).

preconditioner is still viable. However, PETSc does not support the ILU preconditioner in parallel, hence we employ the additive Schwarz method (ASM) preconditioner when running in parallel. For the more fine cases we make the switch to the LU direct solver, which takes some extra time but converges within a single iteration. As PETSc does not have parallel support for LU either we employ the MUMPS library. In this way all cases can smoothly be run in parallel to speed up the computations.

5.3.3.2 Validation of the backward facing step test case

The validation of the backward facing step case is important to show how well the S-A equation can capture fundamental flow phenomena. For the backward facing step case these are flow separation and reattachment near the step. Even though it is not the goal of this thesis to show how well the S-A model performs compared to experiments, the validation of the backward facing step case does demonstrate the engineering interest for these kind of flows.



Figure 5.26: Streamlines and magnitude of velocity contours throughout the domain for the fine mesh case.

Before showing the comparison of the numerical results to the experiment we first show a plot of the streamlines and velocity magnitude throughout the channel in figure 5.26. The result shown here is from the computation on the fine mesh. The imposed uniform inflow velocity at $\frac{x}{H} = -130$ can be observed. On close examination the boundary layer can be seen to grow from $\frac{x}{H} = -110$ where the wall starts, up to the step. After the step the flow separates and a region of recirculating flow emerges. The flow separates and reattaches in the region where the mesh is the finest, and can be seen to smoothly exit the domain at $\frac{x}{H} = 50$. In figure 5.27 the fine mesh velocity profile results are plotted against experimental data at several locations throughout the channel. The experimental data is taken from the experiment performed by Driver and Seegmiller [28]. From the velocity profile upstream of the step it can be observed that it comes across with the experimental data, though the inlet velocity could be set roughly 1% higher such that u_{ref} comes across better with its actually value. Similarly in figures 5.27c and 5.27d good correspondence with the experimental results can be observed. At $\frac{x}{H} = 1$ in figure 5.27b quite some modeling error can be noted in the section below the step, as the flow velocity in negative y-direction in the recirculating region is predicted too high.



Figure 5.27: *u*-velocity profile normalized by u_{ref} at several locations for fine mesh computation and experiment.

For the following graphs we define the skin friction coefficient as,

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho u_{ref}^2} \tag{5.15}$$

with τ_w the local wall shear stress defined in equation (A.1), $\rho = \rho_{ref} = 1.18 kg/m^3$ and $u_{ref} = 44.2m/s$. The calculation of the velocity derivative $\frac{\partial u}{\partial y}\Big|_{y=0}$ for the wall shear stress is performed with respect to the first nodal value adjacent to the wall.

The pressure coefficient is defined as,

$$C_{p} = \frac{p - p_{ref}}{\frac{1}{2}\rho_{ref}u_{ref}^{2}}$$
(5.16)

with p_{ref} the value of the pressure at the reference point in the domain. The pressure coefficient data found has been uniformly shifted such that C_p is zero at $\frac{x}{H} = 37.5$ for both the experiment and the computed values, as performed by Eça et al. [32, 30, 33].

Plots of the lower wall skin friction coefficient and pressure coefficient are given in figure 5.28. Again it can be noted that the error is largest in the region where the flow is separated. The prediction of the flow reattachment is computed close to its experimental value. For the experiment the location with zero skin friction is found to be $\frac{x}{H} = 6.26 \pm 0.10$, while the numerically computed value of flow reattachment is at $\frac{x}{H} = 6.05$.

Finally, as a verification exercise, it is observed that these results correspond very well to those found by three independent CFD codes when running the S-A model for this case on an equally fine mesh. The results when using NASA's CFL3D, FUN3D and WIND code are also presented on the NASA Langley Research Center website [15].



Figure 5.28: Lower wall skin friction (left) and pressure coefficient (right), computed fine mesh solution versus experiment.

5.3.3.3 Refinement study for the backward facing step test case

In this section the coarse mesh results using the Galerkin method as well as several stabilized formulations for the S-A model will be compared to the fine mesh solution. As the backward facing step provides a more challenging test case than the turbulent channel flow problem it is interesting to see if the Galerkin method for the S-A model will still work well on the coarse mesh.

In figure 5.29 the normalized velocity profile at $\frac{x}{H} = -4$ is given. This location is upstream of the step and hence in a region of turbulent channel flow. We already saw for the channel flow problem that the Galerkin method performs slightly better at mediumcoarse meshes compared to the traditional multiscale method, indicated by MS in the figure. Since the results are very close some portions of the figure are magnified in order to give a clearer picture of the differences. It can be observed the traditional multiscale method is slightly over diffusive compared to the Galerkin method, but the differences are small. The SUPG method is close to equal to the Galerkin method, while Khurram's method shows bad results due to an under-prediction of the modified turbulent viscosity. For Khurram's method this leads to under diffusive behavior of the velocity solution.

The velocity profile downstream of the step at $\frac{x}{H} = 1$ is given in figure 5.30. In the region of recirculating flow close to the wall the traditional multiscale method shows improved results compared to the Galerkin method, the same holds for the SUPG method.

Further away from the wall the slightly over diffusive behavior of the multiscale method is present again.



Figure 5.29: *u*-velocity profile normalized by u_{ref} upstream of the step at x/H = -4 on fine and coarse mesh for several S-A formulations.



Figure 5.30: *u*-velocity profile normalized by u_{ref} downstream of the step at x/H = 1 on fine and coarse mesh for several S-A formulations.

Figures 5.31 and 5.32 at locations $\frac{x}{H} = 4$ and $\frac{x}{H} = 6$, respectively, show similar results as the previous figures. The results in the recirculating flow region are very close, while in the region further away from the wall the multiscale method slightly over predicts the flow velocity. The method by Khurram et al. [52] performs bad overall. From these figures it can already be noted that the Galerkin method performs well for the S-A model and is also robust on the coarse mesh for this more challenging test case.



Figure 5.31: *u*-velocity profile normalized by u_{ref} downstream of the step at x/H = 4 on fine and coarse mesh for several S-A formulations.



Figure 5.32: *u*-velocity profile normalized by u_{ref} downstream of the step at x/H = 6 on fine and coarse mesh for several S-A formulations.

The coarse to fine mesh comparison for the lower wall skin friction coefficient is given in figures 5.33. In the channel flow region upstream of the step an increased skin friction coefficient for the traditional multiscale method is observed. Downstream of the step the multiscale method shows a slightly improved skin friction coefficient compared to the Galerkin and SUPG method. The reattachment point of the flow is computed to be $\frac{x}{H} = 6.08$ for the Galerkin, multiscale and SUPG method, compared to $\frac{x}{H} = 6.05$ for the fine mesh solution. Using Khurram's method the reattachment point is predicted at $\frac{x}{H} = 7.31$.

Figure 5.34 presents the lower wall pressure coefficient. The coarse mesh Galerkin,

multiscale and SUPG method show very similar results. It can be observed none of the stabilized formulations can provide an improved solution closer to the fine mesh solution.



Figure 5.33: Lower wall skin friction coefficient on fine and coarse mesh for several S-A formulations.



Figure 5.34: Lower wall pressure coefficient on fine and coarse mesh for several S-A formulations.

The modified turbulent viscosity profiles normalized by the kinematic viscosity are presented in figures 5.35 and 5.36. ν_e can be seen to grow along the boundary of the channel upstream of the step. After the step a peak is observed in the region with separated flow. From figure 5.36d it can clearly be observed Khurram's method under predicts the magnitude of ν_e , resulting in the under diffusive behavior that was observed in the figures above. The Galerkin, multiscale and SUPG method all show ν_e distributions fairly equal to those obtained on the fine mesh.



Figure 5.35: Modified turbulent viscosity field normalized by the kinematic viscosity for the fine grid backward facing step case.



Figure 5.36: Modified turbulent viscosity field normalized by the kinematic viscosity for the coarse grid backward facing step case using several S-A formulations.

On closer inspection of figure 5.36 it can be observed the ν_e distribution for the Galerkin and SUPG method is not as smooth as for the multiscale and Khurram's method. In fact, the solution is oscillatory at certain regions when using the Galerkin and SUPG method for the S-A model. To show this, a line plot is presented in figure 5.37 of the nodal values of ν_e normalized by ν midway the channel upstream of the step. The highly oscillatory behavior of the solution for the Galerkin and SUPG method can be seen here. It is noted that only after three uniform refinements of the mesh the oscillatory behavior of the solution is resolved using the Galerkin method for the S-A model. Both the traditional multiscale and Khurram's method that have a form of stabilization for the reaction terms show a smooth solution behavior for ν_e . The traditional multiscale method follows the lower part of the oscillatory Galerkin solution quite precisely.



Figure 5.37: Line plot of the modified turbulent vicosity normalized by the kinematic viscosity midway the channel (y/H = 5) upstream of the step. The results are given on fine and coarse mesh for several S-A formulations.

The reason for the oscillatory behavior in the solution for ν_e using the Galerkin and SUPG method can be argued to stem from the lack of diffusion in the region where no gradient and second gradient of ν_e is present. The magnitude of the diffusion term in the S-A equation upstream of the step if plotted in figure 5.38 at several locations. The diffusion term can be seen to equal zero at the parts where the solution for ν_e is oscillatory for the Galerkin and SUPG method on the coarse mesh. Due to a lack of diffusion we are effectively solving the advection-reaction equation. The advection in the middle of the channel upstream of the step is small but nonzero since there is a small linear gradient for ν_e in the flow direction for the fine mesh case. The advection-reaction equation is known to suffer from instabilities due to a lack of diffusion for large enough Damköhler numbers, as shown by Hauke et al. [40]. Since the advection term is near zero the Damköhler number can get very large for this case, as well as the turbulent channel flow case.



Figure 5.38: Line plot of the magnitude of the diffusion term for the backward facing step case at several locations upstream of the step for the fine mesh solution (values have been interpolated).

From the above it can be concluded that the Galerkin and SUPG method do not have a robust enough S-A model formulation to provide sufficient stabilization in the absence of diffusion. The traditional multiscale and Khurram's method both provide formulations that are sufficiently robust to deal with a lack of diffusion in the domain on coarser meshes. The reaction coefficient in the weighting function of the stabilizing term is thus required to produce a more robust stabilized formulation. Only the traditional multiscale formulation for the S-A model provides reasonably accurate results compared to the Galerkin and SUPG formulations. In the presence of diffusion the latter two methods are robust and produce good results, also on coarser meshes.
Chapter 6

Conclusions & Recommendations

The present study was intended to find a more robust finite element formulation for the S-A turbulence model. We have presented two seperate multiscale formulations, for which the scalar field is decomposed into coarse and fine scales, and one SUPG type formulation, using only the advective stabilizing weighting term of the traditional multiscale formulation. One multiscale formulation is based on the work of Khurram et al. [52], in which the multiscale approach is applied to the A-D-R equation instead. The other is based on a more traditional variational multiscale approach applied directly to the Spalart-Allmaras equation. The two approaches are compared and found partially equal in that they both neglect fine scale products. A few distinct differences are however found in the weighting function slot, as well as the solution slot of the multiscale stabilizing term. Next to this, Khurram's method ignores the highly nonlinear reaction term coefficients and assumes them constant, while in the traditional multiscale the nonlinearity is taken into account.

After studying a turbulent channel flow problem we find the unstabilized S-A formulation performs very well on coarse, medium, and fine meshes. A more careful examination reveals that the element Péclet numbers for the channel flow problem are well below 1 due to the definition of the advection coefficient of the S-A model. It is well known that the Galerkin method is stable for element Péclet number below 1. It can therefore be argued no form of advection stabilization is required for this sort of problem. It is shown the largest error in the solution can be found near the wall, resulting from the highly nonlinear behavior of the reaction terms in this region.

When using the SUPG type stabilization, we thus note no improvement of the result. The results differ very little from using only the Galerkin terms since the advection term of the S-A model is found to be rather small for channel flow. Both multiscale methods for the S-A model are observed to have deteriorated results for coarse meshes using 32 to 64 elements normal to the direction of flow. When using medium fine meshes of 128 elements and above, both multiscale methods can be seen to improve again. While the Khurram method does not improve compared to Galerkin, we observe a slightly improved result using the traditional multiscale method for the fine meshes. Reasons for these results are argued to be:

• The definition of the stabilization parameter used for the S-A turbulence model was derived for the linear advection-diffusion-reaction equation. Hence when the mesh

is too coarse we may not be in a range appropriate for linearization, since the S-A equation is highly nonlinear.

- Higher-order fine scale terms have been neglected. For coarse meshes these are more important as the coarse scale residual is much larger compared to the coarse scale residual for fine meshes. This can be seen from the fact that as u' becomes small, using 128 elements and above, the traditional multiscale method starts improving compared to the unstabilized S-A formulation.
- Another simplification stems from the assumption of a diagonal τ matrix for the coupled system of equations. By specifying off-diagonal terms, the residual of all equations are taken into account for each of the fine scale unknowns.

A backward facing step problem was also studied and again we find using only the Galerkin terms on a coarse mesh for the S-A equation results in very good results compared to the fine mesh reference solution. Equally good results are found using the traditional multiscale method, while the results using Khurram's multiscale method for the S-A equation are found to be much worse. Hence it is argued that the method by Khurram et al. [52] is neglecting stabilization terms compared to our derived formulation that should not be neglected. Using the SUPG stabilized S-A model formulation again shows very similar behavior compared to the Galerkin method.

Though the general results found from the unstabilized and SUPG formulation are good – such as the determination of the reattachment point of the flow downstream of the step – oscillatory behavior is observed for the modified turbulent viscosity field. These oscillations are mainly present in the coarsest part of the mesh, such as midchannel upstream and far downstream of the step. It is noted that only after 3 uniform mesh refinements of the coarsest mesh the oscillations are resolved using the unstabilized formulation. The oscillatory behavior is argued to stem from the lack of diffusion in this region of the flow, resulting in the advection-reaction equation effectively being solved. Due to a very small amount of advection in the region mid-channel as well, the Damköhler number in this region is high. It is known that for these kind of flows stability issues ensue when using the Galerkin and SUPG method, as shown by Hauke et al. [40]. For the backward facing step case though, these stability issues do not notably lead to deteriorated results in the pressure and velocity field.

Both multiscale formulations obtain a smooth solution for ν_e on the coarsest mesh, showing that these formulations possess additional robustness compared to the Galerkin formulation. Therefore, the added reaction stabilization term notably contributes to the additional robustness. Only the traditional multiscale method formulation for the S-A model provides reasonably accurate results compared to the Galerkin and SUPG formulations. Additional accuracy may be obtained, as noted above for the channel flow case, by inclusion of the higher order fine scale terms. Inclusion of \mathbf{u}' in the S-A equation should also improve results for this case in the regions where the advection term $\mathbf{u} \cdot \nabla \nu_e$ is nonzero.

In conclusion, the unstabilized S-A formulation is quite robust and accurate by itself for the turbulent channel flow and backward facing step case. The SUPG method does not provide any additional robustness and has equal accuracy. Using both multiscale formulations for the S-A model provides additional robustness and stability in regions lacking any form of diffusion, of which only the traditional multiscale method mainly preserves accuracy.

Finally, we note some improvements that can be done for future projects in this line of work and for the developed code in general. These are listed as follows:

- Inclusion of higher order fine scale terms to improve coarse mesh accuracy.
- Inclusion of the fine scale velocity term in the S-A equation.
- Use of a nondiagonal τ to include the residual of each equation for each unknown for the coupled system of equations.
- Perform tests for even more challenging test cases to find the limits of the robustness of the unstabilized S-A formulation.
- Inclusion of additional stabilizing terms for the flow equations in order to run a true multiscale model.
- Adapting the convective term in the flow equations to its divergence form.
- Inclusion of a third spatial dimension for a more general code to also be used for other applications than RANS.

References

- James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large data visualization, 2005.
- [2] S.R. Allmaras, F.T. Johnson, and P.R. Spalart. Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. Seventh International Conference on Computational Fluid Dynamics (ICCFD7), 2012.
- [3] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [4] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.
- [5] Kerstin Avila, David Moxey, Alberto de Lozar, Marc Avila, Dwight Barkley, and Björn Hof. The onset of turbulence in pipe flow. *Science*, 333(6039):192–196, 2011.
- [6] Ivo Babuška. The finite element method with penalty. Mathematics of computation, 27(122):221–228, 1973.
- [7] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, and Hong Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2015.
- [8] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1):173–201, 2007.
- [9] Eric Becker, Graham Carey, and J. Oden. Finite Elements. An Introduction. PH, 1981.

- [10] D. Borello, A. Corsini, and F. Rispoli. A finite element overlapping scheme for turbomachinery flows on parallel platforms. *Comput. & Fluids*, 32:1017–1047, 2003.
- [11] F. Brezzi, L.P. Franca, T.J.R. Hughes, and A. Russo. $b = \int g$. Comput. Methods Appl. Mech. Engrg., 17:1–19, 1997.
- [12] Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32(1):199–259, 1982.
- [13] A. Celić and E. Hirschel. Comparison of eddy-viscosity turbulence models in flows with adverse pressure gradient. AIAA Journal, 44(10):2156–2169, 2006.
- [14] Alan Celic and Ernst H. Hirschel. Comparison of eddy-viscosity turbulence models in flows with adverse pressure gradient. AIAA journal, 44(10):2156–2169, 2006.
- [15] NASA Langley Research Center. Turbulence modeling resource. http:// turbmodels.larc.nasa.gov/, 2016. Accessed: 14-06-2016.
- [16] Ward Cheney. Analysis for applied mathematics, volume 208. Springer Science & Business Media, 2013.
- [17] R. Codina. Comparison of some finite element methods for solving the diffusionconvection-reaction equation. Comp. Meth. Appl. Mech. Engrg., 156:185–210, 1998.
- [18] R. Codina. On stabilized finite element methods for linear systems of convectiondiffusion-reaction equations. Comp. Meth. Appl. Mech. Engrg., 188:61–88, 2000.
- [19] R. Codina. Stabilized finite element method for generalized stationary incompressible flows. Comp. Meth. Appl. Mech. Engrg., 190:2681–2706, 2001.
- [20] Ramon Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Computer Methods in Applied Mechanics and Engineering, 191(39):4295–4321, 2002.
- [21] D.E. Coles. The law of the wake in the turbulent boundary layer. J. Fluid Mech. 1, pages 191–226, 1956.
- [22] A. Corsini, F. Menichini, F. Rispoli, A. Santoriello, and T.E. Tezduyar. A Multiscale Finite Element Formulation With Discontinuity Capturing for Turbulence Models With Dominant Reactionlike Terms. *Journal of Applied Mechanics*, 76:021211, 2009.
- [23] A. Corsini, F. Rispoli, and A. Santoriello. A variational multiscale higher-order finite element formulation for turbomachinery flow computations. *Comput. Methods Appl. Mech. Engrg.*, 194:4797–4823, 2005.
- [24] A. Corsini, F. Rispoli, A. Santoriello, and T.E. Tezduyar. Improved discontinuitycapturing finite element techniques for reaction effects in turbulence computation. *Comput. Mech.*, 38:356–364, 2006.
- [25] T.J. Craft, B.E. Launder, and K. Suga. Development and Application of a Cubic Eddy-Viscosity Model of Turbulence. Int. J. Heat Fluid Flow, 17:108–155, 1996.

- [26] Juan C. del Álamo and Javier Jiménez. Spectra of the very large anisotropic scales in turbulent channels. *Physics of Fluids*, 15(6):L41, 2003.
- [27] Jean Donea and Antonio Huerta. Finite element methods for flow problems. John Wiley & Sons, 2003.
- [28] David M. Driver and H. Lee Seegmiller. Features of a reattaching turbulent shear layer in divergent channelflow. AIAA journal, 23(2):163–171, 1985.
- [29] P.A. Durbin. Separated Flow computations with the $k \epsilon v^2$ Model. AIAA J., 33:659–664, 1995.
- [30] L Eça, M Hoekstra, and Patrick J Roache. Verification of calculations: an overview of the 2nd lisbon workshop. In Second Workshop on CFD Uncertainty Analysis, AIAA Computational Fluid Dynamics Conference, 2007.
- [31] Luis Eça, M. Hoekstra, A. Hay, and D. Pelletier. A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow. *International Journal of Computational Fluid Dynamics*, 21(3-4):175–188, 2007.
- [32] Luís Eça, Martin Hoekstra, and Patrick J Roache. Verification of calculations: an overview of the lisbon workshop. In AIAA Computational Fluid Dynamics Conference, 2005.
- [33] Luis Eça, Martin Hoekstra, Patrick J Roache, and Hugh Coleman. Code verification, solution verification and validation: an overview of the 3rd lisbon workshop. In the Proceedings of 19th AIAA Computational Fluid Dynamics Conference, San Antonio, Texas, USA, 2009.
- [34] L. Eça, M. Hoekstra, and G. Vaz. Manufactured solutions for steady-flow Reynoldsaveraged Navier-Stokes solvers. *International Journal of Computational Fluid Dynamics*, 26(5):313–332, 2012.
- [35] Leopoldo P. Franca and A. Nesliturk. On a two-level finite element method for the incompressible Navier–Stokes equations. International Journal for Numerical Methods in Engineering, 52(4):433–453, 2001.
- [36] L.P. Franca and C. Farhat. Bubble functions prompt unusual stabilized finite element methods. Comput. Methods Appl. Mech. Engrg., 123:299–308, 1995.
- [37] L.P. Franca and F. Valentin. On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation. *Comput. Methods Appl. Mech. Engrg.*, 190:1785–1800, 2000.
- [38] G. Hauke. A simple subgrid scale stabilized method for the advection-diffusion-reaction equation. *Comput. Methods Appl. Mech. Engrg.*, 191:2925–2947, 2002.
- [39] G. Hauke and M.H. Doweidar. Exploring new subgrid scale stabilized methods for advection-diffusion-reaction. In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, *ECCOMAS*, 2004. Jyväkyä.

- [40] G. Hauke and A. García-Olivares. Variational subgrid scale formulations for the advection-diffusion-reaction equation. *Comput. Methods Appl. Mech. Engrg.*, 190:6847–6865, 2001.
- [41] G. Hauke, G. Sangalli, and M.H. Doweidar. Combining adjoint stabilized methods for the advection-diffusion-reaction problem. *Math. Models Methods Appl. Sci.*, 17:305– 326, 2007.
- [42] M.-C. Hsu, Yuri Bazilevs, Victor M. Calo, Tayfun E. Tezduyar, and Thomas J.R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199(13):828–840, 2010.
- [43] Thomas J.R. Hughes. The finite element method: linear static and dynamic finite element analysis. Courier Corporation, 2012.
- [44] Thomas J.R. Hughes, Leopoldo P. Franca, and Marc Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the babuška-brezzi condition: A stable petrov-galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1):85–99, 1986.
- [45] Thomas J.R. Hughes and Michel Mallet. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. Computer Methods in Applied Mechanics and Engineering, 58(3):305–328, 1986.
- [46] Thomas J.R. Hughes, Guglielmo Scovazzi, and Leopoldo P. Franca. Multiscale and stabilized methods. Wiley Online Library, 2004.
- [47] T.J.R. Hughes. Multiscale phenomena: Greens functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origin of stabilized methods. *Comput. Methods Appl. Mech. Engrg.*, 127:387–401, 1995.
- [48] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J. Quincy. The variational multiscale method-a paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166:3–24, 1998.
- [49] S.J. Hulshoff. CFD II Part 2: Computation and Modelling of Turbulence. Faculty of Aerospace Engineering, Delft University of Technology, 2012.
- [50] S.J. Hulshoff, I. Akkerman, and D. Ruflin. Stabilized discretizations for RANS using the Spalart-Allmaras and Menter SST models. Presentation. Faculty of Aerospace Engineering, Delft University of Technology.
- [51] R.A. Khurram and W.G. Habashi. Multiscale/stabilized finite element method for Spalart-Allmaras turbulence model, in: W.A. Wall, V. Gravemeier (Eds.), International Conference on Finite Elements in Flow Problems (FEF 2011). In *Book of Abstracts*, page 120, March 23–25 2011.

- [52] R.A. Khurram, Y. Zhang, and W.G. Habashi. Multiscale finite element method applied to the Spalart-Allmaras turbulence model for 3D detached-eddy simulation. *Comput. Methods Appl. Mech. Engrg.*, 233-236:180–193, 2012.
- [53] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engi*neering with Computers, 22(3-4):237-254, 2006. http://dx.doi.org/10.1007/ s00366-006-0049-3.
- [54] R.G. Lerner and G.L. Trigg. *Encyclopaedia of Physics*. VHC publishers, 2nd edition, 1991. ISBN (Verlagsgesellschaft) 3-527-26954-1, ISBN (VHC Inc.) 0-89573-752-3.
- [55] A. Masud and R.A. Khurram. A multiscale/stabilized finite element method for the advection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, 193:1997–2018, 2004.
- [56] A. Masud and R.A. Khurram. A multiscale finite element method for the incompressible Navier–Stokes equations. Computer Methods in Applied Mechanics and Engineering, 195(13):1750–1777, 2006.
- [57] F.R. Menter. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. AIAA, 32:1598–1605, 1994.
- [58] P.J. Roache. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, 124(1):4–10, 2002.
- [59] C.J. Roy. Review of code and solution verification procedures for computational simulation. Journal of Computational Physics, 205(1):131–156, 2002.
- [60] Ir. A. Segal. Finite element methods for the incompressible Navier-Stokes equations. Delft University of Technology, 2012.
- [61] P.R. Spalart and S.R. Allmaras. A One-Equation Turbulence Model for Aerodynamic Flows. La Rech. Aérospatiale 1, pages 5–21, 1994.
- [62] Roger Temam. Navier-Stokes equations: theory and numerical analysis, volume 343. American Mathematical Soc., 2001.
- [63] T.E. Tezduyar and Y.J. Park. Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 59:307–325, 1986.
- [64] T. von Kármán. Mechanische ähnlichkeit und Turbulenz. Nachr. Ges. Wiss. Göttingen, Math. Nat. Kl., pages 58–76, 1930.
- [65] D.C. Wilcox. Re-assessment of the scale-determining equation for advanced turbulence models. AIAA Journal, 26(11):1299–1310, 1988.
- [66] Y. Zhang, R.A. Khurram, and W.G. Habashi. Multiscale finite element method applied to detached-eddy simulation for computational wind engineering. *Wind Struct.*, 145:329–339, 2013.

Appendix A

Law of the wall

It has been shown experimentally that the flow near a wall behaves differently from the flow farther away from the flow. A turbulent boundary layer is typically divided into two parts: the inner layer and the outer layer. The outer layer makes up about 80 - 90% of the boundary layer and consists of regions of intermittent laminar and turbulent flow. Its edge is strongly rippled and mixes with the external laminar flow. The inner layer, closest to the wall, makes up the remaining 10 - 20% of the boundary layer. This layer can be further subdivided into the viscous sublayer and the buffer layer. An intermediate layer can also be distinguished, making up for the part in between the inner and outer layer. This subdivision is shown in figure A.1, where the velocity u is plotted against y^+ . The definition of y^+ is given in the next paragraph.

In order to compare different wall-bounded flows to each other, a self-similar solution can be derived for the region close to the wall. This solution is expressed using nondimensional quantities for direct comparison. We define the friction velocity u_{τ} and viscous length l_{ν} scale as,

$$u_{\tau} = \sqrt{\frac{\tau_w}{\rho}} \qquad \qquad l_{\nu} = \sqrt{\frac{\nu}{\frac{\partial \bar{u}}{\partial y}\Big|_{y=0}}} \qquad (A.1)$$

where τ_w is the wall shear stress, and is defined as,

$$\tau_w = \nu \rho \left. \frac{\partial \bar{u}}{\partial y} \right|_{y=0} \,. \tag{A.2}$$

The non-dimensionalized quantities are referred to as wall units and can be written in terms of the above parameters as,

$$y^{+} = \frac{y}{l_{\nu}} = \frac{yu_{\tau}}{\nu}$$
 $u^{+} = \frac{\bar{u}}{u_{\tau}}$ (A.3)

with y the dimensional distance from the wall.

Using suitable assumptions one can derive expressions for u^+ as a function of y^+ by examining the flow in a channel. The first expression is valid for the viscous sublayer region and one can note that velocities are low here. Viscous stresses thus dominate the



Figure A.1: Nomenclature for wall-bounded turbulent flows [49].

flow in this region. It can be derived that u^+ and y^+ are approximately equal in the region below 5 wall units:

$$u^+ = y^+$$
 for $y^+ < 5$. (A.4)

In the intermediate layer instead the flow is dominated by turbulent stresses and the viscous stresses are negligible. Von Kármán was first to publish the so called log law for this region, see [64]. The equation is valid for the region above 30 wall units and stretches throughout the intermediate layer. It is given as,

$$u^{+} = \frac{1}{\kappa} \ln y^{+} + \beta \tag{A.5}$$

with κ and β experimentally obtained constants. κ is termed the von Kármán constant and its value is about 0.41, β can range from 5 to 5.5. Note that in the buffer layer, $5 < y^+ < 30$, neither law holds.

In the outer layer one can obtain a velocity defect law by dimensional analysis and for a zero pressure gradient:

$$\frac{u_e - \bar{u}}{u_\tau} = f\left(\frac{y}{\delta}\right) \tag{A.6}$$

where u_e is the external flow velocity and δ is the boundary layer thickness. A popular defect law was introduced by Coles and is given in [21].

Appendix B

Penalty method

Application of the Dirichlet boundary conditions is done through use of the penalty method. This method was put in a finite element context by Babuška [6]. See also Becker et al. (pp 121-123) [9] for the treatment of implementation.

Using the penalty method for the N-S equations the Dirichlet boundary condition given in equation (2.8) is replaced by a penalized Robin-type boundary condition. This adjusted boundary conditions looks as follows,

$$\mathbf{t}'(\mathbf{x}) + \beta \mathbf{u}(\mathbf{x}) = \beta \mathbf{u}_D(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_R$$
 (B.1)

where the boundary condition is assumed steady, Γ_R refers to the penalized Robin portion of the boundary, and β acts as a penalty constant. It can be seen that as $\beta \to \infty$ the boundary condition reduces to the Dirichlet boundary condition.

Taking a look at the boundary term in equation (3.4), we split the boundary into the Neumann portion Γ_N , and the penalized Robin portion Γ_R . Since the velocity on Γ_R is now deemed unknown, because of the change made in equation (B.1), the test function **w** is nonzero on this part of the boundary as well. The boundary term splits as follows,

$$(\mathbf{w}, (-p\mathbf{n} + \nu (\mathbf{n} \cdot \nabla) \mathbf{u}))_{\Gamma} = (\mathbf{w}, \mathbf{t}')_{\Gamma_{N}} + (\mathbf{w}, \mathbf{t}')_{\Gamma_{R}}$$

= $(\mathbf{w}, \mathbf{t}')_{\Gamma_{N}} + (\mathbf{w}, \beta \mathbf{u}_{D} - \beta \mathbf{u})_{\Gamma_{R}}$ (B.2)

where equation (B.1) has been used. Therefore the momentum equation (3.9) changes as follows using the penalty method,

$$\begin{pmatrix} \mathbf{w}, \frac{\partial \mathbf{u}}{\partial t} \end{pmatrix} + c \left(\mathbf{u}; \mathbf{w}, \mathbf{u} \right) + a \left(\mathbf{w}, \mathbf{u} \right) - b \left(\mathbf{w}, p \right) + \left(\mathbf{w}, \beta \mathbf{u} \right)_{\Gamma_R}$$

$$= \left(\mathbf{w}, \mathbf{f} \right) + \left(\mathbf{w}, \mathbf{t'} \right)_{\Gamma_N} + \left(\mathbf{w}, \beta \mathbf{u}_D \right)_{\Gamma_R} .$$
(B.3)

Note that in the final discretized system of equations the penalty terms are not multiplied by the time step in order to not make the penalty dependent on the time step size. Since the penalty method is simply a mathematical trick this does not invalidate the method.

Because all variables are assumed unknown on the boundary, the Dirichlet boundary nodes also makes up part of the left-hand-side vector of unknowns. Let us consider as an example the matrix-vector system $\mathbf{K}\boldsymbol{u} = \mathbf{f}$, with \mathbf{K} the Jacobian matrix, \boldsymbol{u} the vector

of nodal unknowns and ${\bf f}$ the right-hand-side vector. Using the penalty method we can write,

$$\mathbf{K}\boldsymbol{u} + \beta \tilde{\mathbf{K}}_c \boldsymbol{u} = \mathbf{f} + \beta \tilde{\mathbf{K}}_c \boldsymbol{u}_D \tag{B.4}$$

where \mathbf{K}_c is a diagonal constraint matrix which has unit entries only for the Dirichlet boundary nodes. Looking at the equation for Dirichlet boundary node p we get,

$$\sum_{j=1}^{n_{np}} K_{pj} u_j + \beta u_p = f_p + \beta u_{Dp}.$$
 (B.5)

Since β is very large the second term on the left- and right-hand-side will dominate this equation, hence obtaining the correct Dirichlet boundary condition for node p.

The value of β is taken 1e10 in libMesh. A disadvantage of the penalty method can be that large values of β can cause ill-conditioning of the matrix. However, this problem is now largely alleviated by today's use of preconditioners and linear system of equation solvers. This method can thus be seen as an easy and robust approach for applying Dirichlet boundary conditions.

Appendix C

FEM discretized Navier-Stokes equations

In the following the Galerkin terms are presented for building the element matrices of the Navier-Stokes equations resulting from applying the finite element method. The summations signs, $\sum_{g=1}^{n_{qp}} \sum_{a=1}^{n_{en}} \sum_{b=1}^{n_{en}}$, are omitted below to keep to keep the formulation more clear. They should however be placed in front of every term (omit the summation over b for the right-hand-side vector). For the pressure unknown one should loop over the amount of pressure nodes \hat{n}_{en} instead, in case this differs from the amount of velocity nodes. The terms below that contain the velocity and pressure at the current time step are known from the previous Newton iteration. Finally, it is noted that the Galerkin term for the continuity equation changes when PSPG stabilization is added to the equation (i.e., it is multiplied by Δt and the theta method is applied as well).

Right-hand-side vector

$$(\mathbf{b}_{a}^{e})_{u} = W_{|g}J_{|g}\left(\hat{u}_{|g}\phi_{a|g} + \Delta t \left\{ (1-\theta) \left(- \left(\mathbf{\hat{u}}_{|g} \cdot \nabla \hat{u}_{|g}\right)\phi_{a|g} + \hat{p}_{|g}\phi_{a,x|g} - \left(\nabla \hat{u}_{|g} \cdot \nabla \phi_{a|g}\right)\nu\right) \right. \\ \left. + \theta \left(\mathbf{u}_{|g} \cdot \nabla u_{|g}\right)\phi_{a|g} + \phi_{a|g}f_{x|g} + \left[\phi_{a|g}t'_{x|g}\right]_{\Gamma_{N}} \right\} \right)$$

$$(C.1)$$

$$(\mathbf{b}_{a}^{e})_{v} = W_{|g}J_{|g}\left(\hat{v}_{|g}\phi_{a|g} + \Delta t \left\{ (1-\theta) \left(- \left(\mathbf{\hat{u}}_{|g} \cdot \nabla \hat{v}_{|g}\right)\phi_{a|g} + \hat{p}_{|g}\phi_{a,y|g} - \left(\nabla \hat{v}_{|g} \cdot \nabla \phi_{a|g}\right)\nu\right) \right. \\ \left. + \theta \left(\mathbf{u}_{|\mathbf{g}} \cdot \nabla v_{|g}\right)\phi_{a|g} + \phi_{a|g}f_{y|g} + \left[\phi_{a|g}t'_{y|g}\right]_{\Gamma_{N}} \right\} \right)$$

$$(C.2)$$

`

uu-coupling

$$(\mathbf{M}_{ab}^e)_{uu} = W_{|g} J_{|g} \phi_{a|g} \phi_{b|g} \tag{C.3}$$

$$(\mathbf{K}^{e}_{ab})_{uu} = \nu \theta \Delta t W_{|g} J_{|g} \left(\nabla \phi_{a|g} \cdot \nabla \phi_{b|g} \right)$$
(C.4)

$$(\mathbf{C}^{e}_{ab})_{uu} = \theta \Delta t W_{|g} J_{|g} \bigg(\phi_{a|g} \left(\mathbf{u}_{|g} \cdot \nabla \phi_{b|g} \right) + u_{,x|g} \phi_{a|g} \phi_{b|g} \bigg)$$
(C.5)

uv-coupling

$$(\mathbf{C}^{e}_{ab})_{uv} = \theta \Delta t W_{|g} J_{|g} u_{,y|g} \phi_{a|g} \phi_{b|g} \tag{C.6}$$

vv-coupling

$$(\mathbf{M}_{ab}^e)_{vv} = W_{|g} J_{|g} \phi_{a|g} \phi_{b|g} \tag{C.7}$$

$$(\mathbf{K}_{ab}^{e})_{vv} = \nu \theta \Delta t W_{|g} J_{|g} \left(\nabla \phi_{a|g} \cdot \nabla \phi_{b|g} \right)$$
(C.8)

$$(\mathbf{C}^{e}_{ab})_{vv} = \theta \Delta t W_{|g} J_{|g} \bigg(\phi_{a|g} \left(\mathbf{u}_{|g} \cdot \nabla \phi_{b|g} \right) + v_{,y|g} \phi_{a|g} \phi_{b|g} \bigg)$$
(C.9)

vu-coupling

$$(\mathbf{C}^{e}_{ab})_{vu} = \theta \Delta t W_{|g} J_{|g} v_{,x|g} \phi_{a|g} \phi_{b|g} \tag{C.10}$$

up-coupling

$$\left(\mathbf{G}_{a\hat{b}}^{e}\right)_{up} = -\theta \Delta t W_{|g} J_{|g} \phi_{a,x|g} \varphi_{\hat{b}|g} \tag{C.11}$$

vp-coupling

$$\left(\mathbf{G}^{e}_{a\hat{b}}\right)_{vp} = -\theta \Delta t W_{|g} J_{|g} \phi_{a,y|g} \varphi_{\hat{b}|g} \tag{C.12}$$

 $\underline{pu} ext{-coupling}$

$$\left(\mathbf{G}_{\hat{a}b}^{Te}\right)_{pu} = W_{|g}J_{|g}\varphi_{\hat{a}|g}\phi_{b,x|g} \tag{C.13}$$

pv-coupling

$$\left(\mathbf{G}_{\hat{a}\hat{b}}^{Te}\right)_{pv} = W_{|g}J_{|g}\varphi_{\hat{a}|g}\phi_{b,y|g} \tag{C.14}$$

Appendix D

FEM discretized Navier-Stokes stabilization terms

In the following the Navier-Stokes stabilization terms are presented for building the element matrices. These are added to the Galerkin terms in appendix C to obtain the stabilized formulation. The submatrices corresponding to the stabilization terms will be referred to below as **S**. The summations signs, $\sum_{g=1}^{n_{qp}} \sum_{a=1}^{n_{en}} \sum_{b=1}^{n_{en}}$, are again omitted below to keep to keep the formulation more clear. They should however be placed in front of every term (omit the summation over b for the right-hand-side vector). For the pressure unknown one should loop over the amount of pressure nodes \hat{n}_{en} instead, in case this differs from the amount of velocity nodes. The terms below that contain the velocity and pressure at the current time step are known from the previous Newton iteration. Finally, since the Galerkin terms for the continuity equation (**G**^T matrix) changed, they will be restated below as well.

Right-hand-side vector

$$\begin{aligned} (\mathbf{b}_{a}^{e})_{u} &= W_{|g}J_{|g} \bigg(-\tau_{c}\Delta t(1-\theta)\phi_{a,x|g}\nabla\cdot\hat{\mathbf{u}}_{|g} + \tau_{m}\left(\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right)u_{|g} \\ &+ \tau_{m}\theta\Delta t\bigg\{ \left(\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right)\left[2\left(\nabla u_{|g}\cdot\mathbf{u}_{|g}\right) + \nu\nabla^{2}u_{|g} + p_{,x|g}\right]\bigg\}\bigg) \end{aligned} \tag{D.1} \\ (\mathbf{b}_{a}^{e})_{v} &= W_{|g}J_{|g}\bigg(-\tau_{c}\Delta t(1-\theta)\phi_{a,y|g}\nabla\cdot\hat{\mathbf{u}}_{|g} + \tau_{m}\left(\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right)v_{|g} \\ &+ \tau_{m}\theta\Delta t\bigg\{ \left(\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right)\left[2\left(\nabla v_{|g}\cdot\mathbf{u}_{|g}\right) + \nu\nabla^{2}v_{|g} + p_{,y|g}\right]\bigg\}\bigg) \end{aligned}$$

$$\begin{aligned} (\mathbf{b}_{\hat{a}}^{e})_{p} &= W_{|g} J_{|g} \left(\overbrace{-\Delta t(1-\theta)\varphi_{\hat{a}|g}\nabla\cdot\hat{\mathbf{u}}_{|g}}^{\text{Galerkin term}} + \tau_{m} \left((\hat{\mathbf{u}}_{|g}\cdot\nabla\varphi_{\hat{a}|g}) + \Delta t\theta \left[\nabla\varphi_{\hat{a}|g}(\nabla\mathbf{u}_{|g}\cdot\mathbf{u}_{|g}) \right] \right) \\ &+ \tau_{m} \Delta t(1-\theta) \left\{ \nabla\varphi_{\hat{a}|g} \left(- (\nabla\hat{\mathbf{u}}_{|g}\cdot\hat{\mathbf{u}}_{|g}) + \nu\nabla^{2}\hat{\mathbf{u}}_{|g} - \nabla\hat{p} \right) \right\} \right) \end{aligned}$$
(D.3)

 $\underline{uu} ext{-coupling}$

$$(\mathbf{S}_{ab}^{e})_{uu} = W_{|g}J_{|g} \bigg(\tau_{c}\Delta t\theta \phi_{a,x|g}\phi_{b,x|g} + \tau_{m} \bigg[(u_{|g} - \hat{u}_{|g})\phi_{a,x|g}\phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g})\phi_{b|g} \bigg]$$

$$+ \tau_{m}\Delta t\theta \bigg\{ (\nabla u_{|g} \cdot \mathbf{u}_{|g})\phi_{a,x|g}\phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \big[(\mathbf{u}_{|g} \cdot \nabla \phi_{b|g}) + u_{,x|g}\phi_{b|g} \big]$$

$$- \nu \big[\nabla^{2}u_{|g}\phi_{a,x|g}\phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g})\nabla^{2}\phi_{b|g} \big] + p_{,x|g}\phi_{a,x|g}\phi_{b|g} \bigg\}$$

$$+ \tau_{m}\Delta t(1-\theta)\phi_{a,x}\phi_{b|g} \bigg\{ \nabla \hat{u}_{|g} \cdot \hat{\mathbf{u}}_{|g} - \nu \nabla^{2}\hat{u}_{|g} + \hat{p}_{,x|g} \bigg\} - \tau_{m}\Delta t\phi_{a,x}\phi_{b|g}f_{x|g} \bigg)$$

$$(D.4)$$

uv-coupling

$$\begin{aligned} (\mathbf{S}_{ab}^{e})_{uv} &= W_{|g} J_{|g} \bigg(\tau_{c} \Delta t \theta \phi_{a,x|g} \phi_{b,y|g} + \tau_{m} (u_{|g} - \hat{u}_{|g}) \phi_{a,y|g} \phi_{b|g} \\ &+ \tau_{m} \Delta t \theta \Big\{ (\nabla u_{|g} \cdot \mathbf{u}_{|g}) \phi_{a,y|g} \phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) u_{,y|g} \phi_{b|g} \\ &- \nu \nabla^{2} u_{|g} \phi_{a,y|g} \phi_{b|g} + p_{,x|g} \phi_{a,y|g} \phi_{b|g} \Big\} \\ &+ \tau_{m} \Delta t (1 - \theta) \phi_{a,y} \phi_{b|g} \Big\{ \nabla \hat{u}_{|g} \cdot \hat{\mathbf{u}}_{|g} - \nu \nabla^{2} \hat{u}_{|g} + \hat{p}_{,x|g} \Big\} - \tau_{m} \Delta t \phi_{a,y} \phi_{b|g} f_{x|g} \bigg) \end{aligned}$$
(D.5)

vv-coupling

$$(\mathbf{S}_{ab}^{e})_{vv} = W_{|g}J_{|g} \bigg(\tau_{c}\Delta t\theta \phi_{a,y|g} \phi_{b,y|g} + \tau_{m} \bigg[(v_{|g} - \hat{v}_{|g})\phi_{a,y|g} \phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g})\phi_{b|g} \bigg]$$

$$+ \tau_{m}\Delta t\theta \bigg\{ (\nabla v_{|g} \cdot \mathbf{u}_{|g})\phi_{a,y|g} \phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \big[(\mathbf{u}_{|g} \cdot \nabla \phi_{b|g}) + v_{,y|g} \phi_{b|g} \big]$$

$$- \nu \big[\nabla^{2}v_{|g} \phi_{a,y|g} \phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \nabla^{2} \phi_{b|g} \big] + p_{,y|g} \phi_{a,y|g} \phi_{b|g} \bigg\}$$

$$+ \tau_{m}\Delta t (1 - \theta) \phi_{a,y} \phi_{b|g} \bigg\{ \nabla \hat{v}_{|g} \cdot \hat{\mathbf{u}}_{|g} - \nu \nabla^{2} \hat{v}_{|g} + \hat{p}_{,y|g} \bigg\} - \tau_{m}\Delta t \phi_{a,y} \phi_{b|g} f_{y|g} \bigg)$$

$$(D.6)$$

vu-coupling

$$(\mathbf{S}_{ab}^{e})_{vu} = W_{|g}J_{|g} \bigg(\tau_{c}\Delta t\theta \phi_{a,y|g}\phi_{b,x|g} + \tau_{m}(v_{|g} - \hat{v}_{|g})\phi_{a,x|g}\phi_{b|g} + \tau_{m}\Delta t\theta \bigg\{ (\nabla v_{|g} \cdot \mathbf{u}_{|g})\phi_{a,x|g}\phi_{b|g} + (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g})v_{,x|g}\phi_{b|g} - \nu\nabla^{2}v_{|g}\phi_{a,x|g}\phi_{b|g} + p_{,y|g}\phi_{a,x|g}\phi_{b|g} \bigg\} + \tau_{m}\Delta t(1-\theta)\phi_{a,x}\phi_{b|g} \bigg\{ \nabla \hat{v}_{|g} \cdot \hat{\mathbf{u}}_{|g} - \nu\nabla^{2}\hat{v}_{|g} + \hat{p}_{,y|g} \bigg\} - \tau_{m}\Delta t\phi_{a,x}\phi_{b|g}f_{y|g} \bigg)$$
(D.7)

up-coupling

$$\left(\mathbf{S}^{e}_{a\hat{b}}\right)_{up} = \theta \Delta t W_{|g} J_{|g} \tau_{m} (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \varphi_{\hat{b}, x|g}$$
(D.8)

vp-coupling

$$\left(\mathbf{S}_{a\hat{b}}^{e}\right)_{vp} = \theta \Delta t W_{|g} J_{|g} \tau_{m} (\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \varphi_{\hat{b},y|g}$$
(D.9)

 $\underline{pu} ext{-coupling}$

$$\left(\mathbf{G}_{\hat{a}\hat{b}}^{Te}\right)_{pu} = W_{|g}J_{|g}\Delta t\theta\varphi_{\hat{a}|g}\phi_{b,x|g} \tag{D.10}$$

$$(\mathbf{S}^{e}_{\hat{a}b})_{pu} = W_{|g}J_{|g}\tau_{m} \bigg(\varphi_{\hat{a},x|g}\phi_{b|g} + \Delta t\theta \bigg\{ (\mathbf{u}_{|g} \cdot \nabla \phi_{b,g})\varphi_{\hat{a},x|g} + (\mathbf{u}_{,x|g}\nabla \varphi_{\hat{a}|g}\phi_{b|g}) - \nu \varphi_{\hat{a},x|g}\nabla^{2}\phi_{b|g} \bigg\} \bigg)$$
(D.11)

pv-coupling

$$\left(\mathbf{G}_{\hat{a}b}^{Te}\right)_{pv} = W_{|g}J_{|g}\Delta t\theta\varphi_{\hat{a}|g}\phi_{b,y|g} \tag{D.12}$$

$$(\mathbf{S}^{e}_{\hat{a}b})_{pv} = W_{|g}J_{|g}\tau_{m} \bigg(\varphi_{\hat{a},y|g}\phi_{b|g} + \Delta t\theta \bigg\{ (\mathbf{u}_{|g} \cdot \nabla \phi_{b,g})\varphi_{\hat{a},y|g} + (\mathbf{u}_{,y|g}\nabla \varphi_{\hat{a}|g})\phi_{b|g} - \nu \varphi_{\hat{a},y|g}\nabla^{2}\phi_{b|g} \bigg\} \bigg)$$
(D.13)

pp-coupling

$$(\mathbf{S}^{e}_{\hat{a}\hat{b}})_{pp} = W_{|g}J_{|g}\tau_{m}\Delta t\theta(\nabla\varphi_{\hat{a}|g}\cdot\nabla\varphi_{\hat{b}|g})$$
(D.14)

Appendix E

FEM discretized RANS diffusion terms

In the following the RANS diffusion terms are presented for building the element matrices. When replacing these Galerkin and stabilization terms by the Navier-Stokes diffusion terms, presented in appendices C and D, we obtain the element level-equations for the Reynolds-averaged Navier-Stokes equations. Similarly as for the Navier-Stokes element matrices, the summations signs, $\sum_{g=1}^{n_{en}} \sum_{a=1}^{n_{en}} \sum_{b=1}^{n_{en}}$, are omitted below to keep to keep the formulation more clear. They should however be placed in front of every term (omit the summation over *b* for the right-hand-side vector). The terms below that contain the velocity at the current time step are known from the previous Newton iteration.

Right-hand-side vector

$$(\mathbf{b}_{a}^{e})_{u} = W_{|g}J_{|g}\Delta t \left(-(1-\theta)(\nu+\nu_{t|g}) \left[\nabla \phi_{a|g} \cdot (\nabla \hat{u}_{|g} + \hat{\mathbf{u}}_{,x|g}) \right] - \tau_{m}\theta \left((\mathbf{u}_{|g} \cdot \nabla \phi_{a|g}) \left((\nu+\nu_{t|g}) \nabla^{2} u_{|g} + \left[\nabla \nu_{t|g} \cdot (\nabla u_{|g} + \mathbf{u}_{,x|g}) \right] \right) \right)$$

$$(E.1)$$

$$(\mathbf{b}_{a}^{e})_{v} = W_{|g}J_{|g}\Delta t \left(-(1-\theta)(\nu+\nu_{t|g})\nabla\phi_{a|g}\cdot(\nabla\hat{v}_{|g}+\hat{\mathbf{u}}_{,y|g}) -\tau_{m}\theta\left((\mathbf{u}_{|g}\cdot\nabla\phi_{a|g})\left((\nu+\nu_{t|g})\nabla^{2}v_{|g}+\left[\nabla\nu_{t|g}\cdot(\nabla v_{|g}+\mathbf{u}_{,y|g})\right]\right) \right)$$
(E.2)

$$(\mathbf{b}_{\hat{a}}^{e})_{p} = W_{|g}J_{|g}\Delta t(1-\theta)\tau_{m} \left(\nabla\varphi_{\hat{a}|g} \cdot \left((\nu+\nu_{t|g})\nabla^{2}\mathbf{u}_{|g} + \left[\nabla\mathbf{u}_{|g} + \left(\nabla\mathbf{u}_{|g}\right)^{T}\right] \cdot \nabla\nu_{t|g}\right)\right)$$
(E.3)

<u>uu-coupling</u>

$$(\mathbf{K}^{e}_{ab})_{uu} = W_{|g} J_{|g} \theta \Delta t (\nu + \nu_{t|g}) \left(\nabla \phi_{a|g} \cdot \nabla \phi_{b|g} + \phi_{a,x|g} \phi_{b,x|g} \right)$$
(E.4)

$$(\mathbf{S}_{ab}^{e})_{uu} = -W_{|g}J_{|g}\Delta t\tau_{m} \left(\theta \left[\phi_{a,x|g}\phi_{b|g} \left(\nabla^{2}u_{|g}(\nu + \nu_{t|g}) + \nabla\nu_{t|g} \cdot (\nabla u_{|g} + \mathbf{u}_{,x|g}) \right) \right. \\ \left. + \left[\mathbf{u}_{|g} \cdot \nabla\phi_{a|g} \right] \left(\nabla^{2}\phi_{b|g}(\nu + \nu_{t|g}) + \nabla\phi_{b|g} \cdot \nabla\nu_{t|g} + \phi_{b,x|g}\nu_{t,x|g} \right) \right]$$

$$\left. + (1 - \theta)\phi_{a,x|g}\phi_{b|g} \left(\nabla^{2}\hat{u}_{|g}(\nu + \nu_{t|g}) + \nabla\nu_{t|g} \cdot (\nabla\hat{u}_{|g} + \hat{\mathbf{u}}_{,x|g}) \right) \right)$$

$$(E.5)$$

uv-coupling

$$(\mathbf{K}^{e}_{ab})_{uv} = W_{|g} J_{|g} \theta \Delta t (\nu + \nu_{t|g}) \left(\phi_{a,y|g} \phi_{b,x|g} \right)$$
(E.6)

$$(\mathbf{S}_{ab}^{e})_{uv} = -W_{|g}J_{|g}\Delta t\tau_{m} \left(\theta \left[\phi_{a,y|g}\phi_{b|g} \left(\nabla^{2}u_{|g}(\nu+\nu_{t|g}) + \nabla\nu_{t|g} \cdot \left(\nabla u_{|g} + \mathbf{u}_{,x|g}\right)\right) + \left[\mathbf{u}_{|g} \cdot \nabla\phi_{a|g}\right] \left(\phi_{b,x|g}\nu_{t,y|g}\right)\right]$$

$$+ \left[\mathbf{u}_{|g} \cdot \nabla\phi_{a|g}\right] \left(\phi_{b,x|g}\nu_{t,y|g}\right) + \left(1 - \theta\right)\phi_{a,y|g}\phi_{b|g} \left(\nabla^{2}\hat{u}_{|g}(\nu+\nu_{t|g}) + \nabla\nu_{t|g} \cdot \left(\nabla\hat{u}_{|g} + \hat{\mathbf{u}}_{,x|g}\right)\right) \right)$$

$$(E.7)$$

vv-coupling

$$(\mathbf{K}_{ab}^{e})_{vv} = W_{|g}J_{|g}\theta\Delta t(\nu+\nu_{t|g})\left(\nabla\phi_{a|g}\cdot\nabla\phi_{b|g}+\phi_{a,y|g}\phi_{b,y|g}\right)$$
(E.8)

$$(\mathbf{S}_{ab}^{e})_{vv} = -W_{|g}J_{|g}\Delta t\tau_{m} \left(\theta \left[\phi_{a,y|g}\phi_{b|g}\left(\nabla^{2}v_{|g}(\nu+\nu_{t|g})+\nabla\nu_{t|g}\cdot(\nabla v_{|g}+\mathbf{u}_{,y|g})\right)\right. \\ \left. + \left[\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right]\left(\nabla^{2}\phi_{b|g}(\nu+\nu_{t|g})+\nabla\phi_{b|g}\cdot\nabla\nu_{t|g}+\phi_{b,y|g}\nu_{t,y|g}\right)\right]$$

$$+ \left(1-\theta\right)\phi_{a,y|g}\phi_{b|g}\left(\nabla^{2}\hat{v}_{|g}(\nu+\nu_{t|g})+\nabla\nu_{t|g}\cdot(\nabla\hat{v}_{|g}+\hat{\mathbf{u}}_{,y|g})\right)\right)$$

$$(E.9)$$

vu-coupling

$$(\mathbf{K}^{e}_{ab})_{vu} = W_{|g} J_{|g} \theta \Delta t (\nu + \nu_{t|g}) \left(\phi_{a,x|g} \phi_{b,y|g} \right)$$
(E.10)

$$\begin{aligned} (\mathbf{S}_{ab}^{e})_{vu} &= -W_{|g}J_{|g}\Delta t\tau_{m} \left(\theta \left[\phi_{a,x|g}\phi_{b|g} \left(\nabla^{2}v_{|g}(\nu+\nu_{t|g})+\nabla\nu_{t|g}\cdot\left(\nabla v_{|g}+\mathbf{u}_{,y|g}\right)\right)\right. \\ &+ \left[\mathbf{u}_{|g}\cdot\nabla\phi_{a|g}\right] \left(\phi_{b,y|g}\nu_{t,x|g}\right)\right] \\ &+ \left(1-\theta\right)\phi_{a,x|g}\phi_{b|g} \left(\nabla^{2}\hat{v}_{|g}(\nu+\nu_{t|g})+\nabla\nu_{t|g}\cdot\left(\nabla\hat{v}_{|g}+\hat{\mathbf{u}}_{,y|g}\right)\right)\right) \end{aligned}$$
(E.11)

$\underline{pu} ext{-coupling}$

$$(\mathbf{S}^{e}_{\hat{a}b})_{pu} = -W_{|g}J_{|g}\theta\Delta t\tau_{m} \left(\varphi_{\hat{a},x|g}\left((\nu+\nu_{t|g})\nabla^{2}\phi_{b|g}+\nabla\nu_{t|g}\cdot\nabla\phi_{b|g}\right)\right)$$
$$+\nu_{t,x|g}\left(\nabla\varphi_{\hat{a}|g}\cdot\nabla\phi_{b|g}\right)\right)$$
(E.12)

pv-coupling

$$(\mathbf{S}^{e}_{\hat{a}b})_{pv} = -W_{|g}J_{|g}\theta\Delta t\tau_{m} \bigg(\varphi_{\hat{a},y|g} \Big((\nu + \nu_{t|g})\nabla^{2}\phi_{b|g} + \nabla\nu_{t|g}\cdot\nabla\phi_{b|g} \Big) + \nu_{t,y|g} \big(\nabla\varphi_{\hat{a}|g}\cdot\nabla\phi_{b|g}\big) \bigg)$$
(E.13)

Appendix F

FEM discretized S-A turbulence model

In the following the Galerkin and multiscale stabilizing terms are presented for building the element matrices of the advection-diffusion-reaction equation, resulting from applying the finite element method. The summations signs, $\sum_{g=1}^{n_{en}} \sum_{a=1}^{n_{en}} \sum_{b=1}^{n_{en}}$, are omitted below to keep to keep the formulation more clear. They should however be placed in front of every term (omit the summation over b for the right-hand-side vector). The terms below that contain the modified turbulent viscosity at the current time step are known from the previous Newton iteration. The parameters of the model are substituted such that they represent the S-A turbulence model, these are given in equations (4.50)-(4.52) for the baseline model, and equations (4.71)-(4.73) for the negative S-A model. Below the right-handside element vector is termed \mathbf{b}_{a}^{e} , while the element matrix is termed \mathbf{K}_{ab}^{e} .

Right-hand-side vector

$$\mathbf{b}_{a}^{e} = W_{|g}J_{|g}f_{|g}\left(\phi_{a|g} + \tau_{mfv}\left[\mathbf{a}_{|g} \cdot \nabla\phi_{a|g} + \kappa_{|g}\nabla^{2}\phi_{a|g} - s_{|g}\phi_{a|g}\right]\right)$$
(F.1)

<u>Element matrix</u>

$$\mathbf{K}_{ab}^{e} = W_{|g} J_{|g} \bigg(\phi_{a|g} \big(\mathbf{a}_{|g} \cdot \nabla \phi_{b|g} \big) + \kappa_{|g} \big(\nabla \phi_{a|g} \cdot \nabla \phi_{b|g} \big) + s_{|g} \phi_{a|g} \phi_{b|g} + \tau_{mfv} \big(\mathbf{a}_{|g} \cdot \nabla \phi_{a|g} + \kappa_{|g} \nabla^{2} \phi_{a|g} - s_{|g} \phi_{a|g} \big) \big(\mathbf{a}_{|g} \cdot \nabla \phi_{b|g} - \kappa \nabla^{2} \phi_{b|g} + s_{|g} \phi_{b|g} \big) \bigg)$$
(F.2)

Appendix G

Multiscale method applied to Spalart-Allmaras reaction terms

For the application of the multiscale method to the Spalart-Allmaras turbulence model reaction terms use is made of the binomial theorem, where for x close to zero we have,

$$\frac{1}{1+x} \approx 1 - x \,. \tag{G.1}$$

Next to this all higher-order fine scale terms will be ignored below for convenience.

The Spalart-Allmaras baseline model and its coefficients are given in section 2.3.2. Let us start with the term f_{v1} :

$$f_{v1} = \frac{\nu_e^3}{\nu_e^3 + \nu^3 c_{v1}^3} \,. \tag{G.2}$$

Applying a splitting of the unknown into coarse and fine scales, $\nu_e = \overline{\nu_e} + \nu'_e$, we obtain,

$$f_{v1} = \frac{(\overline{\nu_e} + \nu'_e)^3}{(\overline{\nu_e} + \nu'_e)^3 + \nu^3 c_{v1}^3}$$
(G.3)

$$\approx \underbrace{\frac{\overline{\nu_e}^3 + 3\overline{\nu_e}^2 \nu'_e}{\nu_e^3 c_{v_1}^3 + \overline{\nu_e}^3} + 3\overline{\nu_e}^2 \nu'_e}_{=\frac{1}{a}} \tag{G.4}$$

$$=\frac{a\left(\overline{\nu_e}^3 + 3\overline{\nu_e}^2\nu'_e\right)}{1 + 3a\overline{\nu_e}^2\nu'_e} \tag{G.5}$$

$$\approx \left(a\overline{\nu_e}^3 + 3a\overline{\nu_e}^2\nu_e'\right)\left(1 - 3a\overline{\nu_e}^2\nu_e'\right) \tag{G.6}$$

$$\approx a\overline{\nu_e}^3 + \left(\underbrace{3a\overline{\nu_e}^2 - 3a^2\overline{\nu_e}^5}_{=b}\right)\nu'_e \tag{G.7}$$

$$=\overline{f}_{v1} + b\nu'_e \tag{G.8}$$

where \overline{f}_{v1} is calculated using the coarse scale modified turbulent viscosity.

The term f_{v2} is defined as,

$$f_{v2} = 1 - \frac{\nu_e}{\nu + \nu_e f_{v1}} \,. \tag{G.9}$$

Applying a splitting of the unknown into coarse and fine scales and inserting f_{v1} from equation (G.8) we obtain,

$$f_{v2} = 1 - \frac{\overline{\nu_e} + \nu'_e}{\nu + (\overline{\nu_e} + \nu'_e) \left(\overline{f}_{v1} + b\nu'_e\right)} \tag{G.10}$$

$$\approx 1 - \underbrace{\frac{\overline{\nu_e} + \nu'_e}{\nu + \overline{\nu_e}\overline{f_{v1}}}_{=\frac{1}{c}} + \left(\overline{f_{v1}} + b\overline{\nu_e}\right)\nu'_e}_{=\frac{1}{c}} \tag{G.11}$$

$$\approx 1 - c\left(\overline{\nu_e} + \nu'_e\right) \left(1 - c\left(\overline{f}_{v1} + b\overline{\nu_e}\right)\nu'_e\right) \tag{G.12}$$

$$\approx 1 - c\overline{\nu_e} - \left(\underbrace{c - c^2\overline{\nu_e}\left(\overline{f}_{v1} + b\overline{\nu_e}\right)}_{=b}\right)\nu'_e \tag{G.13}$$

$$=\overline{f}_{v2} - h\nu'_e \tag{G.14}$$

where \overline{f}_{v2} is calculated using the coarse scale modified turbulent viscosity.

The modified magnitude of vorticity is given by,

$$S_e = S + \frac{\nu_e}{\kappa^2 d^2} f_{v2} \,.$$
 (G.15)

Applying a splitting of the unknown into coarse and fine scales and inserting f_{v2} from equation (G.14) we obtain,

$$S_e = S + \frac{\overline{\nu_e} + \nu'_e}{\kappa^2 d^2} \left(\overline{f}_{v2} - h\nu'_e \right) \tag{G.16}$$

$$\approx S + \frac{\overline{\nu_e}}{\kappa^2 d^2} \overline{f}_{\nu 2} + \underbrace{\frac{\overline{f}_{\nu 2} - h\overline{\nu_e}}{\kappa^2 d^2}}_{=j_1} \nu'_e \tag{G.17}$$

$$=\overline{S}_e + j_1 \nu'_e. \tag{G.18}$$

where \overline{S}_e is calculated using the coarse scale modified turbulent viscosity.

In section 2.3.3 modifications to the baseline S-A model were introduced. In case $\frac{\nu_e}{\kappa^2 d^2} f_{v2} < c_{v2}S$ we instead calculate S_e as,

$$S_e = S + \frac{S\left(c_{v2}^2 S + c_{v3}\frac{\nu_e}{\kappa^2 d^2} f_{v2}\right)}{\left(c_{v3} - 2c_{v2}\right)S - \frac{\nu_e}{\kappa^2 d^2} f_{v2}}.$$
(G.19)

Using equation (G.18) we obtain,

$$S_{e} = S + \frac{S\left(c_{v2}^{2}S + c_{v3}\left(\frac{\overline{\nu_{e}}}{\kappa^{2}d^{2}}\overline{f}_{v2} + j_{1}\nu'_{e}\right)\right)}{(c_{v3} - 2c_{v2})S - \frac{\overline{\nu_{e}}}{\kappa^{2}d^{2}}\overline{f}_{v2}} - j_{1}\nu'_{e}}$$

$$= \frac{1}{z}$$

$$\approx S + zS\left(c_{v2}^{2}S + c_{v3}\frac{\overline{\nu_{e}}}{\kappa^{2}d^{2}}\overline{f}_{v2} + c_{v3}j_{1}\nu'_{e}}\right)\left(1 + zj_{1}\nu'_{e}\right)$$
(G.20)
(G.21)

$$\approx \overline{S}_e + \underbrace{zj_1\left(\frac{S\left(c_{v2}^2S + c_{v3}\frac{\overline{\nu_e}}{\kappa^2 d^2}\overline{f}_{v2}\right)}{\left(c_{v3} - 2c_{v2}\right)S - \frac{\overline{\nu_e}}{\kappa^2 d^2}\overline{f}_{v2}} + Sc_{v3}\right)}_{=j_2}\nu'_e \qquad (G.22)$$

$$=\overline{S}_e + j_2\nu'_e \tag{G.23}$$

The contribution of the reaction term $-c_{b1}S_e\nu_e$ through application of the multiscale method can now be assessed to be,

$$-c_{b1}S_e\nu_e = -c_{b1}\left(\overline{\nu_e} + \nu'_e\right)\left(\overline{S}_e + j\nu'_e\right) \tag{G.24}$$

$$\approx -c_{b1} \left(\overline{S}_e \overline{\nu_e} + \left(\overline{S}_e + j \overline{\nu_e} \right) \nu'_e \right) \tag{G.25}$$

with j equaling j_1 or j_2 depending on how S_e is calculated.

Next, the term r is defined as,

$$r = \min\left[\frac{\nu_e}{\kappa^2 d^2 S_e}, 10\right] \,. \tag{G.26}$$

Let us discuss the case where $\frac{\nu_e}{\kappa^2 d^2 S_e} < 10$, in any other case r = 10 will be taken and no splitting of scales is necessary. Applying a splitting of the unknown into coarse and fine scales and inserting S_e from equations (G.18) and (G.23) we obtain,

$$r = \underbrace{\frac{\overline{\nu_e} + \nu'_e}{\underbrace{\kappa^2 d^2 \overline{S}_e}_{=\frac{1}{x}} + \kappa^2 d^2 j \nu'_e}}_{=\frac{1}{x}}$$
(G.27)

$$\approx n\left(\overline{\nu_e} + \nu'_e\right)\left(1 - n\kappa^2 d^2 j\nu'_e\right) \tag{G.28}$$

$$\approx n\overline{\nu_e} + \left(\underbrace{n - n^2 \kappa^2 d^2 j \overline{\nu_e}}_{=m}\right) \nu'_e \tag{G.29}$$

$$= \overline{r} + m\nu'_e \tag{G.30}$$

where \overline{r} is calculated using $\overline{\nu_e}$. Continuing for the term g and inserting r from the equation above:

$$g = r + c_{w2} \left(r^6 - r \right) \tag{G.31}$$

$$= \bar{r} + m\nu'_{e} + c_{w2} \left(\left(\bar{r} + m\nu'_{e} \right)^{6} - \bar{r} - m\nu'_{e} \right)$$
(G.32)

$$\approx \overline{r} + m\nu'_e + c_{w2} \left(\overline{r}^6 + 6\overline{r}^5 m\nu'_e - \overline{r} - m\nu'_e \right) \tag{G.33}$$

$$=\overline{r} + c_{w2}\left(\overline{r}^{6} - \overline{r}\right) + \left(\underbrace{m + 6c_{w2}\overline{r}^{5}m - c_{w2}m}_{=q}\right)\nu'_{e} \tag{G.34}$$

$$= \overline{g} + q\nu'_e \tag{G.35}$$

where \overline{g} is calculated using \overline{r} .

The wall function f_w is defined as,

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}} . \tag{G.36}$$

Focusing at the part between brackets first we obtain,

$$\frac{1+c_{w3}^6}{g^6+c_{w3}^6} = \frac{1+c_{w3}^6}{\underline{\overline{g}}^6+c_{w3}^6} + 6\overline{g}^5q\nu_e' \tag{G.37}$$

$$= \alpha \left(1 + c_{w3}^6\right) \left(1 - 6\alpha \overline{g}^5 q \nu_e'\right) \tag{G.38}$$

$$=\underbrace{\left(1+c_{w3}^{6}\right)\alpha}_{=\beta} + \left(\underbrace{-6\alpha^{2}\left(1+c_{w3}^{6}\right)\overline{g}^{5}q}_{=\gamma}\right)\nu_{e}^{\prime} \tag{G.39}$$

$$= \beta + \gamma \nu'_e. \tag{G.40}$$

Therefore we have,

$$\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{\frac{1}{6}} = \left(\beta + \gamma\nu_e'\right)^{\frac{1}{6}} \tag{G.41}$$

$$=\beta^{\frac{1}{6}} \left(1 + \frac{\gamma}{\beta}\nu'_e\right)^{\frac{1}{6}}.$$
 (G.42)

Observe that for a function $f(x) = x^{\frac{1}{6}}$ with $x = 1 + \epsilon$, ϵ being any small number, the Taylor series around x = 1 is given as follows:

$$f(1+\epsilon) \approx f(1) + \left. \frac{\partial f}{\partial x} \right|_1 \epsilon + \left. \frac{\partial^2 f}{\partial x^2} \right|_1 \frac{\epsilon^2}{2} + \dots$$
 (G.43)

$$= 1 + \frac{\epsilon}{6} - \frac{5\epsilon^2}{72} \,. \tag{G.44}$$

Applying this result to equation (G.42) with $\epsilon = \frac{\gamma}{\beta}\nu'_e$ and again ignoring higher order terms results in,

$$\beta^{\frac{1}{6}} \left(1 + \frac{\gamma}{\beta} \nu'_e \right)^{\frac{1}{6}} \approx \beta^{\frac{1}{6}} \left(1 + \frac{\gamma}{6\beta} \nu'_e \right) . \tag{G.45}$$

Taking into account the definition of f_w given in equation (G.36) we obtain,

$$f_w \approx \left(\overline{g} + q\nu'_e\right)\beta^{\frac{1}{6}} \left(1 + \frac{\gamma}{6\beta}\nu'_e\right) \tag{G.46}$$

$$\approx \overline{g}\beta^{\frac{1}{6}} + \left(\underbrace{q\beta^{\frac{1}{6}} + \frac{\overline{g}\gamma\beta^{\frac{1}{6}}}{6\beta}}_{=\eta}\right)\nu'_{e} \tag{G.47}$$

$$=\overline{f}_w + \eta\nu'_e \tag{G.48}$$

where \overline{f}_w is calculated using \overline{g} .

Finally, the contribution of the reaction term $c_{w1}f_w\left(\frac{\nu_e}{d}\right)^2$ through application of the multiscale method can now be determined to be,

$$c_{w1}f_w\left(\frac{\nu_e}{d}\right)^2 \approx \frac{c_{w1}}{d^2}\left(\overline{f}_w + \eta\nu'_e\right)\left(\overline{\nu_e}^2 + 2\overline{\nu_e}\nu'_e\right) \tag{G.49}$$

$$\approx c_{w1}\overline{f}_w \left(\frac{\overline{\nu_e}}{d}\right)^2 + \frac{c_{w1}}{d^2} \left(\eta \overline{\nu_e}^2 + 2\overline{f}_w \overline{\nu_e}\right) \nu'_e. \tag{G.50}$$

Note that in the negative S-A model introduced in section 2.3.3 the terms f_w and S_e are not present in the reaction terms, hence no extra contribution from the multiscale method needs to be determined from these.

Appendix H

MMS source terms for RANS and Spalart-Allmaras

Source terms for the Reynolds-averaged Navier-Stokes manufactured solution:

$$\begin{split} f_x(x,y) =& 1 - 2x - (\nu + \nu_t) \left(2 \ (1-x)^2 \ (2 \ y - 6 \ y^2 + 4 \ y^3) - 8 \ x \ (1-x) \ (2 \ y - 6 \ y^2 + 4 \ y^3) \right) \\ &+ 2 \ x^2 \ (2 \ y - 6 \ y^2 + 4 \ y^3) \left) - 2 \ \frac{\partial \nu_t}{\partial x} \ \left(2 \ x \ (1-x)^2 \ (2 \ y - 6 \ y^2 + 4 \ y^3) \right) \\ &- 2 \ x^2 \ (1-x) \ (2 \ y - 6 \ y^2 + 4 \ y^3) \right) - (\nu + \nu_t) \ x^2 \ (1-x)^2 \ (-12 + 24 \ y) \\ &- \frac{\partial \nu_t}{\partial y} \ \left(x^2 \ (1-x)^2 \ (2 - 12 \ y + 12 \ y^2) - y^2 \ (1-y)^2 \ (2 - 12 \ x + 12 \ x^2) \right) \\ &+ x^2 \ (1-x)^2 \ (2 \ y - 6 \ y^2 + 4 \ y^3) \ \left(2 \ x \ (1-x)^2 \ (2 \ y - 6 \ y^2 + 4 \ y^3) \right) \\ &- 2 \ x^2 \ (1-x) \ (2 \ y - 6 \ y^2 + 4 \ y^3) \ \right) \\ &- y^2 \ (1-y)^2 \ (2 \ x - 6 \ x^2 + 4 \ x^3) \ x^2 \ (1-x)^2 \ (2 - 12 \ y + 12 \ y^2) \ , \end{split}$$

$$\begin{split} f_y(x,y) &= (\nu + \nu_t) y^2 (1-y)^2 (-12+24 \, x) - \frac{\partial \nu_t}{\partial x} \left(x^2 (1-x)^2 \left(2-12 \, y+12 \, y^2 \right) \right. \\ &\quad - y^2 \left(1-y \right)^2 \left(2-12 \, x+12 \, x^2 \right) \left. \right) - (\nu + \nu_t) \left(-2 \, \left(1-y \right)^2 \left(2 \, x-6 \, x^2+4 \, x^3 \right) \right. \\ &\quad + 8 \, y \left(1-y \right) \left(2 \, x-6 \, x^2+4 \, x^3 \right) - 2 \, y^2 \left(2 \, x-6 \, x^2+4 \, x^3 \right) \right) \\ &\quad - 2 \, \frac{\partial \nu_t}{\partial y} \left(-2 \, y \left(1-y \right)^2 \left(2 \, x-6 \, x^2+4 \, x^3 \right) + 2 \, y^2 \left(1-y \right) \left(2 \, x-6 \, x^2+4 \, x^3 \right) \right) \\ &\quad - x^2 \left(1-x \right)^2 \left(2 \, y-6 \, y^2+4 \, y^3 \right) y^2 \left(1-y \right)^2 \left(2-12 \, x+12 \, x^2 \right) \\ &\quad - y^2 \left(1-y \right)^2 \left(2 \, x-6 \, x^2+4 \, x^3 \right) \left(-2 \, y \left(1-y \right)^2 \left(2 \, x-6 \, x^2+4 \, x^3 \right) \right) \\ &\quad + 2 \, y^2 \left(1-y \right) \left(2 \, x-6 \, x^2+4 \, x^3 \right) \right), \end{split}$$

$$f_p(x,y) = 0$$

,

The following manufactured solution source term is added to the already existing source given in equation (4.52) for the Spalart-Allmaras model:

$$f = \Delta t \left(0.05 \cos(\pi x) \pi \sin(\pi y) u + 0.05 \sin(\pi x) \cos(\pi y) \pi v - 0.05 c_{b1} S_e \sin(\pi x) \sin(\pi y) \right. \\ \left. + \frac{0.0025 c_{w1} \left(r + c_{w2} \left(r^6 - r\right)\right) \sqrt[6]{\frac{1 + c_{w3}^6}{(r + c_{w2} \left(r^6 - r\right))^6 + c_{w3}^6}} \left(\sin(\pi x)\right)^2 \left(\sin(\pi y)\right)^2}{d^2} \right. \\ \left. - \frac{0.0025 \left(\cos(\pi x)\right)^2 \pi^2 \left(\sin(\pi y)\right)^2 - 0.05 \left(\nu + 0.05 \sin(\pi x) \sin(\pi y)\right) \sin(\pi x) \pi^2 \sin(\pi y)}{\sigma} \right. \\ \left. - \frac{0.0025 \left(\sin(\pi x)\right)^2 \left(\cos(\pi y)\right)^2 \pi^2 - 0.05 \left(\nu + 0.05 \sin(\pi x) \sin(\pi y)\right) \sin(\pi x) \pi^2 \sin(\pi y)}{\sigma} \right. \\ \left. - \frac{c_{b2} \left(0.0025 \left(\cos(\pi x)\right)^2 \pi^2 \left(\sin(\pi y)\right)^2 + 0.0025 \left(\sin(\pi x)\right)^2 \left(\cos(\pi y)\right)^2 \pi^2\right)}{\sigma} \right).$$

Note that the multiplication by the time step for this source term is consistent with the multiplication by Δt performed in equation (4.41). Furthermore we have,

$$S_e = S + \overline{S} \,,$$

$$\overline{S} = \frac{0.05\,\sin\left(\pi\,x\right)\sin\left(\pi\,y\right)\left(1 - \frac{0.05\,\sin(\pi\,x)\sin(\pi\,y)}{\nu\left(\frac{1+0.0000625\,(\sin(\pi\,x))^4(\sin(\pi\,y))^4}{\nu^4\left(0.000125\,\frac{(\sin(\pi\,x))^3(\sin(\pi\,y))^3}{\nu^3}+357.911\right)}\right)}{\kappa^2 d^2}\right)}{\kappa^2 d^2}$$

and

$$r = \min\left[\frac{0.05\sin(\pi x)\sin(\pi y)}{\kappa^2 d^2 S_e}, 10\right] \,.$$