11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

NUMERICAL METHODS FOR THE OPTIMIZATION OF NONLINEAR RESIDUAL-BASED SUBGRID-SCALE MODELS USING THE VARIATIONAL GERMANO IDENTITY

Gabriel D. Maher*, **Steven J. Hulshoff**†

* Faculty of Aerospace Engineering
Delft University of Technology
Kluyverweg 1, 2629HS Delft, The Netherlands
e-mail: gabriel.d.maher@gmail.com, web page: www.lr.tudelft.nl/aerodynamics

†Faculty of Aerospace Engineering
Delft University of Technology
Kluyverweg 2, 2629HT Delft, The Netherlands
e-mail: s.j.hulshoff@tudelft.nl - web page: www.lr.tudelft.nl/aerodynamics

Key words: Subgrid-scale model, Variational Multiscale Method, Variational Germano Identity, Optimization, Turbulence

Abstract. The Variational Germano Identity [1, 2] is used to optimize the coefficients of residual-based subgrid-scale models that arise from the application of a Variational Multiscale Method [3, 4]. It is demonstrated that numerical iterative methods can be used to solve the Germano relations to obtain values for the parameters of subgrid-scale models that are nonlinear in their coefficients. Specifically, the Newton-Raphson method is employed. A least-squares minimization formulation of the Germano Identity is developed to resolve issues that occur when the residual is positive and negative over different regions of the domain. In this case a Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is used to solve the minimization problem.

The developed method is applied to the one-dimensional unsteady forced Burgers' equation and the two-dimensional steady Stokes' equations. It is shown that the Newton-Raphson method and BFGS algorithm generally solve, or minimize the residual of, the Germano relations in a relatively small number of iterations. The optimized subgrid-scale models are shown to outperform standard SGS models with respect to a L_2 error. Additionally, the nonlinear SGS models tend to achieve lower L_2 errors than the linear models.

1 INTRODUCTION

The numerical solution of multiscale Partial Differential Equations (PDEs), i.e. PDEs where the solution contains a large range of different scales that are dynamically important for predicting quantities of interest, can be a challenging task. This is due to the fact that the computational effort to compute all the necessary scales may be intractable. In order to deal with multiscales PDEs, Hughes et al introduced a Variational Multiscale Method (VMM) in [3, 4]. In the VMM, the solution of the PDE $\boldsymbol{u}$, is sum-decomposed as $\boldsymbol{u} = \boldsymbol{u}^h + \boldsymbol{u}'$, where $\boldsymbol{u}^h$ denote the numerically resolved scales and $\boldsymbol{u}'$ the unresolved scales. Since the effect of $\boldsymbol{u}'$ is non-negligible, by not computing $\boldsymbol{u}'$, $\boldsymbol{u}^h$ is not accurately computed. Therefore the influence of $\boldsymbol{u}'$ on $\boldsymbol{u}^h$ must be represented.

One approach for accounting for the influence of $\boldsymbol{u}'$ is to model $\boldsymbol{u}'$ using a subgrid-scale (SGS) model. In most cases only approximate models are available for $\boldsymbol{u}'$ which depend on a number of parameters. Often optimal values for the model parameters cannot be exactly determined. As a solution to this problem, Oberai et al showed that the variational Germano method (VGM) [5, 1] could be used to dynamically calibrate parameters in numerical methods. However the VGM has only seen limited application in combination with the VMM [2] and in most cases only SGS models are used for which the relations appearing in the VGM can be solved analytically.

There is evidence that improved SGS models for use with the VMM can be obtained by including nonlinear dependencies and spatial variations, as shown by Gravemeier [6] and Calo [7]. In doing so, it may no longer be possible to solve the VGM relations analytically for the SGS Model parameters. This motivates the need for a framework in which the VGM can be applied to more general SGS models. Therefore, in this paper two methods are proposed and tested which can numerically solve the VGM relations for arbitrary SGS Models. The combination of VGM and VMM is then investigated by considering the unsteady forced Burgers' equation and Stokes equations.

2 VARIATIONAL MULTISCALE METHOD

The following problem is considered:

$$\begin{aligned}&\text{find } \boldsymbol{u} \in \mathcal{V}: \\ &\mathcal{L}\boldsymbol{u} = \boldsymbol{f}, \text{ on } \Omega\end{aligned} \tag{1}$$

with appropriate boundary and initial conditions. $\mathcal{L}$ represents a potentially non-linear differential operator, $\mathcal{V}$ is the function space that contains the solution, Ω is the domain and $\boldsymbol{f}$ is a specified forcing function. The Galerkin variational form of (1) is shown in (2).

$$\begin{aligned}&\text{find } \boldsymbol{u} \in \mathcal{V}: \\ &B(\boldsymbol{w}, \boldsymbol{u}) = (\boldsymbol{w}, \boldsymbol{f}), \ \forall \boldsymbol{w} \in \mathcal{V}\end{aligned} \tag{2}$$

In this case $(\cdot, \cdot)$ is the standard L^2 inner-product on Ω and $B(\boldsymbol{w}, \boldsymbol{u}) := (\boldsymbol{w}, \mathcal{L}\boldsymbol{u})$.

In the VMM as proposed by Hughes et al [3, 4], a numerical solution to (2) is sought in the finite dimensional function space $\mathcal{V}^h \subset \mathcal{V}$, where h represents a characteristic length of the discretization. This defines the projector $\mathbb{P}^h : \mathcal{V} \to \mathcal{V}^h$. The resolved scales are the part of $\boldsymbol{u}$ that can be represented in $\mathcal{V}^h$ and are given by $\boldsymbol{u}^h = \mathbb{P}^h \boldsymbol{u}$. The unresolved scales denote the part of $\boldsymbol{u}$ that is not contained in $\mathcal{V}^h$ and are given by:

$$\boldsymbol{u}' := \boldsymbol{u} - \mathbb{P}^h \boldsymbol{u} = (\mathbb{I} - \mathbb{P}^h)\boldsymbol{u} \in \mathcal{V}' := \mathcal{V} \backslash \mathcal{V}^h \tag{3}$$

A similar decomposition into resolved and unresolved scales can be made for $\boldsymbol{w}$.

For linear PDEs it can be shown that $\boldsymbol{u}'$ is governed by an unresolved scale Green's function [3]. For nonlinear PDEs this represents a first order perturbation approximation for $\boldsymbol{u}'$ as shown by Scovazzi [8]. In both cases the following expression can be used:

$$\begin{aligned} &\boldsymbol{u}' \approx -\boldsymbol{\tau} \boldsymbol{R} \\ &\boldsymbol{R} := \mathcal{L}\boldsymbol{u}^h - \boldsymbol{f} \end{aligned} \tag{4}$$

where $\boldsymbol{R}$ is the resolved scale residual and $\boldsymbol{\tau}$ is an approximate unresolved scale Green's function. Substituting $\boldsymbol{u} = \boldsymbol{u}^h + \boldsymbol{u}' = \boldsymbol{u}^h - \boldsymbol{\tau}\boldsymbol{R}$ in (2) allows $\boldsymbol{u}^h$ to be determined by solving the variational problem in (5).

$$\begin{aligned} &\text{find } \boldsymbol{u}^h \in \mathcal{V}^h : \\ &B(\boldsymbol{w}^h, \boldsymbol{u}^h) + M(\boldsymbol{w}^h, \boldsymbol{u}^h; \vec{c}, \boldsymbol{f}, h) = (\boldsymbol{w}^h, \boldsymbol{f}), \ \forall \boldsymbol{w}^h \in \mathcal{V}^h \end{aligned} \tag{5}$$

$\vec{c}$ is a vector of constants that $\boldsymbol{\tau}$ depends on and $M(\boldsymbol{w}^h, \boldsymbol{u}^h; \vec{c}, \boldsymbol{f}, h)$ represents the additional terms that occur due to the substitution.

3 VARIATIONAL GERMANO METHOD AND NUMERICAL SOLUTION PROCEDURES

Application of the VMM leads to a subgrid-scale model for $\boldsymbol{u}'$ that depends on a vector of parameters $\vec{c}$. The definition of $\boldsymbol{u}'$ depends on the definition of $\mathcal{V}^h$. Therefore $\vec{c}$ will have values that depend on the numerical discretization used. In many cases optimal values for $\vec{c}$ are not known. The problem of determining $\vec{c}$ can be solved by using the variational Germano method as proposed by Oberai et al [5, 1, 2]. The VGM does not require the exact solution of the PDE problem to be known, making it a suitable procedure for general nonlinear problems.

In the VGM a series of coarse nested function spaces, $\mathcal{V}^{h_N} \subset \mathcal{V}^{h_{N-1}} \subset ... \subset \mathcal{V}^{h_1} \subset \mathcal{V}^h$ are defined. On each $\mathcal{V}^{h_i}$ a coarse space solution $\boldsymbol{u}^{h_i}$ can be defined. The optimal $\boldsymbol{u}^{h_i}$ will be given by $\mathbb{P}^{h_i}\boldsymbol{u}$, where $\mathbb{P}^{h_i} : \mathcal{V} \to \mathcal{V}^{h_i}$ is a projector onto the coarse function space $\mathcal{V}^{h_i}$. As $\boldsymbol{u}$ is generally not known, $\boldsymbol{u}^{h_i}$ can instead be obtained via $\boldsymbol{u}^{h_i} = \mathbb{P}^{h_i}\boldsymbol{u}^h$. If $\boldsymbol{u}^h = \mathbb{P}^h\boldsymbol{u}$ then $\boldsymbol{u}^{h_i} = \mathbb{P}^{h_i}\mathbb{P}^h\boldsymbol{u}$. Therefore the $\boldsymbol{u}^{h_i}$ will still be optimal provided $\mathbb{P}^{h_i}\mathbb{P}^h = \mathbb{P}^{h_i}$ [1]. There are a number of projectors with this property. In this paper the L_2 projector is used.

Optimal values for $\vec{c}$ can then be found by solving the variational inverse problem:

$$\text{find } \vec{c}: \quad B(\boldsymbol{w}^{h_i}, \boldsymbol{u}^{h_i}) + M(\boldsymbol{w}^{h_i}, \boldsymbol{u}^{h_i}; \vec{c}, \boldsymbol{f}, h_i) - (\boldsymbol{w}^{h_i}, \boldsymbol{f}) = 0, \ \forall \boldsymbol{w}^{h_i} \in \mathcal{V}^{h_i}, \ i = 1, ..., N \tag{6}$$

(6) is known as the Germano identity and generally has more equations than there are coefficients $\vec{c}$. The amount of equations can be reduced by interpreting (6) in a global sense, i.e. taking all inner products over the entire domain Ω [5, 1]. In this way (6) reduces to one system of equations per $\mathcal{V}^{h_i}$. The equations in each system can further be summed to produce one equation. Then as many $\mathcal{V}^{h_i}$ must be defined as there are components of $\vec{c}$.

Once $\vec{c}$ has been obtained from the VGM it can be used to compute a new $\boldsymbol{u}^h$, which can then be used to repeat the VGM. This produces an iterative procedure for optimizing $\vec{c}$ [5]. In this paper the Germano procedure was only performed once per time step for unsteady equations. For steady equations the procedure was repeated until $\vec{c}$ no longer changed per iteration, or 100 iterations were exceeded.

3.1 Least-squares Germano identity

In the case where $\boldsymbol{w}^{h_i} = \sum_j \boldsymbol{\phi}_j^{h_i}$, where $\boldsymbol{\phi}_j^{h_i} \in \mathcal{V}^{h_i}$ are local basis functions, (6) can be split into local components as in (7).

$$\boldsymbol{r}_j^i := B(\boldsymbol{\phi}_j^{h_i}, \boldsymbol{u}^{h_i}) + M(\boldsymbol{\phi}_j^{h_i}, \boldsymbol{u}^{h_i}; \vec{c}, \boldsymbol{f}, h_i) - (\boldsymbol{\phi}_j^{h_i}, \boldsymbol{f}) \tag{7}$$

For each $\mathcal{V}^{h_i}$ in (6) the residual of the Germano identity can then be assembled as $\sum_j \boldsymbol{r}_j^i$. However, it is possible for the $\boldsymbol{r}_j^i$ to be individually large while their sum may be small due to differing signs. This would make it seem that the chosen $\vec{c}$ satisfies (6) when it does not. The solution of (6) can then result in no further change for $\vec{c}$. Furthermore, numerical floating point errors may occur if the $\boldsymbol{r}_i^j$ have differing signs and very different magnitudes. These issues can be resolved by using the least-squares form of the Germano identity as proposed by Oberai and Wang in [1]. Instead of assembling individual equations per $\mathcal{V}^{h_i}$, a least-squares residual, R_G, is constructed from all the $\boldsymbol{r}_j^i$:

$$R_G := \sum_{i=1}^{N} \sum_j (\boldsymbol{r}_j^i \cdot \boldsymbol{r}_j^i) \tag{8}$$

$\vec{c}$ is found by determining values that minimize R_G.

In this paper a further modification of the least-squares Germano identity is proposed. When the PDE problem being solved consists of a system of PDEs, $\boldsymbol{r}_j^i$ will be a vector with one component per equation, i.e. $\boldsymbol{r}_j^i := [r_{1,j}^i, r_{2,j}^i, ..., r_{N_{eq},j}^i]$ where N_{eq} is the number of equations. In this case a separate least-squares residual can be assembled per equation:

$$R_{k,G} := \sum_{i=1}^{N} \sum_j (r_{k,j}^i)^2 \tag{9}$$

$\vec{c}$ is then found by determining values that minimize all, or some of the $R_{k,G}$.

3.2 Newton-Raphson algorithm

When $\boldsymbol{\tau}$, and hence $M(\boldsymbol{w}^{h_i}, \boldsymbol{u}^{h_i}; \vec{c}, \boldsymbol{f}, h_i)$, depends nonlinearly on $\vec{c}$, it may not be possible to derive an analytical expression for $\vec{c}$ from (6). However, a numerical procedure can then be used to solve the system of equations. In this paper the Newton-Raphson algorithm is proposed.

The global Germano identity on $\mathcal{V}^{h_i}$ is now denoted by $G^{h_i} := B(\boldsymbol{w}^{h_i}, \boldsymbol{u}^{h_i}) + M(\boldsymbol{w}^{h_i}, \boldsymbol{u}^{h_i}; \vec{c}, \boldsymbol{f}, h_i) - (\boldsymbol{w}^{h_i}, \boldsymbol{f})$. The system of Germano identities in (6) can then be represented as the vector $\boldsymbol{G} := [G^{h_1}, ..., G^{h_N}]^T$. (6) is then equivalent to solving $\boldsymbol{G} = 0$. Taking a first order Taylor expansion of $\boldsymbol{G} = 0$ produces:

$$\boldsymbol{G}(\vec{c}) \approx \boldsymbol{G}(\vec{c}_0) + \boldsymbol{J}(\vec{c}_0)(\vec{c} - \vec{c}_0) = 0, \ \boldsymbol{J} = \begin{pmatrix} \frac{\partial G^{h_1}}{\partial c_1} & \cdots & \frac{\partial G^{h_1}}{\partial c_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial G^{h_N}}{\partial c_1} & \cdots & \frac{\partial G^{h_N}}{\partial c_N} \end{pmatrix} \tag{10}$$

where $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{G}$ with respect to $\vec{c}$. In this paper exact algebraic expressions for the entires of $\boldsymbol{J}$ are used. Replacing $\vec{c}$ by $\vec{c}_{n+1}$ and $\vec{c}_0$ by $\vec{c}_n$, where n indicates the nth iteration, allows an iterative procedure for solving $\boldsymbol{G} = 0$ to be derived:

$$\vec{c}_{n+1} = \vec{c}_n - \boldsymbol{J}^{-1}(\vec{c}_n)\boldsymbol{G}(\vec{c}_n) \tag{11}$$

Here (11) is stopped when $||\boldsymbol{G}||_{L_2} < 1 \cdot 10^{-10}$.

3.3 BFGS Algorithm

The least-squares formulation of the Germano identity requires the minimization of R_G, or $R_{k,G}$ to solve for $\vec{c}$. If a general least-squares residual R is considered, then R is minimized when $\nabla R = 0$. An efficient procedure for solving $\nabla R = 0$ is the BFGS algorithm. In this procedure, ∇R is approximated with a first order Taylor expansion:

$$\nabla R(\vec{c}) \approx \nabla R(\vec{c}_0) + B(\vec{c}_0)(\vec{c} - \vec{c}_0) = 0 \tag{12}$$

where B is the hessian of R with respect to $\vec{c}$. Again replacing $\vec{c}$ by $\vec{c}_{n+1}$ and $\vec{c}_0$ by $\vec{c}_n$, the following iterative procedure is produced:

$$\vec{c}_{n+1} = \vec{c}_n - \alpha_n B_n^{-1} \nabla R(\vec{c}_n) \tag{13}$$

α_n is a parameter that controls the step length and is obtained in this paper from an inexact line search. B_n is an approximation to the Hessian at step n. B_n is obtained by first setting $B_0 = I$, the identity matrix, and then updated to B_{n+1} according to:

$$\begin{aligned} B_{n+1} &= B_n + \frac{y_n y_n^T}{y_n^T s_n} - \frac{B_n s_n s_n^T B_n}{s_n^T B_n s_n} \\ s_n &= \alpha_n(\vec{c}_{n+1} - \vec{c}_n) \\ y_n &= \nabla R(\vec{c}_{n+1}) - \nabla R(\vec{c}_n) \end{aligned} \tag{14}$$

To complete the algorithm it is noted that ∇R is evaluated using forward finite differences:

$$\frac{\partial R}{\partial c_i} \approx \frac{R(c_i+\epsilon)-R(c_i)}{\epsilon} \tag{15}$$

with $\epsilon = 1 \cdot 10^{-5}$. The algorithm is stopped when $||\nabla R||_{L_2} < 1 \cdot 10^{-7}$.

4 APPLICATION 1: BURGERS' EQUATION

In this section the VGM, as defined by (6), will be used to optimize subgrid-scale models for the one-dimensional unsteady forced Burgers' equation. The problem is defined as:

$$\begin{aligned}
&\text{find } u(x,t) \in \mathcal{V}: \\
&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} = f, \text{ on } \Omega\times]0,T[\\
&u = 0, \text{ on } \delta\Omega \\
&u(x,0) = 0
\end{aligned} \tag{16}$$

where $\Omega :=]0,1[$ is the unit line, $\delta\Omega$ is the boundary of Ω, T is the final time and Re is a dimensionless parameter. $T = 25$ and $Re = 512$ are used here. The forcing function f is defined to be a spatially and temporally varying sine wave with an additional constant term:

$$f := 10\sin(t)\sin(2\pi x) + 11 \tag{17}$$

this ensures that u will form a sharp layer near the right boundary of the domain.

Defining $\mathcal{C}^h$ as the partition of Ω into a grid of elements with size h, $\mathcal{V}^h$ can then be defined as the space of standard linear finite element functions on $\mathcal{C}^h$. Application of the VMM, as described in Section 2, to (16) then results in the following variational problem:

$$\begin{aligned}
&\text{find } u^h \in \mathcal{V}^h: \\
&B(w^h,u^h) + M(w^h,u^h;\vec{c},f,h) = (w^h,f), \ \forall w^h \in \mathcal{V}^h \\
&u^h = 0, \text{ on } \delta\Omega \\
&u^h(x,0) = 0 \\
&B(w^h,u^h) := \left(w^h, \frac{\partial u^h}{\partial t}\right) - \frac{1}{2}\left(\frac{\partial w^h}{\partial x}, (u^h)^2\right) + \frac{1}{Re}\left(\frac{\partial w^h}{\partial x}, \frac{\partial u^h}{\partial x}\right) \\
&M(w^h,u^h;\vec{c},f,h) := \left(\frac{\partial w^h}{\partial x}, u^h\tau\mathcal{R}\right) \\
&\mathcal{R} := \frac{\partial u^h}{\partial t} + u^h\frac{\partial u^h}{\partial x} - f
\end{aligned} \tag{18}$$

where second order products of u' have been neglected and all second order spatial derivatives vanish due to the use of linear elements. Once τ is defined, (18) is integrated in time using the generalized-α method as proposed by Jansen et al in [9]. In this paper the time-step $\Delta t = 0.25$ is used.

The definition of τ has a large influence on the quality of the results obtained from (18). For nonlinear advective-diffusive equations such as Burgers' equation, it is known that τ must be a function of h [10, 11, 12]. Therefore the following definitions for τ are proposed, with coefficients c_n to be optimized with the VGM:

$$\tau_{quadratic} := c_1 h + c_2 h^2 \tag{19}$$

$$\tau_{cubic} := c_1 h + c_2 h^2 + c_3 h^3 \tag{20}$$

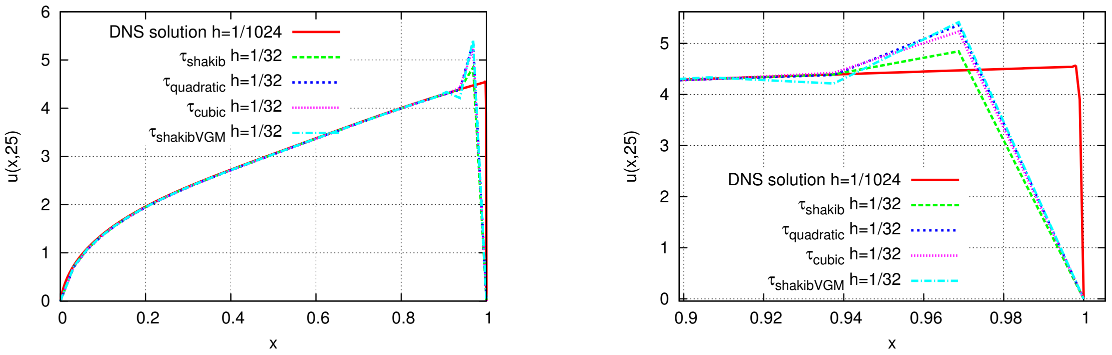
$$\tau_{shakibVGM} := \left(\frac{4}{\Delta t^2} + c_1 \left(\frac{u}{h}\right)^2 + 9 \left(\frac{4}{Reh^2}\right)^2 \right)^{-\frac{1}{2}} \tag{21}$$

For the VGM the coarse functions spaces $\mathcal{V}^{h_i}$ are defined as the standard linear finite element functions on $\mathcal{C}^{h_i}$, where $\mathcal{C}^{h_i}$ denotes the partition of Ω into a grid with elements of size $2^i h$. The VGM optimization is started after five time steps have been performed with the initial coefficient values. As a benchmark, the definition of τ as proposed by Shakib in [10] is used.

$$\tau_{shakib} := \left(\frac{4}{\Delta t^2} + 4 \left(\frac{u}{h}\right)^2 + 9 \left(\frac{4}{Reh^2}\right)^2 \right)^{-\frac{1}{2}} \tag{22}$$

4.1 Numerical results

Now numerical results are presented for the forced Burgers' equation. For each definition of τ used, results are computed on uniform grids with size $h = 1/32, 1/64, 1/128$ and $h = 1/256$. For comparison a simulation with $\tau = 0$ on a grid with size $h = 1/1024$ is also performed and is referred to as the Direct Numerical Simulation (DNS) solution. The instantaneous DNS solution and the solutions with $h = 1/32$, at the final time T are shown in Figure 1. It is evident that the main differences between the solutions obtained with the different definitions of τ, occur near the right boundary $x = 1$. The solutions where τ has been optimized using the VGM appear to be slightly more oscillatory than that obtained with τ_{shakib}. A graph of the space-time L_2 error, with respect to the DNS solution, is shown for each τ definition in Figure 2a. The errors produced with each τ are close together, indicating the ability of the VGM to identify suitable subgrid-scale model parameters. In particular the error obtained when using $\tau_{shakibVGM}$ is lower than when using the other τs including τ_{shakib}. This demonstrates the potential for improved performance by optimizing nonlinear τs with the VGM. In Figure 2b the values of the coefficient vector $\vec{c}$ are shown for each time step, for results with $h = 1/32$. In updating the coefficients each time step the Newton Algorithm converged within two iterations for $\tau_{quadratic}$ and τ_{cubic} and three to four iterations for $\tau_{shakibVGM}$. Note that the value of c_1 for



(a) Instantaneous solutions full domain

(b) Instantaneous solutions near $x = 1$

Figure 1: Instantaneous DNS solution and solutions with grid size $h = 1/32$ for the forced Burgers' equation at final time $T = 25$.

$\tau_{shakibVGM}$ is more than twice the standard value used in τ_{Shakib}. Finally the ratio of time taken for each simulation with and without the VGM is presented in Figure 2c. As can be seen the extra computation time increases as more coefficients need to be optimized. The computation time ratio also increases as the number of elements increases. However, for the simulations conducted in Figure 2c, the proposed numerical method for the VGM does not result in a substantial increase in required computation time.

5 APPLICATION 2: STOKES EQUATIONS

In this section the steady two-dimensional Stokes equations are considered:

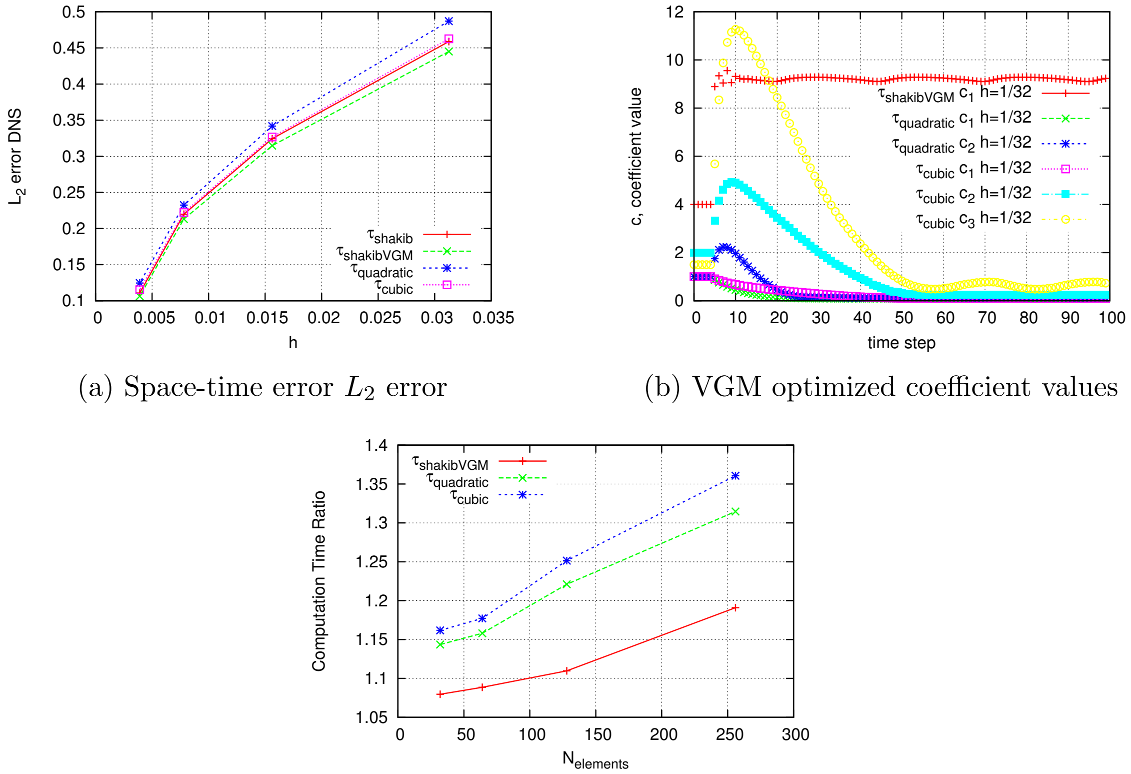
$$\begin{aligned}
&-\nabla\cdot(2\nu\nabla^s\boldsymbol{u})+\nabla p=\boldsymbol{f},\ \text{on }\Omega\\
&\nabla\cdot\boldsymbol{u}=0,\ \text{on }\Omega\\
&\boldsymbol{u}=0,\ \text{on }\delta\Omega,\quad\int_\Omega p\,\mathrm{d}\Omega=0
\end{aligned}\tag{23}$$

where $\Omega :=]0,1[\times]0,1[$ is the unit-square and $\nu\in\mathbb{R}$ is the viscosity parameter. $\nabla^s\boldsymbol{u} := 1/2(\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^T)$ denotes the symmetric velocity gradient.

The VMM is applied to (23), where both $\boldsymbol{u}$ and p contain resolved and unresolved scales such that $\boldsymbol{u}=\boldsymbol{u}^h+\boldsymbol{u}'$ and $p=p^h+p'$. The unresolved scales are then defined as:

$$\begin{aligned}
&\boldsymbol{u}' := -\tau_M\boldsymbol{R}_m,\quad \boldsymbol{R}_m := \nabla p^h-\boldsymbol{f}\\
&p' := -\tau_C R_c,\quad R_c := \nabla\cdot\boldsymbol{u}
\end{aligned}\tag{24}$$

Defining $\mathcal{V}^h$ and $\mathcal{Q}^h$ as spaces of standard bilinear finite element basis functions with $v\in\mathcal{V}^h : v=0$ on $\delta\Omega$ and $v\in\mathcal{Q}^h : \int_\Omega v\,\mathrm{d}\Omega=0$, results in the following variational



(a) Space-time error L_2 error

(b) VGM optimized coefficient values

(c) Computation time ratio when using VGM

Figure 2: Further results for Burgers' equation: L_2 error with respect to DNS solution. Time history of VGM optimized coefficients on grids with $h = 1/32$. Ratio of time taken per simulation when using VGM compared to without

problem formulation:

$$
\begin{aligned}
&\text{find } \boldsymbol{u}^h \in \mathcal{V}^h \times \mathcal{V}^h, p^h \in \mathcal{Q}^h : \\
&(\nabla^s \boldsymbol{w}^h, 2\nu \nabla^s \boldsymbol{u}^h) - (\nabla \cdot \boldsymbol{w}^h, p^h) + (q^h, \nabla \cdot \boldsymbol{u}^h) + (\nabla \cdot \boldsymbol{w}^h, \tau_C \nabla \cdot \boldsymbol{u}^h) \\
&+ (\boldsymbol{\nabla} q^h, \tau_M(\boldsymbol{\nabla} p^h - \boldsymbol{f})) = (\boldsymbol{w}^h, \boldsymbol{f}),\ \forall\ \boldsymbol{w}^h \in \mathcal{V}^h \times \mathcal{V}^h, q^h \in \mathcal{Q}^h
\end{aligned}
\tag{25}
$$

For the VGM optimization of τ_M and τ_C the coarse spaces are defined in a manner similar to the procedure used for the forced Burgers' equation, where each coarse space Ω is partitioned by $\mathcal{C}^{h_i}$ into elements with edge length $2^i h$. The least-squares version of the Germano identity is used. In this case the Germano residuals are split into residuals of the momentum and continuity equations. If the x-momentum, y-momentum and continuity equations are given the indexes 1, 2 and 3 respectively, then τ_M is optimized by minimizing $R_{3,G}$ and τ_C is optimized by minimizing $R_{1,G} + R_{2,G}$, where $R_{k,G}$ is defined by (9).

Two definitions for τ_M and τ_C are selected from literature, with one additional definition

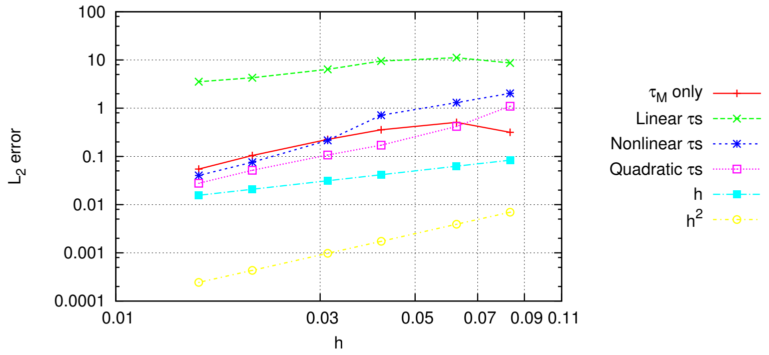


Figure 3: L_2 error convergence with h for VGM optimized Stokes equations

proposed in this paper:

$$\tau_M := c_1 \frac{h^2}{24\sqrt{2}\nu}, \quad \tau_C := c_2 \nu \tag{26}$$

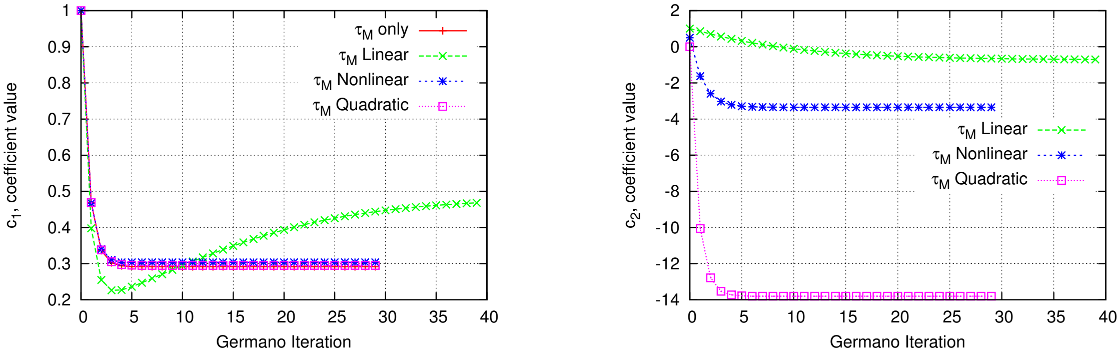
$$\tau_M := c_1 \frac{h^2}{24\sqrt{2}\nu}, \quad \tau_C := \frac{h\sqrt{\boldsymbol{u}\cdot\boldsymbol{u}}}{4} \tag{27}$$

$$\tau_M := c_1 \frac{h^2}{24\sqrt{2}\nu}, \quad \tau_C := c_2 h^2 \tag{28}$$

where (26) was proposed by Franca and Hughes in [13], (23) was proposed by Taylor et al in [14] and (28) is the definition proposed here. (26), (27) and (28) are referred to as the linear τs, nonlinear τs and quadratic τs from this point onwards. An additional case is defined by setting $\tau_C = 0$, which will be referred to as "τ_M only".

5.1 Numerical results

Numerical results are computed for the Stokes equations on grids with sizes: $h = 1/12, 1/16, 1/24, 1/32, 1/48$ and $h = 1/64$. The forcing function $\boldsymbol{f}$ is defined by inserting a manufactured solution of $\sin(4\pi x)\sin(4\pi y)$ for each solution variable. The manufactured solution can also be used for computing the L_2 error of each result. A graph of the L_2 error for the different mesh sizes and τ definitions is shown in Figure 3. It is directly evident that the different definitions of τ can result in very different error and convergence properties. The linear τs have the highest error and first-order convergence, whereas the other τs have lower errors and have approximately second-order convergence. The nonlinear τs perform better than the linear τs, but not as well as the quadratic τs. This indicates that including a nonlinear dependence in τ may lead to improved performance, but that a correct parametrization of τ with h remains important. Figure 4 shows the values of c_1 and c_2 obtained after each Germano iteration, for each τ definition, for the case $h = 1/24$. All τ definitions converge within five iterations, except for the linear τs which take approximately 35 iterations to converge. Additionally, the BFGS algorithm converged to the specified tolerance for all τs. This is evidence that the BFGS algorithm is a suitable method for minimizing the least-squares Germano residuals.



(a) c_1 coefficient value

(b) c_2 coefficient value

Figure 4: Values of c_1 and c_2 obtained with VGM iterations $h = 1/24$

6 CONCLUSION

This paper demonstrated how the Newton and BFGS algorithm could be used to solve the standard VGM relations and least-squares formulation respectively, for arbitrary forms of the τ parameter, including nonlinear τs, appearing in the VMM. When applied to the forced unsteady Burgers' and Stokes equations, both algorithms always converged within the alloted amount of iterations. This indicates that the proposed framework for solving the VGM relations numerically is a promising approach for determining the coefficients of general SGS models arising from the VMM.

The numerical results further showed that the inclusion of nonlinearities in the SGS model can lead to improved performance. However for the Stokes equations it was possible to construct a τ that depended only linearly on its coefficients that outperformed the proposed nonlinear τs. This indicates the strong dependence of the combined VGM and VMM approach on the parametrization of τ.

REFERENCES

[1] A. A. Oberai and Z. Wang, "Optimal numerical solutions of PDEs using the variational Germano identity," *Computer methods in applied mechanics and engineering*, vol. 197, pp. 2948–2962, 2008.

[2] A. A. Oberai and D. Sondak, "Application of the variational Germano identity to the variational multiscale formulation," *Communications in numerical methods in engineering*, vol. 27, pp. 335–344, 2011.

[3] T. J. R. Hughes, "Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods," *Computer methods in applied mechanics and engineering*, vol. 127, pp. 387–401, 1995.

[4] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J. Quincy, "The variational multiscale method - a paradigm for computational mechanics," *Computer methods in applied mechanics and engineering*, vol. 166, pp. 3–24, 1998.

[5] A. A. Oberai and J. Wanderer, "A dynamic approach for evaluating parameters in a numerical method," *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 50–71, 2005.

[6] V. Gravemeier, *The variational multiscale method for laminar and turbulent incompressible flow.* PhD thesis, Institut für Baustatik der Universität Stuttgart, 2003.

[7] V. M. Calo, *Residual-based multiscale turbulence modeling: finite volume simulations of bypass transition.* PhD thesis, Stanford University, 2004.

[8] G. Scovazzi, *Multiscale methods in science and engineering.* PhD thesis, Stanford university, 2004.

[9] K. E. Jansen, C. Whiting, and G. M. Hulbert, "A generalized-alpha method for integrating the filtered Navier-Stokes equations with a stabilized finite element method," 2000.

[10] F. Shakib, *Finite element analysis of the compressible Euler and Navier-Stokes equations.* PhD thesis, Stanford University, 1988.

[11] T. J. R. Hughes and G. Sangalli, "Variational multiscale analysis: the fine-scale green's function, projection, optimization, localization, and stabilized methods," *SIAM Journal of numerical analysis*, vol. 45, pp. 539–557, 2007.

[12] I. Akkerman, K. G. van der Zee, and S. J. Hulshoff, "A variational Germano approach for stabilized finite element methods," *Computer methods in applied mechanics and engineering*, vol. 199, pp. 502–513, 2010.

[13] L. P. Franca and T. J. R. Hughes, "Convergence analyses of Galerking least-squares methods for symmetric advective-diffusive forms of the Stokes and incompressible Navier-Stokes equations," *Computer methods in applied mechanics and engineering*, vol. 105, pp. 285–298, 1993.

[14] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins, "Finite element modeling of blood flow in arteries," *Computer methods in applied mechanics and engineering*, vol. 158, pp. 155–196, 1998.