Development of a regularization term in a TLES code in OpenFOAM

Samuel Maloney

Computational Science and Engineering Master

Seminar in Fluid Dynamics for CSE HS 2017

Institute of Fluid Dynamics
ETH Zürich

Supervisor: Daniel Oberle

Professor: Patrick Jenny

Abstract

Temporal large-eddy simulation (TLES) is a promising method to reduce computational costs compared to DNS for turbulent flows, and offers several potential advantages over conventional spatial LES. A TLES method based termed the temporal exact deconvolution model (TEDM) has previously been developed in OpenFAOM but found to be unstable for large filter widths. In this project, a regularization term was designed, implemented, and tested to investigate its potential for stabilization of the method. Concurrently, a divergence cleaning method using the projection scheme was also added to the solver to test if non-zero divergence might be a source of the instabilities. Demonstration of the base method and the new stabilization strategies was realized using the Pitz-Daily backwards-facing step geometry provided in OpenFOAM, and a variety of different parameter combinations and model configurations were analyzed for both stability and for deviation from a stable baseline simulation.

Contents

1 Introduction

Achieving accurate simulations of turbulent fluid flows is a challenging computational task, due to the large range of different length scales that must be resolved for proper representation of the physical processes in such high Reynolds Number (Re) flows. It has been shown that the ratio between the largest and smallest spatial features (at the Kolmogorov length scale) in a turbulent flow varies as $Re^{3/4}$ [1]. Since turbulence is an inherently 3-dimensional phenomenon, this means that full Direct Numerical Simulation (DNS) of turbulent flows have a computational cost which scales as Re^3 for the three spatial dimensions and time. This scaling imposes a very strong limitation on the magnitudes of Re which can be handled by DNS, and therefore various models for turbulence which are less computationally intensive have been developed.

Large Eddy Simulations (LES) are one such class of methods, where only the large scale features of the flow are resolved directly, thus allowing for a coarser mesh to be used, while the effects of smaller scale flow features are modelled. Low-pass filtering of the velocity fields is used to damp out the high wave number oscillations that are unresolved by the coarse mesh, and applying such a filtering operations to the momentum equation results in a closure problem for the residual stresses of the unresolved subfilter scale (SFS) flows. To date, most LES type simulations have been performed by filtering in the spatial-domain, and then using closures such as Smagorinsky eddy-viscosity models [2], deconvolution methods [3], and dynamic modelling [4].

More recently, Pruett [5] developed an LES type scheme using filtering in the time, rather than the spatial, domain. This new technique, termed Temporal LES (TLES), is based on the idea that removing high-frequency temporal fluctuations of the flow will correspondlingly remove high wavenumber spatial features and thus still allow for the use of coarser resolutions for simulation. Several potential advantages of TLES over conventional LES are elucidated by Pruett [1] and briefly summarized here.

First, because methods based on the Reynolds Averaged Navier-Stokes (RANS) equations also utilize time-domain filtering, the linkage between RANS and TLES is more natural than for spatial LES. Second, as most experimental data of turbulence are acquired in the time-domain, it might also be more natural to carry out computations in the time-domain. Third, "differentiation-operator/filter-operator commutation error is problematic for spatial filtering on finite domains or highly stretched grids."[1] Fourth, the filter width for LES should be significantly larger than the grid spacing for proper separation of resolved and unresolved scales, but this is often not possible in practice, while for TLES the filter width is a theoretically tunable parameter from zero to infinity. Finally, TLES is much more amenable to time-dependent point sources which are sometimes used for manipulation of engineering problems.

A brief overview of some of the basic theory underpinning TLES is presented next, adapting primarily from the work done by Pruett [1] and Jenny [6].

1.1 TLES Theory

Since for TLES filtering is done during the course of the numerical experiments, the filtering operation must be *causal*, ie. it must depend only upon the current and previous values of the quantity being filtered and not on any future values. Letting an overbar denote a time-filtered quantity and T denote the characteristic filter width, one obtains the following causal time-filtering operation for some filter kernal $G(t, T)$:

$$\bar{f}(t, T) = \int_{-\infty}^{t} G(\tau - t; T) f(\tau) \, \mathrm{d}\tau \tag{1.1}$$

For this project only the exponential filter was used, namely:

$$\begin{aligned} G(t; T) &= \frac{\exp(t/T)}{T} \\ \bar{f}(t, T) &= \frac{1}{T} \int_{-\infty}^{t} \exp\left(\frac{\tau - t}{T}\right) f(\tau) \, \mathrm{d}\tau \end{aligned} \tag{1.2}$$

which is a second order low-pass filter and has the transfer function:

$$H(\omega, T) = \frac{1}{1 + iT\omega} \tag{1.3}$$

Since the integral formulation would require storage of the quanitity at all previous time points in the simulation, the following equivalent differential form is used for implementation, as it can be integrated using standard time-marching schemes to update the filtered quantities at each step (where the explicit time-dependence of the quanitites is now dropped for convenience):

$$\frac{\partial \bar{f}}{\partial t} = \frac{f - \bar{f}}{T} \tag{1.4}$$

Applying such a filtering operation to the incompressible Navier-Stokes equations gives the following Temporally-Filtered Navier-Stokes (TFNS) system for the evolution of the temporally-filtered velocity fields, using the same initial condition as for the unfiltered fields $\bar{u}(0, x; T) = u(0, x)$:

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \tag{1.5}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial(\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{1.6}$$

where u is the fluid velocity, p is the pressure, ν is the kinematic pressure, and subscripts $i, j \in \{1, 2, 3\}$ indicate the 3D cartesian direction. The element τ_{ij} in the final term is referred to as the *temporal stress tensor* and is defined as:

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j \tag{1.7}$$

Solving the temporally-filtered Navier-Stokes equations requires a residual-stress model to handle the unkown value $\overline{u_i u_j}$ and close the system. Stolz and Adams [7] published an approximate deconvolution model (ADM) for for spatial LES which was then adapted for TLES by Pruett [1]. For this project, however, a new method termed the Temporal Exact Deconvolution Model (TEDM) developed by Jenny [6] is used to provide the closure.

For the TEDM method, it is observed that by inserting the velocity field into the differential form of the filter given in Eq. (1.4) the unfiltered field can be recovered as follows:

$$u_i = \bar{u}_i + T\frac{\partial \bar{u}_i}{\partial t} \tag{1.8}$$

The same Eq. (1.4) can then be applied to the unknown quantity $\overline{u_i u_j}$ and the just obtained relation for the unfiltered field in Eq. (1.8) can be inserted in the resulting expression to obtain the following:

$$\begin{aligned}\frac{\partial \overline{u_i u_j}}{\partial t} &= \frac{u_i u_j - \overline{u_i u_j}}{T} \\ &= \frac{\left(\bar{u}_i + T\frac{\partial \bar{u}_i}{\partial t}\right)\left(\bar{u}_j + T\frac{\partial \bar{u}_j}{\partial t}\right) - \overline{u_i u_j}}{T} \\ &= \frac{\bar{u}_i \bar{u}_j - \overline{u_i u_j}}{T} + \bar{u}_i\frac{\partial \bar{u}_j}{\partial t} + \bar{u}_j\frac{\partial \bar{u}_i}{\partial t} + T\frac{\partial \bar{u}_i}{\partial t}\frac{\partial \bar{u}_j}{\partial t}\end{aligned} \tag{1.9}$$

This can then be rearranged to produce an equation for the time evolution of the temporal stress tensor τ_{ij} containing only the known filtered velocities:

$$\frac{\partial \tau_{ij}}{\partial t} = -\frac{\tau_{ij}}{T} + T\frac{\partial \bar{u}_i}{\partial t}\frac{\partial \bar{u}_j}{\partial t} \tag{1.10}$$

Suitable time integration schemes can then be used to solve Eqs. (1.6) and (1.10) in alternation to evolve the system in time.

1.2 Regularization

In order to sufficiently damp the higher frequency oscillations and allow the use of coarser grids for simulations, the filter width T must be made large enough, often several orders of magnitude above the timestep. However, for such large filter widths the method becomes highly non-dissipative, which leads to instability in the numerical evolution. For this reason, a regularization term based on work by Åkervik et al. [8] was investigated as a potential means to improve the stability of the method, and is briefly outlined here.

The form of the regularization term is that of a proportional feedback control mechanism, well known in control theory. It manifests itself in the TFNS system Eq. (1.6) as an additional linear term on the right hand side:

$$-\chi(\bar{u} - \tilde{u}) \tag{1.11}$$

where χ is the control coefficient for the strength of the regularization effect and $\tilde{u}$ represents the velocity field after being temporally filtered using a filter width $\tilde{T}$ which is larger than the original filter width T used to compute $\bar{u}$. This $\tilde{T} > T$ relation means that $\tilde{u}$ should contain only lower frequency variations compared to $\bar{u}$ and therefore provides a slower varying target solution towards whcih $\bar{u}$ is forced, damping out high frequency changes in its value and, hopefully, also in the growth of instabilities.

Both χ and $\tilde{T}$ are free parameters that can be tuned to try and stabliize a given system. It is noted that larger values of χ would be expected to slow down the evolution of the system and it is therefore desirable to set χ close to the minimal value for which the simulation is stable, in order to eliminate unecessary increases in the time required for the system to evolve.

1.3 Divergence Cleaning

An implicit condition of the TFNS system given in Eq. (1.5) is that the divergence of the velocity field must be zero. However, numerical discretization schemes generally lead to non-zero values of the divergence which can be a potential source of instability during simulations. As well, since Eq. (1.5) is not explicitly included in the standard TLES algorithm of alternating solutions of Eqs. (1.6) and (1.10), any non-zero divergence resulting from each numerical timestep is simply allowed to accrue in the system. To counter this, explicit divergence cleaning (DC) using the projection scheme, following the procedure outlined in section 5 of the work by Tóth [9] for DC of magnetic fields, was also considered for possible stabilizing effects.

The mathematical basis for the projection scheme lies in the ability to decompose any vector field into the sum of a curl and a gradient according to the Helmholtz-Hodge decomposition:

$$\bar{u}^* = \nabla \times A + \nabla \phi \tag{1.12}$$

where $\bar{u}^*$ is the filtered velocity field computed by advancing the numerical discretization of Eq. (1.6) at the current timestep, A is a vector potential containing the phsyically menaingful (ie. divergence free) component of $\bar{u}^*$, and ϕ is a scalar field contatining the non-zero divergence that is to be removed. Taking the divergence of both sides gives:

$$\nabla^2 \phi = \nabla \cdot \bar{u}^* \tag{1.13}$$

which is a Poisson equation for the non-physical component ϕ. Therefore, our TLES method can be augmented with DC by solving this Poisson equation during each timestep immediately after computing $\bar{u}^*$ and then correcting the filtered velocity field as:

$$\bar{u} = \bar{u}^* - \nabla \phi \tag{1.14}$$

It is noted that while the projection scheme is able to zero out the divergence to machine precision in the given discretization scheme, in general the divergence will still be non-zero in any other discretization. Therefore, it is usually sufficient to make the divergence merely *close* to zero in the given discretization, and save the computational effort required for a very precise solution of the Poisson problem. This is easily achieved in practice by using iterative solution methods and setting the tolerance to a much less stringent condition than is used for solving the other equations in the TLES method.

2 Implementation

For this project, all implementation of the regularization term and divergence cleaning scheme was carried out in C++ using the open source OpenFOAM CFD code. A solver developed during a previous project at the institute, which implemented the fundamental TLES methodology, was used as a starting code. This solver already contained the necessary code for discretizing and solving Eq. (1.6) in the TFNS system for the filtered velocity, Eq. (1.10) for the temporal stress tensor, as well as computing the deconvoluted velocity field according to the TEDM and the mean filtered velocity field for statistical comparison and validation of the methods.

For a numerical discretization scheme using a timestep Δt, the filter width T is naturally parameterized by defining a filter width ratio r as:

$$r = \frac{T}{\Delta t} \tag{2.1}$$

A similar quanitity can also be defined for the filter used in generating the regularization quantity $\tilde{u}$ by defining a regularization filter width ratio $\tilde{r}$ as:

$$\tilde{r} = \frac{\tilde{T}}{T} \tag{2.2}$$

This definition of the ratios implies the restriction $\tilde{r} > 1$ for logical consistency with the regularization theory, while in theory any $r > 0$ is permissible. However, as stated in section 1.1, it is deisrable to have $T > \Delta t$ and so $r > 1$ can be considered as a practical limitation for the method to be of much use.

As suggested in section 1.3, the DC method can be implemented in a computationally efficient manner quite easily in OpenFOAM by specifying an iterative solver for the Poisson Eq. (1.13) in the 'fvSolution' file with a relatively non-stringent tolerance. For this project, the same solver type was used as for the regular filtered velocity field, but the tolerance was relaxed from 10^{-5} to 10^{-2} as this did not noticeably affect the size of the increased range of stability, but did greatly reduce the observed computational time.

3 Results and Discussion

All simulations were carried out on the standard Pitz-Daily backward facing step geometry provided with OpenFOAM. The kinematic viscosity of $\nu = 10^{-5}\,\mathrm{m^2/s}$ and the inlet velocity of $u = 10\,\mathrm{m/s}$ were kept constant for all runs. Using this inlet velocity and the channel width $L = 50.8\,\mathrm{mm}$, the Reynolds number of the flow can be approximated as:

$$Re = \frac{uL}{\nu} = 50800$$

In order to provide a somewhat objective measure of comparison between the results of the method using different sets of parameters, a vertical line profile of the mean magnitude of the unfiltered velocity field was taken at the location shown in Fig. 3.1 after each run. As well, a simulation using a simple direct solution of the Navier-Stokes equations was performed as a baseline. It must be noted that this is not a true DNS comparison, as the simulation was performed on the same grid as used for all of the following TLES testing, and so is not fine enough to resolve the full range of scales of the system in general. Since such a fully-resolved DNS simulation was prohibitively time-consuming for this project, the under-resolved baseline case was considered acceptable to at least check for any large deviations resulting from the methods being tested, and for simplicity is referred to as DNS in the following analysis. The location of the line was chosen to be at the minimum value of the recirculation vortex in the DNS baseline, so as to have a distinct feature to compare with between runs.

To start with, several simulations were run using only the basic TLES implementation, with no regularization or DC. This gave one free parameter to be varied, the filter width ratio r, and the resulting profiles were taken at an end time of $t_{\text{final}} = 0.1$ using a time step of $\Delta t = 10^{-6}$ for all runs. It was observed that the simulation becomes unstable if the filter width is made too large, and so this time step was chosen as it gave a small range of values, up to $r \approx 28$, for which the base method was stable and any effects of varying r in isolation could be observed.

The results of this initial study are shown in Fig. 3.2 and the profiles from the various runs are seen to lie almost completely atop one another. This would seem to suggest that the effect of r is minimal within the range of values which are stable without regularization.

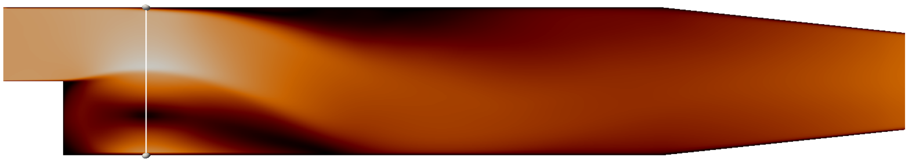

Figure 3.1: Representative image of a $|\bar{u}|_{\text{mean}}$ velocity field with a white line showing the location $x = 28.6\,\mathrm{mm}$ past the step at which the comparison profiles were captured

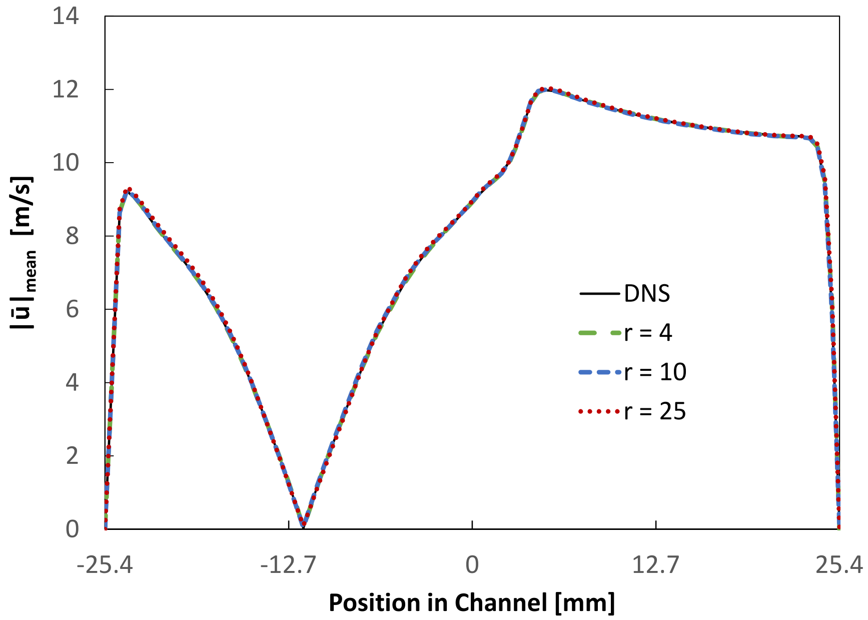

Figure 3.2: Comparison of line profiles from the $|\bar{u}|_{\text{mean}}$ velocity field at $x = 28.6\,\text{mm}$ for various values of filter width r, taken at $t_{\text{final}} = 0.1$ using $\Delta t = 10^{-6}$

3.1 Divergence Cleaning

The effects of adding DC to the method as per the projection method outlined in section 1.3 were investigated next. A small increase in the stable range of the filter width r was observed, with the largest stable value for r increasing from $r = 4.4$ to $r = 4.7$ when $\Delta t = 10^{-5}$, and from $r = 28$ to $r = 29$ for a $\Delta t = 10^{-6}$ time step. Simulations were run with DC included in the simulations for the same already stable values of r as in the previous set as well as for the now stabilized $r = 29$ case, and the results are shown in Fig. 3.3. This allowed for potential comparison between the effects of DC both when it was and was not required to achieve stability in the first instance.

Unlike with the base TLES simulations, all of the runs with DC are seen to be no longer as perfectly concident with the baseline simulation, although the deviation is relatively minor and the variations of r still appear to have little discernible impact on the results. Additionally, the runtime of the simulations increased by only between 5-11% for the values of r with a counterpart simulation not using DC, indicating decent efficiency of the DC implementation using an iterative solver with a relaxed tolerance, as discussed in section 1.3 and chapter 2. Since DC was only able to provide stabilization over a small range of additional filter widths, this would suggest both that non-zero divergence is indeed present in the simulations, since DC did have an effect, but also that it is not itself the cause of the instability, since then DC would have been expected to provide a much greater degree of stabilization. It is therefore posited that the stabilizing effects of DC are possibly due simply to reducing the overall velocity magnitude when removing

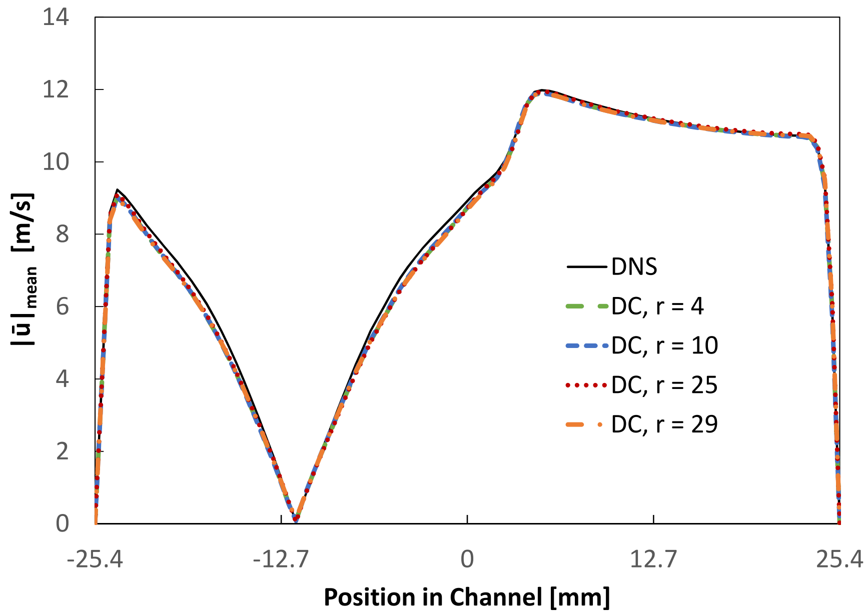

Figure 3.3: Comparison of line profiles from the $|\bar{u}|_{\text{mean}}$ velocity field at $x = 28.6\,\text{mm}$ for various values of filter width r with divergence cleaning, taken at $t_{\text{final}} = 0.1$ using $\Delta t = 10^{-6}$

the non-physical component, which for instabilities whose growth rates were but slightly greater than one was enough to bring them back inside the stable region.

3.2 Regularization

In Fig. 3.4 is shown the result of a run at $r = 30$, using regularization with a control value of $\chi = 32000$ to stabilize the simulation. This value of χ was chosen so as to be slightly larger than the minimal value for which the simulation was found to be stable for $r = 30$, as discussed later in this chapter. It is observed that while the shape of the profile is very similar to the baseline case, the data shown was collected only after propogating the simulation through 87X as many timesteps, to a $t_{\text{final}} = 8.4$ compared to 0.1 for the baseline (and all other profiles shown.) The final time for comparison was chosen to be the time when the minimum of the recirculation vortex in the regularized run reached the same location as in the compared against baseline profile. Such good agreement between the profiles when matching such a recognizable feature as the vortex minimum suggests that the regularization stabilized simulation seems to evolve through the same sequence as without regularization, only at a *much* slower rate.

Two slight modifications of the regularization term in Eq. (1.11) were also briefly investigated. The first used the de-convoluted velocity in place of the filtered velocity as $-\chi(u - \tilde{u})$ but it was found to provide less of a stabilizing influence. In particular, making χ too large caused the simulation to again become unstable, meaning there was a limited range of acceptable χ

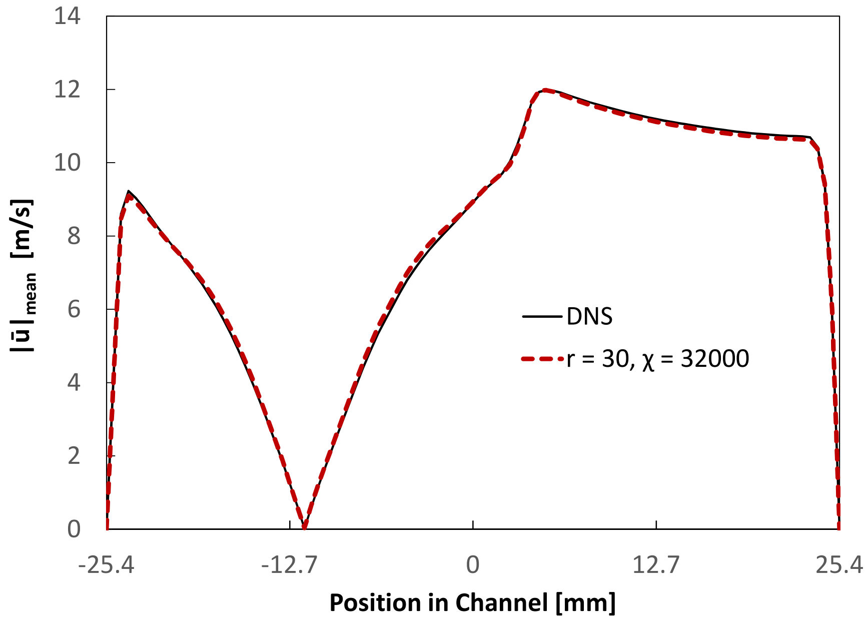

Figure 3.4: Line profile from the $|\bar{u}|_{\text{mean}}$ velocity field at $x = 28.6\,\text{mm}$ for $r = 30$ with regularization. $\chi = 32000$, $\tilde{r} = 100$, $\Delta t = 10^{-6}$, $t_{\text{final}} = 0.1$ for DNS and $t_{\text{final}} = 8.4$ for $r = 30$

which which provided stability, while when using the filtered velocity no such upper bound was encountered during testing. As well, this range of acceptable χ values contracted with increasing filter widths, eventually vanishing entirely for large enough r, a limitation that was also not encountered for the formulation of Eq. (1.11), which was able to stabilize all tested values of r simply through comensurate increase in χ.

The second modification changed the computation of $\tilde{u}$. Instead of filtering the de-convoluted value, the filtered value $\bar{u}$ was simply filtered again with the same condition $\tilde{T} > T$ on the filter width of the second filtering operation, in effect producing a regulariztion term $-\chi(\bar{u} - \tilde{\bar{u}})$. This set-up followed more closely the methodology of Åkervik et al. [8], but was not investigated very deeply due to time constraints, and as it was found to produce almost exactly the same results as Eq. (1.11), no immediate benefits were obvious.

3.2.1 Minimum Stabilizing χ

Since most cases of instability in the simulation manifested within the first few dozen time steps (approx. 30-40 or less), and given the large number of parameter sets that required testing, for the purposes of the following section a simulation was considered ‘stable’ if it ran for at least 100 timesteps without failing. It is noted that a small number of the runs performed crashed even after 80+ timesteps, so this 100 timestep rule is certainly not an ironclad guarantee of stability over a much longer run. However, as in all of these cases it was the point of transition from unstable to stable that was under investigation, it suffices to warn that the reported minimal χ

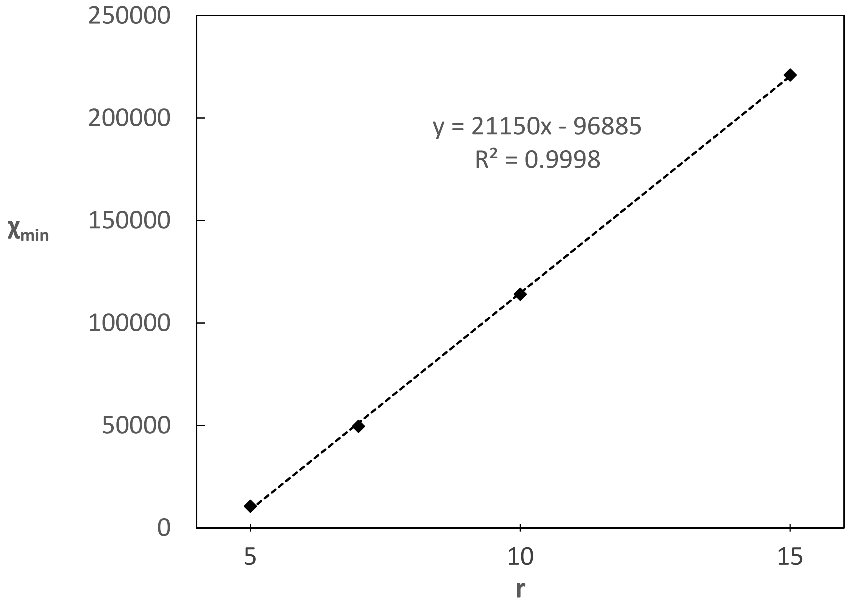

Figure 3.5: Minimum χ values required to stabilize simulation for various values of r, using $\Delta t = 10^{-5}$ and $\tilde{r} = 100$, and showing the linear least squares regression

values required for stability should be regarded as providing only marginal stability, and slightly larger values might be safer in real simulations to provide some margin of safety.

To investigate any potential relationship between the smallest value of χ for which a given set-up was stable and the other simulation parameters (r, $\tilde{r}$, and Δt) three series of simulations were run where r was varied while maintaining $\tilde{r}$ and Δt constant. Results from the first of these series is found in Fig. 3.5 for $\Delta t = 10^{-5}$ and $\tilde{r} = 100$. A clear linear relation is observed between $\chi_{\min}$ and r, with the trendline and equation of the linear least squares regression also shown on the chart, along with the R^2 measure of the goodness of fit (in this case an excellently high 0.9998.) The value of the regression slope is quite large, at just over 21000, indicative of a high degree of regularization required to achieve stability for large filter widths and underscoring the reason for the slow time evolution of the stabilized systems, as Åkervik et al. [8] clearly demonstrate that large values of χ will greatly reduce the convergence rate of the system towards its steady state.

Fig. 3.6 shows the second series of data with the time step still at $\Delta t = 10^{-5}$ but reducing the regularization filter width ratio to $\tilde{r} = 10$. An extremely linear relation is observed, with only a slight increase in the slope and slight decrease in the intercept, and both changes being less than 2.5% compared to the order of magnitude decrease in $\tilde{r}$. This suggests only a weak relation between the required χ and $\tilde{r}$ but would ccertainly require more data points to confirm, particularly for values of $\tilde{r}$ which are very large and those close to the base TLES filter width where $\tilde{r}$ approaches one.

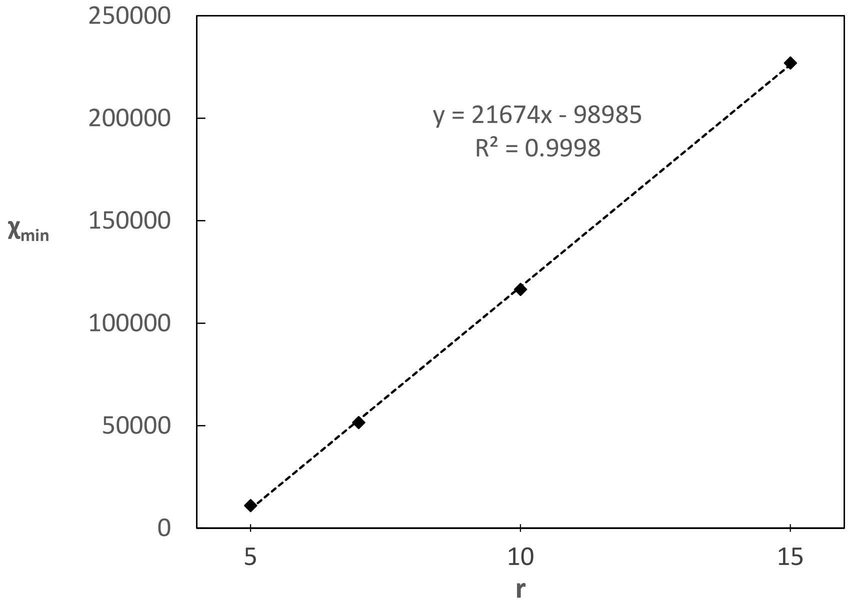

Figure 3.6: Minimum χ values required to stabilize simulation for various values of r, using $\Delta t = 10^{-5}$ and $\tilde{r} = 10$, and showing the linear least squares regression

Data for the third and final series can be found in Fig. 3.7, this time returning the regularization filter width ratio to $\tilde{r} = 100$ but now decreasing the time step to $\Delta t = 10^{-6}$. Once again the trend is linear, with the slope for this case being somewhat lower at 19377, but still only a decrease of 8.4% compared to an order of magnitude decrease in the time step, suggesting that the slope has only a weak dependence on the time step similarly as on $\tilde{r}$. On the other hand, the intercept displays a much stronger dependence on change in Δt, decreasing almost exactly sixfold, but additional data points would again be necessary to try and quantify the exact nature of the dependence relation.

It was observed that when analyzing the data output step-by-step for an unstable simulation, the problematic growth is observed to occur precisely at the sharp corner of the backwards facing step in the geometry. Such an obvious focal point would suggest a likely strong dependence of the stability behaviour for the TLES solver on the specific geometry under consideration. As such, any functional relationship between $\chi_{mathrmmin}$ and the other method parameters would most likely be valid only for the geometry (and probably also the specific mesh) on which the data was measured.

As well, the initial conditions used to initialize the simulation can also play a large role. While not thoroughly investigated by any means, it was found that using the data from the end of the baseline simulation for the starting values of a new simulation allowed extremely large filter widths to be used stably without regularization. For a time step of $\Delta t = 10^{-6}$, filter width ratios of over $r = 1200$ were able to run through 100 steps without failing compared to a maximum of $r = 28$ with a uniform zero initial velocity condition, although a test at $r = 1400$ still crashed

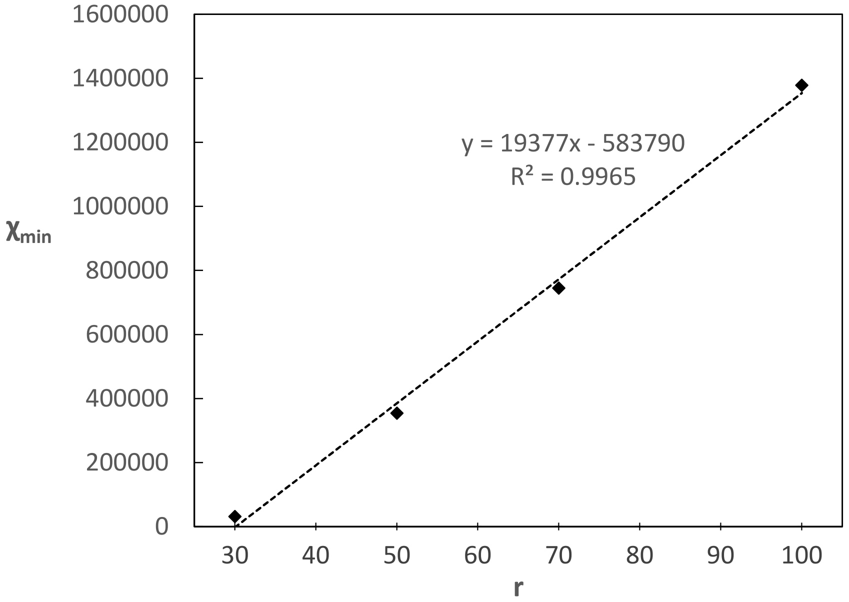

Figure 3.7: Minimum χ values required to stabilize simulation for various values of r, using $\Delta t = 10^{-6}$ and $\tilde{r} = 100$, and showing the linear least squares regression

so it is certainly not a complete cure for the instability. And while using such a well-developed baseline as a starting condition would be expensive in general (and rather defeat the purpose of an LES based model), since most runs failed within only a few dozen time steps or else not at all, it could be fruitful to investigate whether even just a few tens of interations at the start of a run with a stable parameter set could be used to greatly increase the stability of a full simulation with a more desirable but less stable model configuration.

Finally, while the investigation focused on achieving stabilization when increasing the filter width, it was also observed that a simulation using a time step of $\Delta t = 10^{-4}$, which was not stable for any filter width under the base method, was also able to be stabilized with a large enough χ. This could offer one potential avenue to mitigate some of the effects of the slower evolution of the regularized system by allowing the use of larger time steps, although further investigation would again be necessary to determine the viability of such an approach.

4 Conclusion

A stabilizing regularization term and divergence cleaning via the projection method have been implemented and integrated into a previously implemented temporally filtered large-eddy simulation (TLES) solver in OpenFOAM. The stability and output of the base method under different model parameters was studied and used as comparison against tests performed using the newly coded features to determine any stabilizing or other effects on the performance of the method.

Divergence cleaning was able to be efficiently implemented using an iterative solver with a relaxed tolerance, and was found to provide a measurable but small increase in the stable range of the filter width, along with very minor changes in the shape of its velocity line profile. The increase in stability seems to indicate that some non-zero divergence is present in the fields of the base method, but also that this was not the root cause of the instability, or at least not the sole root cause.

Regularization using the addition of a linear forcing term to the filtered Navier-Stokes momentum equation was found to be quite effective at stabilizing the solution. Any value of the filter width tested was able to be simulated successfully by setting the control parameter χ of the regularization sufficiently large. The evolution of the system towards a steady state was found to be greatly slowed by the regularization, but did produce a very similar velocity line profile when a recognizable feature was propagated to the same location in the geometry as in the non-regularized runs. A simulation using a time step for which no filter width was stable in the base method was also able to be stabilized by the regularization, and it was additionally found that starting the simulation with already time-evolved initial data allowed for successful simulation with some parameters that were unstable for uniform initial conditions. These two results, while needing additional investigation, suggest possibilities for mitigating the slow system evolution, either by using a larger time step, or by starting a simulation with a number of iterations using a stable parameter set in order to produce a more stable starting point for the full simulation using the desired, but potentially unstable, configuration of the method.

Bibliography

[1] C. Pruett, “Temporal large-eddy simulation: theory and implementation,” Theor. Comput. Fluid Dyn. **22**, 275 (2008).

[2] J. Smagorinsky, “General circulation experiments with the primitive equations. I. The basic experiment,” Mon. Weather Rev. **91**, 99 (1963).

[3] S. Stolz, and N.A. Adams, “An approximate deconvolution procedure for large-eddy simulation,” Phys. Fluids **11**, 1699 (1999).

[4] M. Germano, U. Piomelli, P. Moin and W.H. Cabot, “A dynamic subgrid-scale eddy viscosity model,” Phys. Fluids **3**, 1760 (1991).

[5] C. Pruett, “Eulerian Time-Domain Filtering for Spatial Large-Eddy Simulation,” AIAA J. **38**, 1634 (2000).

[6] P. Jenny, “Unsteady RANS closure,” Unpublished (2016).

[7] S. Stolz, N.A. Adams, and L. Kleiser, “An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows,” Phys. Fluids **13**, 997 (2001).

[8] A. Åkervik, L Brandt, D. S. Henningson, J. Hœpffner, O. Marxen, and P. Schlatter, “Steady solutions of the Navier-Stokes equations by selective frequency damping,” Phys. Fluids **18**, 068102 (2006).

[9] G. Tóth, “The $\nabla \cdot B = 0$ Constraint in Shock-Capturing Magnetohydrodynamics Codes,” J. Comput. Fluids **161**, 605 (2000).