CSE380: Tools and Techniques of Computational Science

Final Project Proposal

Gopal Yalla

November 28, 2016

Contents

1 Problem Formulation

It is well known that the Navier-Stokes equations,

$$\begin{aligned} \partial_t u_i + u_j \partial_j u_i &= \frac{1}{\rho}\partial_i P + \nu \Delta u_i \\ \partial_i u_i &= 0 \end{aligned} , \tag{1}$$

govern viscous incompressible flow; however, at high Reynolds number, it is computationally infeasible to use (1) to simulate fluid flow. Therefore, one must develop turbulent models to provide tools for computing complex turbulent flows. Often times, a statistical approach is taken where the goal is to describe the statistics of the velocity field, without access to realizations of the random flow [1]. For example, the total velocity can be decomposed into a sum of its mean and fluctuations $u(\mathbf{x}, t) = U(\mathbf{x}, t) + \tilde{u}(\mathbf{x}, t)$, where $U \equiv \overline{u}$. Substituting this decomposition into (1) gives,

$$\begin{aligned} \partial_t(U_i + \tilde{u}_i) + (U_j + \tilde{u}_j)\partial_j(U_i + \tilde{u}_i) &= \frac{-1}{\rho}\partial_i(P + p) + \nu\Delta(U_i + \tilde{u}_i) \\ \partial_i(U_i + \tilde{u}_i) &= 0 \end{aligned} .$$

The average of these equations is obtained by noting $\overline{U} = U$ and $\overline{\tilde{u}} = 0$:

$$\begin{aligned} \partial_t U_i + U_j \partial_j U_i &= \frac{-1}{\rho}\partial_i P + \nu\Delta U_i - \partial_j \overline{\tilde{u}_j \tilde{u}_i} \\ \partial_i U_i &= 0 \end{aligned} ; \tag{2}$$

These are the Reynolds Averaged Navier-Stokes (RANS) equations, which are unclosed primarily due to the terms in the Reynolds stress tensor, $\partial_j \overline{\tilde{u}_j \tilde{u}_i}$. Thus, to close these equations, a (semi-empirical) model for the Reynolds stress tensor is required.

1.1 $\overline{v^2} - f$ Model

Closing the RANS equations is accomplished by various turbulence models. Here, we will focus on the $\overline{v^2} - f$ model, first proposed by Durbin in 1995 [2]. The $\overline{v^2} - f$ model is the primary RANS model used by the PECOS group in their hybrid RANS/LES models [3].

The $\overline{v^2}-f$ equations are:

$$\partial_t U_i + U_j d_j U_i = \frac{-1}{\rho}\partial_i\left(P+\frac{2}{3}\rho k\right)+\partial_j\left[(\nu+\nu_T)(\partial_j U_i+\partial_i U_j)\right]] \tag{3}$$

$$\partial_t k + U_j d_j k = \mathcal{P} - \epsilon + \partial_j\left((\nu+\nu_T)d_j k\right) \tag{4}$$

$$\partial_t \epsilon + U_j d_j \epsilon = \frac{C_{\epsilon 1}\mathcal{P} - C_{\epsilon 2}\epsilon}{T} + \partial_j\left((\nu+\frac{\nu_T}{\sigma_\epsilon})\partial_j \epsilon\right) \tag{5}$$

$$\partial_t \overline{v^2} + U_j d_j \overline{v^2} + \epsilon\frac{\overline{v^2}}{k} = kf + \partial_k[\nu_T \partial_k \overline{v^2}] + v\nabla^2\overline{v^2} \tag{6}$$

$$L^2\nabla^2 f - f = -C_2\frac{\mathcal{P}}{k} + \frac{C_1}{T}\left(\frac{\overline{v^2}}{k}-\frac{2}{3}\right) \tag{7}$$

Below we summarize the various unknowns, constants, and other terms.
Unknowns:

- U_i — i^{th} component of the mean velocity.
- $\nu_T = C_\mu \overline{v^2} T$
  — eddy viscosity.
- k — turbulent kinetic energy.
- ϵ — energy dissipation.
- $\overline{v^2}$ — Velocity scale to capture anisotropy near wall
- f — "redistribution term."

Other Terms:

- $\mathcal{P} = \nu_T(\partial_i U_j + \partial_j U_i)\partial_i U_j$
  — rate at which mean flow is converted to turbulent.
- $T = \max\left\{\frac{k}{\epsilon}, 6\left(\frac{v}{\epsilon}\right)^{\frac{1}{2}}\right\}$
  — turbulent length scale.
- $L = \max C_L\left\{\frac{k^{3/2}}{\epsilon}, C_\eta(\frac{v^3}{\epsilon})^{1/4}\right\}$
  — Length Scale.
- ν — viscosity.
- ρ — fluid density.

Constants:

1. $C_\mu = 0.19$
2. $\sigma_\epsilon = 1.3$
3. $C_2 = 0.3$
4. $C_L = 0.3$

5. $C_1 = 0.4$
6. $C_{\epsilon 1} = 1.55$
7. $C_{\epsilon 2} = 1.9$
8. $C_\eta = 70$

1.2 Fully Developed Channel Flow

For this project, we only consider fully developed channel flow through a rectangular duct of height $h = 2\delta$. The bottom and top walls are at $y = 0$ and $y = 2\delta$ with the mid plane being $y = \delta$. In the fully developed region, the flow is entirely in the axial direction, i.e, statistics no longer vary with x [4]. Hence, the fully developed channel flow being considered is statistically stationary and statistically one-dimensional, with velocity statistics depending only on y [4]. Moreover, the flow is statistically symmetric about the mid-plane $y = \delta$. The steady state $\overline{v^2} - f$ equations for fully developed flow are given by,

$$(\nu + \nu_T)\frac{\partial^2 U}{\partial y^2} + \frac{\partial U}{\partial y}\frac{\partial \nu_T}{\partial y} - \frac{1}{\rho}\frac{\partial P}{\partial x} = 0 \tag{8}$$

$$\mathcal{P} - \epsilon + (\nu + \nu_T)\frac{\partial^2 k}{\partial y^2} + \frac{\partial k}{\partial y}\frac{\partial \nu_T}{\partial y} = 0 \tag{9}$$

$$\frac{C_{\epsilon 1}\mathcal{P} - C_{\epsilon 2}\epsilon}{T} + \left(\nu + \frac{\nu_T}{\sigma_\epsilon}\right)\frac{\partial^2 \epsilon}{\partial y^2} + \frac{1}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial y}\frac{\partial \nu_T}{\partial y} = 0 \tag{10}$$

$$kf + (\nu + \nu_T)\frac{\partial^2 \overline{v^2}}{\partial y^2} + \frac{\partial \overline{v^2}}{\partial y}\frac{\partial \nu_T}{\partial y} - \frac{\epsilon \overline{v^2}}{k} = 0 \tag{11}$$

$$L^2\frac{d^2 f}{dy^2} - f + c_2\frac{\mathcal{P}}{k} - \frac{c_1}{T}\left(\frac{\overline{v^2}}{k} - \frac{2}{3}\right) = 0 \tag{12}$$

with boundary conditions at $y = 0$ being,

$$U = k = \overline{v^2} = 0,\ \epsilon \to 2\nu\frac{k}{y^2},\ f \to -\frac{20\nu^2\overline{v^2}}{\epsilon(0)y^4}\ .$$

and boundary conditions at $y = \delta$ being,

$$\frac{\partial U}{\partial y} = \frac{\partial k}{\partial y} = \frac{\partial \overline{v^2}}{\partial y} = \frac{\partial \epsilon}{\partial y} = \frac{\partial f}{\partial y} = 0\ .$$

There is a "code-friendly" version of equation (12) mentioned in [5] to alleviate the instability that may arise in the boundary condition for $f(0)$ (due to the y^4 term), but it is also mentioned that when coupled implicit solvers are used, the code-friendly version may be unnecessary, so we will only incorporate this if need be.

1.3 Non-Dimensionalization

Before solving equations (8)-(12), we must non-dimensionalize the equations. Below we list each dimensional variable and corresponding unit. In table 1, L and T represent the fundamental physical dimensions of length and time (not to be confused with the L and T above.)

Variable	U	ϵ	k	ν_T	$\mathcal{P}$	f	$\overline{v^2}$
Units	$\frac{L}{T}$	$\frac{L^2}{T^3}$	$\frac{L^2}{T^2}$	$\frac{L^2}{T}$	$\frac{L^2}{T^3}$	$\frac{1}{T}$	$\frac{L^2}{T^2}$

Table 1: Dimensional Variables and Physical Units

We will normalize length by the layer thickness, δ, and normalize velocity by the velocity scale,

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} = \sqrt{\frac{\mu\frac{\partial U}{\partial y}}{\rho}} = \sqrt{\nu\frac{\partial U}{\partial y}}\ ,$$

where τ_w is the stress at the wall. Note that this normalization is chosen so that

$$\frac{1}{\rho}\frac{\partial P}{\partial x} = -1\ .$$

Using this normalization we can define our non-dimensional variables based on table 1, e.g., $U^+ = U/u_\tau, \epsilon^+ = \epsilon\delta/u_\tau^3$, etc. Moreover, we define the parameter $Re_\tau = u_\tau\delta/\nu$ so that $\nu^+ = 1/Re_\tau$. From here on we will abuse notation and only use the $(^+)$ notation on U, e.g., $U^+ = U/u_\tau$ but $\epsilon = \epsilon\delta/u_\tau^3$. Lastly, we define the variable $\eta = y/\delta$ so that $\eta \in (0,1)$. This leads to the non-dimensional $\overline{v^2} - f$ equations for fully developed flow:

$$\left(\frac{1}{Re_\tau} + \nu_T\right)\frac{\partial^2 U^+}{\partial \eta^2} + \frac{\partial U^+}{\partial \eta}\frac{\partial \nu_T}{\partial \eta} + 1 = 0 \tag{13}$$

$$\mathcal{P} - \epsilon + \left(\frac{1}{Re_\tau} + \nu_T\right)\frac{\partial^2 k}{\partial \eta^2} + \frac{\partial k}{\partial \eta}\frac{\partial \nu_T}{\partial \eta} = 0 \tag{14}$$

$$\frac{C_{\epsilon 1}\mathcal{P} - C_{\epsilon 2}\epsilon}{T} + \left(\frac{1}{Re_\tau} + \frac{\nu_T}{\sigma_\epsilon}\right)\frac{\partial^2 \epsilon}{\partial \eta^2} + \frac{1}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial \eta}\frac{\partial \nu_T}{\partial \eta} = 0 \tag{15}$$

$$kf + \left(\frac{1}{Re_\tau} + \nu_T\right)\frac{\partial^2 \overline{v^2}}{\partial \eta^2} + \frac{\partial \overline{v^2}}{\partial \eta}\frac{\partial \nu_T}{\partial \eta} - \frac{\epsilon\overline{v^2}}{k} = 0 \tag{16}$$

$$L^2\frac{d^2 f}{d\eta^2} - f + c_2\frac{\mathcal{P}}{k} - \frac{c_1}{T}\left(\frac{\overline{v^2}}{k} - \frac{2}{3}\right) = 0 \tag{17}$$

where,

$$T = \max\left\{\frac{k}{\epsilon}, 6\sqrt{\frac{1}{Re_\tau\epsilon}}\right\}$$

$$L = C_L \max\left\{\frac{k^{3/2}}{\epsilon}, C_\eta\left(\frac{1}{Re_\tau^3\epsilon}\right)^{1/4}\right\}$$

$$\mathcal{P} = \nu_T\left(\frac{\partial U^+}{\partial \eta}\right)^2$$

with boundary conditions at $\eta = 0$ being,

$$U^+ = k = \overline{v^2} = 0, \; \epsilon \to 2\frac{k}{Re_\tau \eta^2}, \; f \to -\frac{20\overline{v^2}}{Re_\tau^2 \epsilon(0)\eta^4} \; .$$

and boundary conditions at $y = \delta$ being,

$$\frac{\partial U}{\partial \eta} = \frac{\partial k}{\partial \eta} = \frac{\partial \overline{v^2}}{\partial \eta} = \frac{\partial \epsilon}{\partial \eta} = \frac{\partial f}{\partial \eta} = 0 \; .$$

2 Numerical Methods

2.1 Time-Marching

As described in the previous section, we want to solve the steady state $\overline{v^2} - f$ equations for fully developed channel flow. We could try and do this by discretizing the steady equations and writing a nonlinear solver, but the equations are sensitive enough that the most naive solvers (e.g., Newton's method) will diverge unless one starts with a very accurate initial guess. One way around this problem is to use some sort of continuation method, often done by time marching. Essentially, we are looking to march in time from our initial guess to the steady state equations. Backward Euler is a good choice of time scheme because it is implicit (so one can take large steps) and because accuracy in time is not of importance. In fact, because we do not care about temporal accuracy, there is no need to fully solve the nonlinear system at each step,i.e., we can just take a single Newton iteration and go to the next step. [6]

To be more specific, if $R(U) = 0$ is the system to solve, we start with an initial guess $U = U_0$ and solve $\partial U/\partial t + R(U) = 0$ forward in time until $\partial U/\partial t = 0$. Generally, this works best if you start with a small time step and then gradually increase the time step until it is very large at which point you're essentially trying to solve the steady problem from a different initial guess. [6]

2.2 Finite Difference Scheme

In order to solve equations (13) - (17), we will use a (second-order) centered difference scheme in space and backward Euler in time for time marching. Let $\Delta\eta$ and Δt be the space and time step size, respectively. Moreover, for each variable ξ, define $\xi_i^n = \xi(x_i, t_n)$ where $x_i = i \cdot \Delta\eta$, and $t_n = n \cdot \Delta t$, and define $I = 1/\Delta\eta$. Let ξ^{n+1} be the vector of unknowns at $t = (n+1) \cdot \Delta t$, i.e.,

$$\xi^{n+1} = \begin{bmatrix} \epsilon_0^{n+1} \\ f_0^{n+1} \\ {U^+}_1^{n+1} \\ k_1^{n+1} \\ \epsilon_1^{n+1} \\ \overline{v^2}_1^{n+1} \\ f_1^{n+1} \\ {U^+}_2^{n+1} \\ k_2^{n+1} \\ \vdots \\ f_{I-1}^{n+1} \\ {U^+}_I^{n+1} \\ k_I^{n+1} \\ \epsilon_I^{n+1} \\ \overline{v^2}_I^{n+1} \\ f_I^{n+1} \end{bmatrix}$$

Then, from equations (13)- (17) plus time marching, we wish to solve $F(\xi^{n+1}) = 0$ for each time step, where

$$F(\xi^{n+1}) = \begin{pmatrix} \epsilon_0^{n+1} - \frac{2k_1^{n+1}}{Re_\tau \Delta\eta^2} \\ f_0^{n+1} - \frac{20\overline{v^2}_1^{n+1}}{Re_\tau^3 \epsilon_0^{n+1} \Delta\eta^4} \\ \vdots \\ \frac{{U^+}_i^{n+1} - {U^+}_i^{n}}{\Delta t} + \left(\frac{1}{Re_\tau} + {\nu_T}_i^{n+1}\right)\left(\frac{{U^+}_{i+1}^{n+1} - 2{U^+}_i^{n+1} + {U^+}_{i-1}^{n+1}}{\Delta\eta^2}\right) + \left(\frac{{U^+}_{i+1}^{n+1} - {U^+}_{i-1}^{n+1}}{2\Delta\eta}\right)\left(\frac{{\nu_T}_{i+1}^{n+1} - {\nu_T}_{i-1}^{n+1}}{2\Delta\eta}\right) + 1 \\ \frac{k_i^{n+1} - k_i^n}{\Delta t} + \mathcal{P}_i^{n+1} - \epsilon_i^{n+1} + \left(\frac{1}{Re_\tau} + {\nu_T}_i^{n+1}\right)\left(\frac{k_{i+1}^{n+1} - 2k_i^{n+1} + k_{i-1}^{n+1}}{\Delta\eta^2}\right) + \left(\frac{k_{i+1}^{n+1} - k_{i-1}^{n+1}}{2\Delta\eta}\right)\left(\frac{{\nu_T}_{i+1}^{n+1} - {\nu_T}_{i-1}^{n+1}}{2\Delta\eta}\right) \\ \frac{\epsilon_i^{n+1} - \epsilon_i^n}{\Delta t} + \frac{C_{\epsilon1}\mathcal{P}_i^{n+1} - C_{\epsilon2}\epsilon_i^{n+1}}{T_i^{n+1}} + \left(\frac{1}{Re_\tau} + \frac{{\nu_T}_i^{n+1}}{\sigma_\epsilon}\right)\left(\frac{\epsilon_{i+1}^{n+1} - 2\epsilon_i^{n+1} + \epsilon_{i-1}^{n+1}}{\Delta\eta^2}\right) + \frac{1}{\sigma_\epsilon}\left(\frac{\epsilon_{i+1}^{n+1} - \epsilon_{i-1}^{n+1}}{2\Delta\eta}\right)\left(\frac{{\nu_T}_{i+1}^{n+1} - {\nu_T}_{i-1}^{n+1}}{2\Delta\eta}\right) \\ \frac{\overline{v^2}_i^{n+1} - \overline{v^2}_i^{n}}{\Delta t} + k_i^{n+1} f_i^{n+1} + \left(\frac{1}{Re_\tau} + {\nu_T}_i^{n+1}\right)\left(\frac{\overline{v^2}_{i+1}^{n+1} - 2\overline{v^2}_i^{n+1} + \overline{v^2}_{i-1}^{n+1}}{\Delta\eta^2}\right) + \left(\frac{\overline{v^2}_{i+1}^{n+1} - \overline{v^2}_{i-1}^{n+1}}{2\Delta\eta}\right)\left(\frac{{\nu_T}_{i+1}^{n+1} - {\nu_T}_{i-1}^{n+1}}{2\Delta\eta}\right) - \epsilon_i^{n+1}\frac{\overline{v^2}_i^{n+1}}{k_i^{n+1}} \\ \frac{f_i^{n+1} - f_i^n}{\Delta t} + (L_i^{n+1})^2\left(\frac{f_{i+1}^{n+1} - 2f_i^{n+1} + f_{i-1}^{n+1}}{\Delta\eta^2}\right) - f_i^{n+1} + C_2\frac{\mathcal{P}_i^{n+1}}{k_i^{n+1}} - \frac{C_1}{T_i^{n+1}}\left(\frac{\overline{v^2}_i^{n+1}}{k_i^{n+1}} - \frac{2}{3}\right) \\ \vdots \\ \left(\frac{{U^+}_I^{n+1} - {U^+}_I^{n}}{\Delta t}\right) + \left(\frac{1}{Re_\tau} + {\nu_T}_I^{n+1}\right)\left(\frac{2{U^+}_{I-1}^{n+1} - 2{U^+}_I^{n+1}}{\Delta\eta^2}\right) + 1 \\ \left(\frac{k_I^{n+1} - k_I^n}{\Delta t}\right) - \epsilon_I^{n+1} + \left(\frac{1}{Re_\tau} + {\nu_T}_I^{n+1}\right)\left(\frac{2k_{I-1}^{n+1} - 2k_I^{n+1}}{\Delta\eta^2}\right) \\ \left(\frac{\epsilon_I^{n+1} - \epsilon_I^n}{\Delta t}\right) - \frac{C_{\epsilon2}\epsilon_I^{n+1}}{T_I^{n+1}} + \left(\frac{1}{Re_\tau} + \frac{{\nu_T}_I^{n+1}}{\sigma\epsilon}\right)\left(\frac{2\epsilon_{I-1}^{n+1} - 2\epsilon_I^{n+1}}{\Delta\eta^2}\right) \\ \left(\frac{\overline{v^2}_I^{n+1} - \overline{v^2}_I^{n}}{\Delta t}\right) + k_I^{n+1} f_I^{n+1} + \left(\frac{1}{Re_\tau} + {\nu_T}_I^{n+1}\right)\left(\frac{2\overline{v^2}_{I-1}^{n+1} - 2\overline{v^2}_I^{n+1}}{\Delta\eta^2}\right) \\ \left(\frac{f_I^{n+1} - f_I^n}{\Delta t}\right) + \left(L_i^{n+1}\right)^2\left(\frac{2f_{I-1}^{n+1} - 2f_I^{n+1}}{\Delta\eta^2}\right) - f_I^{n+1} - \frac{C_1}{T_I^{n+1}}\left(\frac{\overline{v^2}_I^{n+1}}{k_I^{n+1}} - \frac{2}{3}\right) \end{pmatrix}$$

The initial conditions will be interpolated from the direct numerical simulation data performed by Dr. Moser and Dr. Lee on turbulence.ices.utexas.edu.

3 Design Considerations

3.1 External Software

We plan to make use of GSL's multi-dimensional root solvers (e.g. GSL's newton solver) and GRVY for input parsing. Regarding input options we will allow a choice of model parameters and constants, step size, and make use of various log levels to control standard output.

3.2 Regression Tests

In order to test the accuracy of the numerical method, we will perform asymptotic error analysis. To do this we compute the solution for a large number of steps, N, and use this as our "true" solution. Then we compute the solution for small N and plot the error vs N on a log-log plot. The slope of the line should be the order of the numerical method (2 in this case). Moreover, we can save the computed solutions to make sure we can recover them in the future.

Another test for the numerical method would be Jacobian verification. We could test the numeric jacobian vs. the analytical one and make sure they agree within some tolerance.

To test the actual physics opposed to the numerical scheme, we will check the convergence of certain statistical quantities; however, more research needs to be done to determine which quantities to check.

Lastly, if time permits (or possibly over break), we will implement a method of manufactured solutions — a technique by which numerical methods within codes can be verified to ensure that they have been coded correctly and do, in fact, solve the equations that they claim to solve.

3.3 Unit Tests

Unit testing will play a major role in the project, as there will be several routines that compute quantities, which we will need to test. The best way to do this is to test each function with dummy quantities and compare the result to the true value, which will be computed by hand.

3.4 Build Systems

We will use GNU make to link in all 3rd party libraries that the application requires. It will have an option to run a suite of the aforementioned regression tests via the "make check" check target.

3.5 Version Control

Git will be used for version control.

3.6 Profiling and Performance Optimization

Once the code is complete and we are confident in the answer, we will worry about performance optimization. First off, performance timing measurements for key kernels within the code will be included using the GRVY library. If the program takes longer to run than desired, we will use gprof for profiling to see which parts of the code to optimize.

3.7 Documentation

Putting this document together has taken over 3 weeks to complete, and will serve as the primary form of documentation for the project. Having started with zero experience in fluid, I have already gained a lot from this project. Moreover, I understand how important it is to fully lay out a problem before writing any code.

References

[1] P. A. Durbin and B. P. Reif, *Statistical theory and modeling for turbulent flows.* John Wiley & Sons, 2011.

[2] P. Durbin, “Separated flow computations with the k-epsilon-v-squared model,” *AIAA journal*, vol. 33, no. 4, pp. 659–664, 1995.

[3] S. W. Haering *et al.*, *Anisotropic hybrid turbulence modeling with specific application to the simulation of pulse-actuated dynamic stall control.* PhD thesis, 2015.

[4] S. B. Pope, *Turbulent flows.* IOP Publishing, 2001.

[5] F. Lien and P. Durbin, “Non-linear k–v2 modelling with application to high-lift,” *Proceedings of the Summer Program Centre for Turbulence Research. Stanford University*, pp. 5–22, 1996.

[6] Todd