2.29 Final Project

DNS/LES 1D Combustion Simulation

Yang Chen

Introduction

The objective of current project is to incorporate LES compressible viscous reactive flow simulation with detailed chemical reactor solver, Cantera, to accurately calculate emissions levels out of gas turbine combustor. As for this class project, the procedure consists of three steps. Firstly, a 1D DNS simulation for compressible viscous reactive flow with simplified G flamelet model with be performed in MATLAB. Then, a 1D LES simulation will be constructed based on the DNS simulation. Once, the LES simulation is validated by the DNS simulation, full 3D simulation will be performed using OpenFoam. As of the time this report is written, the author is struggling with resolving the stability issue in the first step 1D DNS simulation. Thus, what would be presented in this report will be the formulation of governing equation and finite volume discretization of the first step 1D simulation. Also, I will present the result obtained from this first step simulation with a few attempts in trying debug the code and the suppressed the instability. Information on LES simulation will not be shown in this report as they are presented in slides.

Governing Equations

The governing equations for compressible viscous flow with simplified G flamelet model are from previous studies.[1][2] The one dimensional formulation of the corresponding equations are shown as follows.

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho u}{\partial x}$$

$$\frac{\partial \rho u}{\partial t} = -\frac{\partial P}{dx} - \frac{\partial (\rho u^2)}{\partial x} + \frac{4}{3}\mu \frac{\partial^2 u}{\partial^2 x}$$

$$\frac{\partial \rho E}{\partial t} = -\frac{\partial (E\rho + P)u}{dx} + \frac{4}{3}\mu \frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + k\frac{\partial^2 T}{\partial x^2}$$

$$\frac{\partial \rho G}{\partial t} = -\frac{\partial \rho u G}{\partial x} - \rho S_L sign\left(\frac{\partial G}{\partial x}\right)\frac{\partial G}{\partial x}$$

$$P = \rho RT \quad E = C_v T + \Delta h_f G + \frac{1}{2}u^2 \quad S_L = 10 + 3.71 \cdot 10^{-4} \cdot T^2$$

Only heat condition is utilized in this study for simplicity; however, more comprehensive heat transfer models will be used in future study. Unstretched laminar flame speed of stoichiometric

methane combustion is used[3] for simplicity, while in future work, more robust flame speed model can be used, which is shown as follows.

$$S_L = S_{L,ref}\left(\frac{T_u}{T_{u,ref}}\right)^{\gamma}\left(\frac{P}{P_{ref}}\right)^{\beta}(1-2.1Y_{dil})$$

$$S_{Lref} = B_M + B_2(\phi-\phi_M)^2$$

$$\gamma = 2.18 - 0.8(\phi-1)$$

$$\beta = -0.16 + 0.22(\phi-1)$$

$$T_{uref} = 298K\ P_{ref} = 1atm$$

$$Propane: \phi_M = 1.11\ B_M = 36.92\ B_2 = -140.51$$

Ydil is the diluent in air fuel mixture that account for the recirculation of burnt product. As this effect will be taken cared by the advection term in governing equation. Ydil of 0 will be used. This more accurate flame speed model will be used in future study.

Moreover, for the G flamelet model, as indicated by the previous study[1], the actual G equation contains a heavy side component which is omitted for stability reason. Thus, the simulation is only limited to narrow flame front for 3D simulation.

Finite Volume Discretization

The discretized equations are shown as follows.

$$\frac{d\rho_P}{dt} = \frac{1}{\Delta x}(\rho u_\omega - \rho u_e)$$

$$\frac{d\rho u_P}{dt} = \frac{1}{\Delta x}(P_\omega - P_e) + \frac{1}{\Delta x}(\rho u_\omega^2 - \rho u_e^2) + \frac{4}{3}\frac{\mu}{\Delta x}\left(\frac{\partial u}{\partial x_e} - \frac{\partial u}{\partial x_\omega}\right)$$

$$\frac{d\rho E_P}{dt} = \frac{1}{\Delta x}\left((\rho E + P)u_\omega - (\rho E + P)u_e\right) + \frac{4}{3}\frac{\mu}{\Delta x}\left(u\frac{\partial u}{\partial x_e} - u\frac{\partial u}{\partial x_\omega}\right) + \frac{k}{\Delta x}\left(\frac{\partial T}{\partial x_e} - \frac{\partial T}{\partial x_\omega}\right)$$

$$\frac{d\rho G_P}{dt} = \frac{1}{\Delta x}(\rho u G_\omega - \rho u G_e) - \rho_P S_{L_P} Sign\left(\frac{\partial G}{\partial x_P}\right)\frac{\partial G}{\partial x_P}$$

Cell center value approximation is used to represent the cell averaged state variable. That means all the scheme used in this simulation shall be at least second order for the approximation to be effective. For all the advection terms, quick scheme is used as follows.

$$\rho u_{\omega\ or\ e} = \frac{6}{8}\rho_u u_u + \frac{3}{8}\rho_d u_d - \frac{1}{8}\rho_{uu} u_{uu}$$

$$\rho u^2_{\omega\ or\ e} = \frac{6}{8}\rho_u u_u^2 + \frac{3}{8}\rho_d u_d^2 - \frac{1}{8}\rho_{uu} u_{uu}^2$$

$$(E\rho + P)u_{\omega\ or\ e} = \frac{6}{8}(E_u\rho_u + P_u)u_u + \frac{3}{8}(E_d\rho_d + P_d)u_d - \frac{1}{8}(E_{uu}\rho_{uu} + P_{uu})u_{uu}$$

$$\rho u G_{\omega\ or\ e} = \frac{6}{8}\rho_u u_u G_u + \frac{3}{8}\rho_d u_d G_d - \frac{1}{8}\rho_{uu} u_{uu} G_{uu}$$

For all the diffusion, pressure, and temperature terms, central difference is used as follows.

$$P_\omega = \frac{P_W + P_P}{2}$$

$$\frac{\partial u}{\partial x}_\omega = \frac{u_P - u_W}{\Delta x}$$

$$u\frac{\partial u}{\partial x}_\omega = \frac{u_P^2 - u_W^2}{2\Delta x}$$

$$\frac{\partial T}{\partial x}_\omega = \frac{T_P - T_W}{\Delta x}$$

The $u\frac{\partial u}{\partial x}_\omega$ term is basically just the CDS combination of u and $\frac{\partial u}{\partial x}$ terms. The G source term is modeled as follows.

$$\frac{\partial G}{\partial x}_P = \frac{G_P - G_W}{\Delta x}\ for\ (G_E > G_P > G_W)\ \ or\ (G_E < G_P > G_W \text{ but } G_P - G_W \geq G_P - G_E)$$

$$= \frac{G_E - G_P}{\Delta x}\ for\ (G_E < G_P < G_W)\ \ or\ (G_E < G_P > G_W \text{ but } G_P - G_W < G_P - G_E)$$

$$= \quad 0 \quad for\ (G_E > G_P < G_W)$$

Here, first order UDS is used for simplicity. Higher order scheme will be used in future to enhance accuracy. When G equals to 0, it represents product. When G equals to 1, it is fuel. Thus, the third case, $(G_E > G_P < G_W)$, will avoid the creation of multiple flame fronts from numerical oscillation of the G equation, which is also realistic. However, if $\frac{\partial G}{\partial x}_P$ equals to 0 in the third case, there must be at least 1 cell in the domain where G is equal to 0, pure product, or else the fuel in domain will not be fully consumed even run till steady state.

For G source term, the unwind direction is determined based on the G gradient, while for all other advection term, the upwind direction is by the flow velocity gradient.

For time matching, 3rd order Runge-Kutta from a previous study is used[4]. The procedure is shown as follows.

$$Step\ 1: U_1 = U_n + \Delta t \cdot Res_n$$

$$Step\ 2: U_2 = \frac{3}{4}U_n + \frac{1}{4}U_1 + \frac{1}{4} \cdot \Delta t \cdot Res_1$$

$$Step\ 3: U_{n+1} = \frac{1}{3}U_n + \frac{2}{3}U_2 + \frac{2}{3} \cdot \Delta t \cdot Res_2$$

$$Res_n = \frac{d}{dt}(U_n)$$

Spatial domain is set to be 0 to 1 m for simplicity. Time domain starts at 0 sec while the total simulation time is not determined due to instability observed. Ideally, this total time simulated shall be longer than the smallest turbulent time scale, because the final idea is to compare how LES simulated smallest turbulent flow matches with the actual DNS simulated smallest turbulent flow.

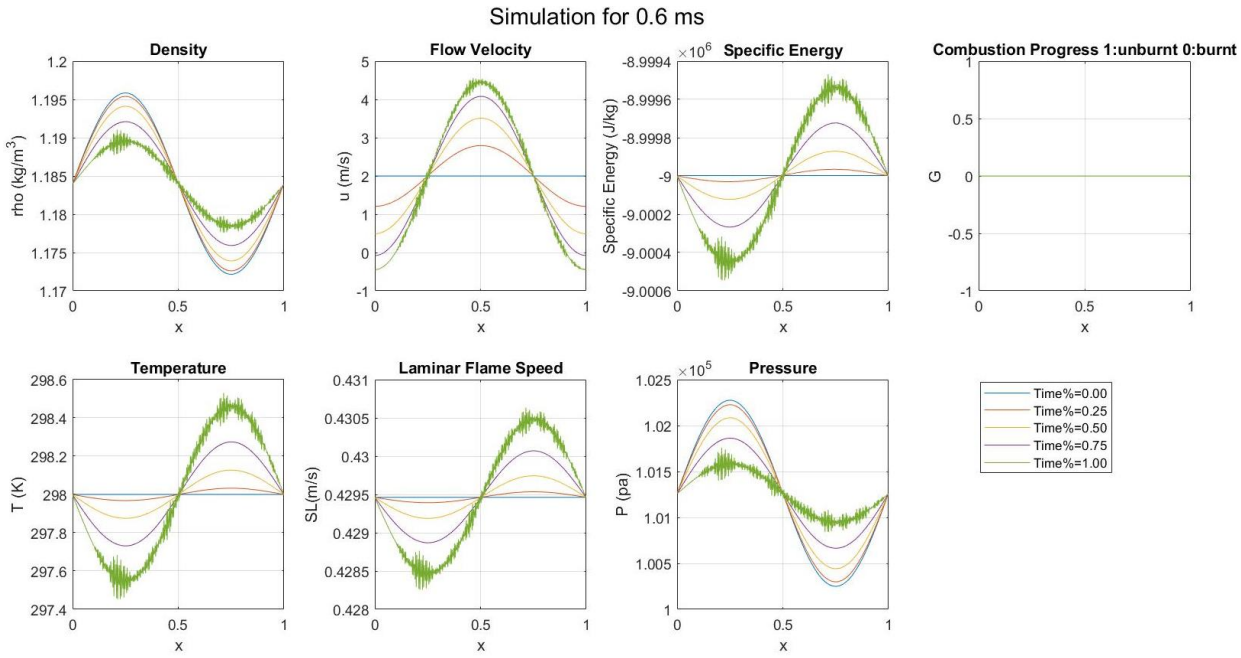
In term boundary condition, a periodic boundary condition is used. This formulation of boundary condition can avoid the blowing out or flashing back of flame front when the mean flow velocity is higher or lower than the mean flow velocity which is very likely to happen.

Results:

Some parameters are assumed constant and drawn from different online sources. For air at 2000K, the parameters are shown as follows.

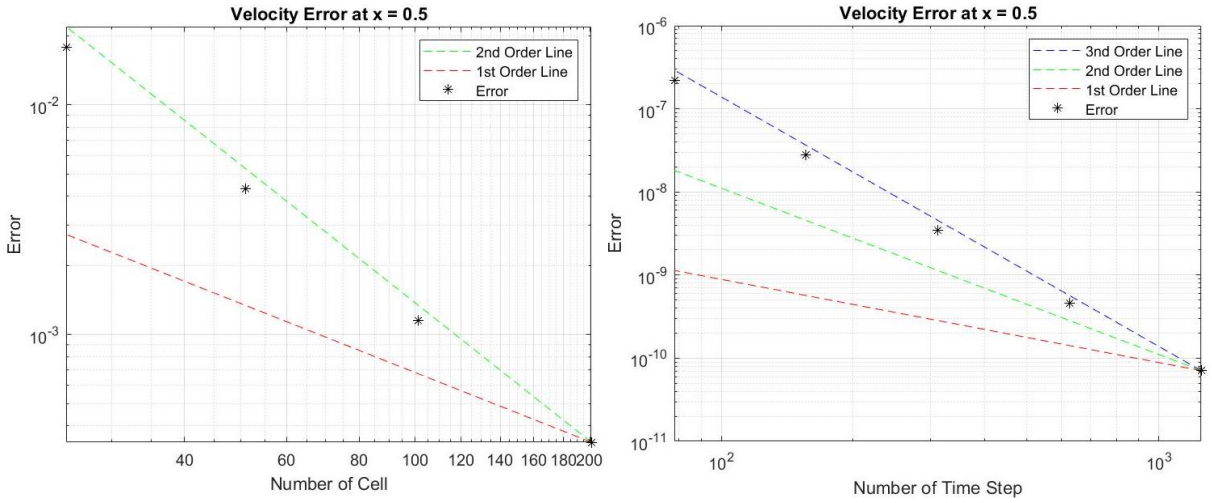
$$\mathrm{C_v} = 1000\,\frac{\mathrm{J}}{\mathrm{kg \cdot K}}\ ; \mathrm{R} = 287\,\frac{\mathrm{J}}{\mathrm{kg \cdot K}}; \mu = 1.81 \cdot 10^{-5}\,\frac{kg}{m \cdot s}; k_{heat} = 0.023\,\frac{W}{m \cdot K}; \rho = 1.184\,\frac{kg}{m^3}$$

An Oscillatory simulation is shown as follows.

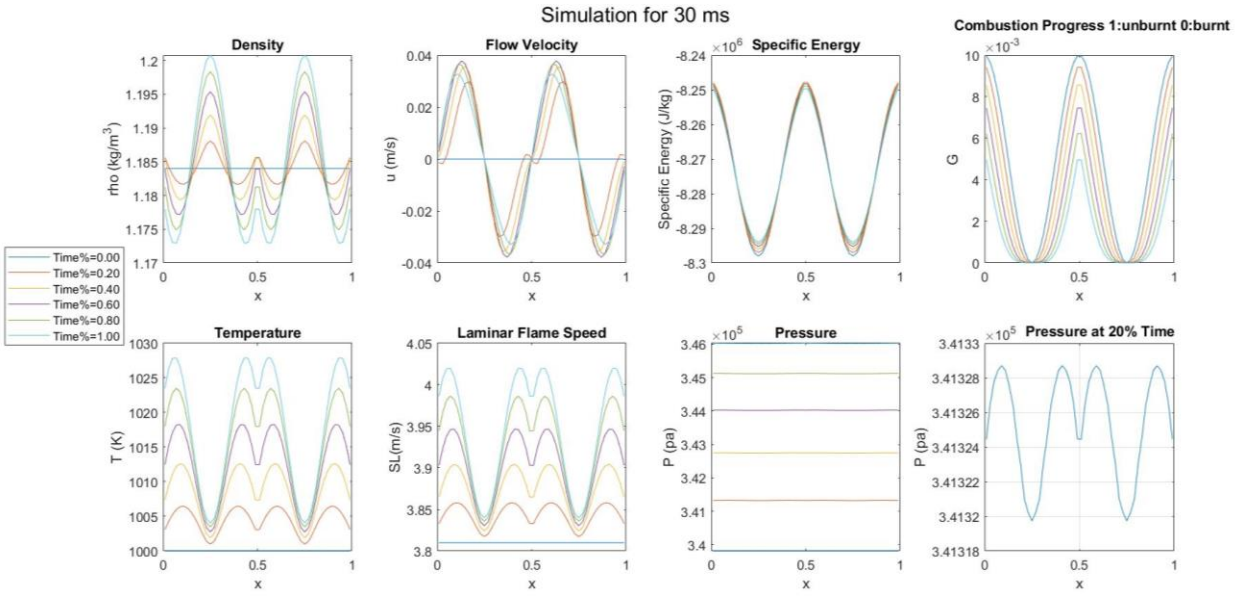


This simulation starts with a sinusoidal density and pressure distribution with 10000 time steps and 401 spatial cells. It can be observed that the oscillation starts to show up after 75% of the total time. It is also observed that this oscillation is not caused by CFL condition as keep increasing the time refinement does not have an effect on it. And the actually CFL impact can be seen when the number of time step is reduced to a very low value as 20 or 40 that the solution suddenly blows up.

Putting the oscillation aside which will be discussed in later section, the order of convergence can be observed by calculating the error based on the velocity at x = 0.5m and t = last time step that does not have oscillation. The convergence test result is shown as follows.



It can be observed that spatial convergence is 2 and time convergence is 3, matching the with 2nd order CDS and 3rd order Runge-Kutta. A combustion test is also performed as follows.



A cos(2*pi*x)^2 shape initial G concentration is used to generate the combustion test. The ^2 is used because G cannot be smaller than 0. An unrealistically high viscosity of 7kg/(m s) is used to suppressed oscillation. It can be observed that the trends of physics matches with intuition, which I elaborated in presentation and will not be repeated here.

Manufacture Solution for Pressure Term:

To verify the pressure term that is causing the oscillation, a standing wave manufacture solution is used to test it. More test will be done on future study during summer, but currently I need some time to finish the revision of the my master thesis draft that my advisor just sent back to me:).

Solution is manufactured as follows:

$$\rho_{ini} = 1.184 \frac{kg}{m^3}$$

$$u_{ini} = 0 \frac{m}{s}$$

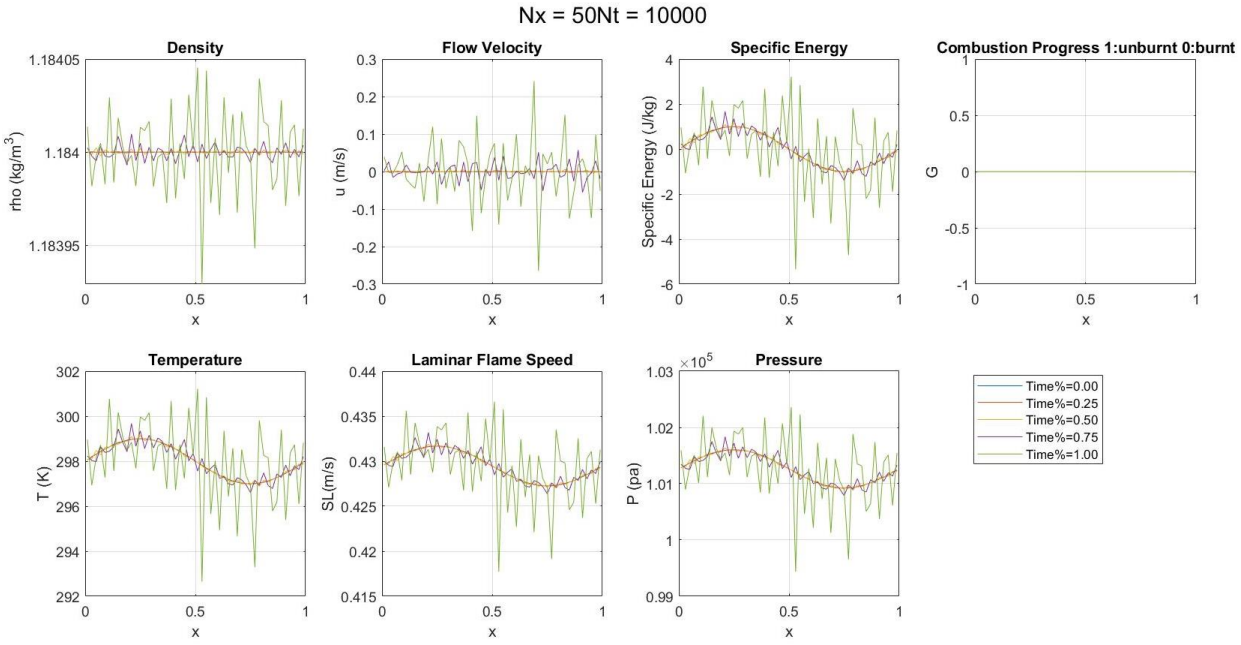
$$Ene_{ini} = \sin(2\pi x)$$

$$G_{ini} = 0$$

These will result in the following residuals.

$$Res_{\rho} = 0$$
$$Res_{\rho u} = 2 \cdot \pi \cdot 1.184 \cdot R \cdot cos(2\pi x)$$
$$Res_{\rho Ene} = 0$$
$$Res_{\rho G} = 0$$

So, we subtract the residues from the right hand side of the equation, we shall see a standing still pressure curves. The result after 2 ms with 50 cells and 10000 time steps is shown as follows.



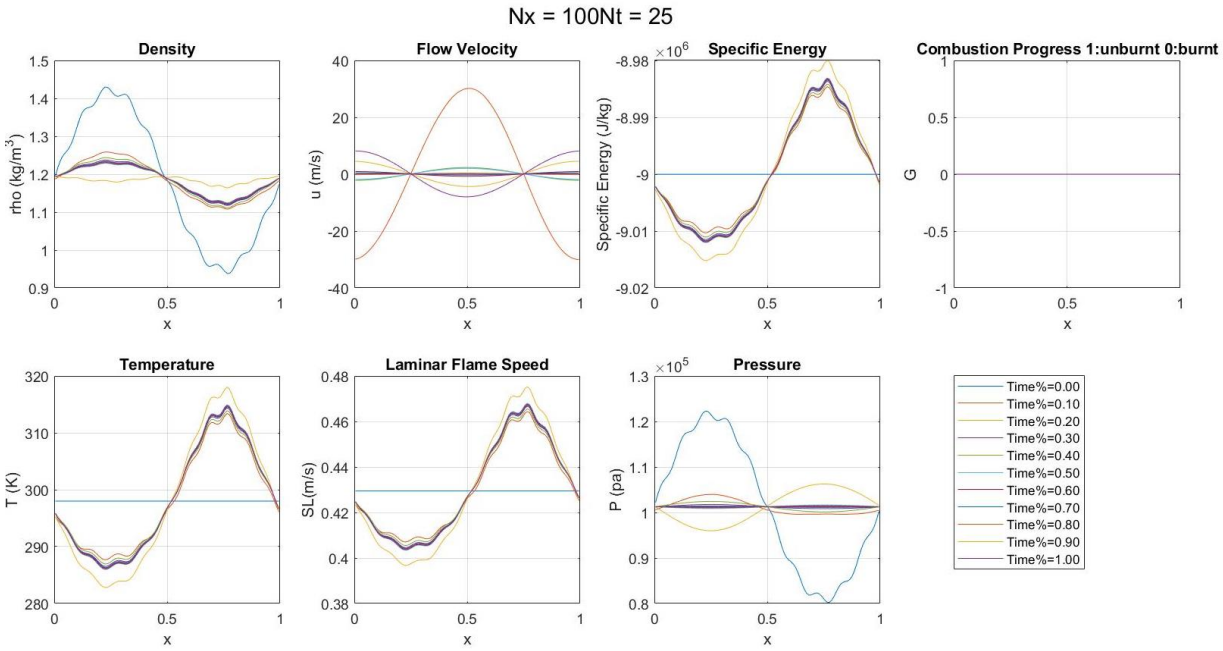
The average wave is observed to be standing still with the oscillation developed. The exact details will be further study in the future study during the summer time.

Implicit Solver:

We can also use implicit solver to reduce oscillation. However, as I said, the instability does not seem to be caused by the CFL condition. It does not seem to be helpful that much. To implement the implicit time matching for mass energy and momentum equations (Not the G equation as it will be unstable), all the lefthand side 1st order backward Euler terms are move to the righthand side with the residues. Then, the resulted equations are set as objectives for the MATLAB implicit solver fsolve. The code is shown as follows.

```
function F = Objective(Xnp1,Xn,Switch,dx,dt,hf0,dhf,Cv,R,mu)
    rho_np1    = Xnp1(1:(1/3)*length(Xnp1));
    rhou_np1   = Xnp1((1/3)*length(Xnp1)+1:(2/3)*length(Xnp1));
    rhoEne_np1 = Xnp1((2/3)*length(Xnp1)+1:end);
    rho_n      = Xn(1:(1/3)*length(Xn));
    rhou_n     = Xn((1/3)*length(Xn)+1:(2/3)*length(Xn));
    rhoEne_n   = Xn((2/3)*length(Xn)+1:end);
    u_np1      = rhou_np1./rho_np1;
    Ene_np1    = rhoEne_np1./rho_np1;
    [Resrho,Resrhou,ResrhoEne,~] = Residual(Switch,rho_np1,u_np1,Ene_np1,dx,hf0,dhf,Cv,R,mu);
    F1 = (1/dt)*(rho_np1 - rho_n) - Resrho;
    F2 = (1/dt)*(rhou_np1 - rhou_n) - Resrhou;
    F3 = (1/dt)*(rhoEne_np1 - rhoEne_n) - ResrhoEne;
    F = [F1,F2,F3];
end
```

Sample result is shown as follows.

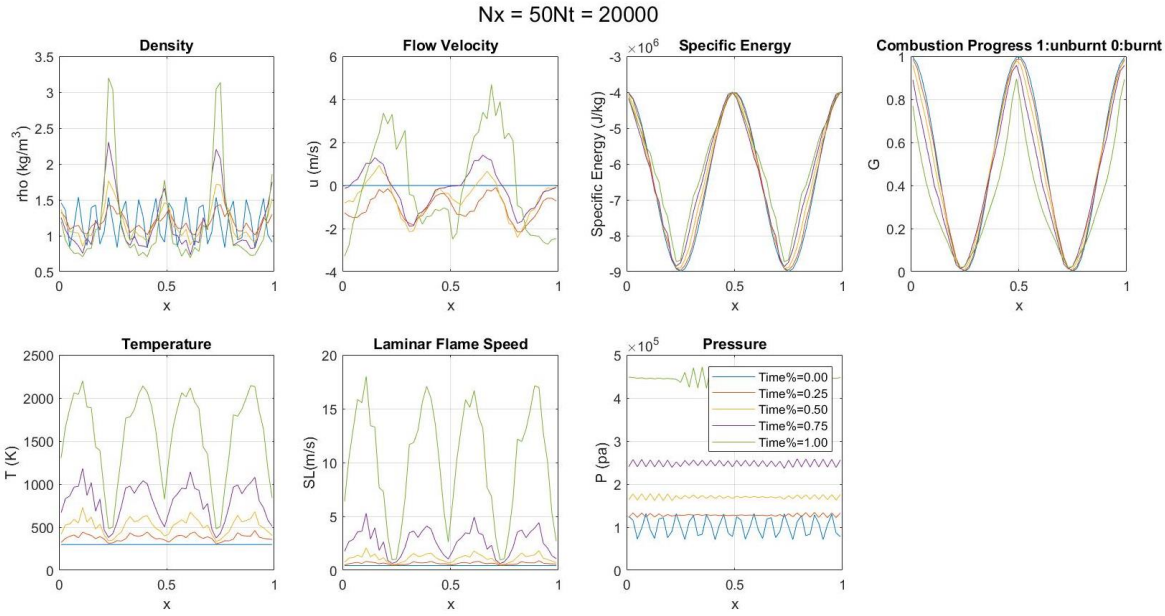


It can be seen that even oscillatory initial condition is given, the solution can be resolved. The current total time show above is for 10 ms. However, with the same time step, simulation as long as 4 sec can be complete. And the moving distance of the wave is also confirmed to be consistent with the advection speed. However, it is found that the solution can only be resolved with certain initial condition because the first step implicit calculation is very hard to converge given a trivial initial guess. In contrast, so long as the first step converges, the rest of the steps seems to be alright. Many attempts have been tested to generate good initial conditions, including 3rd order Runge Kutta and first order Euler or reduced time step implicit method. However, non of these methods seems to generate good enough initial guess for the first stage to converges for certain initial condition. Thus, this subject needs to be studied further.

Filter:

What if I use a filter to smooth out the solution? would that avoid instability? Another method I try is to use Matlab built in filter, smoothdata(). There are three way I try to apply the filter. 1. Apply filter to velocity every certain numbers of time steps. 2.Apply filter to the velocity term when the step variance is larger than certain value. 3.Apply filter to the pressure residue in the momentum equation locally at region where we observed a W shape pressure, similar to a flux limiter.

The 3rd method does not seem to work from testing as the oscillation seems to preserve. For the 1st method, I smooth the velocity every 10 steps for a total time step of 20000 for a total time of 30 ms. The combustion result can be seen as follows without the used of extremely high viscosity.

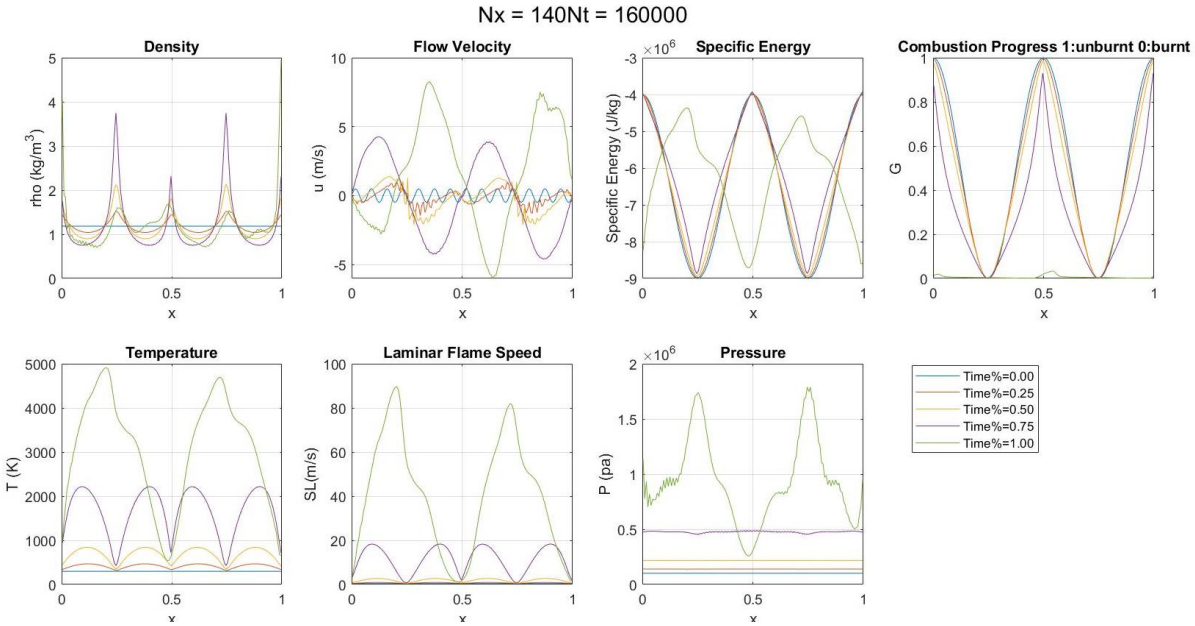


The general physics can see be seen but a lot of quantity like specific total energy and kinetic energy will not be conserved. However, one thing better than using a super high viscosity in this case is that the general physics can still be captured. The fluid will barely move under high viscosity and cannot be used to represent the performance of gas. However, when using filter, the fluid can still flow freely without significant viscosity, just the exact quantitative motion can be captured inaccurately.

To minimize the number of filters applied, I decide to use velocity mean variance to determine when to apply filter. The best case scenario is to keep large scale perturbation in generating turbulence but kill all the super small scale perturbation to avoid instability. The code is implemented as follows.

```
        ulstnp1      = rhoulstnp1./rholstnp1;
        TV = sum(abs(diff(ulstnp1)))/length(ulstnp1);
        if TV > 0.3
%             rholstnp1     = smoothdata(rholstnp1);
            TVCount = TVCount+1;
            rhoulstnp1    = smoothdata(rhoulstnp1);
%             rhoEnelstnp1  = smoothdata(rhoEnelstnp1);
%             rhoGlstnp1    = smoothdata(rhoGlstnp1);
        end
```

When the mean variance TV is higher than 0.3 m/s (imperially tested), a filter to the velocity is applied. The results can be seen as follows with a total time of 40ms and 160000 time steps.



Again, the general physics can be captured while the exact quantity is not correct because of the filter. However, supposedly, this way of filter might perform better than the previous one where filter is applied after certain time steps. In this case, filter is applied only when necessary. Still, after all these attempts, there is no satisfactory solution found. Maybe only resolving the instability from the numerical scheme standpoint can really eradicate the problem, which is again subject to future study this summer.

Ending:

I intend to make this report only a complementary document to my presentation. So, some information from the presentation is not repeated in this report, including the LES formulation and turbulence initiator. This report only discusses different potential failed attempts in trying resolve the instability problem. Though failed, lessons are learnt from them and can be used for future study.

References:

[1] Kim, W.W., Menon, S. and Mongia, H.C., 1999. Large-eddy simulation of a gas turbine combustor flow. Combustion Science and technology, 143(1-6), pp.25-62.

[2] Chandrashekar, P., 2013. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. Communications in Computational Physics, 14(5), pp.1252-1286.

[3] Turns, S.R., 1996. Introduction to combustion (Vol. 287, p. 569). New York, NY, USA: McGraw-Hill Companies.

[4] Li, Y. and Wang, Z.J., 2016. A priori and a posteriori evaluations of sub-grid scale models for the Burgers' equation. Computers & Fluids, 139, pp.92-104.

[5] Brady, P.T., Herrmann, M. and Lopez, J.M., 2012. Code verification for finite volume multiphase scalar equations using the method of manufactured solutions. Journal of Computational Physics, 231(7), pp.2924-2944.