Turbulence Assignment 3

Seyed Mohammadamin Taleghani

January 2024

LES of the Viscous Burgers' Equation using a Spectral Solver

Contents

List of Figures

1. Solving Burgers' Equation using Spectral Method

In this chapter, we aim to solve the viscous Burgers' equation using a spectral method. The significance of the viscous Burgers' equation lies in its resemblance to the momentum equation of the Navier-Stokes (NS) equation, albeit without the complexities of three-dimensional flow and pressure-velocity coupling issues.

Turbulence exhibits spectral characteristics, making spectral decomposition an advantageous method for solving turbulence-related problems. A spectral solver decomposes a Partial Differential Equation (PDE) into an infinite sum of Ordinary Differential Equations (ODE) in Fourier space. Each ODE is then solved independently, and the overall solution of the PDE is obtained by summing the solutions in Fourier space.

Given that we truncate the Fourier modes up to a certain wave number (K_N), a turbulence model is necessary when the cutoff wave number does not cover the majority of the energy spectrum. Consequently, we introduce a spectral eddy-viscosity model to address this issue when solving the Burgers' equation.

In the appendix, we provide a fast and computationally efficient MATLAB code that avoids using *if/else* conditions.

1.1. Governing Equations

The viscous Burgers' equation without a forcing term is given by:

$$\partial_t u + u\partial_x u = \frac{1}{Re}\partial_{xx} u \tag{1.1}$$

The transformation of this equation to Fourier space is already discussed in the class and lecture notes, yielding the final equation:

$$\partial_t \hat{u}_k + \sum_{k=p+q} \hat{u}_p iq \hat{u}_q = -\nu k^2 \hat{u}_k \quad k = 0, \ldots, N \tag{1.2}$$

where $\nu = \frac{1}{Re}$.

When employing a Large Eddy Simulation (LES) with a spectral Subgrid-Scale (SGS) model, the velocity $\hat{u}$ becomes filtered ($\hat{\bar{u}}$), and the diffusive term is modified as:

$$\partial_t \hat{u}_k + \sum_{k=p+q} \hat{u}_p iq \hat{u}_q = -k^2 \nu_{eff}(k) \hat{u}_k \quad k = 0, \ldots, N \tag{1.3}$$

where $\nu\text{eff}(k) = \nu + \nu_t(k)$, and ν_t is the turbulent viscosity.

The turbulent viscosity ν_t is calculated using the eddy-viscosity model proposed by Metais and Lesieur:

$$\begin{cases} \nu_t = \nu_t^{+\infty} \left(\dfrac{E_{k_N}}{k_N} \right)^{1/2} \nu_t^* \left(\dfrac{k}{k_N} \right) \\ \nu_t^{+\infty} = 0.31 \dfrac{5-m}{m+1} \sqrt{3-m} C_K^{-3/2} \\ \nu_t^* = 1 + 34.5 e^{-3.03(k_N/k)} \end{cases} \tag{1.4}$$

where k_N is the cutoff wave number, E_{k_N} is its corresponding energy, m is the slope of the inertial region of the energy spectrum, and C_K is the Kolmogorov constant. For the 1D Burgers equation $m = 2$ and $C_k \approx 0.4523$.

1.2. Numerical Method

The forward Euler formula can easily discretize equations 1.2 and 1.3. However, the triadic convection term should be split into a summation in terms of p and q for the sake of coding. For any $0 \leq k \leq N$, $\sum_{k=p+q} = \sum_{p=k-N}^{N} \Big|_{q=k-p}$. Therefore, Equation 1.3 can be discretized as:

$$\hat{u}_k^{n+1} = \hat{u}_k^n + \Delta t \cdot \left(- \sum_{p=k-N}^{N} \Big|_{q=k-p} \hat{u}_p^n iq \hat{u}_q^n - k^2 \nu_{eff}(k) \hat{u}_k^n \right) \quad k = 0, \ldots, N \tag{1.5}$$

At each time step, we use the conjugate property to calculate $\hat{u}_k^n$ for $-N \leq k \leq -1$:

$$\hat{u}_k^n = -\hat{u}_{-k}^n, \quad k = -N, \ldots, -1 \tag{1.6}$$

The energy at each time step is calculated using:

$$\hat{E}_k^n = \hat{u}_k^n \overline{\hat{u}_k^n} \tag{1.7}$$

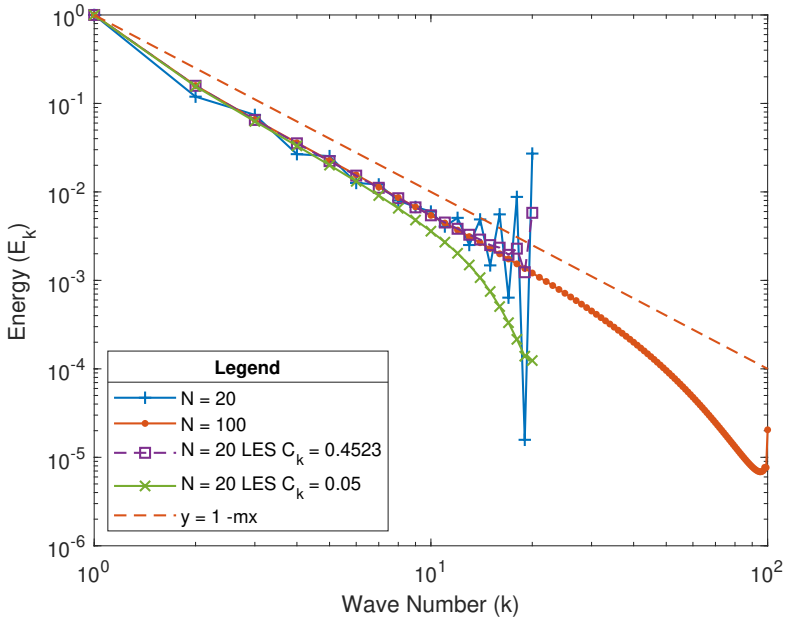

Figure 1.1.: Energy spectrum of the steady-state solution of the Burgers equation for different cases.

1.3. Results and Discussion

Figure 1.1 presents the energy spectrum for different scenarios. For $Re = 40$, a cutoff wave number of 100 captures the energy spectrum up to the dissipation region. Conversely, the $N = 20$ case is limited to the inertial region, necessitating a turbulent viscosity model.

The two LES cases illustrate the effect of C_K on the spectrum. A too-small value for C_K results in the dissipative region starting from smaller wave numbers, making the flow excessively dissipative, as observed in the $C_K = 0.05$ case.

Figure 1.2 demonstrates a parametric study on the effect of Re number on the energy spectrum. Increasing Re makes the dissipative region appear at higher wave numbers, while the inertial region remains unaffected. This observation aligns with the expected behavior: for a given k, an increase in Re leads to a decrease in viscosity, resulting in reduced diffusion.:

for a given k: $Re \uparrow \quad \rightarrow \quad \nu \downarrow \quad \rightarrow \quad \nu \frac{\partial^2 u}{\partial x^2} \downarrow$

1.4. Conclusion

In this study, we explored the numerical solution of the viscous Burgers' equation using a spectral element method. The significance of this equation lies in its similarity to the momentum equation of the Navier-Stokes equation, making it a valuable model for understanding turbulence dynamics without the complexities of three-dimensional flow and pressure-velocity coupling.

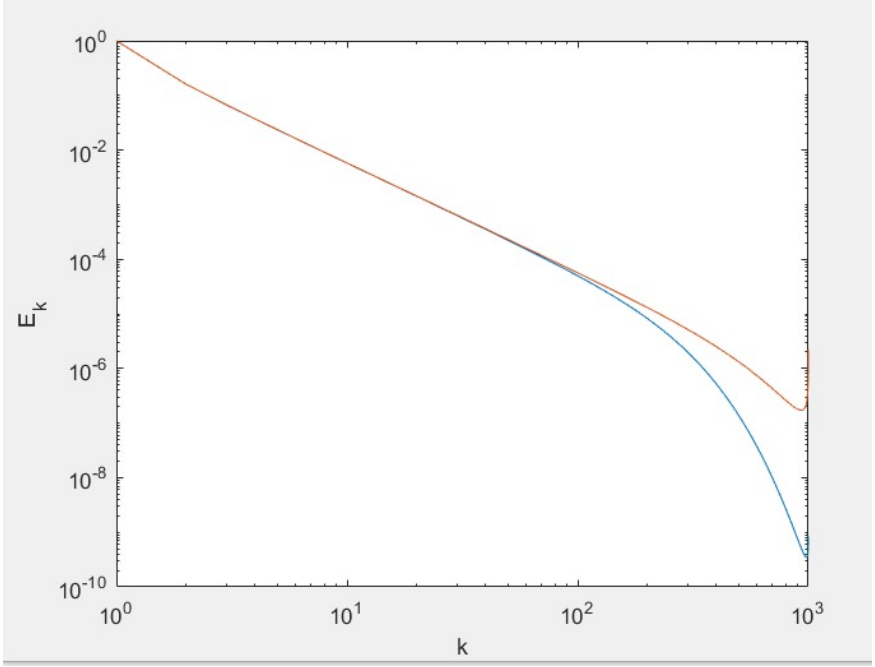

Figure 1.2.: Effect of Re number on the energy spectrum. Blue curve: $Re = 200$, Orange curve: $Re = 500$.

We found that employing a spectral method offers several advantages for solving turbulence problems, given the spectral nature of turbulence. By decomposing the partial differential equation into an infinite sum of ordinary differential equations in Fourier space, we can efficiently solve for the system's dynamics.

Furthermore, we introduced a spectral eddy-viscosity model to account for unresolved turbulent motions, demonstrating the importance of incorporating turbulence modeling in simulations where the cutoff wave number does not capture the entire energy spectrum.

Our numerical results revealed insights into the behavior of the energy spectrum under different scenarios. Notably, we observed the influence of the Reynolds number on the appearance of the dissipative region in the spectrum, highlighting the importance of Reynolds number considerations in turbulence studies.

In conclusion, our study sheds light on the numerical solution of Burgers' equation and its implications for understanding turbulence dynamics. Future research could explore more sophisticated turbulence models and investigate the applicability of spectral methods to more complex flow configurations.

A. MATLAB code

```
clear
%close
clc

% --- 1. Input Data ---------------------------------------
% --- 1.a. Physical ---------------------------------------
Re = 40;

% --- 1.b. Numerical --------------------------------------
N = 20; % Cutoff Fourrier mode
c = 0.1; % CFL condition
dt = c * Re/(N^2);
maxIter=1e5; % Maximum number of iterations for the solver
maxRes=1e-3; % Solver residual |Ax-b|<maxRes
sw = 0; % Turbulence model ON/OFF

% --- 2. Previous Calculation and Vector Allocation ------

kvect = (-N:N)'; % allocating vector for Fourrier modes
ki = @(i) i + N+1; % a simple function that links the k values to their corresponding index

uhat = zeros(length(kvect),maxIter); % allocating uhat for all ks and all timesteps
uhat(:,1) = 1./kvect; % Setting initial values for uhat
uhat(ki(0),1)=0;

conv = zeros(1,N); % Vector for convective term;
dif = zeros(1,N); % Variable for diffusive term
nu_t_star = zeros(1,N);
nu_t = zeros(1,N);

Ek = zeros(N,maxIter); % Allocating energy (only for positive mode)
Ek(1:N,1) = uhat(ki(1:N),1).*conj(uhat(ki(1:N),1)); % Energy for first time-step


% --- 3. Solve using GS ----------------------------------
res = 1 + maxRes;
iter = 1;
t = 0;

% ---- 3.a. The time loop ---------------------------------
while res>maxRes && iter<maxIter

    uhat(:,iter+1) = uhat(:,iter);

    % ---- 3.b. The k loop ------------------------------
    for k = 2:N

        % ---- 3.b.1 The convective term -------------------
```

```
        conv(k) = 0;

        for p =-(N-k):N
            q = k-p;
            conv(k) = conv(k) + uhat(ki(p),iter+1)*1i*q*uhat(ki(q),iter+1);
        end
        % ---- 3.b.2 LES SGS model --------------------------

        % ---- 3.b.2 nu_t -----------------------------------
        m=2;
        Ck = 1; %0.4523;
        nu_t_inf = 0.31*(5-m)/(m+1)*sqrt(3-m)*Ck^-(3/2);
        nu_t_star(k) = 1+34.5*exp(-3.03*(N/k));
        E_end = conj(uhat(ki(N),iter+1))*uhat(ki(N),iter+1);
        nu_t(k) = nu_t_inf*(E_end/N)^0.5*nu_t_star(k);

        % ---- 3.b.3 Diffusive term  ------------------------------------
        dif(k) = - (1/Re+sw*nu_t(k))*k^2*uhat(ki(k),iter+1);

        % ---- 3.b.4 uhat at next timestep
        uhat(ki(k),iter+1) = uhat(ki(k),iter+1) + dt*(-conv(k) + dif(k));

        uhat(ki(1),iter+1) = uhat(ki(1),iter);
        uhat(ki(-1),iter+1) = uhat(ki(-1),iter);

        % ---- 3.b.5 Calculating Ek
        Ek(k,iter+1) = uhat(ki(k),iter+1)*conj(uhat(ki(k),iter+1)); % Energy for↙
first time-step
        Ek(1,iter+1) = 1;
    end


    % ---- 3.b.5 Using the uhat(-k) = conj(uhat(k)) to calculate uhat
    % for negative k's
    uhat(ki(-N:-1),iter+1) = conj(uhat(ki(N:-1:1),iter+1));

    for k = 1:N
    % ENERGY EQUATION
    %Ek(k,ite+1) =   uhat(ki(k),ite+1) * conj(uhat(ki(k),ite+1));
    end


    % ---- 4. calculating the new time
    res = max( abs(uhat(:,iter+1)-uhat(:,iter))/dt )
    t = dt + t;
    iter = iter + 1;
end

iter = iter - 1;
```

```
t = 0:dt:t;

%%
figure(1)
loglog(1:N,Ek(:,iter))

%%

x= -10:0.1:10;

% Creating and opening video files for animating P-h and T-s diagrams
v1 = VideoWriter('burgersUnderResolved.avi');

open(v1)
figure(2)
u = zeros(length(x),length(t));
for j = 1:3:iter
    for i = 1:length(x)
    u(i,j) = sum(uhat(:,j).*exp(1i*kvect*x(i)));
    end
    subplot(1,2,1)
    plot(x,u(:,j))
    xlabel('x')
    ylabel('u')
    title('t = ',num2str(t(j)))

    subplot(1,2,2)
    loglog(1:N,Ek(:,j))
    xlabel('k')
    ylabel('E_k')
    frame=getframe(gcf);
    writeVideo(v1,frame);
end

close(v1);
% closing video files
```