Universitat Politècnica de Catalunya

ESCUELA SUPERIOR DE INGENIERÍA INDUSTRIAL, AEROESPACIAL Y AUDIOVISUAL DE TERRASSA (ESEIAAT)

ASSIGNMENT 4: TURBULENCE. BURGERS' EQUATION IN THE FOURIER SPACE

Computational Engineering - Space and Aeronautical Engineering MSc

Author:

Jorge SIMÓN AZNAR

Contents

1 Proposal

The primary goal of this task is to gain a fundamental understanding of turbulence by exploring interactions between different scales, and the significance of convective and diffusive terms.

To achieve this, we will solve Burgers' equation in Fourier space. The initial examples will involve Reynolds number (Re) values of 40 and two different grid resolutions, namely N=20 and N=100. Subsequently, we will consider additional configurations.

In the second phase, we will introduce the proposed Large Eddy Simulation (LES) model into our analysis.

2 Introduction

Turbulence is characterized by the presence of chaotic and unpredictable flow states that introduce instability in various properties such as pressure, velocity, and temperature within a fluid. In contrast, the Reynolds number serves as a dimensionless parameter that quantifies the balance between inertial and viscous forces in a fluid, helping determine whether the flow is in a laminar state (when Re is below the critical value) or turbulent (when Re exceeds this critical threshold).

In the context of this problem, the Burger's equation serves as a valuable tool for comprehending the energy transfer mechanisms within turbulent flow.

To initiate our exploration, we consider the Navier-Stokes equations, which offer a suitable model for describing the nonlinear dynamics of turbulence, particularly in the case of incompressible flow:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\nabla)\mathbf{u} = \frac{1}{Re}\nabla^2\mathbf{u} - \nabla p \tag{2.1}$$

$$\nabla \mathbf{u} = 0 \tag{2.2}$$

Where Re is the dimensionless Reynolds number defined as

$$Re = \frac{V_0 L}{\nu} \tag{2.3}$$

where V_0 is the free stream velocity, L the characteristic length and ν the kinematic viscosity.

It's crucial to emphasize that the convective term within the Navier-Stokes equations leads to the generation of numerous dynamically significant scales of motion. This poses a significant challenge when attempting to conduct Direct Numerical Simulations (DNS) for the Navier-Stokes equations, as it demands substantial computational resources.

To address this challenge, an alternative simplified model that preserves several essential features of the Navier-Stokes equations is proposed: the Burgers' equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} + f \tag{2.4}$$

The study of the eq.2.4 is better carried out in Fourier space. Considering the equation on an interval with periodic boundary conditions, after some development, the Burgers' equation reads:

$$\frac{\partial \hat{u}_k}{\partial t} + \sum_{p+q=k} \hat{u}_p i q \hat{u}_q = -\frac{k^2}{Re}\hat{u}_k + \hat{F}_k \tag{2.5}$$

Moreover, considering that $u(x,t) \in \mathbb{R}$, it becomes necessary to enforce:

$$\hat{u}_k = \overline{\hat{u}_{-k}} \tag{2.6}$$

This condition implies that for $k < 0$, the velocity value at that particular k is the complex conjugate of the velocity at the corresponding k with a positive value. Consequently, we only need to solve for values of $k > 0$.

The second term of eq.2.5 represents the non-linear component of the Burgers equation, specifically the convective term. Essentially, the algorithm outlined in this study aims to solve the equation provided above. Considering that $\hat{u}_k$ can be a complex number, the analysis of the results will be conducted in terms of the energy E_k:

$$E_k = \hat{u}_k \overline{\hat{u}_k} \in \mathbb{R} \tag{2.7}$$

2.1 Large-Eddy Simulations

To enhance the accuracy of results for small values of N in Direct Numerical Simulations (DNS) of the Burgers equation, Large-Eddy Simulation (LES) will be explored. We will adopt the approach proposed by Métais and Lesieu [1], involving a modification of the ν value (linked to the Reynolds number, as shown in eq. 2.3):

$$\nu_t\left(\frac{k}{k_N}\right) = \nu_t^{+\infty}\left(\frac{E_{k_N}}{k_N}\right)^{\frac{1}{2}} \nu_t^*\left(\frac{k}{k_N}\right) \tag{2.8}$$

where

$$\nu_t^{+\infty} = 0.31\frac{5-m}{m+1}\sqrt{3-m}C_K^{\frac{-3}{2}} \tag{2.9}$$

$$\nu_t^*\left(\frac{k}{k_N}\right) = 1 + 34.5e^{-3.03(k_N/k)} \tag{2.10}$$

In this context, m represents the slope of the energy spectrum, which is $m = 2$ in our case. E_{k_N} stands for the energy at the cutoff frequency k_N, and C_K denotes the Kolmogorov constant. The final kinematic viscosity is then given by:

$$\nu_{eff}(k) = \nu + \nu_t(k) \tag{2.11}$$

3 Problem Description

Before delving into the solution of eq. 2.5, it's important to address some specific details regarding the problem. As previously mentioned, it has been truncated to $2N$ modes, effectively dividing the k space into small segments ranging from $-N$ to N. This truncation is a crucial aspect of the problem formulation.

The temporal evolution of the velocity is addressed through an explicit scheme, expressed as:

$$\frac{\hat{u}_k^{n+1} - \hat{u}_k^n}{\Delta t} \quad ; \quad \Delta t < C_1 \frac{Re}{N^2} \tag{3.1}$$

where $0 < C_1 < 1$. Using the above expression eq. 2.5 can be discretized in time as it follows:

$$\hat{u}_k^{n+1} = \hat{u}_k^n + \Delta t \left[-\frac{k^2}{Re} \hat{u}_k^n + C_{onv}(q,p)^n \right] \tag{3.2}$$

Here $C_{onv}(q,p) \equiv \sum_{p+q=k} \hat{u}_p iq \hat{u}_q$ represents the convective term. The initial state is set as $\hat{u}_k^0 = 1/k$ with the assumption that there is no interaction of $k = 0$ mode so that $\hat{u}_0^n = 0$. In addition, we will impose that $\hat{u}_1^n = 1$.

Ultimately, we will engage in a concise examination of the convective term, $C_{onv}(p,q)$. While p and q can assume values from $-N$ to N, our focus will be solely on computing the convective terms for those specific values of p and q falling within the lines of $k = 0$ and $k = N$ in the p vs q diagram, as illustrated in the following figure:

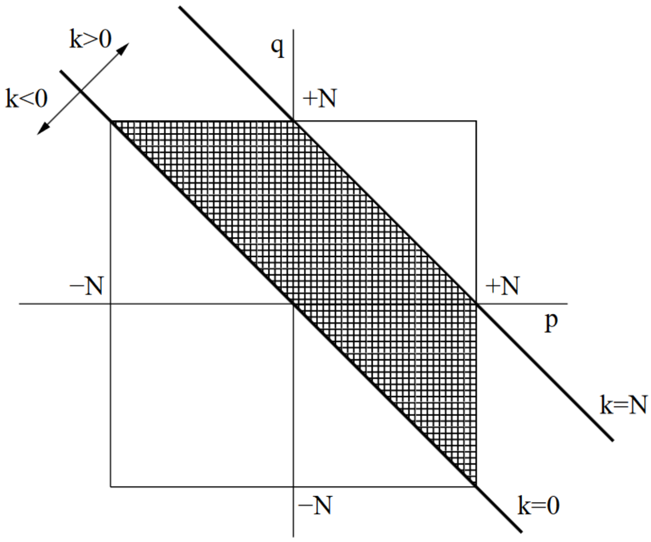


Figure 3.1: Possible triadic interaction between modes.

4 Code Structure

In this section, we outline the structure of the code written in Python3. The code implementation involves the following steps:

1. Define input data, categorizing it into two sections; physical such as Reynolds number (Re), kinematic viscosity (ν), etc., and numerical parameters like the number of modes (N), convergence criteria (δ), etc.

2. Define complex vectors for the velocity $\hat{u}_k^n$ and $\hat{u}_k^0$ and initialize them, along with the energy spectrum E_k.

3. New time step: $t^{n+1} = t^n + \Delta t$.

4. Compute the convective term $C_{onv}(q,p) \equiv \sum_{p+q=k} \hat{u}_p i q \hat{u}_q$.

5. Compute the updated velocity $\hat{u}_k^{n+1}$ using eq. 3.2.

6. Check for steady state: if $|\hat{u}_k^{n+1} - \hat{u}_k^n| < \delta$ is fulfilled, then the steady state has been obtained, and you can proceed to step 7. If not, set $\hat{u}_k^n = \hat{u}_k^{n+1}$ and go back to step 3.

7. Compute the energy spectrum E_k using eq. 2.7.

8. Final calculations and print results.

5 Code Verification

Prior to presenting the achieved outcomes in this study, it is imperative to conduct an analysis ensuring the proper functionality of the code. A comparison will be made with the results presented in figures 2 and 3 of the CTTC report [2].

Initially, a DNS calculation was carried out for $Re = 40$ and $C_1 = 0.02$ with two distinct mode quantities: $N = 20$ and $N = 100$. The resulting energy spectrum is depicted in Fig. 5.1. It is discernible that the solution for $N = 20$ is under-resolved, while the solution for $N = 100$ is fully resolved. This behavior aligns with observations in the second figure of the CTTC report. Notably, for the solution with $N = 100$ at the last mode, a small peak is observed. This phenomenon is a numerical issue where there is an energy transfer to modes beyond N, consistent with observations in the CTTC report. Lastly, it is noteworthy that for high k values in DNS with $N = 100$, energy dissipation occurs, resulting in a deviation from the reference slope of $m = -2$. This effect is induced by the diffusive term in eq. 3.2, introducing damping in the energy spectrum.

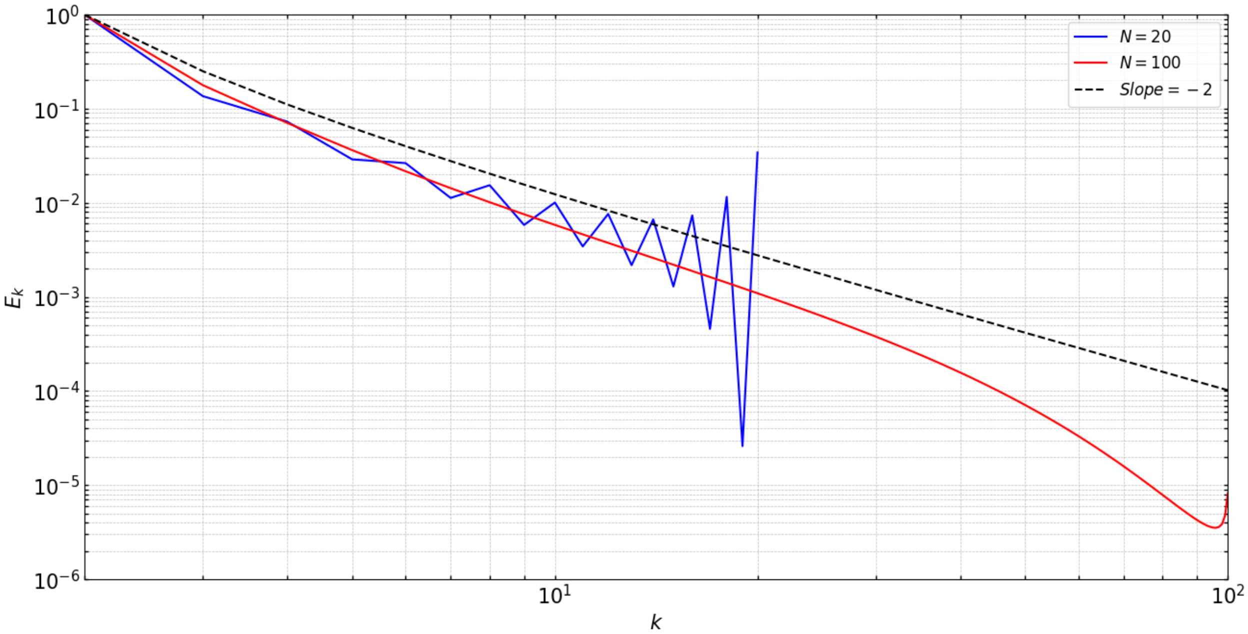
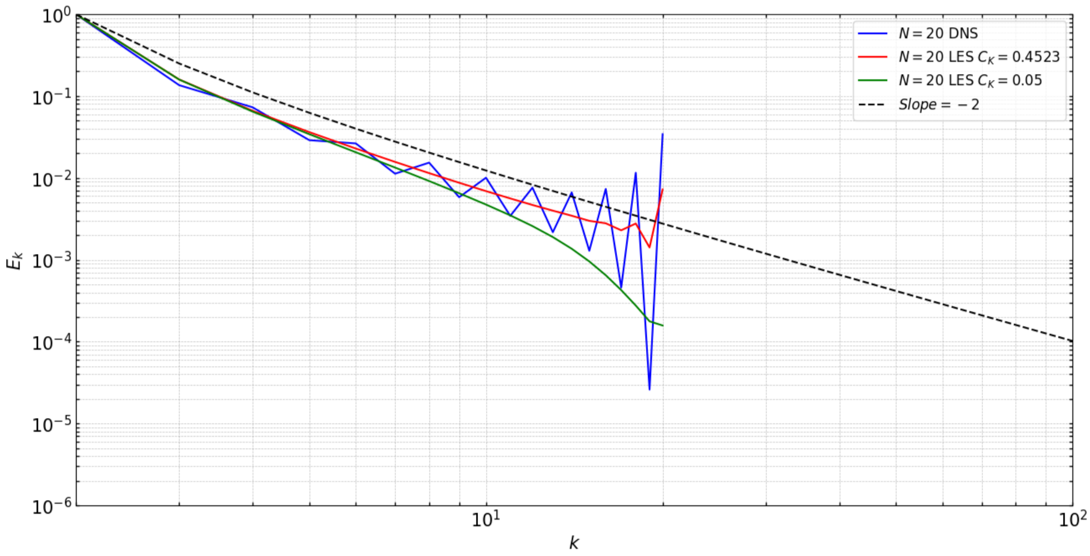
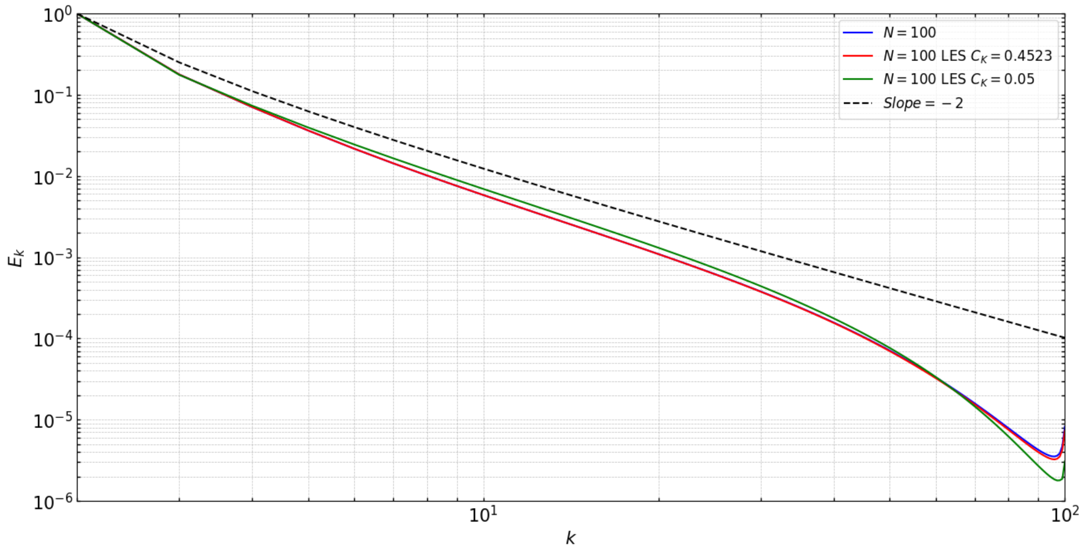


Figure 5.1: DNS solution for $N = 20$ and $N = 100$.

Subsequently, LES calculations were conducted for $N = 20$ and $N = 100$, $Re = 40$, and $C_1 = 0.02$, employing two different Kolmogorov constants, $C_k = 0.4523$ and $C_k = 0.05$. The resulting energy spectrums are presented in Fig. 5.2, with the energy spectra overlaying the results shown in Fig. 5.1. It is clear that LES enhances the solution for a small number of modes, a trend also observed in the reference result from the CTTC (figure 3).



(a) N = 20



(b) N = 100

Figure 5.2: Comparison of DNS and LES solution for $N = 20$ and $N = 100$.

Despite achieving similar results, the reference values display smaller errors compared to the DNS $N = 100$ result, which could be influenced by the choice of the C_1 value. Additionally, it is intriguing to observe how the LES solution varies with the Kolmogorov constant value. For smaller values of C_k, we notice the results deviating from the DNS $N = 100$ line at smaller E_k values. Conversely, with larger C_k values, the deviation occurs at higher E_k values. In this scenario, three smaller peaks (compared to those of DNS $N = 20$) are obtained, consistent with the CTTC reference solution.

6 Results

After validating the algorithm, the subsequent phase involves a comprehensive investigation of the influence of problem parameters on the solution. This examination is conducted through both DNS and LES and alterations are introduced in both the number of Fourier modes and the Reynolds number.

The Reynolds number is linked to the diffusive term in the kinetic energy transport equation, as shown in [2]: $(-2k^2/Re)E_k$. Modifying its value induces changes in the damping of the energy spectrum.

Then, at first, we will see the comparison of DNS solutions for $Re = (10, 70)$ and $N = (20, 100)$:

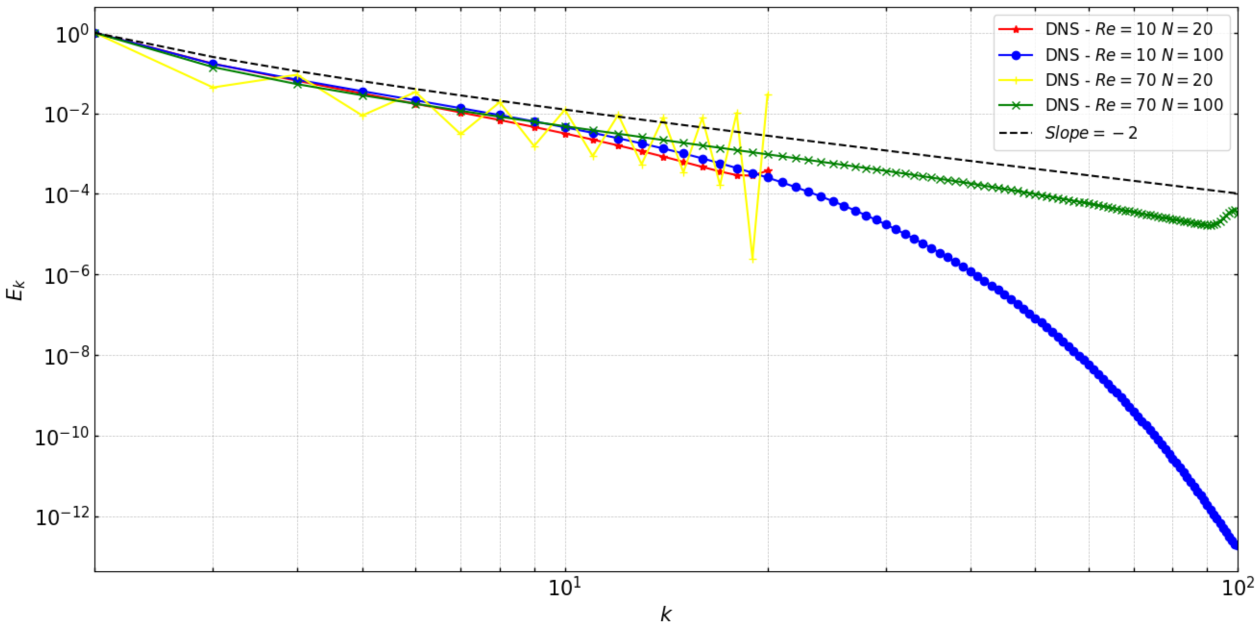


Figure 6.1: DNS solution with $Re = (10, 70)$ for $N = 20$ and $N = 100$.

In high k modes, significant changes in energy dissipation are observed for different Reynolds numbers, as anticipated. Notably, the lowest dissipation is associated with the highest Reynolds number, whereas the highest energy dissipation occurs with the lowest Reynolds number. Furthermore, the energy spectrum at the highest Reynolds number closely follows the reference slope of $m = -2$. In summary, it is clear that alterations in the Reynolds number have a direct impact on the diffusive term of the Burgers equation, thereby influencing the dissipative aspect of the energy spectrum.

As for the LES mode, we have carried out a similar approach, obtaining the results for a Reynolds number of 10 and 70, varying the number of Fourier modes from 20 to 100 and using the two Kolmogorov constants seen in the previous section.

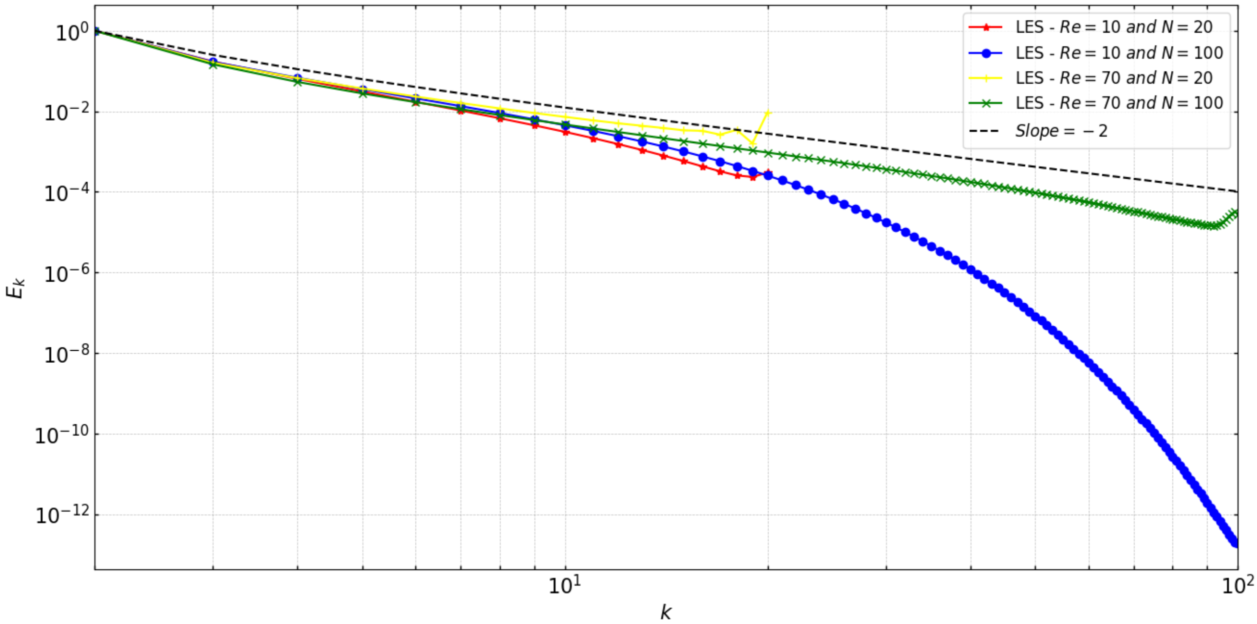


Figure 6.2: LES solution with $Re = (10, 70)$ for $N = 20$ and $N = 100$ and $C_k = 0.4523$.

In the lower k modes, a consistent behavior is observed across all four scenarios, maintaining a shared trend. Nevertheless, as we move to higher wavenumber modes, solutions with larger Reynolds numbers tend to produce higher energy spectrum values, whereas lower Reynolds numbers lead to lower energy spectrum values.

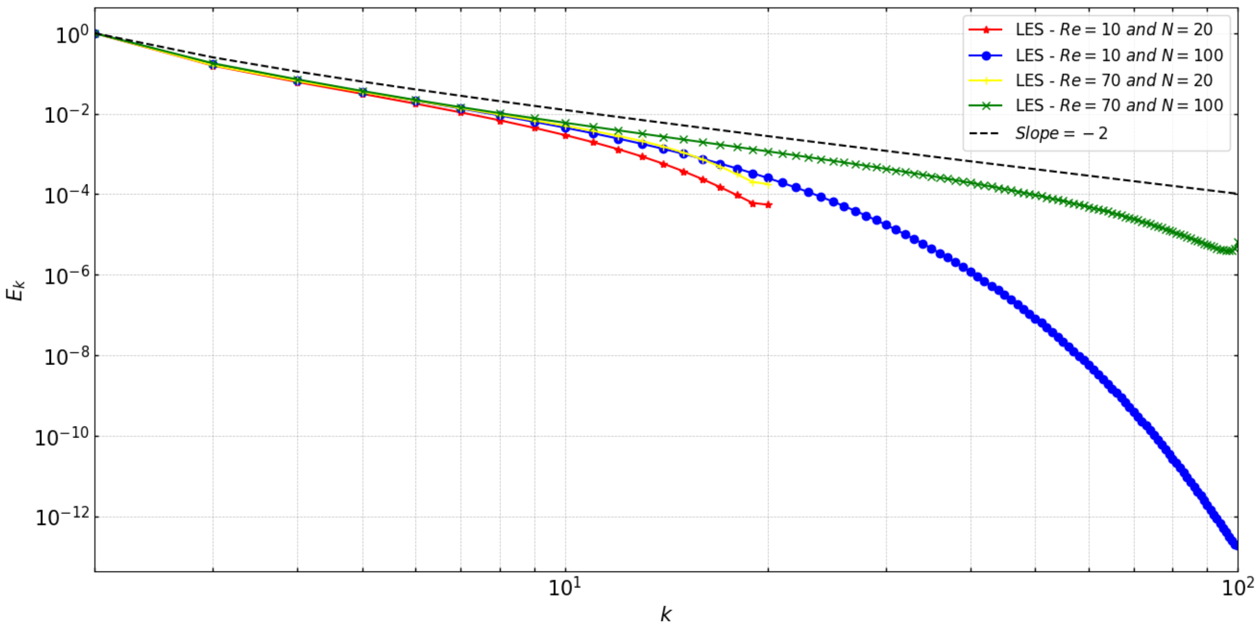


Figure 6.3: LES solution with $Re = (10, 70)$ for $N = 20$ and $N = 100$ and $C_k = 0.05$.

Let's mention the difference when varying Kolmogorov constant using the same values of N and Re. Similar to the Reynolds variation case, in the lower k modes, all results exhibit consistent behavior, maintaining a shared trend. However, in higher modes, solutions with larger C_k values tend to yield higher energy spectrum values, while lower C_k values result in lower energy spectrum values. Therefore, adjusting the C_k values offers a means to tailor our results to the desired outcomes.

7 Conclusions

In this study, we explored the Burgers equation in Fourier space and developed a code to conduct both Direct Numerical Simulation (DNS) and Large-Eddy Simulations (LES).

To ensure the reliability of our code, we performed a thorough validation by comparing our results with the data from the CTTC report [2]. We conducted DNS and LES simulations with varying mode quantities, revealing distinctions between under-resolved and fully resolved solutions.

We have focused on investigating the influence of Reynolds number, number of Fourier modes and Kolmogorov constant for LES mode on the Burgers equation. DNS simulations (with $N = 100$) at different Reynolds numbers unveiled an anticipated trend: higher Reynolds numbers led to reduced energy dissipation in high k modes, while lower Reynolds numbers resulted in increased dissipation. Notably, the energy spectrum at the highest Reynolds number closely matched the expected slope of $m = -2$, illustrating the impact of Reynolds number on the Burgers equation.

In the context of LES, consistent behavior has been observed in the lower k modes across all four scenarios. As we progressed to higher k modes, solutions with higher Reynolds numbers tended to exhibit higher energy spectrum values, while lower Reynolds numbers were associated with lower energy spectrum values. Furthermore, we extended our analysis to consider different values of C_k, revealing a similar trend. In the lower k modes, all results displayed consistent behavior, and in the higher k modes, larger C_k values generally corresponded to higher energy spectrum values, while lower C_k values were associated with lower energy spectrum values.

8 Code

In this section I show the code for the Burgers' equation problem. I will also send it by ATENEA in .zip format.

```
import matplotlib.pyplot as plt
import numpy as np

## INPUT DATA
# Numeric
N        = [20,100,20,100] # Number of Fourier modes
Ck       = [0.05,0.05, 0.05, 0.05]  # Kolmogorov
C        = 0.02  # Constant for explicit scheme integration, 0<C<1
time_max = 100000000000
delta      = 1.0E-06 # Accuracy
m        = 2
LES      = 1 # If LES=1 we compute LES, if not DNS

# Physical
L  = 1.0 # Characteristic lenght
U  = 1.0 # Characteristic viscosity
nu = [1.0/10.0, 1.0/10.0, 1.0/70.0, 1.0/70.0]  # Kinematic viscosity, Re in denominator
i  = 1j

# Solver
for Z in range(len(N)):
    delta_T  = (C * (1.0/nu[Z])) / N[Z]**2 # Time step

    uk      = np.zeros(N[Z], dtype=complex)
    uk0 = np.zeros(N[Z], dtype=complex)
    Ek      = np.zeros(N[Z])

    # Only k>0 values
    uk[0]      = 0.0
    uk0[0]  = 0.0
    uk[1]      = 1.0
    uk0[1]  = 1.0
    for k in range(2, N[Z]):
        uk0[k] = 1.0 / float(k+1)

    # Begin time loop
    for time in range(time_max):
        t0 = np.abs(np.sum(uk0))
        for k in range(2, N[Z]):

            # Convective term
            conv = 0
            for p in range(-N[Z]+1,N[Z]):
                q = k - p
                if (q < -N[Z] + 1) or (q >= N[Z]):
                    uq = 0.0
                    up = 0.0
                else:
                    uq = np.conjugate(uk0[np.abs(q)]) if q < 0 else uk0[q]
                    up = np.conjugate(uk0[np.abs(p)]) if p < 0 else uk0[p]
                conv = conv + q * i * uq * up

            # n+1 velocity
            if LES == 0:
                uk[k] = uk0[k] - delta_T * (nu[Z] * k**2 * uk0[k] + conv)
            elif LES == 1 :
                nu_inf = 0.31 * ((5 - m) / (m + 1)) * np.sqrt(3 - m) * Ck[Z] ** (-3.0 / 2.0)
                EkN    = (uk0[-1] * np.conjugate(uk0[-1])).real
```

```
                nu_a   = 1.0 + 34.5 * np.exp(-3.03 * (N[Z] / k) )
                nu_t   = nu_inf * ((EkN / N[Z]) ** (0.5) * nu_a)
                nu_eff = nu[Z] + nu_t
                uk[k] = uk0[k] - delta_T * (nu_eff * k**2 * uk0[k] + conv)

        t = np.abs(np.sum(uk))
        if np.abs(t-t0) < delta:
            break
        else:
            uk0 = uk

    # Energy spectrum
    for k in range(N[Z]):
        Ek[k] = (uk[k] * np.conjugate(uk[k])).real
    if Z == 0 and LES == 1:
        k_cv1 = np.linspace(1, N[0], N[0])
        Ek1   = np.zeros(N[0])
        Ek1   = Ek
    elif Z == 1 and LES == 1:
        k_cv2 = np.linspace(1, N[1], N[1])
        Ek2 = np.zeros(N[1])
        Ek2 = Ek
    elif Z == 2 and LES == 1:
        k_cv3 = np.linspace(1, N[0], N[0])
        Ek3   = np.zeros(N[0])
        Ek3   = Ek
    else:
        k_cv4 = np.linspace(1, N[1], N[1])
        Ek4 = np.zeros(N[1])
        Ek4 = Ek

## PLOT
# Reference lines
X = np.linspace(1, N[1], N[1])
Y = np.linspace(1, N[1], N[1])
for i in range(1,N[1]):
    Y[i] = (X[i])**-2

# Figure specifications
fontsize=15

# Start plot
plt.figure(1)
plt.figure(figsize = (10,8))
plt.tick_params(axis='both', which='both',length=3, width=1.0,
labelsize=15, right=True, top=True, direction='in')

# Labels
plt.ylabel(r"$E_{k}$", fontsize=fontsize)
plt.xlabel(r"$k$", fontsize=fontsize)

# Limits
plt.xlim(k_cv1[1], N[1])

# Grid
plt.grid(True, which='both', linestyle='--', linewidth=0.5, color='gray', alpha=0.5)

# Plot
plt.loglog(k_cv1[1:], Ek1[1:],'-*', color='red', label=r"LES␣-␣$Re␣=␣10␣~␣and␣~␣N␣=␣20$")
plt.loglog(k_cv2[1:], Ek2[1:], '-o', color='blue', label=r"LES␣-␣$Re␣=␣10␣~␣and␣~␣N␣=␣100$")
plt.loglog(k_cv3[1:], Ek3[1:],'-+', color='yellow', label=r"LES␣-␣$Re␣=␣70␣~␣and␣~␣N␣=␣20$")
plt.loglog(k_cv4[1:], Ek4[1:], '-x', color='green', label=r"LES␣-␣$Re␣=␣70␣~␣and␣~␣N␣=␣100$")
plt.plot(X + 1, Y, linestyle="dashed", color="black", label=r"$Slope=-2$")

```

```
# Legend
plt.legend(fontsize=12, loc='upper right')

plt.show()
plt.close(1)
```

Bibliography

[1] O. Métais and M. Lesieur, “Spectral large-eddy simulation of isotropic and stably stratified turbulence”, Journal of Fluid Mechanics **239**, 157–194 (1992).

[2] CTTC, “Burgers equation in fourier space”, (November 17 th 2014).