LES of Turbulent Flows: Lecture 17

Dr. Jeremy A. Gibbs

Department of Mechanical Engineering
University of Utah

Fall 2016

Overview

1. Stochastic Burgers Equation

Stochastic Burgers Equation

- Originally conceived by Dutch scientist J.M. Burgers in the 1930s
- One of the first attempts to arrive at the statistical theory of turbulent fluid motion

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$$

This represents a very simplified model that describes the interaction of non-linear inertial terms and dissipation in the motion of a fluid

Stochastic Burgers Equation

The original Burgers equation shares a lot in common with the N-S equations

- advective nonlinearity
- diffusion (can compute Re)
- invariance and conservation laws, such as translation in space and time

Stochastic Burgers Equation

- There were downsides discovered
- The equation can be integrated explicitly – meaning that it does not share the N-S equations sensitivity to small changes in initial conditions in presence of boundaries, forcing, and at high Re
- Thus Burgers equation is not an ideal model for the chaotic nature of turbulence
- It was also found that shock waves form in the limit of vanishing viscosity

Stochastic Burgers Equation

- The use of Burgers with a forcing term has been popular because the original model is an incomplete description of a turbulent system
- The forcing term can account for the neglected effects
- For instance, one may perturb the system with a stochastic process that is stationary in time and space (this preserves translational invariance)
- One example is white noise - which preserves Galilean invariance

Stochastic Burgers Equation

- Project #1 is based on a useful model of the Navier-Stokes equations: the 1D Stochastic Burgers Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} + \eta(x,t)$$

- This analog has been found to be a useful one for the study of turbulent like nonlinear systems (e.g., Basu, 2009 and references contained within).
- Although 1D, this equation has some of the most important characteristics of a turbulent flow – making it a good model case

Stochastic Burgers Equabasution

- In short, the Burgers system has been popular because it allowed one to gain insight into turbulence structure before having to generalize for the fully-3D case
- It shares many characteristics of 3D turbulence, such as nonlinearity, energy spectrum, intermittent energy dissipation
- The system is also super-cheap computationally

Stochastic Burgers Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} + \eta(x,t)$$

- In the above equation, the "new" term is η – which is called the stochastic term
- $\eta(x,t)$ should be white noise in time, but spatially correlated

Stochastic Burgers Equation

- Here we use

$$\eta(x,t) = \sqrt{\frac{2D_0}{\Delta t}}\mathfrak{F}^{-1}\left\{|k|^{\beta/2}\hat{f}(k)\right\}$$

where

$$\begin{aligned} D_0 &= \text{noise amplitude} \\ \Delta t &= \text{time step} \\ \mathfrak{F}^{-1} &= \text{inverse Fourier transform} \\ f &= \text{Gaussian random noise with mean} = 0 \text{ and} \\ &\quad \text{standard deviation} = \sqrt{N} \text{ (where } N \text{ is \# of points)} \\ \beta &= \text{spectral slope of the noise (taken as } -3/4 \text{ here)} \end{aligned}$$

Stochastic Burgers Equation

- Many solutions exist for stochastic Burgers equation
- Here we follow Basu (2009) – Fourier collocation
- Basically, use Fourier methods but advance time in real space (compare to Galerkin)
- To do this, the main numerical methods we need to know are how to calculate <u>derivatives</u> and how to <u>advance time</u>

Fourier Derivatives

- Mathematically the discrete Fourier transform pair is (see also Lecture 3 as a supplement)

$$f(x_j) = \sum_{m=-N/2}^{N/2-1} \hat{f}(k_m) e^{ik_m x_j} \rightarrow \text{backward transform} \quad (1)$$

$$\underbrace{\hat{f}(k_m)}_{\substack{\text{Fourier}\\ \text{coeffs}}} = \frac{1}{N} \sum_{j=1}^{N} f(x_j) e^{-ik_m x_j} \rightarrow \text{forward transform} \quad (2)$$

where

$$k_m = \frac{2\pi m}{N \Delta x} \rightarrow \text{wave number (wave period)}$$

recall

$$e^{-ik_m x_j} = \cos(k_m x_j) + i \sin(k_m x_j)$$

Fourier Derivatives

- How is this used numerically to calculate a derivative?
- A Fourier series can be used to interpolate $f(x_j)$ at any point x in the flow and at any time t
- If we differentiate the Fourier representation of $f(x_j)$ (Eq. 1) with respect to x

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\left[\sum_{m=-N/2}^{N/2-1} \hat{f}(k_m) e^{ik_m x_j}\right]$$

$$= \sum_{m=-N/2}^{N/2-1} \hat{f}(k_m) \frac{\partial e^{ik_m x_j}}{\partial x}$$

$$\boxed{= \sum_{m=-N/2}^{N/2-1} ik_m \hat{f}(k_m) e^{ik_m x_j}}$$

Fourier Derivatives

- If we compare this to Eq. (1), we notice that we have

$$\frac{\partial f}{\partial x} = g = \sum_{m=-N/2}^{N/2-1} \underbrace{ik_m \hat{f}(k_m)}_{\hat{g}(k_m)} e^{ik_m x_j}$$

$$\boxed{= \sum_{m=-N/2}^{N/2-1} \hat{g}(k_m) e^{ik_m x_j}}$$

Fourier Derivatives

Procedurally, we can use this to find $\left.\frac{\partial f}{\partial x_j}\right|_j$ given $f(x_j)$ as follows:

- calculate $\hat{f}(k_m)$ by the forward transform (Eq. 2)
- multiply by ik_m to get $\hat{g}(k_m)$, and then
- perform a backward (inverse) transform (Eq. 1) to get $\left.\frac{\partial f}{\partial x_j}\right|_j$

The method easily generalizes to any order derivative

Fourier Derivatives

Although Fourier methods are quite attractive due to their high accuracy and near-exact representation of derivatives, they have a few important limitations

- $f(x_j)$ must be continuously differentiable
- $f(x_j)$ must be periodic
- grid spacing must be uniform

Time Advancement

- Time advancement in this code is accomplished using a 2nd-order Adams-Bashforth scheme
- This is a basic extension of the Euler method – multipoint (in time), with the idea to fit a polynomial of desired order (*e.g.*, 2nd) through 3 points in time to get

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{2}\left[3f(t^n, \phi^n) - f(t^{n-1}, \phi^{n-1})\right]$$

- For specifics on how these things are implemented, see the Matlab code on the course website or on Canvas