fractal and fractional

Article

A Parallel Computational–Statistical Framework for Simulation of Turbulence: Applications to Data-Driven Fractional Modeling

Ali Akhavan-Safaei [1] **and Mohsen Zayernouri** [2,*]

[1] Departments of Mechanical Engineering & Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA; akhavans@msu.edu

[2] Departments of Mechanical Engineering & Statistics and Probability, Michigan State University, East Lansing, MI 48824, USA

* Correspondence: zayern@msu.edu

Abstract: In this work, an open-source computational–statistical platform to obtain synthetic homogeneous isotropic turbulent flow and passive scalar transport is presented. A parallel implementation of the well-known pseudo-spectral method in addition to the comprehensive record of the statistical and small-scale quantities of the turbulent transport are offered for executing on distributed memory CPU-based supercomputers. The user-friendly workflow and easy-to-run design of the developed package are disclosed through an extensive and step-by-step example. The resulting low- and high-order statistical records vividly verify a well-established and fully developed turbulent state as well as the seamless statistical balance of conservation laws. The post-processing tools provided in this platform would allow the user to easily construct multiple important transport quantities from primitive turbulent fields. The programming codes for this tool are accessible through GitHub (see Data Availability Statement).

Keywords: turbulent transport; passive scalars; statistical analysis; pseudo-spectral method; high-performance computing

check for updates

Citation: Akhavan-Safaei, A.; Zayernouri, M. A Parallel Computational–Statistical Framework for Simulation of Turbulence: Applications to Data-Driven Fractional Modeling. *Fractal Fract.* **2023**, *7*, 488. https://doi.org/10.3390/fractalfract7060488

Academic Editor: Riccardo Caponetto

Received: 3 May 2023
Revised: 9 June 2023
Accepted: 15 June 2023
Published: 20 June 2023

1. Introduction

Understanding the complex and random nature of turbulent flows, mixing, and transport is a vital step in predictions and the design of systems interacting with such a heterogeneous medium. Turbulence inherently consists of multi-scale processes that require highly accurate measurements at the smallest scales of transport [1,2]. The direct numerical simulation (DNS) of turbulent transport as a rigorous scientific tool is supposed to fully resolve the smallest scales of the motion resulting from fluctuating fields in the spatial domain while maintaining a high-order of temporal accuracy as turbulence evolves in time [3]. Therefore, developing an open-source, sustainable, portably parallel, and integrated computational–statistical framework with high-order spatial and temporal accuracy provides a useful academic ground for a better understanding of the complex standard to anomalous turbulent transport across a multitude of scales. Moreover, from an educational point of view, developing such a user-friendly scientific software will essentially fill the existing training gap in the subjective trinity, i.e., fluid mechanics, computational fluid dynamics, and turbulent transport; hence, leading to a more cohesive ramp-up in training the future generation of researchers in a variety of academic-to-industrial disciplines.

Among the current open-source computational frameworks, `Nektar++` [4,5] (the spectral/hp element method flow software), `HERCULIS` [6] and `Xcompact3D` [7] (the high-order finite difference flow solvers), `GRINS` [8] (the adaptive mesh refinement finite element method software), `spectralDNS` [9] (the spectral method computational package for DNS), and `OpenFOAM` [10] are notable contributions to the DNS of turbulent transport. On the other hand, the random nature of turbulence requires a thorough statistical analysis on the fluctuating fields and their gradients so that one can identify when the realistic and fully developed turbulent state is obtained in the DNS during an ongoing simulation. This necessitates the development of a comprehensive computational platform that includes computing and recording such statistical quantities of turbulent transport in a time series format.

In the current work, our goal is to offer an extensible, open-source computational platform that carries out the high-fidelity simulations of homogeneous isotropic turbulence (HIT) for an incompressible flow, as well as the transport of a passive scalar (temperature or concentration of species) within that flow medium [11,12]. Moreover, we aim to keep track of various statistical quantities of turbulent transport while the simulation is running. These statistical records serve as comprehensive metrics to ensure the accuracy and physical validity of the simulated turbulent fields over a specified time span. Here, we numerically solve the incompressible Navier–Stokes (NS) equations in addition to the advection–diffusion (AD) equation to sufficiently resolve the fluid velocity and passive scalar fields concurrently. The spatial homogeneity of fluctuating fields makes this problem well-suited for the pseudo-spectral implementation of the NS and AD equations based on the Fourier collocation discretization, as employed in our work. This computational platform is based on the programming in `PYTHON` and leveraging the MPI library for parallel implementation.

The rest of this paper is organized as follows: In Section 2, we describe the details and capabilities of the developed platform as scientific software and briefly point out the theoretical and numerical backgrounds. Furthermore, in Section 3, we go over a comprehensive example, illustrating the results of a fully developed turbulent flow and passive scalar field with proper statistical testing and verification. In Section 4, we outline the broader impacts of the current work in turbulent transport research. Finally, in Section 5, we conclude this paper with a summary and conclusions.

2. Solver Description

2.1. Governing Equations

The incompressible HIT flow considered in the present software is governed by NS equations

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \boldsymbol{\nabla} \boldsymbol{U} = -\boldsymbol{\nabla} p + \nu \, \Delta \boldsymbol{U} + \mathcal{A} \, \boldsymbol{U}; \quad \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0. \tag{1}$$

In (1), $\boldsymbol{U} = (U_1, U_2, U_3)$ and p are the instantaneous velocity and modified pressure (pressure divided by the constant density of fluid) fields in the Cartesian coordinate system $\boldsymbol{x} = (x_1, x_2, x_3)$, respectively. Moreover, ν denotes the dynamic viscosity of the Newtonian fluid, and $\mathcal{A}$ is a dynamically evaluated coefficient that corresponds to the artificial forcing scheme we employed in order to obtain a statistically stationary and fully turbulent state. From the Reynolds decomposition of the instantaneous velocity field, $\boldsymbol{U}(\boldsymbol{x}, t) = \langle \boldsymbol{U}(\boldsymbol{x}, t) \rangle + \boldsymbol{u}(\boldsymbol{x}, t)$, where $\langle \cdot \rangle$ represents the ensemble-averaging operator, and $\boldsymbol{u}(\boldsymbol{x}, t)$ denotes the fluctuating part of the velocity field. In HIT, $\langle \boldsymbol{U}(\boldsymbol{x}, t) \rangle = 0$; therefore, the instantaneous velocity field equals the fluctuating part that is governed by (1). Introducing a passive scalar $\Phi(\boldsymbol{x}, t)$ transported in the considered fully developed HIT flow, the AD equation governing the passive scalar concentration may be formulated as

$$\frac{\partial \Phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \Phi = \mathcal{D} \, \Delta \Phi, \tag{2}$$

where $\mathcal{D}$ denotes the diffusivity of the passive scalar. We apply the Reynolds decomposition to the total passive scalar, $\Phi = \langle \Phi \rangle + \phi$, while ϕ denotes the fluctuating part of the scalar concentration. Considering a uniform mean gradient for the passive scalar as $\boldsymbol{\nabla} \langle \Phi \rangle = (0, \beta, 0)$, where β is a constant, the AD equation in (2) may be rewritten as

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \phi = -\beta \, u_2 + \mathcal{D} \, \Delta \phi. \tag{3}$$

2.2. Fourier Pseudo-Spectral Method

Here, we consider spatial homogeneity for the fluctuating velocity and scalar concentration, which allows choosing the periodic boundary conditions for these fluctuating fields as

$$\boldsymbol{u}(\boldsymbol{x}+\mathcal{L}\,\mathrm{e}_i,t)=\boldsymbol{u}(\boldsymbol{x},t),\quad \phi(\boldsymbol{x}+\mathcal{L}\,\mathrm{e}_i,t)=\phi(\boldsymbol{x},t). \tag{4}$$

In (4), $\mathrm{e}_{i,\ i=1,2,3}$, is the unit vector along the i-th direction in the Cartesian coordinate, and $\mathcal{L}$ denotes the periodicity length, specifying the spatial domain as $\boldsymbol{\Omega}=[0,\mathcal{L}]^3$. Discretizing $\boldsymbol{\Omega}$ using a uniform three-dimensional grid returns N^3 grid points with $\Delta x=\mathcal{L}/N$ as the grid spacing along each direction. Transforming this discretization into the spectral domain returns a standard pseudo-spectral representation of the governing equations given in (1) and (3). Subsequently, $\boldsymbol{k}=(k_1,k_2,k_3)$ indicates the coordinate system in the spectral space. By employing the Fourier collocation method, the discretized representation of $\boldsymbol{k}$ would be $k_i=(-N/2+1,\dots,N/2),\, i=1,2,3$. Accordingly, the discrete Fourier transform of any field variable, such as $\phi(\boldsymbol{x},t)$, is written as

$$\phi(\boldsymbol{x},t)=\frac{1}{N^3}\sum_{\boldsymbol{k}}\hat{\phi}_{\boldsymbol{k}}(t)\,e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}, \tag{5}$$

where $\mathrm{i}=\sqrt{-1}$, and $e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$ are the Fourier basis functions. As a result, the Fourier coefficients associated with $\boldsymbol{k}$ are represented as $\hat{\phi}_{\boldsymbol{k}}(t)=\sum_{\boldsymbol{x}}\phi(\boldsymbol{x},t)\,e^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}$. The standard pseudo-spectral formulation of the NS equations based on the Fourier collocation method is obtained after taking the Fourier transform of (1),

$$\begin{aligned}\frac{d\hat{\boldsymbol{u}}_{\boldsymbol{k}}}{dt}+(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}})_{\boldsymbol{k}}&=-\mathrm{i}\boldsymbol{k}\,\hat{p}_{\boldsymbol{k}}-\nu|\boldsymbol{k}|^2\hat{\boldsymbol{u}}_{\boldsymbol{k}}+\mathcal{A}\hat{\boldsymbol{u}}_{\boldsymbol{k}};\\ \mathrm{i}\,\boldsymbol{k}\cdot\hat{\boldsymbol{u}}_{\boldsymbol{k}}&=0,\end{aligned} \tag{6}$$

By taking the divergence of momentum equation in (6) and applying the continuity, the modified pressure is explicitly expressed in terms of the velocity field. Considering that $\boldsymbol{k}\cdot\boldsymbol{k}=|\boldsymbol{k}|^2$, one can derive that $\hat{p}_{\boldsymbol{k}}=\mathrm{i}\boldsymbol{k}\cdot(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}})_{\boldsymbol{k}}/|\boldsymbol{k}|^2$; hence, (6) may be reformulated as

$$\frac{d\hat{\boldsymbol{u}}_{\boldsymbol{k}}}{dt}+(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}})_{\boldsymbol{k}}=\boldsymbol{k}\frac{\boldsymbol{k}\cdot(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}})_{\boldsymbol{k}}}{|\boldsymbol{k}|^2}-\nu|\boldsymbol{k}|^2\hat{\boldsymbol{u}}_{\boldsymbol{k}}+\mathcal{A}\hat{\boldsymbol{u}}_{\boldsymbol{k}}. \tag{7}$$

Similarly, the pseudo-spectral representation of the AD equation for passive scalar (3) is obtained as

$$\frac{d\hat{\phi}_{\boldsymbol{k}}}{dt}+(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\phi})_{\boldsymbol{k}}=-\beta\,\hat{u}_{k_2}-\mathcal{D}|\boldsymbol{k}|^2\hat{\phi}_{\boldsymbol{k}}. \tag{8}$$

We employ a fourth-order Runge–Kutta (RK4) scheme, to perform the time-stepping procedure for both NS and AD equations. Since the nonlinear (advective) terms are evaluated in the physical space and then transformed into the spectral space, the time-stepping is essentially explicit.

2.3. Programming Architecture

The structure of the presented software is schematically illustrated in Figure 1. According to Figure 1a, the user starts from a pre-processing step, where the isotropic and random velocity initial conditions (I.C.) are constructed based on a prescribed spectrum for turbulent kinetic energy (TKE). The procedure is a straightforward implementation of the well-known work by Rogallo for generating a divergence-free isotropic velocity state [13]. According to Lamorgese et al. [14], the initial TKE spectrum is set to

$$E(\kappa,0) = \frac{u_{rms}^2}{k_F} \times \begin{cases} (\kappa/k_F)^2, & \text{if } \kappa \le k_F, \\ (\kappa/k_F)^{-5/3}, & \text{if } \kappa > k_F. \end{cases} \tag{9}$$

In this initialization of TKE, κ represents the wavenumber associated with spherical shells, k_F denotes the maximum wavenumber of the TKE shell that we apply artificial forcing to, and u_{rms} specifies the initial root-mean-square (rms) intensity of velocity fluctuations. In the construction of the velocity I.C., u_{rms} is set to unity while k_F and the number of Fourier collocation points, N, are taken as input parameters. In a `UNIX/LINUX` environment, these inputs are taken as arguments in the execution command line, which are imported through the `sys` library in `PYTHON`. Once the velocity I.C. is obtained, it is partitioned into N_p slabs according to the "slab" decomposition method. The domain decomposition and the parallel implementation of the forward and inverse three-dimensional fast Fourier transform (FFT) in `PYTHON` programming language are adopted from Mortensen and Langtangen's work in [9]. Here, the MPI communications depend on the `mpi4py` library [15–17].

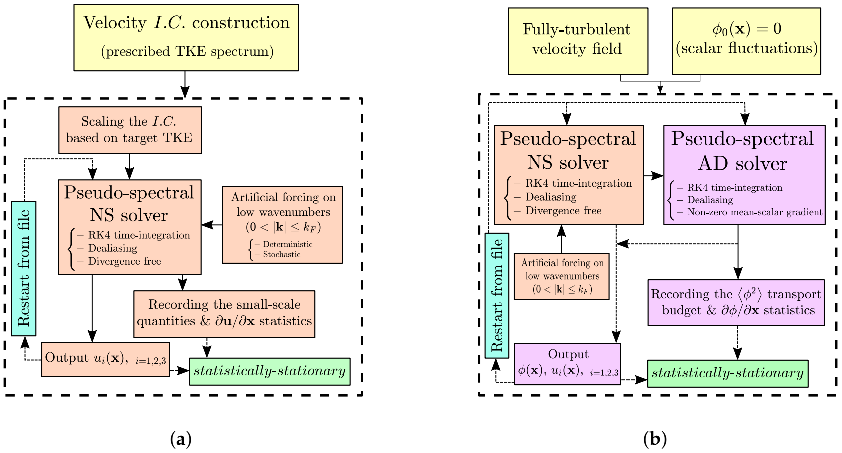

Velocity $I.C.$ construction
(prescribed TKE spectrum)
Scaling the $I.C.$ based on target TKE
Pseudo-spectral NS solver
– RK4 time-integration
– Dealiasing
– Divergence free
Artificial forcing on low wavenumbers $(0 < |\mathbf{k}| \le k_F)$
– Deterministic
– Stochastic
Restart from file
Recording the small-scale quantities & $\partial\mathbf{u}/\partial\mathbf{x}$ statistics
Output $u_i(\mathbf{x}),\ _{i=1,2,3}$
statistically-stationary
(a)
Fully-turbulent velocity field
$\phi_0(\mathbf{x}) = 0$ (scalar fluctuations)
Pseudo-spectral NS solver
– RK4 time-integration
– Dealiasing
– Divergence free
Pseudo-spectral AD solver
– RK4 time-integration
– Dealiasing
– Non-zero mean-scalar gradient
Restart from file
Artificial forcing on low wavenumbers $(0 < |\mathbf{k}| \le k_F)$
Recording the $\langle\phi^2\rangle$ transport budget & $\partial\phi/\partial\mathbf{x}$ statistics
Output $\phi(\mathbf{x}),\ u_i(\mathbf{x}),\ _{i=1,2,3}$
statistically-stationary
(b)

Figure 1. Schematic of the architecture of the software. (**a**) Illustrates the pseudo-spectral NS solver to archive the fully developed turbulent state, statistical records, velocity output, and restart the simulation from the output file. (**b**) Shows the pseudo-spectral NS and AD solvers to reach the fully developed turbulent scalar state.

Having the partitioned velocity I.C. prepared in the pre-processing step, it is fed into the main body of the software, where the initial velocity field might be magnified by a user-defined input argument so that a target TKE level could be considered for the simulation. Moreover, the viscosity of thefluid, ν, is also taken as another user-defined input argument. Next, the magnified velocity field is passed into the solver, where $\hat{\boldsymbol{u}}_{\boldsymbol{k}}$ and $(\widehat{\boldsymbol{\nabla u}})_{\boldsymbol{k}} = \mathrm{i}\,\boldsymbol{k}\,\hat{\boldsymbol{u}}_{\boldsymbol{k}}$ are separately transformed back into the physical space. As a result, the nonlinear term $\boldsymbol{u} \cdot \boldsymbol{\nabla u}$ is simply computed and then it is transformed into the Fourier space. The aliasing error that appears due to this procedure is removed by phase-shifting and truncation, according to $2\sqrt{2}N/3$ as the maximum wavenumber (see [18]). Afterward, all of the terms in Equation (7) are directly evaluated in every stage of the RK4 time integration; however, the last term on the right-hand side of (7) is only evaluated after the final stage during the method called artificial forcing. In artificial forcing, we manage to keep the energy content in the low wavenumbers constant; in the spectral domain, this is associated with the sphere of $0 < |\boldsymbol{k}| \le k_F$. In this procedure, $\mathcal{A}$ is computed in a way that the dissipated energy of turbulent motion is injected into the large scales. This scheme prevents the flow from undergoing a decay process before the realistic and fully developed turbulent state is achieved; nevertheless, the artificial forcing scheme could be turned off through a user-defined input argument if one seeks to obtain decaying HIT data. The forcing coefficient could be determined either deterministically [19] or stochastically [20,21], and both of these methods are supported in the software, as would be specified through parsing an input argument. Moreover, regarding the stable time integration, the Courant–Friedrichs–Lewy (CFL) number is dynamically checked through a user-defined time frequency. According to Eswaran and Pope's work [20], the CFL for this problem is demonstrated as

$$\mathrm{CFL} := \frac{\Delta t}{\Delta x} \max(|u_1| + |u_2| + |u_3|). \tag{10}$$

In (10), Δx is the uniform grid spacing in each direction and Δt is the user-defined constant time interval that is used in the RK4 time-stepping. In practice, CFL is required to be less than unity to ensure stable time integration.

Since the fully developed turbulent state is characterized by the meticulous tracking of statistical quantities of the flow, the present software provides a comprehensive framework for computing and recording the statistical quantities of turbulent flow. Given the homogeneity of the fluctuating fields, spatial averaging is employed for computing these records at user-defined time intervals. These statistical quantities are categorized into turbulent characteristics of small-scale motion reported in Table 1, and high-order central moments of diagonal components of the velocity gradient tensor (VGT), $\boldsymbol{\nabla u}$. For instance,

$$\mathcal{S}_{u_{1,1}} = \left\langle \left(\frac{\partial u_1}{\partial x_1} \right)^3 \right\rangle \bigg/ \left\langle \left(\frac{\partial u_1}{\partial x_1} \right)^2 \right\rangle^{3/2}, \tag{11}$$

$$\mathcal{K}_{u_{1,1}} = \left\langle \left(\frac{\partial u_1}{\partial x_1} \right)^4 \right\rangle \bigg/ \left\langle \left(\frac{\partial u_1}{\partial x_1} \right)^2 \right\rangle^{2}, \tag{12}$$

where $\mathcal{S}_{u_{1,1}}$ and $\mathcal{K}_{u_{1,1}}$ are the skewness factor and flatness factor (or kurtosis) associated with the first diagonal component of VGT, $u_{1,1} = \partial u_1 / \partial x_1$, respectively. The fully turbulent flow state would be identified when the time series of these records reach a statistically stationary state after a long enough time integration, i.e., approximately 10 to 15 large-eddy turnover times (see Table 1). The parallel implementation for computing and collecting these statistical quantities and later recording them as time series is performed by utilizing `point-to-point` and `collective` MPI directives.

Table 1. Statistical characteristics of the turbulent flow to be recorded from the NS solver as time series within user-defined time intervals.

TKE	$K = \frac{1}{2} \langle \boldsymbol{u} \cdot \boldsymbol{u} \rangle$
Turbulent dissipation	$\varepsilon = 2\nu \langle \boldsymbol{S} : \boldsymbol{S} \rangle$
Kolmogorov length-scale	$\eta = \left(\nu^3 / \varepsilon\right)^{1/4}$
Taylor scale Reynolds number	$Re_\lambda = \lambda\, u_{rms} / \nu$
Large-eddy turnover time	$T_e = l_o / u_{rms}$

$\boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^{\mathrm{T}}\right), u_{rms} = \sqrt{2K/3}, \lambda = u_{rms}\sqrt{15\nu/\varepsilon}, l_o = u_{rms}^3/\varepsilon.$

Furthermore, the velocity and pressure fields might be written as output files stored in directories named `Out_*`, based on a user-defined time interval that could be used for post-processing after the flow reaches a fully developed turbulent state. A "restart from file" capability is also designated so that once the statistical record is written out to a file, the latest state of the velocity field and its particular time integration information are also written to files, which are stored in a directory named `Restart`. Starting a simulation from either a prescribed I.C. or restarting it to continue an ongoing simulation that was stopped before is specified by a user-defined input argument. The parallel I/O to store the velocity and pressure fields is done by employing the `scipy.io` library by using `loadmat()` and `savemat()` routines for each partition of the data resolved within each MPI process. These I/O routines provide partitioned outputs in the compressed format and with the machine precision accuracy. This is essentially a fast and efficient I/O approach that maintains the simplicity for users to perform any post-processing steps, such as time-averaging on the statistically stationary turbulent data.

According to Figure 1b, once the fully turbulent velocity state is achieved, the user may be able to use a restart or output instance as the velocity I.C. to introduce a passive scalar transport to the simulation. The passive scalar field is added with a directional constant mean gradient as described in (3), while the fluctuating concentration is assumed to be zero, $\phi_0(\boldsymbol{x}) = 0$. Here, the goal is to resolve the fluctuating scalar concentration field, Equation (8), transported on the fully turbulent incompressible flow for a long enough time span so that the fully developed and realistic turbulent state for the passive scalar is obtained. Subsequently, a similar procedure as described in the schematic shown in Figure 1a is followed, while the pseudo-spectral AD solver is fed by the resolved velocity field from the NS solver. The diffusivity of the passive scalar, $\mathcal{D}$, is specified by a user-defined input argument for the Schmidt number, $Sc = \nu/\mathcal{D}$. Accordingly, similar to the NS solver, the advective scalar flux, $(\widehat{\boldsymbol{u}\cdot\boldsymbol{\nabla}\phi})_k$, is computed in the physical space by the inverse FFT of $\hat{\boldsymbol{u}}_{\boldsymbol{k}}$ and $(\widehat{\boldsymbol{\nabla}\phi})_k$, followed by the forward FFT computation of $\boldsymbol{u}\cdot\boldsymbol{\nabla}\phi$. A similar dealiasing procedure, as described for the NS solver, is employed in the pseudo-spectral AD solver, and the RK4 time integration scheme is utilized to numerically perform explicit time-stepping.

In homogeneous scalar turbulence, the time evolution of the scalar variance $\langle\phi^2\rangle$ is governed by

$$\frac{d}{dt}\langle\phi^2\rangle = -\,2\,\langle\boldsymbol{q}\rangle\cdot\boldsymbol{\nabla}\langle\Phi\rangle - \langle\chi\rangle, \tag{13}$$

where $\boldsymbol{q} = \phi\,\boldsymbol{u}$ denotes the scalar flux vector, and the turbulent scalar dissipation is defined as $\chi = 2\,\mathcal{D}\,\boldsymbol{\nabla}\phi\cdot\boldsymbol{\nabla}\phi$ [22]. According to (3), the first term on the right-hand side of (13) is simplified to $-2\,\beta\,\langle\phi\,u_2\rangle$, which denotes the scalar variance production (by the uniform mean scalar gradient, β). The present software is capable of computing and recording a rate of scalar variance in addition to the production and dissipation terms. This is useful for checking if the balance on both sides of Equation (13) holds throughout a simulation so that one ensures that the implementation of the solver works seamlessly. On the other hand, a measure to evaluate whether the statistically stationary state for the passive scalar is achieved is to check if $-2\,\beta\,\langle\phi\,u_2\rangle/\langle\chi\rangle \sim 1$ throughout the simulation. Moreover, recording the skewness and flatness factors for the components of the fluctuating scalar gradient vector (e.g., $\mathcal{S}_{\phi_{,2}}$ and $\mathcal{K}_{\phi_{,2}}$ for $\phi_{,2} = \partial\phi/\partial x_2$, similar to (11) and (12) for VGT) is another statistical indicator measure for the fully developed turbulent passive scalar state. Therefore, in the current computational platform, the user would be able to recognize the statistically stationary state by monitoring the explained time series data that is written out according to the user-defined time interval as a software input.

The field data output and restart capability for the AD solver are designated using a similar strategy to that described for the NS solver, so that the user would be able to resume an interrupted/stopped simulation and use the output field data for desired applications or post-processing.

In the following section, we present a comprehensive example that, step by step, walks through the utilization of the present software.

3. An Illustrative Example

This comprehensive example mainly consists of the construction of isotropic velocity I.C., obtaining a well-resolved fully turbulent velocity field, and simulating well-resolved passive scalar turbulence with an imposed mean scalar gradient.

3.1. I.C. Construction and the DNS of HIT Flow

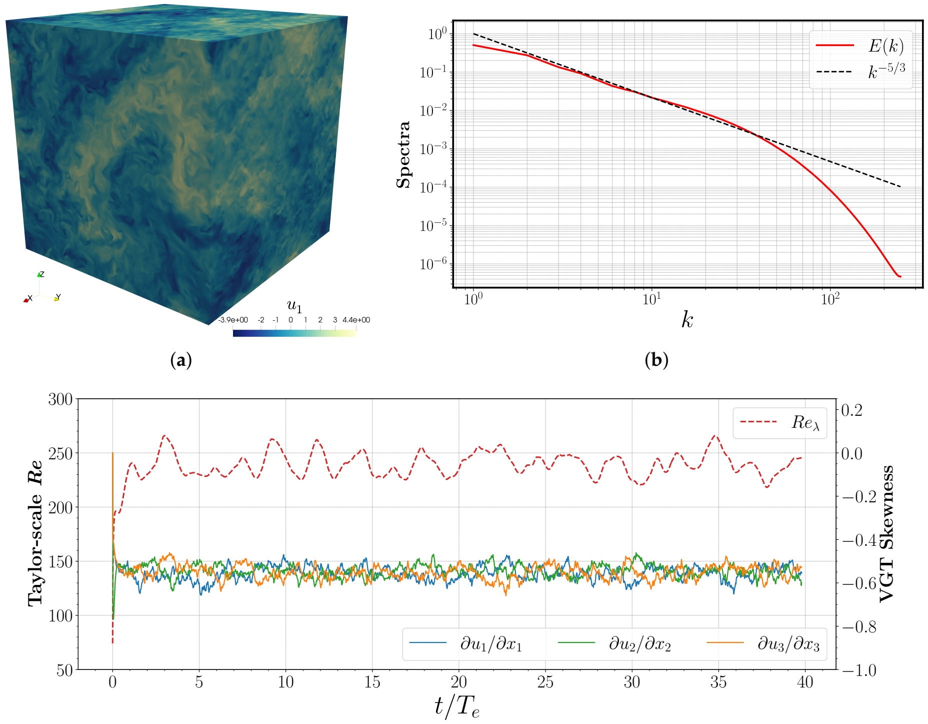
According to the descriptions in Section 2.3, the isotropic and divergence-free velocity I.C. is constructed based on a prescribed energy spectrum given in (9). Considering the periodicity length $\mathcal{L} = 2\pi$, this pre-processing step is done through serial execution of the `Gen_IC.py` script, which takes the following input arguments, respectively: N (spatial resolution along each direction), k_F (forcing wavenumber), and N_p (number of slab partitions). We need to emphasize that N_p must be chosen in a way that N is a multiple of N_p. The resulting velocity field is located in a directory named `IC`, where N_p is the number of `.mat` velocity files that are stored. In this example, we take $N = 520$, $k_F = 2$, and $N_p = 40$. All the components of velocity I.C., in addition to the VGT components, have Gaussian distribution. This velocity I.C. is passed into the NS solver written in `PScHIT.py` script, which takes the following input arguments given in Table 2. Here, `Output_frequency` and `Stats_frequency` are multiplied by the specified Δt. Moreover, the `If_Restart` argument could either be 0 or 1, where 0 indicates that it is a simulation starting from the constructed I.C., while 1 specifies resuming a simulation from the restart files. In this example, we performed the simulation for $t/T_e \sim 15$ to ensure that the fully turbulent flow state was achieved. Figure 2a portrays the first component of the velocity field. Figure 2b shows the radial TKE spectrum averaged over 5 large-eddy turnover times. Moreover, Figure 2c includes the time records of the Taylor scale Reynolds number and VGT skewness factor for diagonal components computed and recorded over 40 large-eddy turnover times. This shows that the statistically stationary state is achieved through the long-time DNS, where $Re_\lambda \approx 240$, $\mathcal{S}_{u_{1,1}} = \mathcal{S}_{u_{2,2}} = \mathcal{S}_{u_{3,3}} \approx -0.55$, and $\mathcal{K}_{u_{1,1}} = \mathcal{K}_{u_{2,2}} = \mathcal{K}_{u_{3,3}} \approx 6.8$ at the fully turbulent state. The statistical records of VGT clearly show that the resolved velocity field is isotropic. Finally, $k_{max}\eta \approx 1.5$ ensures that the small-scale turbulent motions are well-resolved ($k_{max} = \sqrt{2}N/3$) [23].

Table 2. Input arguments for `PScHIT.py` and the specified values for the example case. The order of arguments in the execution command line are listed here.

Input Argument	Value
`t_end`	40
`Output_frequency`	1000
`Stats_frequency`	100
`TKE_magnification`	6.0
ν	0.0008
k_F	2
`forcing_type`	deterministic
N	520
Δt	0.0005
`If_Restart`	0 or 1

3.2. DNS of Passive Scalar Transport

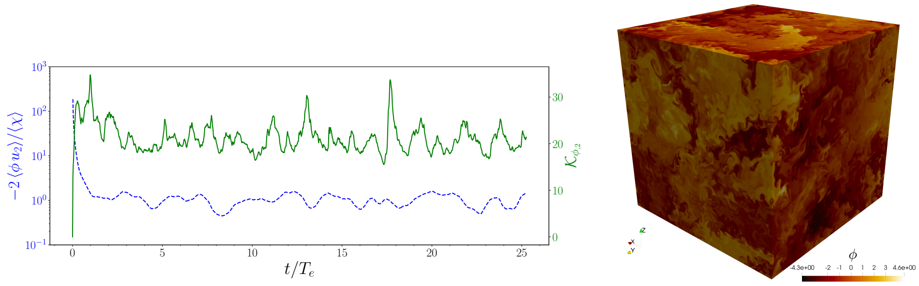
Similar to starting the NS solver from a prescribed velocity I.C., we take a fully turbulent velocity output (velocity state at $t/T_e = 15$ in Section 3.1) and continue the simulation under artificial forcing, while introducing a passive scalar field where its fluctuating part is initialized at zero. The Schmidt number, Sc, is specified by the user through an input argument. According to the problem setting for the mean scalar gradient, we let $\beta = 1$ (mean scalar gradient along x_2 direction). Therefore, for a passive scalar with $Sc = 1$, we aim to obtain the fully turbulent scalar field. We need to note that the spatial resolution required for the passive scalars with $Sc \geq 1$ is defined based on $\eta_B = \eta\, Sc^{-1/2}$ [24] and in this example, the spatial resolution for the velocity field is sufficient for a well-resolved passive scalar. We managed to resolve the passive scalar field for 25 large-eddy turnover times and the rest of the simulation parameters remain the same as values reported in Table 2. Figure 3 shows the records of scalar variance production over the dissipation rate, $-2\,\langle \phi\, u_2 \rangle / \langle \chi \rangle$, and the flatness factor for the scalar gradient along the direction of the mean scalar gradient, $\mathcal{K}_{\phi,2}$. As observed, after resolving the passive scalar field for approximately two large-eddy turnover times, $-2\,\langle \phi\, u_2 \rangle / \langle \chi \rangle \sim 1.0$; this means that the equilibrium state for the passive scalar variance is obtained. Moreover, after approximately three large-eddy turnover times, the high-order statistical moments of the scalar gradient reach a statically stationary state. For instance, $\mathcal{S}_{\phi_{,2}} \approx 1.4$, and $\mathcal{K}_{\phi_{,2}} \approx 20.8$ throughout the time-averaging of these statistical moments when $t/T_e \geq 5$. By resolving the passive scalar field through the AD equation, and for a long enough time after the equilibrium and stationary state, the fully turbulent and realistic scalar field is ensured.



(**a**) (**b**)

(**c**)

Figure 2. (**a**) Snapshot of the fully developed turbulent velocity field, u_1 component. (**b**) Time-averaged TKE spectrum. (**c**) Time series of Re_λ (red dashed line), and VGT skewness factors, $\mathcal{S}_{u_{1,1}}$, $\mathcal{S}_{u_{2,2}}$, $\mathcal{S}_{u_{3,3}}$.



(**a**) (**b**)

Figure 3. (**a**) Time records of production over dissipation of scalar variance (blue dashed line), and the flatness factor for the scalar gradient vector component along the direction of the mean scalar gradient (green solid line). (**b**) Snapshot of the fully developed turbulent passive scalar field.

4. Impact and Applications

Current work offers a framework for obtaining highly accurate spatiotemporal data for homogeneous turbulent transport with proper statistical testing of the recorded quantities. In turbulent transport research, this provides a great source of high-fidelity data for a variety of innovative contributions. In a large-eddy simulation (LES) of turbulence, the exact subgrid-scale (SGS) closure terms appearing in the filtered NS and AD equations are computed from the filtered DNS data and they are utilized in assessing the performances of SGS models (see, e.g., [25]). In particular, this computational–statistical framework has been employed in *the nonlocal* modeling of these SGS closure terms from fractional-order operators. For instance, Akhavan-Safaei et al. [26] utilized a rich dataset of filtered DNS data for the SGS scalar flux to show that the dynamics of SGS transport are essentially "nonlocal". Later, they utilized this dataset to infer the parametric space of the fractional-order SGS model they developed, as well to test its performance.

4.1. Fractional LES SGS Modeling for Scalar Turbulence

The filtered AD equation for the total passive scalar concentration, developed from the filtered kinetic BTE with an α-stable Lévy distribution model, yields a fractional-order SGS scalar flux model at the continuum level. The aforementioned filtered AD equation reads

$$\frac{\partial \widetilde{\Phi}}{\partial t} + \widetilde{\boldsymbol{U}} \cdot \boldsymbol{\nabla} \widetilde{\Phi} = \mathcal{D}\, \Delta \widetilde{\Phi} + \mathcal{D}_{\alpha}\, (-\Delta)^{\alpha} \widetilde{\Phi}, \tag{14}$$

where $\widetilde{(\cdot)}$ represents the filtering operator. Through a proper choice for the fractional Laplacian order α, the developed model optimally works in an LES setting. Applying the Reynolds decomposition and considering the passive scalar with an imposed uniform mean gradient, (14) fully recovers the filtered transport equation for the transport of the filtered scalar fluctuations, $\widetilde{\phi}$.

4.2. Nonlocal Spectral Transfer Model and Scaling Law for Scalar Turbulence

In a subsequent work by Akhavan-Safaei and Zayernouri [27], the DNS conducted by the current framework was used to show that a revised nonlocal spectral transfer model for passive scalars (driven by a large-scale source of anisotropy) would yield an additional fractional-order turbulent diffusion term in the AD equation represented in physical domain. They proposed a physically meaningful modification to the scaling of the scalar variance cascade, given by

$$E_{\phi}(k) = \chi\, \varepsilon^{-1/3}\, k^{-2/3}\, (k^{2} + \mathcal{C}_{\alpha} k^{2\alpha})^{-1/2}. \tag{15}$$

This generalizes the $-5/3$ law, which corresponds to $\alpha = 0$. Modifying the present DNS framework to accommodate this fractional-order diffusion term during the time-stepping, they showed that their recorded time-averaged high-order statistics for the scalar gradient were as precise as the similar computational works reported in the literature with an extremely resolved DNS setup (see [28]). In fact, they demonstrated that their fractional-order DNS approach provides a highly accurate resolved field while efficiently reducing the computational cost.

In another study by Samiee et al. [29], the current DNS framework was utilized for modeling the SGS stress term in the filtered NS equation in terms of a tempered fractional Laplacian operator, and the DNS framework was modified for the LES tests with the tempered fractional-order SGS model. Furthermore, Seyedi and Zayernouri [30] developed a dynamic version of the fractional-order SGS stress model for solving the filtered NS equation through a data-driven approach that was supported by the DNS results of the current framework. They also tested their dynamic fractional SGS model in a modified version of this DNS framework for LES tests. In a following study, Seyedi et al. extended the dynamic fractional modeling methodology to predict the SGS scalar flux in large-eddy simulations of turbulent passive scalars, and utilized this DNS framework to test their model [31].

On the other hand, given the abundance of data and the emergence of data-driven turbulence models [32], the current computational platform would be a reliable candidate to generate data for training and testing such models (see, e.g., [33–37]). Moreover, high-Reynolds and well-examined high-fidelity turbulent transport data from the present

DNS framework could be directly employed in studying the role of coherent turbulent structures and their effects on turbulence statistics [38,39], investigating topological characteristics of turbulent transport [40,41], and analyzing extreme events and internal intermittency [42–44].

5. Summary and Conclusions

This work presented a computational platform for the DNS of homogeneous turbulent flow and passive scalar transport. This open-source software is based on a pseudo-spectral representation of the NS and AD equations on a triply cubic computational domain with periodic boundary conditions for fluctuating fields. Using the Fourier collocation method, the governing equations are discretized in space; by employing the RK4 scheme, the time-stepping is performed. The software provides a pre-processing step to construct homogeneous and isotropic divergence-free velocity I.C., based on a prescribed energy spectrum, which can be decomposed into user-defined partitions. Using the artificial forcing scheme, the dissipated energy is injected into the low wavenumbers, so that after a long time integration, a statistically stationary state is achieved. In order to examine and identify whether the fully developed turbulent flow is obtained, small-scale statistical quantities of turbulence, in addition to the central moments of VGT components, are computed and recorded. Once the realistic turbulent velocity field is obtained, the user is able to start resolving a passive scalar that is transported with the HIT flow, while a uniform mean scalar gradient is imposed. Resolving the scalar fluctuations for a long enough period of time after reaching the equilibrium and stationary state provides the fully developed turbulent scalar field. Statistical records of the scalar gradients, in addition to records of scalar variance production and dissipation, help the user properly identify when the fully developed scalar turbulence is achieved.

Author Contributions: Conceptualization, A.A.-S. and M.Z.; methodology, A.A.-S. and M.Z.; software, A.A.-S.; validation, A.A.-S. and M.Z.; formal analysis, A.A.-S.; investigation, A.A.-S. and M.Z.; resources, M.Z.; data curation, A.A.-S.; writing—original draft preparation, A.A.-S. and M.Z.; writing—review and editing, A.A.-S. and M.Z.; visualization, A.A.-S.; supervision, M.Z.; project administration, M.Z.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the MURI/ARO award (W911NF-15-1-0562), the ARO Young Investigator Program (YIP) award (W911NF-19-1-0444), and the NSF award (DMS-1923201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The computer program for data generation and processing is available through the following GitHub repository: https://github.com/FMATH-Group/PSc_HIT3D, accessed on 3 May 2023.

Acknowledgments: The HPC resources and services were provided by the Institute for Cyber-Enabled Research (ICER) at Michigan State University.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Sagaut, P.; Cambon, C. *Homogeneous Turbulence Dynamics*; Springer: Berlin/Heidelberg, Germany, 2018.
2. Pope, S.B. *Turbulent Flows*; Cambridge University Press: Cambridge, UK, 2001.
3. Moin, P.; Mahesh, K. Direct Numerical Simulation: A Tool in Turbulence Research. *Annu. Rev. Fluid Mech.* **1998**, *30*, 539–578. [CrossRef]
4. Cantwell, C.D.; Moxey, D.; Comerford, A.; Bolis, A.; Rocco, G.; Mengaldo, G.; De Grazia, D.; Yakovlev, S.; Lombard, J.E.; Ekelschot, D.; et al. Nektar++: An open-source spectral/hp element framework. *Comput. Phys. Commun.* **2015**, *192*, 205–219. [CrossRef]

5. Moxey, D.; Cantwell, C.D.; Bao, Y.; Cassinelli, A.; Castiglioni, G.; Chun, S.; Juda, E.; Kazemi, E.; Lackhove, K.; Marcon, J.; et al. Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. *Comput. Phys. Commun.* **2020**, *249*, 107110. [CrossRef]
6. He, P. A high order finite difference solver for massively parallel simulations of stably stratified turbulent channel flows. *Comput. Fluids* **2016**, *127*, 161–173. [CrossRef]
7. Bartholomew, P.; Deskos, G.; Frantz, R.A.; Schuch, F.N.; Lamballais, E.; Laizet, S. Xcompact3D: An open-source framework for solving turbulence problems on a Cartesian mesh. *SoftwareX* **2020**, *12*, 100550. [CrossRef]
8. Bauman, P.T.; Stogner, R.H. GRINS: A multiphysics framework based on the libmesh finite element library. *SIAM J. Sci. Comput.* **2016**, *38*, S78–S100. [CrossRef]
9. Mortensen, M.; Langtangen, H.P. High performance Python for direct numerical simulations of turbulent flows. *Comput. Phys. Commun.* **2016**, *203*, 53 – 65. [CrossRef]
10. Foundation, T.O. OpenFOAM v8 User Guide. Available online: https://doc.cfd.direct/openfoam/user-guide-v8/ (accessed on 3 May 2023).
11. Warhaft, Z. Passive scalars in turbulent flows. *Annu. Rev. Fluid Mech.* **2000**, *32*, 203–240. [CrossRef]
12. Shraiman, B.I.; Siggia, E.D. Scalar turbulence. *Nature* **2000**, *405*, 639–646. [CrossRef]
13. Rogallo, R. *Numerical Experiments in Homogeneous Turbulence*; NASA Technical Memorandum; National Aeronautics and Space Administration: Washington, DC, USA, 1981.
14. Lamorgese, A.G.; Caughey, D.A.; Pope, S.B. Direct numerical simulation of homogeneous turbulence with hyperviscosity. *Phys. Fluids* **2005**, *17*, 015106. [CrossRef]
15. Dalcín, L.; Paz, R.; Storti, M. MPI for Python. *J. Parallel Distrib. Comput.* **2005**, *65*, 1108–1115. [CrossRef]
16. Dalcín, L.; Paz, R.; Storti, M.; D'Elía, J. MPI for Python: Performance improvements and MPI-2 extensions. *J. Parallel Distrib. Comput.* **2008**, *68*, 655–662. [CrossRef]
17. Dalcin, L.D.; Paz, R.R.; Kler, P.A.; Cosimo, A. Parallel distributed computing using Python. *Adv. Water Resour.* **2011**, *34*, 1124–1139. [CrossRef]
18. Patterson, G.S.; Orszag, S.A. Spectral Calculations of Isotropic Turbulence: Efficient Removal of Aliasing Interactions. *Phys. Fluids* **1971**, *14*, 2538–2541. [CrossRef]
19. Sullivan, N.P.; Mahalingam, S.; Kerr, R.M. Deterministic forcing of homogeneous, isotropic turbulence. *Phys. Fluids* **1994**, *6*, 1612–1614. [CrossRef]
20. Eswaran, V.; Pope, S. An examination of forcing in direct numerical simulations of turbulence. *Comput. Fluids* **1988**, *16*, 257–278. [CrossRef]
21. Alvelius, K. Random forcing of three-dimensional homogeneous turbulence. *Phys. Fluids* **1999**, *11*, 1880–1889. [CrossRef]
22. Monin, A.; Yaglom, A. *Statistical Fluid Mechanics, Volume II: Mechanics of Turbulence*; Dover Books on Physics; Dover Publications: Mineola, NY, USA, 2013.
23. Overholt, M.R.; Pope, S.B. Direct numerical simulation of a passive scalar with imposed mean gradient in isotropic turbulence. *Phys. Fluids* **1996**, *8*, 3128–3148. [CrossRef]
24. Batchelor, G.K. Small-scale variation of convected quantities like temperature in turbulent fluid Part 1. General discussion and the case of small conductivity. *J. Fluid Mech.* **1959**, *5*, 113–133. [CrossRef]
25. Samiee, M.; Akhavan-Safaei, A.; Zayernouri, M. A fractional subgrid-scale model for turbulent flows: Theoretical formulation and a priori study. *Phys. Fluids* **2020**, *32*, 055102. [CrossRef]
26. Akhavan-Safaei, A.; Samiee, M.; Zayernouri, M. Data-driven fractional subgrid-scale modeling for scalar turbulence: A nonlocal LES approach. *J. Comput. Phys.* **2021**, *446*, 110571. [CrossRef]
27. Akhavan-Safaei, A.; Zayernouri, M. A non-local spectral transfer model and new scaling law for scalar turbulence. *J. Fluid Mech.* **2023**, *956*, A26. [CrossRef]
28. Donzis, D.; Yeung, P. Resolution effects and scaling in numerical simulations of passive scalar mixing in turbulence. *Phys. D Nonlinear Phenom.* **2010**, *239*, 1278 – 1287. [CrossRef]
29. Samiee, M.; Akhavan-Safaei, A.; Zayernouri, M. Tempered fractional LES modeling. *J. Fluid Mech.* **2022**, *932*, A4 . [CrossRef]
30. Seyedi, S.H.; Zayernouri, M. A data-driven dynamic nonlocal subgrid-scale model for turbulent flows. *Phys. Fluids* **2022**, *34*, 035104. [CrossRef]
31. Seyedi, S.H.; Akhavan-Safaei, A.; Zayernouri, M. Dynamic nonlocal passive scalar subgrid-scale turbulence modeling. *Phys. Fluids* **2022**, *34*, 105122. [CrossRef]
32. Duraisamy, K.; Iaccarino, G.; Xiao, H. Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **2019**, *51*, 357–377. [CrossRef]
33. Wang, Z.; Luo, K.; Li, D.; Tan, J.; Fan, J. Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation. *Phys. Fluids* **2018**, *30*, 125101. [CrossRef]
34. Beck, A.; Flad, D.; Munz, C.D. Deep neural networks for data-driven LES closure models. *J. Comput. Phys.* **2019**, *398*, 108910. [CrossRef]
35. Mohan, A.T.; Tretiak, D.; Chertkov, M.; Livescu, D. Spatio-temporal deep learning models of 3D turbulence with physics informed diagnostics. *J. Turbul.* **2020**, *21*, 484–524. [CrossRef]

36. Mohan, A.T.; Lubbers, N.; Livescu, D.; Chertkov, M. Embedding hard physical constraints in neural network coarse-graining of 3D turbulence. *arXiv* **2020**, arXiv:2002.00021.
37. Sirignano, J.; MacArt, J.F.; Freund, J.B. DPM: A deep learning PDE augmentation method with application to large-eddy simulation. *J. Comput. Phys.* **2020**, *423*, 109811. [CrossRef]
38. Zayernouri, M.; Metzger, M. Coherent features in the sensitivity field of a planar mixing layer. *Phys. Fluids* **2011**, *23*, 025105. [CrossRef]
39. Akhavan-Safaei, A.; Seyedi, S.H.; Zayernouri, M. Anomalous features in internal cylinder flow instabilities subject to uncertain rotational effects. *Phys. Fluids* **2020**, *32*, 094107. [CrossRef]
40. Gonzalez, M. Kinematic properties of passive scalar gradient predicted by a stochastic Lagrangian model. *Phys. Fluids* **2009**, *21*, 055104. [CrossRef]
41. Meneveau, C. Lagrangian dynamics and models of the velocity gradient tensor in turbulent flows. *Annu. Rev. Fluid Mech.* **2011**, *43*, 219–245. [CrossRef]
42. Yeung, P.; Zhai, X.; Sreenivasan, K.R. Extreme events in computational turbulence. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 12633–12638. [CrossRef] [PubMed]
43. Farazmand, M.; Sapsis, T.P. A variational approach to probing extreme events in turbulent dynamical systems. *Sci. Adv.* **2017**, *3*, e1701533. [CrossRef]
44. Sapsis, T.P. Statistics of extreme events in fluid flows and waves. *Annu. Rev. Fluid Mech.* **2021**, *53*, 85–111. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.