A C++ Library for Turbulent Mixing Simulation Using Hierarchical Parcel Swapping (HiPS)

Masoomeh Behrang[a], Tommy Starick[b], Heiko Schmidt[b], David O. Lignell[a,*]

[a] *Chemical Engineering Department, Brigham Young University, Provo, UT 84602, USA*

[b] *Brandenburg University of Technology Cottbus-Senftenberg, Germany*

Abstract

Turbulence models are crucial for simulating flows at all scales, capturing both large-scale structures and small-scale mixing. Software libraries that implement such models should support modular integration, customization, and scalability across different simulation frameworks. This paper presents Hierarchical Parcel Swapping (HiPS), a C++ library documented with Doxygen and available on GitHub. HiPS supports both mixing and reactions and can be used as a standalone model or as a subgrid model in CFD simulations. The code includes examples for users to run it as a standalone model. Additionally, considerations for using it as a subgrid model are provided.

Keywords: Mixing, Reaction, Simulation, Turbulence

Metadata

Nr.	**Code metadata description**	
C1	Current code version	1.0
C2	Permanent link to code/repository used for this code version	`https://github.com/BYUignite/hips.git`
C3	Permanent link to Reproducible Capsule	
C4	Legal Code License	MIT
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	C++
C7	Compilation requirements, operating environments & dependencies	C++11, CMake 3.15+; optional: Cantera, Sundials, Catch2
C8	If available Link to developer documentation/manual	`https://ignite.byu.edu/hips_documentation`
C9	Support email for questions	byuignite@byu.edu

Table 1: Code metadata

[*] Corresponding author
Email address: `davidlignell@byu.edu` (David O. Lignell)

1. Motivation and Significance

Mixing in turbulent flows is a critical phenomenon, influencing species transport and chemical reaction dynamics. Modeling turbulent mixing requires capturing interactions across scales, from large-scale advection to small-scale diffusion, making it distinct from general turbulence modeling. While turbulence models aim to represent the chaotic nature and broad scales of turbulent flows, mixing models focus specifically on scalar transport, variance reduction, and the interactions between species in turbulent environments.

Many models have been developed to address mixing in turbulent flows, each targeting different complexities. One advanced approach is the transported probability density function (PDF) method [1, 2], which tracks the statistical properties of turbulent flow, but it depends on additional mixing models to address fine-scale scalar transport. Accurate mixing models are essential for understanding complex behaviors in turbulence and in improving engineering analysis.

Several mixing models have been proposed to address these challenges, including Interaction by Exchange with the Mean (IEM) [3], Modified Curl (MC) [4], and the Euclidean Minimum Spanning Trees (EMST) [5], among others. While these models have been widely applied, they do not fully address important mixing criteria, such as that outlined by Subramaniam and Pope [5], and Fox [6]. For example, the scalar PDF in IEM does not relax to Gaussian. Neither IEM or Curl's models satisfy locality in composition space, allowing unphysical mixing of parcels. Conversely, the EMST model does satisfy local mixing, but it does not satisfy linearity and independence conditions. None of the models noted include dependence on the Reynolds number, or account for differential diffusion of scalars. Furthermore, these models do not account for the wide range of time and length scales present in turbulent mixing, which can be important when treating complex reacting flows.

The Hierarchical Parcel-Swapping (HiPS) model, developed by Kerstein [7, 8, 9], addresses the noted challenges in turbulent mixing. HiPS is based on a binary tree structure with length scales decreasing geometrically with increasing level and time scales following inertial range scaling. Turbulent advection is represented by stochastic eddy events involving subtree swaps, resulting in locality of mixing in composition space. A recent extension of the model [10] allows for variable scalar diffusivity and differential diffusion. These features make HiPS particularly well-suited for applications, such as combustion [11] or aerosol transport [12].

This paper presents a C++ implementation of the HiPS model for use as both a standalone model for turbulent mixing as for use as a particle mixing model. The library simplifies the complex aspects of setting up and executing mixing simulations in turbulent flows, enabling users to apply the model without focusing on implementation details. By ensuring consistent performance across various computational environments, the library facilitates broader adoption of the HiPS model in research and industry, allowing users to apply it to specific mixing challenges with minimal effort.

2. Software Description

HiPS is an open-source C++ library for simulating turbulent mixing, with optional support for chemical reactions. This section describes the software structure, compilation, and interfaces. The library may be used to construct standalone programs for turbulent mixing, or used with external computational fluid dynamics (CFD) programs as a subgrid mixing model, described in Sections 3.1, and 3.2, respectively.

2.1. Software Architecture

The HiPS package is organized into several subdirectories:

1. `src` – source code for HiPS;
2. `examples` – demonstration cases, including mixing and combustion simulations;
3. `run` – compiled executables for tests examples;
4. `data` – contains runtime output data and reference data;
5. `post` – contains Python scripts for basic post-processing
6. `test` – test functions using the Catch2 library
7. `docs` – documentation and user resources;
8. `docker` – a Dockerfile for facilitating installation and execution.

The `examples` directory contains three sample cases showcasing HiPS as a standalone model. Specifically, `ex_1.cc` simulates scalar mixing dynamics, whereas `ex_2.cc` and `ex_3.cc` incorporate chemical reactions. These examples can be executed from the command line using the corresponding compiled binaries: `/ex_1.x`, `/ex_2.x`, and `/ex_3.x`, located in the `run` directory following installation. A more detailed explanation of these examples is provided in Section 3.1.

HiPS is built using CMake. To install the library, users should navigate to the `hips` folder, create and enter a `build` directory, and execute the following commands:

1. `cmake ..`
2. `make`
3. `make install`

Compilation options can be edited in the top level `CMakeLists.txt` file, or at the `cmake` command. Documentation can be built using `make docs`. This includes both source documentation and descriptions of the HiPS model, examples, and build process.

For mixing simulations, HiPS does not require any external libraries. If the chemical reaction module is enabled, two additional dependencies are required:

- `Cantera`: Handles chemical kinetics and thermodynamic calculations. Users must provide a detailed chemical mechanism file (e.g., `gri30.yaml`) to define species and reaction pathways.

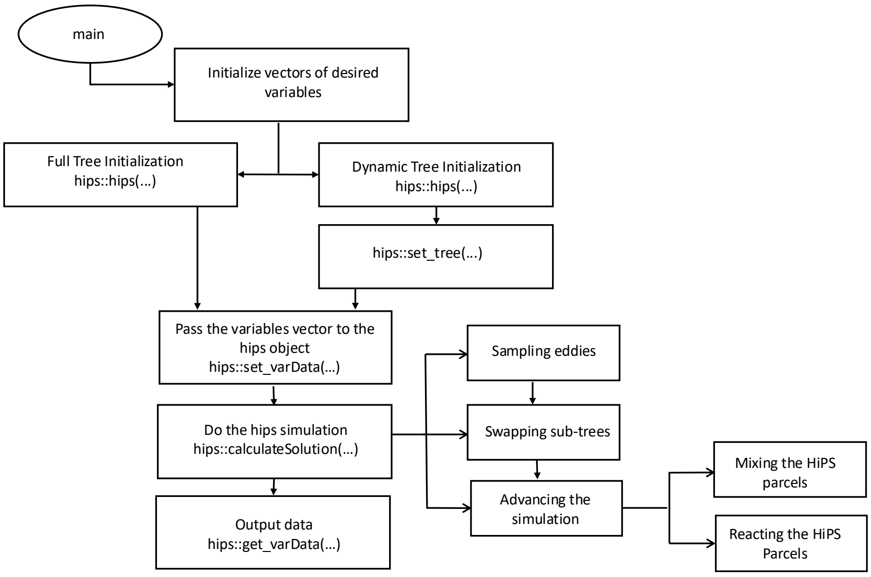
main
Initialize vectors of desired variables
Full Tree Initialization hips::hips(…)
Dynamic Tree Initialization hips::hips(…)
hips::set_tree(…)
Pass the variables vector to the hips object hips::set_varData(…)
Do the hips simulation hips::calculateSolution(…)
Output data hips::get_varData(…)
Sampling eddies
Swapping sub-trees
Advancing the simulation
Mixing the HiPS parcels
Reacting the HiPS Parcels

Figure 1: HiPS workflow diagram.

- `SUNDIALS (CVODE)`: Provides solvers for stiff ordinary differential equations (ODEs), ensuring numerical stability in reaction simulations with multiple timescales.

Unit and integration tests are included in folder `test`. These use the Catch2 library and are built if the `HIPS_BUILD_TEST` compilation flag is enabled, in which case the Catch2 library should be available on the system. The tests focus on the user interface and the code output for mixing and reaction cases. A git hook in `/.githooks/pre-push` runs the tests before the code can be pushed to the repository.

HiPS uses Git for version control and the software is hosted at github.com. Major releases are given annotated tags in the repository using semantic versioning.

2.2. Interfaces and Initialization

The use of HiPS as a standalone model and as a subgrid mixing model are similar and described below. Standalone programs will normally write output files with multiple realizations performed. Mixing models are intended to be used with CFD simulations, such as RANS and LES, especially those using transported PDF methods utilizing a large number of Lagrangian fluid particles. Between global CFD advancement steps, particles in a given CFD grid cell are mixed. To use HiPS as the mixing model, conserved scalars, such as enthalpy and species mass fractions, are projected from the CFD fluid particles to the HiPS parcels; HiPS is then run; and then the HiPS parcels are projected back to the CFD fluid particles for further advancement.

Figure 1 illustrates the structure of the HiPS framework. Programs include the `hips.h` header file. The HiPS simulation workflow begins with initializing vectors of key simulation variables. These variables are passed to a HiPS object, which can be instantiated using one of two constructors, depending on the simulation requirements:

- **Full Tree Initialization**: Constructs the HiPS tree at the start of the simulation, suitable for cases where the domain structure remains static. Key parameters, such as the number of hierarchical levels, domain length, and characteristic time scales are set.
- **Dynamic Tree Initialization**: Designed for simulations requiring frequent updates to the HiPS tree, such as CFD solvers where local tree structures need to be modified at each time step or in specific regions. This approach reduces computational overhead by initializing only the necessary parts of the tree, which are then updated using the `set_tree` function.

After constructing a HiPS object, users interact with the simulation via three primary functions:

- `set_varData`: Initializes variable data to the HiPS object, including projecting CFD particles to the HiPS tree when using as a sugrid model.
- `calculateSolution`: Performs HiPS mixing and (optionally) reaction.
- `get_varData`: Retrieves simulation results including projecting HiPS parcels back to arrays corresponding to CFD particles.

Other public functions listed in `hips.h` control data output and frequency.

The HiPS code provided gives a framework for running HiPS as described in Sections 3.1, 3.2. Users can extend the code or modify it to suite the needs of specific use cases by directly changing the source code, for example, the treatment of chemistry noted at the end of Section 3.1. Updates to the code that may be of broader interest can be incorporated by issuing a pull request on the code repository.

2.3. Model Description

HiPS has been described in the literature [7, 8], with additional details by Behrang et al. [10]. Here, we give a brief summary to motivate the presented software and application. The HiPS mixing formulation presented is applicable to the inertial-advective flow regime under conditions of homogeneous, isotropic turbulence (HIT). The HiPS model uses an auxiliary binary tree structure to represent fluid parcels and turbulent mixing. Figure 2 illustrates the tree, where nodes divide down to smaller scales, storing fluid properties in parcels at the bottom and linking length and time scales to each level. This hierarchical representation mimics the energy cascade process, where larger eddies break into progressively smaller structures.

For an N-level tree, the number of fluid parcels is 2^{N-1}, and the characteristic length scales at levels i decrease by successive factors A, given by

$$L_i = L_0 A^i, \tag{1}$$

where L_0 is the reference length scale, and $A < 1$ depends on the implied spatial dimension. A value of $A = 1/2$ corresponds to one dimension and gives efficient scale resolution. For more details, see Behrang et al. [10].

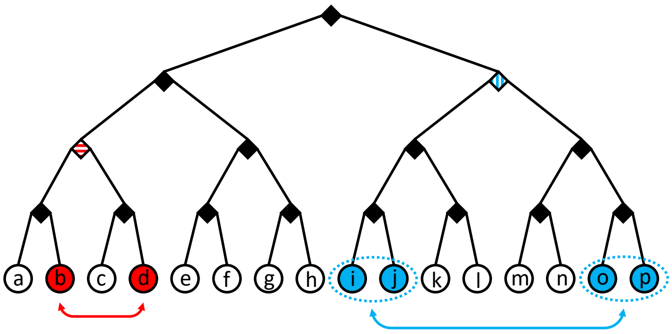

Figure 2: Schematic of the binary HiPS tree with nodes (diamonds) and fluid parcels (circles).

Turbulent advection is modeled as a cascading process that reduces flow structure size across scales. According to Kolmogorov's second hypothesis [13], the energy dissipation rate ϵ remains constant across scales, and the characteristic time scale at each level is related to L_i by $\tau_i \sim L_i^{2/3}$, resulting in

$$\tau_i = \tau_0 A^{2i/3}, \tag{2}$$

where τ_0 is the characteristic time at the largest scale L_0.

2.3.1. Eddy Events and Parcel Swapping

In HiPS, turbulent mixing is modeled by rearranging fluid parcel pairs through a process called an *eddy event*. This swapping mechanism represents turbulent mixing by exchanging fluid properties between different scales in a manner consistent with turbulent scaling processes.

Each eddy event follows these steps:

1. A base node is selected from levels 0 to $N - 3$.
2. A random node or parcel is chosen two levels down on both the left and right branches.
3. The nodes and their subtrees or parcels are swapped instantaneously.

Figure 2 illustrates two types of swaps used in the HiPS model. The first swap, indicated by the red horizontally striped node, involves fluid parcels b and d, changing their pairings from $(a, b) \rightarrow (a, d)$ and $(c, d) \rightarrow (c, b)$. The second swap, shown at the blue vertically striped node, exchanges subtrees containing parcels (i, j) with (o, p). Swaps associated with the green checked node directly modify parcel adjacency, representing micromixing events. In contrast, swaps at the blue striped node contribute to macromixing by reorganizing parcel groupings without altering individual parcel states. Further details on these mechanisms are provided in [10].

HiPS parcel data is stored in `vector` class objects from the C++ standard template library. To accommodate multiple scalars efficiently, an index array is used to track parcel locations. Eddy events only swap the index array `pLoc`, which maps parcel index `i` to memory location `pLoc[i]`. In Fig 2, if we index parcels left to right starting at zero, a scalar v held in vector `v` across nodes would have value `v[pLoc[0]]` in parcel a. Subtrees are selected and swapped following Kerstein [7]. Indexing parcels left to right as noted with integers, the binary representation of the index gives the path to the parcel. For example, parcel n in Fig. 2 would be index 13, or 1101 in binary; As we descend the tree from the top, if we read the binary number left to right, a 1 takes the right branch and a 0 takes the left. So, to get to parcel n, we go right (1), right (1), left (0), right (1). Such indexing, and other operations involving powers of two are conveniently and efficiently done using the right and left shift operators.

2.3.2. Eddy Selection and Timing

The rate of eddy events at level i is determined by the inverse of the time scale,

$$\lambda_i = \frac{2^i}{\tau_i}. \tag{3}$$

Here, 2^i accounts for the increasing number of nodes (parcels) at level i. Since turbulent eddies form and dissipate randomly, their occurrence times are modeled as a Poisson process and follow an exponential distribution with a total rate Λ:

$$p(\Delta t) = \Lambda e^{-\Lambda \Delta t}, \tag{4}$$

where Δt is the time increment to the eddy and is sampled from $p(\Delta t)$ by inverting uniform random variates on the cumulative distribution,

$$\Delta t = -\frac{\ln(P_r)}{\Lambda}, \tag{5}$$

where $P_r \in [0, 1]$ is a uniform random variate. This models the statistically representation of the mixing dynamics.

2.3.3. Schmidt Number Dependence

To understand the impact of Schmidt number on mixing, it is important to first recognize the categorization of eddy events based on length scales. Eddy events are categorized into two regions based on length scales: the inertial range (I) and the viscous scale range (V). The total eddy event rate is given by

$$\Lambda = \Lambda_I + \Lambda_V. \tag{6}$$

A random variate P_r determines the region; if $P_r \leq \Lambda_I/\Lambda$, region I is chosen; otherwise, region V is selected.

In the inertial and viscous ranges, the probability of an eddy event at level i is λ_i/Λ_I, and λ_i/Λ_V, respectively.

In turbulent flows, momentum and scalar diffusivities can differ, affecting how species mix. The Schmidt number (Sc) represents the ratio of momentum diffusivity to species diffusivity and is extended to arbitrary scalars. The HiPS model treats mixing differently depending on whether $Sc > 1$ or $Sc < 1$.

The characteristic mixing length scale l^* at level i^* is used for scalars where $Sc = 1$, representing the smallest scale at which mixing occurs. For scalars with arbitrary Sc, the smallest scale l_s^* at level i_s^* is similarly defined and is proportional to l^*. The relationship between Sc and l_s^* is

$$Sc = \left(\frac{l^*}{l_s^*}\right)^{p_s}, \tag{7}$$

where $p_s = 2$ for $Sc \geq 1$, and $p_s = 3/4$ otherwise.

For levels above i^* (higher level index), the timescale τ equals τ^*, consistent with mixing in the viscous-advective regime. For scalars with $Sc < 1$ or scalars with Sc less than the scalar with the maximum Sc, mixing occurs across multiple parcels, reflecting the enhanced diffusivity of such scalars. This characteristic ensures that mixing is appropriately distributed across parcels, accommodating the variable diffusivity.

For discrete Sc values corresponding to HiPS levels, the relationship between Sc and tree levels i^* and i_s^* is

$$Sc = A^{p_s(i^* - i_s^*)}. \tag{8}$$

Since l_s^* may fall between two HiPS levels, the corresponding level i_s^* is interpolated as

$$i_s^* = i^* - \frac{\log Sc}{p_s \log A}. \tag{9}$$

The mixing probability is weighted based on proximity to the nearest level, ensuring smooth transitions in the HiPS framework.

For scalars with $Sc = 1$, a tree would have N^* levels. For scalars with $Sc > 1$, additional levels are present to represent the final scalar structures, and the total number of levels is N. To characterize the scale separation in HiPS, we define a Reynolds number Re based on Kolmogorov scaling as

$$Re = \left(\frac{L_0}{l^*}\right)^{4/3} = A^{-\frac{4}{3}(N^*-3)}. \tag{10}$$

3. HiPS functionality

3.1. Using HiPS as a Standalone Model

HiPS includes three example codes in the `example` directory that demonstrate how to use HiPS as a standalone model. Output from the examples is contained in directory `data/reference/` so that comparison of the user-generated output can be made to verify the build and execution process. The first example, `ex_1.cc`, simulates scalar mixing dynamics, showing how a basic mixing model behaves. Initially, fluid parcels are divided into two groups: the first half has a scalar value of $\phi = 0$, and the second half has $\phi = 1$. These parcels mix in time to reach an average value. Parameter selection guidance is available in the documentation. The base number of levels is set to 9, and Schmidt numbers of 0.0625 and 16 are used to illustrate the effects of low and high Schmidt numbers on mixing. The low Schmidt number corresponds to high diffusivity, while the high Schmidt number corresponds to low diffusivity. Figure 3 illustrates how mixing behavior changes with different Schmidt numbers. Although the overall mixing rate remains constant, higher Schmidt numbers result in mixing

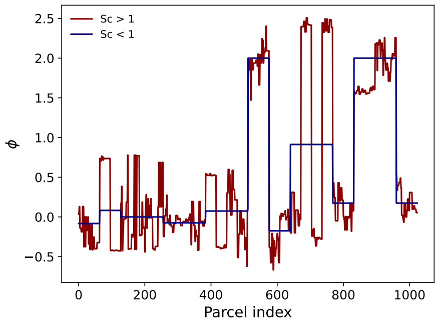

Figure 3: Effect of Schmidt numbers on mixing behavior for high Schmidt numbers, $Sc > 1$, and low Schmidt numbers, $Sc < 1$.

at smaller scales, visible as finer fluctuations in the figure, while lower Schmidt numbers mix at larger scales with coarser variations.

The second and third examples in `ex_2.cc` and `ex_3.cc`, focus on HiPS mixing with chemical reactions, simulating turbulent flames. Both examples use different initialization methods. The difference between the two cases lies solely in the initialization. While `ex_2.cc` is premixed with a stoichiometric ethylene/air mixture, `ex_3.cc` corresponds to a non-premixed configuration, where separate parcels carry fuel and oxidizer. Aside from the initial condition, all other aspects—such as the HiPS domain setup, mixing model, and reaction mechanism—remain identical. This highlights the flexibility of the library to handle different combustion scenarios by simply modifying the initial parcel distribution. In `ex_2.cc`, a stoichiometric ethylene/air mixture is initialized on the HiPS domain with six levels, where 25% of the parcels are pre-combusted, and the remaining 75% contain fresh reactants. All species use $Sc = 1$, and the domain length scale is $L_0 = 0.01$.

The mixing rate, represented by τ_0, influences ignition and extinction in combustion processes. Larger τ_0 values lead to slower mixing while smaller τ_0 increase mixing rates, which can cause extinction. In `ex_2.cc`, two values of τ_0 are shown. Figure 4 shows how mixing rate affects ignition and extinction. In Fig. 4a, a higher τ_0 causes slow mixing, leading to temperatures above 2000 K, indicating ignition. In Fig. 4b, a smaller τ_0 causes fast mixing, resulting in temperatures below 900 K, showing extinction.

HiPS offers two interfaces for integrating chemical reactions. Users can choose either integrator by specifying the method in the HiPS constructor. The first, in the `batchReactor_cvode` class, uses the CVODE solver directly. The second, in the `batchReactor_cantera` class, uses Cantera's interface to CVODE. These two implementations serve as working examples for incorporating chemical kinetics. Users may either use the provided methods or adapt the code to implement their own customized solvers.

3.2. Consideration for HiPS as a subgrid mixing model

HiPS is designed to serve as a subgrid mixing model in turbulent flow simulations, with applications in the transported PDF (TPDF) framework and other

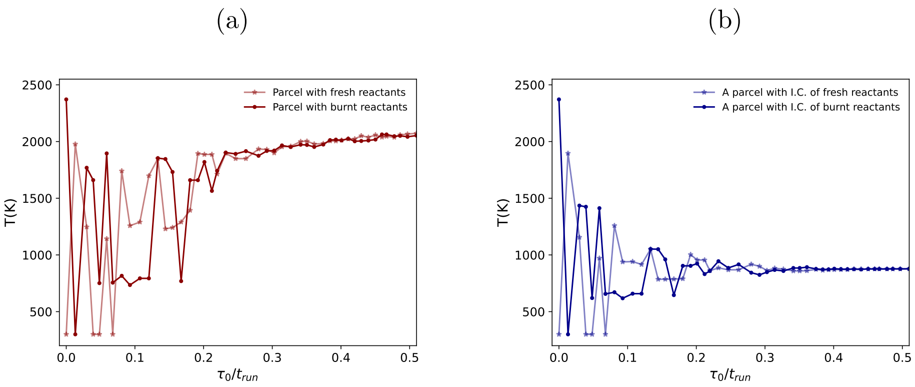

Figure 4: Ignition (a) and extinction (b) in the parcels.

particle-based or hybrid CFD methods. These models normally provide the grid length scale and a local Re, which correspond to L_0 and Re in HiPS. In such cases, the external CFD software handles all aspects of mesh management, discretization, and advancement of the global solution, with HiPS serving as a modular application of particle mixing in CFD cells between advancement steps.

3.2.1. Coupling HiPS parcels and CFD particles

In Lagrangian transport modeling, the number of particles within each grid cell can vary dynamically due to advection, diffusion, and interactions with sources or sinks. As a result, the number of flow particles associated with a given cell may fluctuate. However, for a given Re, Eq. 10, HiPS uses a fixed number of parcels, that is, a power of two.

To address this constraint, a projection step is introduced, as performed using `set_varData` introduced in Section 2.2. Consider a computational domain divided into multiple grid cells. In Fig. 5, one such grid cell is highlighted to illustrate the projection process. Within this cell, flow particles are distributed as shown by black circles. Each flow particle has a weight, w, which reflects the fraction of the physical volume it represents. For visualization purposes, the sizes of the circles are scaled accordingly. The weights are normalized so their sum is unity.

The flow particles and uniformly sized HiPS parcels are arranged spatially as indicated in Fig. 5b,c. The horizontal length of the fluid particles and HiPS parcels correspond to their volumes.

The projection is done so as to conserve the scalar. The flow and HiPS scalar values in particle i, and parcel j are denoted ϕ_i^F, ϕ_j^H, with volumes V_i^F and V_j^H, and there are N_F, and N_H fluid particles and HiPS parcels considered, respectively. Scalar ϕ_j^H is computed as

$$\phi_j^H = \frac{\sum_{i\in I_j} \phi_i^F \hat{V}_{j,i}^F}{V_j^H}. \tag{11}$$

Here, I_j is the set of fluid particle indices that overlap HiPS parcel j. In Fig. 5, fluid particles $i = 0$ and $i = 1$ overlap HiPS parcel $j = 0$ (indexing from left

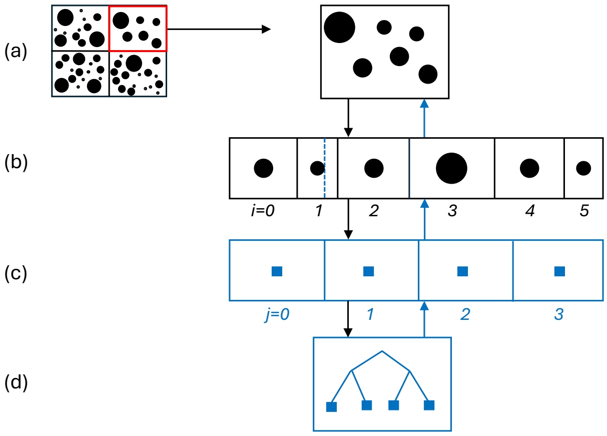

Figure 5: Overview of the HiPS projection process in a selected grid cell. **(a)** A single grid cell is highlighted, showing the distribution of flow particles. Circle sizes are proportional to particle weights, representing the volume each particle occupies. **(b)** Flow particles are distributed within the cell, with sizes indicating their relative weights. **(c)** The cell is partitioned into equally sized HiPS parcels. **(d)** Parcels are organized into a binary tree structure for hierarchical mixing within the HiPS model.

to right beginning at zero). $\hat{V}^F_{j,i}$ is the portion of a fluid particle volume i that overlaps the HiPS particle j. In Fig. 5, $\hat{V}^F_{0,0} = V^F_0$, and $\hat{V}^F_{0,1}/ \approx (2/3)V^F_1$, indicated by the dashed line in Fig. 5b. The implementation proceeds left to right over HiPS parcels in an outer loop, with an inner loop over sequantial fluid particles that overlap the given HiPS parcel. This process can be reversed to project parcel properties back to the fluid particles.

This formulation applies to scalars governed by conservation laws, such as energy, or chemical species mass fractions. For variable density flows, a mass weighting is used.

In some cases, a large discrepancy between the number of flow particles and the number of HiPS parcels can arise—for example, when many FPs are present but the Reynolds number results in a small HP count. This mismatch can reduce statistical accuracy and introduce projection error. To address this, multiple HiPS trees can be assigned per cell. This strategy improves representation and reduces information loss by distributing the projection across several independent HiPS trees.

3.2.2. Continuous Reynolds Numbers

In flow simulations, the Reynolds number (Re) can take any positive value depending on the flow conditions. In contrast, within the HiPS model, Re values are discrete, corresponding to the discrete number of tree levels. As a result, the characteristic scale l^* at level i^* corresponding to a given Re may not coincide exactly with a predefined tree level and may instead fall between two adjacent levels $i_m \equiv \text{floor}(i^*)$, $i_p \equiv \text{ceil}(i^*)$.

To accurately represent flow properties when i^* falls between discrete levels, four methods are available:

1. **Rounding to the nearest level:** This method rounds i^* to the closest

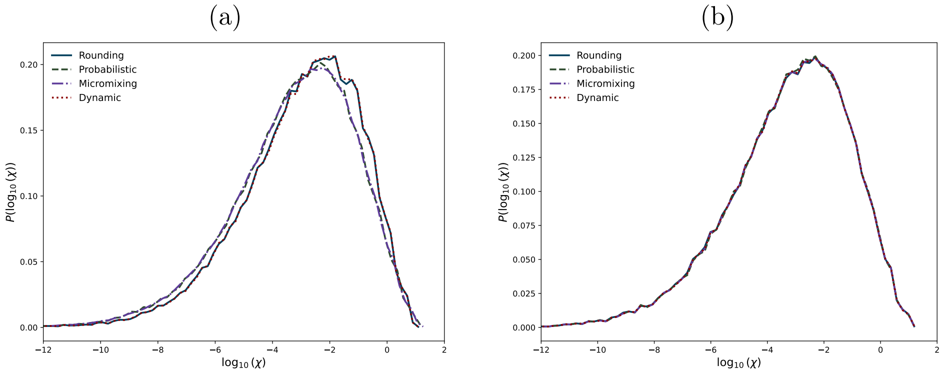

Figure 6: Comparison of the four methods using the probability density function for scalar dissipation rate at (a) $Re = 750$ and (b) $Re = 1500$.

discrete level, either i_m or i_p, and uses the corresponding length and time scales for micromixing.

2. **Probability-based evaluation:** Eddies at level i_p are accepted with a probability $P_{i^*} = i^* - i_m$ based on the relative position of i^* to i_m.

3. **Micromixing at level i_p with adjusted time scale:** Micromixing occurs at level i_p, but the mixing time scale is computed using the adjusted length scale L_s^*. As a result, Eq. 2 is modified to

$$\tau_{i_p} = \tau_0 A^{2i^*/3}.$$

4. **Dynamic adjustment of parameter A:** This method slightly changes the parameter A to dynamically ensure that i^* remains consistent with the hierarchical tree structure. It rounds i^* to the nearest integer and updates A accordingly to maintain consistency with the length scales.

The accuracy of these methods is evaluated using the probability density function (PDF) of the scalar dissipation rate, χ, which quantifies the rate of decay of scalar property fluctuations, e.g., temperature or concentration.

The methods were tested using simulations at Reynolds numbers $Re = 750$ and $Re = 1500$, corresponding to i^* of 7.16 and 7.93, respectively. The results are similar for the four methods. At $Re = 750$, the rounding and dynamic results are similar and evaluate micromixing at level $i^* = 7$ with the dynamic adjusted $A = 0.492$. The probabilitic and micromixing results are similar and evaluate at $i^* = 8$. For $Re = 1500$, all four methods evaluate at $i^* = 8$, and give nearly identical results. The dynamic adjusted $A = 0.504$ for this Re.

4. Impact

The HiPS model has the potential to significantly advance the simulation of turbulent mixing and combustion. It provides a more physically detailed representation of turbulent mixing by accounting for different time and length scales and modeling scalars with variable Schmidt numbers. This level of physical detail is crucial for simulations requiring precise modeling of turbulent flows and chemical reactions.

Furthermore, HiPS is designed to address the inherent challenges of combustion modeling, which involve multiple coupled processes such as turbulence, chemical reactions, radiation, soot formation, and sub-grid scale phenomena. Its comprehensive structure allows these elements to be integrated within a unified framework, making it particularly suited for combustion applications. This capability is useful for CFD users who need to simultaneously capture these interactions, a task often difficult to achieve with standard models.

In addition to its technical advantages, HiPS reduces the need for users to develop and manage separate models for each physical process. By providing an integrated approach, it enables faster progress in both research and industrial applications, making it a valuable tool for the broader scientific community.

HiPS has also been extended for use alongside other simulation models, broadening its applicability. As a standalone tool, it enables investigations of turbulent flows, offering researchers new opportunities to explore turbulence.

5. Conclusions

Modeling turbulent flows with mixing is complex due to the varying time scales, length scales, and species involved. Although various models have shown success, there remains a need for models that accurately capture turbulent mixing while maintaining computational efficiency.

This study presents a C++ library for the Hierarchical Parcel Swapping model, designed to represent multiple time and length scales within turbulent flows and account for variable Schmidt numbers–an essential feature for simulations involving species with differing diffusivities, such as soot. The library operates both as a standalone mixing model and as a subgrid mixing model for computational fluid dynamics (CFD) simulations. We provided examples demonstrating HiPS' capability to mix scalars with different Sc numbers and to model flame ignition and extinction. These examples confirm the model's effectiveness in capturing essential mixing behaviors and chemical reactions, highlighting its versatility and precision in simulating complex turbulent mixing processes.

In addition to the mixing HiPS formulation presented, a flow formulation is possible [8] in which momentum components are solved that dynamically local eddy rates within the tree structure. Such formulation would allow application to more general flow configurations and will be considered in future work.

Acknowledgements

H.S. acknowledges support by the German Federal Government, the Federal Ministry of Research, Technology and Space, and the State of Brandenburg within the framework of the joint project EIZ: Energy Innovation Center (project numbers 85056897 and 03SF0693A) with funds from the Structural Development Act (Strukturst¨arkungsgesetz) for coal-mining regions.

References

[1] S. Pope, PDF methods for turbulent reactive flows, Progress in Energy and Combustion Science 11 (2) (1985) 119–192.

[2] D. C. Haworth, Progress in probability density function methods for turbulent reacting flows, Progress in Energy and Combustion Science 36 (2010) 168–259.

[3] J. Villermaux, J. C. Devillon, Représentation de la coalescence et de la redispersion des domaines de ségrégation dans un fluide par un modèle d'interaction phénoménologique, Proceedings of the 2^{nd} International Symposium on Chemical Reaction Engineering 2 (1972) 1–13.

[4] J. Janicka, W. Kolbe, W. Kollmann, Closure of the transport equation for the probability density function of turbulent scalar fields, Journal of Non-Equilibrium Thermodynamics 4 (1979) 47–66.

[5] S. Subramaniam, S. B. Pope, A mixing model for turbulent reactive flows based on Euclidean minimum spanning trees, cnf 115 (1998) 487–514.

[6] R. Fox, Computational models for turbulent reacting flows, Cambridge University Press, Cambridge, United Kingdom, 2003.

[7] A. R. Kerstein, Hierarchical parcel-swapping representation of turbulent mixing. Part 1. formulation and scaling properties, jstatphys 153 (2013) 142–161.

[8] A. R. Kerstein, Hierarchical parcel swapping representation of turbulent mixing. Part 2. Application to channel flow, Journal of Fluid Mechanics 750 (2014) 421–463.

[9] A. R. Kerstein, Hierarchical parcel-swapping representation of turbulent mixing. Part 3. origins of correlation patterns observed in turbulent boundary layers, Physical Review Fluids 6 (2021) 044611.

[10] M. Behrang, T. Starick, I. Wheeler, H. Schmidt, A. Kerstein, D. Lignell, Turbulent mixing of scalars with nonunity schmidt numbers using hierarchical parcel-swapping, in review, Journal of Fluid Mechanics (2025).

[11] D. O. Lignell, G. C. Fredline, A. D. Lewis, Comparison of one-dimensional turbulence and direct numerical simulations of soot formation and transport in a nonpremixed ethylene jet flame, Proceedings of the Combustion Institute 35 (2015) 1199–1206.

[12] G. Sun, D. O. Lignell, J. C. Hewson, C. R. Gin, Particle dispersion in homogeneous turbulence using the one-dimensional turbulence model, Physics of Fluids 26 (2014) 103301.

[13] S. B. Pope, Turbulent flows, Cambridge University Press, New York, 2000.