Computers and Chemical Engineering 210 (2026) 109615

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Reproducibility of GPU-based Large Eddy Simulations for mixing in stirred tank reactors

Ryan Rautenbach [a], Héctor Maldonado de León [b], Pieter Brorens [b], Michael Schlüter [a], Cees Haringa [b,*]

[a] Institute of Multiphase Flows, Hamburg University of Technology, Eißendorfer Straße 38, Hamburg, 21073, Germany
[b] Department of Biotechnology, Delft University of Technology, Van der Maasweg 9, Delft, 2629 HZ, The Netherlands

ARTICLE INFO

Dataset link: https://doi.org/10.18419/DARUS-5523

Keywords:
CFD
Mixing
Bioreactor
LES
Floating-point
Reproducibility

ABSTRACT

CFD simulations are widely used to quantify the mixing performance of stirred tanks for various applications in chemical engineering and biotechnology. Due to advances in GPU computing, these simulations increasingly employ Large Eddy Simulation (LES), which explicitly resolves the dynamics of large-scale turbulence. Although such simulations are fully deterministic and therefore theoretically reproducible, small numerical variations induced by round-off errors, floating-point arithmetic, and differences in the distribution and ordering of operations in parallel computing lead to separation of trajectories i.e., different flow-field evolutions and consequently to significant run-to-run variability in predicted mixing times, even on the same hardware architecture. This work investigates the impact of repeated simulations, in the form of a case study, on the mixing-time distribution observed in a 30 L stirred tank reactor using two commercial CFD packages operating with representative, production-level solver configurations. The analysis does not aim to assess the general performance of numerical method classes, but rather to quantify run-to-run variability under fixed solver settings and to compare the resulting numerical distributions to experimental variability. The results demonstrate that numerical variability is of comparable magnitude to the experimental spread, highlighting the necessity to treat LES-derived metrics as statistical ensembles rather than deterministic values. It is concluded that the reporting of confidence intervals is essential for methodological rigour in LES-based mixing studies.

1. Introduction

Mixing is a common industrial operation that has been thoroughly researched over many decades, both experimentally (Kramers et al., 1953; Nienow, 2014) and numerically (Ranade et al., 1991; Van den Akker, 2006; Hartmann et al., 2006). Numerous protocols exist to quantify mixing, from invasive single-point measurements to spatially resolved methods (Fitschen et al., 2021). As both the measurement method and the definition of the criterion (e.g. 90 or 95% homogenisation) vary, a unique "mixing time" cannot generally be assigned to a given system. Even with nominally identical setups, inter-facility variability of order 15% has been reported (Kraume and Zehner, 2001). In practice, a mixing time distribution is observed in turbulent stirred vessels; beyond experimental noise, this reflects the stochastic, chained nature of turbulent transport and the influence of macro-instabilities (MIs) (Paglianti et al., 2006; Ducci and Yianneskis, 2007). Hence, experimental results should be reported with confidence intervals.

Computational fluid dynamics (CFD) is algorithmically deterministic and, under identical arithmetic and execution, should in principle yield identical outputs for a fixed input. For Reynolds-averaged Navier–Stokes (RANS) simulations, when rotor-stator interactions and MIs are negligible, frozen-flow approaches further reduce variability. On the other hand, variability due to turbulent fluctuations is eliminated by the Reynolds-Averaging operation, while MIs remain (Haringa et al., 2018a,b). If rotor-stator interactions and MIs are negligible, applying frozen-flow eliminates variability (Coroneo et al., 2011).

In contrast, for dynamic methods that resolve unsteady turbulence notably LES rerunning the same setup produces different instantaneous realisations and, consequently, different single-run mixing times. The present study is thereby intentionally framed as a case study using two widely employed, production-level LES solver configurations implemented with standard GPU-accelerated settings recommended for the respective solvers. The configurations were not optimised to match experimental results or to minimise variability, but instead reflect typical choices used in applied CFD studies. No claims are intended regarding the general performance of specific discretisation classes. Instead, the focus is on quantifying run-to-run variability and reproducibility under

* Corresponding author.
E-mail address: c.haringa@tudelft.nl (C. Haringa).

https://doi.org/10.1016/j.compchemeng.2026.109615
Received 7 November 2025; Received in revised form 30 January 2026; Accepted 1 March 2026
Available online 2 March 2026

fixed, reasonable, and converged LES setups representative of common practice.

While the non-deterministic nature of parallel execution is well established in high-performance computing, its implications for engineering metrics such as mixing time are frequently overlooked in the broader CFD community. This work aims to bridge this gap by providing a detailed quantification of how machine-level perturbations propagate into macroscopic observables. By doing so, we aim to establish a standard for methodological rigour in which LES results are treated as statistical ensembles rather than single-point solutions. This motivates two questions:

1. What is the origin of the observed run-to-run variability in LES mixing simulations?
2. How does this numerical variability compare to experimental mixing time distributions?

2. Origin of variability in LES simulations

Large Eddy Simulations (LES) (and Direct Numerical Simulations - DNS) of turbulent flows exhibit sensitivity to arbitrarily small perturbations. Even when the governing algorithms are deterministic, parallel execution and low-level arithmetic details modify the ordering and rounding of elementary operations, thereby injecting perturbations at machine precision. In a turbulent flow with positive finite-time Lyapunov exponents, such perturbations are amplified until trajectories decorrelate, yielding distinctly different instantaneous flow fields and consequently, variations in single-run mixing times (Senoner et al., 2008). Increasing arithmetic precision delays but does not prevent this divergence. The growth rate is governed by the flow physics rather than by the choice of precision; a non-turbulent flow will not propagate the same way a highly turbulent flow does (Senoner et al., 2008).

2.1. Floating-point model and rounding schedule

Floating-point arithmetic provides the numerical framework for machine computations, enabling efficient representation and calculation. However, a floating-point value generally approximates, rather than equals, the corresponding real number.

Let $\mathrm{RN}(\cdot)$ denote the IEEE 754 (IEEE, 2019) round-to-nearest, ties-to-even operator. For a binary operation $\circ \in \{+, -, \times, \div\}$ the computed result is

$$\mathrm{fl}(x \circ y) = \mathrm{RN}(x \circ y), \tag{1}$$

and for composed expressions the rounding operator is applied after each elementary operation unless otherwise specified by the instruction set and compiler. Since most real numbers are not exactly representable in a fixed-precision binary format, both data conversion and arithmetic incur rounding. Crucially, $\mathrm{fl}(x+y)$ is not associative, meaning in general

$$\mathrm{fl}\big(\mathrm{fl}(x+y)+z\big) \neq \mathrm{fl}\big(x+\mathrm{fl}(y+z)\big), \tag{2}$$

and the discrepancy depends on the magnitudes of (x, y, z) and on the local unit roundoff (Goldberg, 1991; Higham, 2021). In parallel algorithms (e.g. reductions of sums, norms, inner products), the pairing order defines the sequence of applications of (1). Changing that order changes the rounded intermediates and thus the final result.

2.2. Fused multiply-add contraction

Modern GPUs and many CPUs implement the fused multiply-add (FMA) structure, which computes

$$\mathrm{FMA}(a, b, c) = \mathrm{RN}(a\,b + c), \tag{3}$$

with a single final rounding, in contrast to the two-rounding path

$$\mathrm{RN}\big(\,\mathrm{RN}(a\,b) + c\,\big). \tag{4}$$

Eq. (3) is closer to the exact value of $ab + c$ for that expression, but it rounds differently than Eq. (4). Hence, two builds that differ only in their treatment of $ab + c$ can disagree slightly at each such site. This is quantified with "ulp", the unit in the last place which is the tiny step between two adjacent floating-point numbers near a value x. This means that the two results can differ by only a few ulps per contraction site (Goldberg, 1991; Higham, 2021; Lafage, 2020).

2.3. Parallel reductions, library choices, and determinism

On GPUs, reductions are implemented through trees whose pairing order can vary across kernel launches, problem partitions, or library algorithm choices. Because of non-associativity, distinct trees yield distinct rounded outcomes for the same mathematical sum or product (Demmel and Nguyen, 2013; Ahrens et al., 2020). In addition, vendor libraries may select among multiple numerically stable algorithms at runtime. Unless a deterministic mode is forced, run-to-run bitwise identity is not guaranteed.

2.4. Subnormals, mixed precision, and micro-perturbations

Floating-point formats include a subnormal regime, numbers so close to zero that they are represented with reduced precision to bridge the gap between the smallest normal number and zero. Many compilers and runtimes enable flush-to-zero (FTZ) or denormals-are-zero (DAZ), which, for speed, treat these tiny subnormal values as exactly zero. In practical terms, this only changes the last few bits of affected operations, but those bits are enough to shift a result by a few ulps. In a chaotic LES, such shifts provide a persistent source of micro-perturbations that will amplify over time, particularly taking affect if an otherwise relevant value is, for example, multiplied by a value that was tied to flush-to-zero or denormals-are-zero instead of a non-zero value (Harris, 2013; Intel Corporation, 2025; NVIDIA Corporation, 2025).

A similar effect arises with mixed-precision kernels or approximate intrinsics, certain operations are executed at lower precision, so their results are rounded on a coarser grid. This again alters the computation by a few ulps at specific sites, changing the rounding pathway. Individually, these differences are minute, collectively, and over many time-steps, they are sufficient to produce divergent instantaneous flow fields while preserving statistical agreement when properly averaged.

2.5. Implications for mixing time reproducibility

Mixing time, as used in this study, is a statistic inferred from transient LES fields. In practice, it depends not only on the resolved physics but also on how floating-point operations are scheduled on the hardware and compiled by the toolchain. Uncertainties from small numerical differences originating from the above-mentioned mechanisms balloon into a separation of trajectories, yielding variations in mixing time even for nominally identical runs. This means reproducibility of numerical LES mixing studies should be judged statistically, with metrics such as the mixing time requiring a confidence interval much like in experimental procedures. In this study we quantify the magnitude of this uncertainty and whether, given that small numerical perturbations are propagated through the physical LES-algorithm, this variability reflects experimental variability in mixing. The findings are determined for GPU-based software however the implications further apply to parallelised CPUs with only the magnitude of variability changing (Demmel and Nguyen, 2013).

If bitwise reproducibility is required, this can be achieved with a tightly controlled stack and deterministic arithmetic with fixed reduction trees and consistent contraction policies (Demmel and Nguyen, 2013; Ahrens et al., 2020). However, enforcing bitwise reproducibility risks locking onto a single, potentially unrepresentative flow realisation and thus a single fixed bias mixing time.

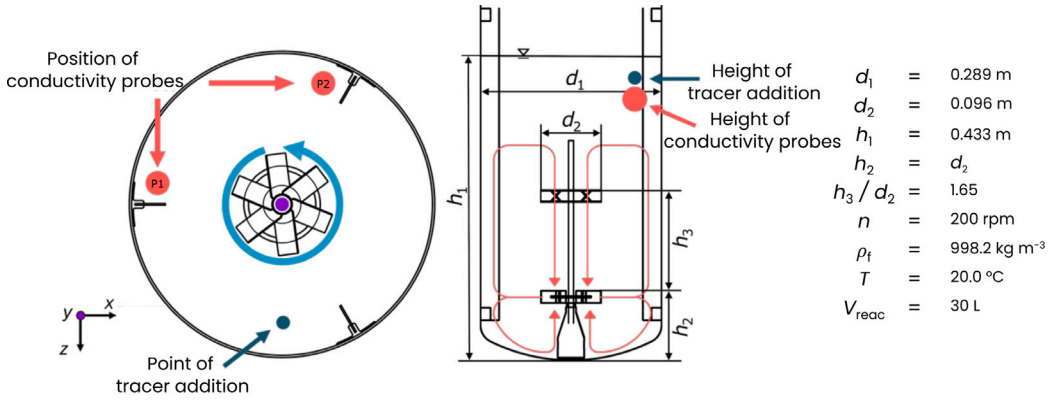

Fig. 1. Bioreactor geometry and instrumentation: Tracer addition point, conductivity probe locations, and principal geometric proportions used in the experiments.

Table 1
Location of the probes and tracer injection location used in the simulations and experiment.

Feature	x/m	y/m	z/m
Probe 1	−0.1270	0.4015	−0.0180
Probe 2	0.0530	0.4015	−0.1180
Injection	0	0.4055	0.1164

3. Quantifying variability: Methodology

To quantify the observed variability in numerically determined mixing times from nominally identical LES reruns, and comparison of observed variability against experimental data, the mixing time was determined in a 30L stirred tank from single-point conductivity probes (Meusel et al., 2020). This choice is motivated by (i) widespread use in the mixing literature, (ii) straightforward data processing, and (iii) rapid sensor response.

3.1. Experimental setup

Eleven independent repeats were performed in a 30 L bioreactor operated at 200 rpm, fitted with a Rushton turbine as the lower and a pitched-blade turbine as the upper impeller. The reactor geometry and instrumentation are illustrated in Fig. 1. Mixing time was defined as τ_{95}, consisting of the time for the conductivity signal to reach 95% of its asymptotic value after tracer injection. For each experiment, a 5 mL tracer solution of 4 M NaOH was injected into the reactor. Two conductivity probes (WTW-IDS TetraCon® 925), were used simultaneously, and the reported τ_{95} for each run was taken as the larger of the two probe-based values. With a response time of 1 s and the expected mixing time of $\tau_{95} \approx 25$ s, the probe dynamics are sufficiently fast that measurement error is expected to contribute only marginally to the observed mixing time distribution. The probe and injection positions are listed in Table 1.

3.2. Numerical setup

Large-eddy simulations (LES) were performed with two commercial solvers: M-Star CFD 3.12.21 (lattice Boltzmann, LB) and ANSYS Fluent 2024R1 (finite volume, FV). In both solvers the Smagorinsky subgrid model was employed with $C_S = 0.1$.

A mesh-dependence study was conducted for both packages in Section 4.4: For M-Star CFD, mesh independence was achieved at a resolution just below 9×10^7 lattice points. For ANSYS Fluent, mesh independence was reached at a cell count of 1.201×10^6, while the discussed results were calculated for a cell count of 3.408×10^6. These thresholds are consistent with prior reports using the same commercial software and with the respective vendor guidelines (Haringa et al., 2018b; Hofmann et al., 2025).

M-Star Setup: A standard D3Q19 lattice with single-relaxation-time (BGK) collision is used, explicit time integration, and Courant number Co = 0.1 (Kuschel et al., 2021). The lattice spacing is $\Delta x = 0.95$ mm, yielding $\approx 9 \times 10^7$ lattice nodes. This resolution is supported by the mesh-dependence study.

One simulation configuration includes ten passive-scalar tracer injections, each applied as a 1 s pulse beginning at $t = 25$ s, with the initial 25 s allowing the flow-field to be sufficiently stable and established. The simulation with the respective configuration is repeated five times, giving 50 independent mixing realisations. All runs are executed in single precision on an NVIDIA A100 GPU (CUDA 12.6). An auxiliary comparison between single and double precision is also performed.

Mixing is quantified via a virtual probe that mirrors the experimental sensor, consisting of a stationary, no-slip shaft geometry submerged in the vessel, with a small control volume located just below the shaft tip to sample the local tracer concentration. This setup replicates the experimental measurement procedure.

Fluent setup: A mesh of 3.41M elements was used, with local refinement at the impellers. The SIMPLE algorithm was set for P-V coupling, second-order discretisation was applied for pressure and tracer dispersion, and bounded central differencing was used for momentum, together with second-order implicit time integration. A sliding mesh method with a fixed timestep size of 0.005s was used. A simulation for flow-field convergence — residuals less than $O(-4)$ — was conducted using Fluent's GPU solver on a workstation equipped with a 16 AMD Ryzen ThreadripperPRO 5955WX CPUs, 64 GB RAM, and an Nvidia GeForce RTX 3090 graphics card, storing flow-fields at 1s intervals between $t = [25, \ldots, 34]$ s. From these files, 10 mixing simulations were set, each with a 1s interval in tracer injection time. Each of the 10 setups was run 5 times on the DelftBlue (Delft High Performance Computing Centre (DHPC), 2024) or Snellius supercomputers, either on an Nvidia V100S or A100 GPUs. The mixing simulations were conducted in double precision mode.

Setup comparison: Fluent vs. M-Star Both workflows analyse 50 tracer injections, but the starting protocols differ. In both packages, we run simulations with ten different starting times, at 1 s intervals. In FLUENT the ten starting files for each run are set up on sequential states taken from a single seed simulation, and each starting file is rerun five times to study variability in mixing starting from the same initial state. In M-Star, the ten injections share one configuration by injecting independent tracers in the same simulation at 1 s intervals, but the five repetitions are each a fully transient rerun starting from $t = 0$ s, evolving independently of previous or subsequent runs.

A second difference is the precision; while mixing in Fluent runs are executed in double precision, the M-Star runs use single precision (with an auxiliary single-vs-double comparison for M-Star). These protocol and precision choices are noted when interpreting variability between the two solvers.

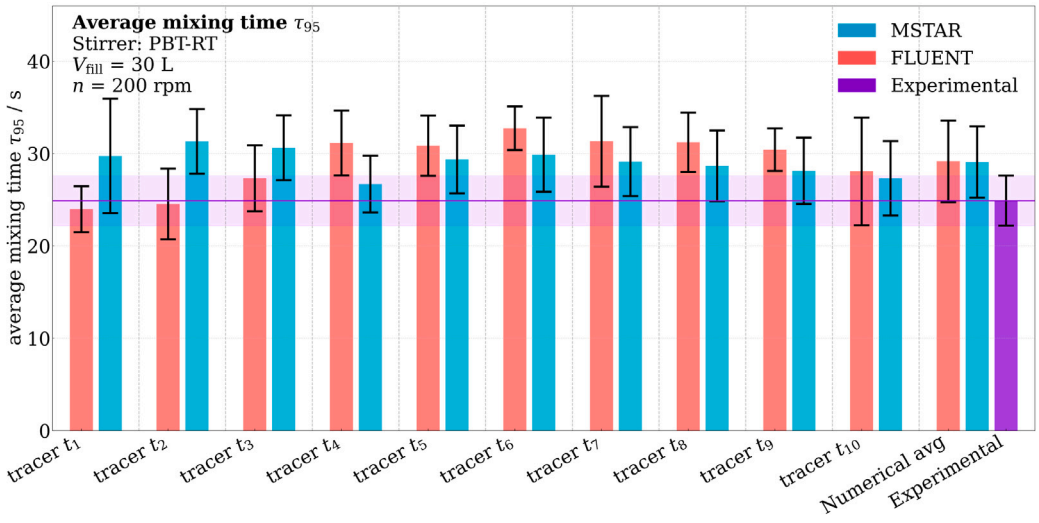

Fig. 2. Mixing times per injection from M-Star and Fluent (ten injections $t_1 - t_{10}$, spaced by 1 s) compared with experimental results (11 repeats), showing the means; error bars indicating standard deviation between numerical and experimental repeats.

4. Results

This section compares numerically determined mixing times from LES with experimental measurements and examines how run-to-run variability manifests across solvers and setups. The results proceed through the injection-time conditioned variability, to aggregate distributions, over trajectory-level evidence, then to grid effects and precision sensitivity via M-Star runs, and finally, the statistical reproducibility is assessed through time-averaged fields.

4.1. Results by injection time

To assess sensitivity to injection timing and capture multiple mixing time realisations during a single simulation, distributions are collected based on the time of tracer injection (t_1-t_{10} at 1 s intervals). Fig. 2 reports means and confidence intervals per tracer injection slot. Fluent shows slightly greater variation between slots than M-Star, however, the impact is small and likely consistent with protocol and precision differences. A quantifiable influence of macro instabilities is not possible with the present results, yet the variability between tracer injection times alludes to the macro instabilities having an influence on the numerical mixing progression as is evident from literature (Paglianti et al., 2006; Roussinova et al., 2003).

4.2. Mixing time distributions

Previous scale-resolving CFD studies have reported substantial variability in tracer responses when using repeated tracer injections. For example, Brown et al. (2021) observed significant variability in individual tracer responses and demonstrated that a sufficiently large number of tracer injections is required to obtain statistically representative average behaviour and meaningful confidence limits. However, these studies primarily treated the observed variability as a sampling and averaging issue, without analysing mixing-time distributions or investigating the origin of run-to-run variability in fully transient simulations. In contrast, the present work explicitly analyses mixing-time distributions obtained from repeated LES simulations and directly compares the numerical distributions with experimental measurements, with particular emphasis on GPU-accelerated workflows.

Overall, Fluent and M-Star produce similar mixing time distributions (Fig. 3). Both predict slightly longer means than the experiments, while the overall spread is comparable. The positive bias is likely due to the tracer model, experimentally a 4 M sodium-hydroxide solution

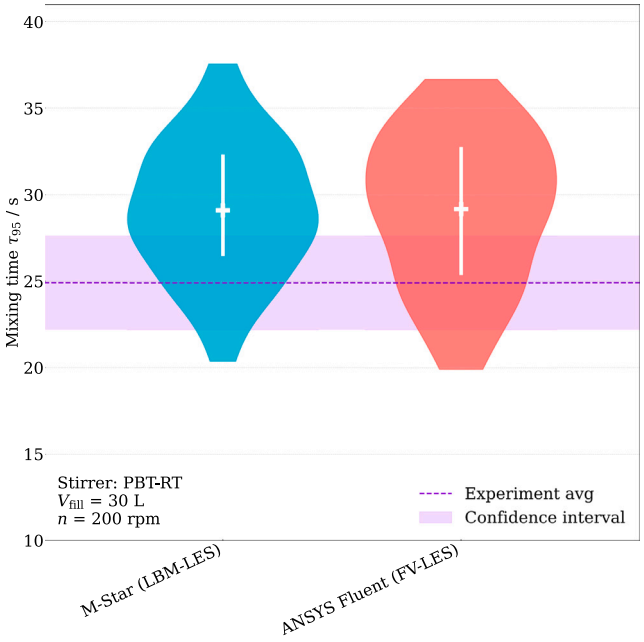

Fig. 3. Kernel-density estimates, average and standard deviation of τ_{95} for ANSYS Fluent (FV-LES) and M-Star CFD (LBM-LES) solver as violin plots, with the experimental average and variability.

with a non-negligible density contrast was injected and relevant initial momentum of addition was used, whereas the simulations used a neutrally buoyant passive scalar. Buoyancy and fluid property effects can accelerate local mixing near the injection point and reduce τ_{95} in the measurements.

The distributions pool all numerical realisations across reruns and multiple tracer injections per run and show that LES run-to-run variability closely mirrors the experimental spread. The variability across injection slots within a single baseline flow is comparable to what would be expected from fifty fully independent simulations, indicating that multiple tracer additions per run can, in part, substitute for separate repeats when estimating the mixing time distribution. Consequently, the results particularly highlight how the variability in GPU based LES procedure captures the underlying physical variability of the

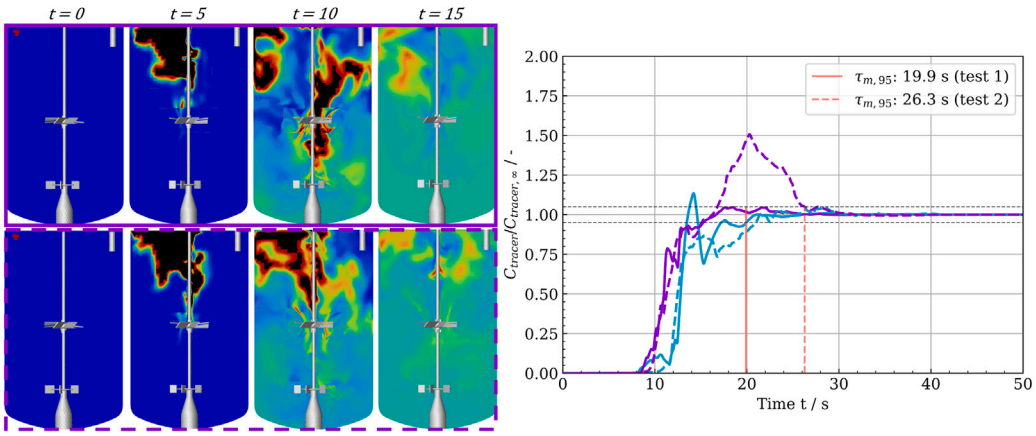

Fig. 4. Left: two mixing realisations (Fluent) with identical injection timing; snapshots at 0, 5, 10, and 15 s after injection. Right: probe traces (cyan: probe 1, purple: probe 2) for the two realisations (solid vs. dashed) and their τ_{95}.

process. This however also indicates that a fixation of the numerical procedure to ensure bitwise identical reruns can lead to false positives or negatives in matching the results to the experiments.

It is important to note that the selection of specific numerical schemes, such as spatial discretisation order or upwinding strategies, can influence absolute mixing-time predictions and consequently their associated variance, which in the present study is of similar magnitude to the variability observed experimentally. However, the objective of the present analysis is not to characterise the sensitivity of the results to the full configuration space of the solvers. Instead, by deliberately fixing the solver and simulation settings to representative, production-level configurations, the aim is to isolate the variability arising solely from repeated executions of nominally identical simulations. The results demonstrate that run-to-run variability is driven by low-level numerical execution effects, such as parallel reductions, FMA and rounding operations, and kernel scheduling, which interact with the chaotic nature of LES turbulence propagation and are intrinsic to the hardware-software coupling on modern parallel architectures, persisting even when solver settings are converged. This finding implies that reported LES metrics should be treated as statistical realisations rather than deterministic values, regardless of the underlying discretisation scheme.

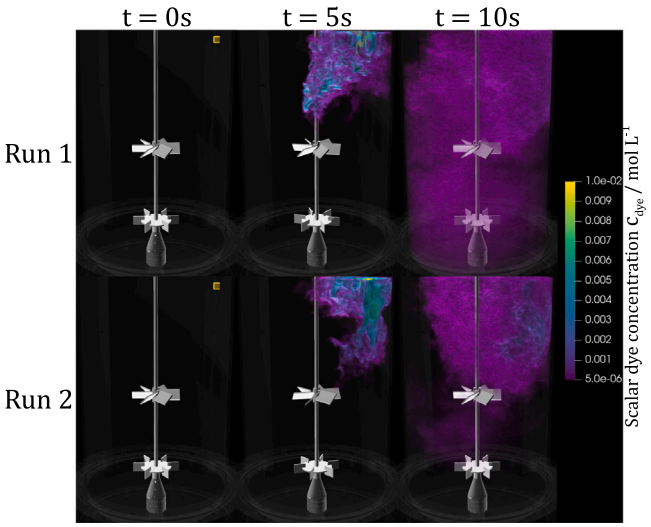

Fig. 5. Visualisation of trajectory separation for the tracer solution in M-Star with identical injection timing and numerical conditions; snapshots at 0, 5, and 10 s after injection.

4.3. Visualisation of trajectory separation

Instantaneous fields from nominally identical reruns diverge over time, yet the extracted τ_{95} values remain comparable. Fig. 4 shows two repeats of the same Fluent case: snapshot sequences at 0, 5, 10, and 15 s after injection (left) and the corresponding probe traces with their τ_{95} (right).

The snapshots make the separation of trajectories explicit. Within the first 5s after injection, runs that start from identical velocity fields already diverge. The resulting concentration fields differ in both pattern and extent throughout the vessel. The probe signals and the extracted τ_{95} quantify this divergence, showing how two nominally identical runs can yield distinctively different mixing times. This highlights the necessity for the use of repeated runs and multiple tracer additions to approximate the numerical mixing time distribution with adequate accuracy.

Similar to the separation of trajectories from ANSYS Fluent this separation is visible for the M-Star CFD runs. Here, the tracer injections between runs did not start on the identical seed velocity field. This separation is shown in 5 with the concentration distribution over time for two nominally identical fully transient runs.

The results highlight for one the already different transient velocity field but also the separation of trajectories after injection. This confirms that trajectory separation is not solver-specific but a consequence of the dynamics and floating-point execution and rounding dynamics occurring under parallel execution using FMA. Accordingly, mixing time statistics should be estimated from repeated realisations (or multiple tracer additions per run) rather than from a single outcome.

4.4. Grid-independence

Grid resolution was assessed separately for the LBM (M-Star) and FV (Fluent) solvers to verify that the reported mixing time statistics are insensitive to further refinement. Two practical criteria were used: (i) the change in the mean τ_{95} between successive grids should be small relative to the run-to-run variability, and (ii) the impeller power number should converge as a proxy for consistent momentum transfer from the moving body to the fluid.

Fig. 6 summarises the M-Star study across four lattice resolutions. Fig. 6(a) shows kernel-density estimates and average mixing times of the 50 mixing time realisations per grid as violin plot, while Fig. 6(b) shows the power number as a function of time.

For M-Star, mesh independence is achieved at $\approx 9 \times 10^7$ lattice nodes, the shift in the mean τ_{95} between the two finest grids is minimal

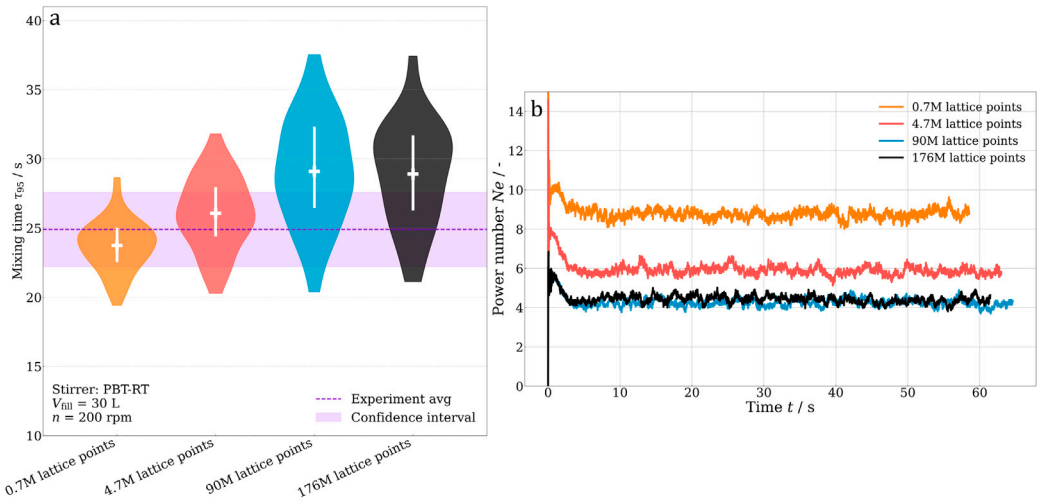

Fig. 6. Grid-independence for M-Star (LBM-LES), showing the (a) kernel-density estimates, average and standard deviation of τ_{95} for four lattice resolutions as violin plots, and (b) Power number $Ne(t)$ over time for the same grids.

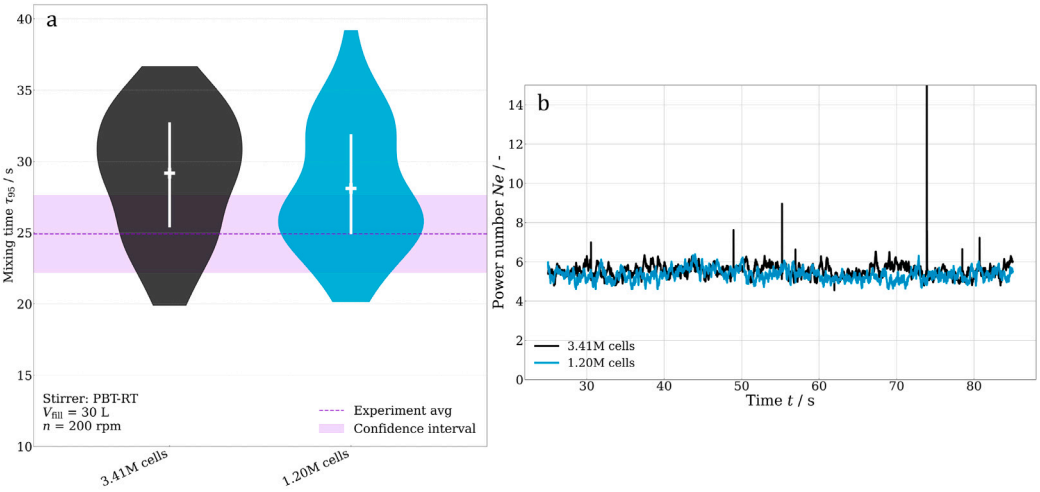

Fig. 7. Grid-independence for Fluent (FV-LES).

and within sampling variability, and the standard deviations are nearly unchanged. The power-number over time exhibits the same behaviour, with curves for the finest two resolutions essentially overlapping, indicating converged momentum transfer at this resolution.

Fig. 7 summarises the ANSYS Fluent study across two lattice resolutions. Fig. 7(a) shows kernel-density estimates and average mixing times of the 50 mixing time realisations per grid as violin plot, while Fig. 7(b) shows the power number as a function of time.

For the Fluent cases, mesh independence is achieved already at the lower resolution of 1.2×10^6 cells. This is evident in Fig. 7(a), where the dispersions, means, and standard deviations of the mixing-time results are closely aligned between the two mesh resolutions. Fig. 7(b) corroborates this via the power number, which shows that the transfer of mechanical energy to the fluid field is convergent at the given resolutions.

4.5. Effect of numerical precision (M-Star)

Sensitivity to arithmetic precision was assessed for the LBM-LES solver at a lattice resolution of $\approx 9 \times 10^7$ nodes by repeating the runs in single and double precision with identical configurations. (The single-precision results correspond to Fig. 3). Across repeats, the single- and double-precision distributions of τ_{95} are closely aligned. A kernel-density estimate highlights the similarity and is shown in Fig. 8. Double precision yields a slightly tighter distribution, consistent with the notion that higher precision delays but does not prevent trajectory separation (Senoner et al., 2008). The means and standard deviations are close to equal.

4.6. Statistical reproducibility of time-averaged fields

To test statistical reproducibility, a rolling average of the domain-averaged velocity was computed for Fluent and M-Star. At each time step, the velocity magnitude v_{mag} was averaged over the whole domain to give a single scalar value. This value was then cumulatively averaged from $t = 25$ s to $t = 75$ s. The window covers the full mixing process used to determine τ_{95}, and the rolling average damps instantaneous fluctuations, enabling a quantitative assessment of the reproducibility.

Each rerun yields a final scalar value of

$$\overline{v}_i = \overline{v}_{\text{roll}}(75\ \text{s}),$$

and a set of $\{\overline{v}_i\}_{i=1}^{5}$ per solver is summarised by the sample mean $\bar{v}$ and respective standard deviation s.

A 95% confidence interval (CI) for the variance is obtained from the chi-squared bounds with $m-1$ degrees of freedom. Through the square

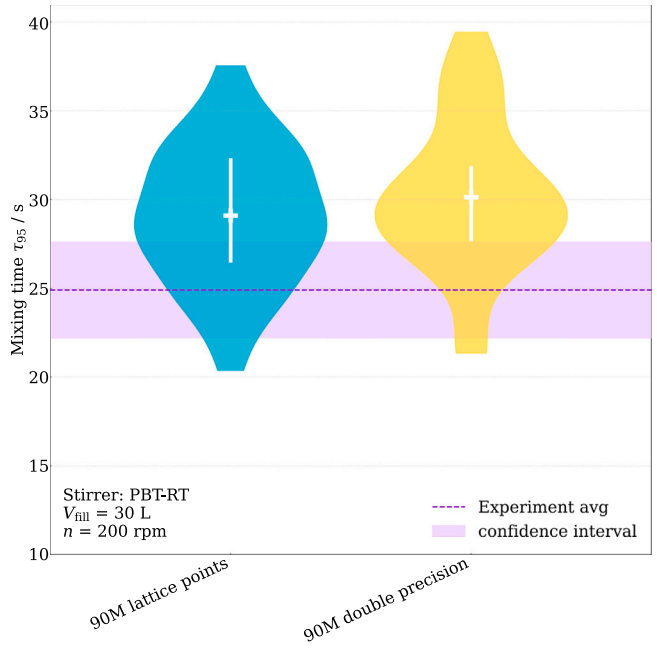

Fig. 8. Kernel-density estimates, average and standard deviation of τ_{95} for four lattice resolutions as violin plots for M-Star in single and double precision, compared to the experimental results.

roots the standard deviation with lower and upper limits s_- and s_+ of the CI is determined. A relative CI width defined as,

$$\delta_{\%} = 100\,\frac{s_+ - s_-}{\bar{v}},$$

details the width of the 95% CI standard deviation s normalised by the mean rolling average.

Table 2 summarises the M-Star and FLUENT reruns and their statistical convergence.

The final rolling averages $\bar{v}_i$ cluster tightly around $\bar{v}$ for both solvers. For M-Star, the relative standard deviation is 0.2371% with a standard deviation of $s = 3.54 \times 10^{-4}$ at an average of $\bar{v} = 0.149463$. The chi-squared bounds (df = 4) give a 95% CI for s of $[2.12\times10^{-4}, 1.02\times10^{-3}]$, yielding a relative CI width $\delta_{\%} = 0.539\%$.

Fluent shows comparable behaviour, with a relative standard deviation of 1.986% and slightly wider relative confidence interval of $\delta_{\%} = 4.516\%$. Together, these small dispersions indicate stable time-averaged speeds across independent reruns despite instantaneous trajectory separation.

5. Conclusions and recommendations

This study compares LES-based mixing times from different commercially available software packages with experiments and analyses how run-to-run variability appears across solvers, precisions, and grids. The conclusions drawn here are restricted to the specific solver configurations and numerical settings employed in this study and are not intended as a general performance assessment of numerical methods or discretisation classes. The central finding is that GPU-based LES produces physically valid and statistically consistent, but not bitwise-identical, flow fields. Instantaneous divergence between nominally identical reruns is evident. Nevertheless, the mixing time τ_{95} distributions from Fluent and M-Star reproduce the experimental spread with only a modest positive bias in the mean, plausibly linked to tracer-property differences.

Precision and grid choices, within the validated ranges, have a minor impact on τ_{95} compared with intrinsic variability. The observed dispersion persists when initiating from a seed velocity field compared to a fully transient flow field. Time-averaged fields and turbulence statistics are reproducible across solvers and repeats, supporting a statistical notion of agreement.

Table 2
Rolling-average statistics of domain-averaged speed for M-Star and Fluent reruns. Values correspond to the mean over the interval t = 25–75 s.

	M-Star (LBM-LES)	Fluent (FV-LES)
Run 1	0.149246	0.160027
Run 2	0.148943	0.156662
Run 3	0.149719	0.152836
Run 4	0.149744	0.159269
Run 5	0.149664	0.154225
Mean $\bar{v}$	0.149463	0.156604
Standard deviation s	0.000354	0.00311
Relative std. (%)	0.237113	1.985687
Lower s_- (95% CI)	0.000212	0.001863
Upper s_+ (95% CI)	0.001018	0.008936
CI width Δs	0.000806	0.007073
Relative CI width $\delta_{\%}$	0.539294	4.516292

In light of these results, reporting should shift from single values used to quantify mixing time to more complete mixing time distributions. Mixing time predictions should be presented with uncertainty considerations and with transparent information on the number of realisations, injection slots, and averaging windows. Reproducibility and validation should focus on statistical benchmarks such as distributions of τ_{95} and time-averaged statistical outputs.

When wall-time is constrained, multiple tracer injections within a run can efficiently sample the distribution where multiple separate runs are not feasible. Numerical settings that materially affect rounding pathways such as the contraction policy, FMA, or reduction order should be acknowledged where necessary while the variability should be quantified. Bitwise-deterministic modes remain useful for applications such as numerical speed benchmarking or software regression tests, but enforcing identical bit patterns is not an accurate approach for stochastic metrics such as the mixing time.

Key findings:

1. LES-based mixing simulations are practically irreproducible in the bitwise sense, consequently, results should be reported with uncertainty measures rather than as single values, highlighting the run-to-run variability.
2. The numerical variability aligns with experimental variability and is of similar magnitude in single- and double-precision runs.
3. The dispersion of τ_{95} is not notably sensitive to the initialisation protocol and multiple tracer injections per run can partially substitute for separate reruns when estimating the distribution.
4. Within validated resolutions, further grid refinement and precision changes do not shift the τ_{95} distribution relative to the spread.

Overall, the results support judging LES by statistical rather than bitwise reproducibility. Although we here focused on GPU-based implementations, similar impacts are expected in (parallelised) CPU implementations. LES yields physically sound yet non-identical instantaneous fields. For applications such as mixing time estimation, the numerical variability closely mirrors the dispersion observed experimentally. Far from being a drawback, this variability can aid realistic process characterisation and operation by sampling the true distribution of outcomes.

An important implication of these results is that the conclusions drawn here for mixing time extend beyond this specific metric. Any LES-derived metric that depends on transient flow realisations such as power number, residence time, scalar statistics, or local shear measures may exhibit comparable run-to-run variability when obtained from a single simulation. Treating such quantities as deterministic outputs of a single LES run therefore risks underestimating uncertainty. In this

Table 3
Details about the hardware per HPC system used for running the simulations in ANSYS Fluent and M-Star CFD.

Solver	System	GPU	CPU
ANSYS Fluent	DelftBlue (DB)	NVIDIA Tesla A100 (80 GB)	AMD EPYC 7402 (24 cores)
ANSYS Fluent	DelftBlue (DB)	NVIDIA Tesla V100S (32 GB)	Intel Xeon E5-6448Y (32 cores)
ANSYS Fluent	Snellius (SN)	NVIDIA Tesla A100 (40 GB)	Intel Xeon Platinum 8360Y (32 cores)
M-Star CFD	Local Cluster	NVIDIA A100 (40 Gb)	AMD EPYC 7702 (64 cores)

sense, LES studies should be conducted and interpreted in replicate, analogous to experimental practice, with statistical characterisation of the resulting observables rather than reliance on single realisations.

CRediT authorship contribution statement

Ryan Rautenbach: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Héctor Maldonado de León:** Writing – review & editing, Visualization, Methodology, Investigation, Formal analysis, Data curation. **Pieter Brorens:** Writing – review & editing, Investigation, Data curation. **Michael Schlüter:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Cees Haringa:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We acknowledge Jürgen Fitschen for initially pointing out this problem, Thi Thai Le, Dieter Bothe, Martin Sommerfeld, Holger Marshall, Matthias Kraume and Johannes Wutz for insightful discussions. RR is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Germany - 427899833. PB is supported by the Dutch National Growth Fund Cellular Agriculture. International exchange is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Germany - SFB 1615 - 503850735. CH, HML, and PB acknowledge the use of computational resources of the DelftBlue supercomputer, provided by Delft High Performance Computing Centre (https://www.tudelft.nl/dhpc), and the supercomputer Snellius (https://www.surf.nl/en/services/snellius-the-national-supercomputer) (Grant no. EINF-11590, Project: "CFD-derived data for surrogate models of bioreactors").

Appendix. Hardware employed

For running the LES simulations, the following hardware systems were used for both solvers (see Table 3).

Data availability

All data is provided via the following repository https://doi.org/10.18419/DARUS-5523.

References

Ahrens, P., Demmel, J., Nguyen, H.D., 2020. Algorithms for efficient reproducible floating point summation. ACM Trans. Math. Software 46, http://dx.doi.org/10.1145/3389360.

Brown, G.J., Fletcher, D.F., Leggoe, J.W., Whyte, D.S., 2021. Application of hybrid RANS-LES models to the prediction of mixing time and residence time distribution: Case study of a draft tube reactor. Chem. Eng. Sci. 240, http://dx.doi.org/10.1016/j.ces.2021.116676.

Coroneo, M., Montante, G., Paglianti, A., Magelli, F., 2011. CFD prediction of fluid flow and mixing in stirred tanks: Numerical issues about the RANS simulations. Comput. Chem. Eng. 35 (10), 1959–1968. http://dx.doi.org/10.1016/j.compchemeng.2010.12.007.

Delft High Performance Computing Centre (DHPC), 2024. Delftblue supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Demmel, J., Nguyen, H.D., 2013. Fast reproducible floating-point summation. In: Proceedings - Symposium on Computer Arithmetic. pp. 163–172. http://dx.doi.org/10.1109/ARITH.2013.9, URL https://dl.acm.org/doi/10.1109/ARITH.2013.9.

Ducci, A., Yianneskis, M., 2007. Vortex identification methodology for feed insertion guidance in fluid mixing processes. Chem. Eng. Res. Des. 85 (5), 543–550. http://dx.doi.org/10.1205/CHERD06192, URL https://www.sciencedirect.com/science/article/pii/S0263876207730812.

Fitschen, J., Hofmann, S., Wutz, J., Kameke, A.V., Hoffmann, M., Wucherpfennig, T., Schlüter, M., 2021. Novel evaluation method to determine the local mixing time distribution in stirred tank reactors. Chem. Eng. Sci.: X 10, 100098. http://dx.doi.org/10.1016/J.CESX.2021.100098.

Goldberg, D., 1991. What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv. 23, 5–48. http://dx.doi.org/10.1145/103162.103163, URL https://dl.acm.org/doi/pdf/10.1145/103162.103163.

Haringa, C., Vandewijer, R., Mudde, R.F., 2018a. Inter-compartment interaction in multi-impeller mixing: Part I. Experiments and multiple reference frame CFD. Chem. Eng. Res. Des. 136, 870–885. http://dx.doi.org/10.1016/J.CHERD.2018.06.005, URL https://www.sciencedirect.com/science/article/pii/S0263876218302909.

Haringa, C., Vandewijer, R., Mudde, R.F., 2018b. Inter-compartment interaction in multi-impeller mixing. Part II. Experiments, sliding mesh and large Eddy simulations. Chem. Eng. Res. Des. 136, 886–899. http://dx.doi.org/10.1016/J.CHERD.2018.06.007, URL https://www.sciencedirect.com/science/article/pii/S0263876218302922.

Harris, M., 2013. CUDA pro tip: Flush denormals with confidence. NVIDIA Technical Blog. URL https://developer.nvidia.com/blog/cuda-pro-tip-flush-denormals-confidence/.

Hartmann, H., Derksen, J.J., van den Akker, H.E.A., 2006. Mixing times in a turbulent stirred tank by means of LES. AIChE J. 52 (11), 3696–3706. http://dx.doi.org/10.1002/aic.10997, URL http://doi.wiley.com/10.1002/aic.10997.

Higham, N.J., 2021. The mathematics of floating-point arithmetic. LMS Newsl. (493), 35–41.

Hofmann, S., Rautenbach, R., Buntkiel, L., Brouwers, I.S., Gaugler, L., Barczyk, J., Fitschen, J., Reinecke, S., Hoffmann, M., Takors, R., Hampel, U., Schlüter, M., 2025. Lagrangian sensor particles for detecting hydrodynamic heterogeneities in industrial bioreactors: Experimental analysis and Lattice-Boltzmann simulations. Chem. Eng. J. Adv. 22, 100744. http://dx.doi.org/10.1016/J.CEJA.2025.100744.

IEEE, 2019. Ieee standard for floating-point arithmetic. http://dx.doi.org/10.1109/IEEESTD.2019.8766229, URL https://ieeexplore.ieee.org/document/8766229.

Intel Corporation, 2025. Set the FTZ and DAZ flags. Intel oneAPI DPC++/C++ Compiler Developer Guide and Reference. URL https://www.intel.com/content/www/us/en/docs/dpcpp-cpp-compiler/developer-guide-reference/2025-0/set-the-ftz-and-daz-flags.html.

Kramers, H., Baars, G.M., Knoll, W.H., 1953. A comparative study on the rate of mixing in stirred tanks. Chem. Eng. Sci. 2 (1), 35–42. http://dx.doi.org/10.1016/0009-2509(53)80006-0.

Kraume, M., Zehner, P., 2001. Experience with experimental standards for measurements of various parameters in stirred tanks: A comparative test. Chem. Eng. Res. Des. 79 (8), 811–818. http://dx.doi.org/10.1205/02638760152721316.

Kuschel, M., Fitschen, J., Hoffmann, M., von Kameke, A., Schlüter, M., Wucherpfennig, T., 2021. Validation of novel Lattice Boltzmann large Eddy simulations (LB LES) for equipment characterization in biopharma. Process. 9 (6), 950. http://dx.doi.org/10.3390/PR9060950, URL https://www.mdpi.com/2227-9717/9/6/950/htm https://www.mdpi.com/2227-9717/9/6/950.

Lafage, V., 2020. Revisiting what every computer scientist should know about floating-point arithmetic. arXiv preprint arXiv:2012.02492.

Meusel, W., Löffelholz, C., Husemann, U., Dreher, T., Greller, G., Kauling, J., Eibl, D., Kleebank, S., Bauer, I., Glöckler, R., Huber, P., Kuhlmann, W., John, G.T., Werner, S., Kaiser, S.C., Pörtner, R., Kraume, M. (Eds.), 2020. Recommendations for Process Engineering Characterisation of Single-Use Bioreactors and Mixing Systems by Using Experimental Methods, second completely revised ed. DECHEMA

Gesellschaft für Chemische Technik und Biotechnologie e.V., Frankfurt am Main, Germany, URL https://dechema.de/dechema_media/Downloads/Positionspapiere/Single_Use_BioReactors_2020-p-20006899.pdf.

Nienow, A.W., 2014. Stirring and stirred-tank reactors. Chem. Ing. Tech. 86 (12), 2063–2074. http://dx.doi.org/10.1002/CITE.201400087, URL https://onlinelibrary.wiley.com/doi/full/10.1002/cite.201400087.

NVIDIA Corporation, 2025. NVIDIA HPC Compilers User's Guide. See Section 2.5 "Floating-point Subnormal" and related environment variables. URL https://docs.nvidia.com/hpc-sdk/compilers/hpc-compilers-user-guide/contents.html.

Paglianti, A., Montante, G., Magelli, F., 2006. Novel experiments and a mechanistic model for macroinstabilities in stirred tanks. AIChE J. 52, 426–437. http://dx.doi.org/10.1002/AIC.10634.

Ranade, V.V., Bourne, J.R., Joshi, J.B., 1991. Fluid mechanics and blending in agitated tanks. Chem. Eng. Sci. 46 (8), 1883–1893. http://dx.doi.org/10.1016/0009-2509(91)80150-W.

Roussinova, V., Kresta, S.M., Weetman, R., 2003. Low frequency macroinstabilities in a stirred tank: scale-up and prediction based on large Eddy simulations. Chem. Eng. Sci. 58, 2297–2311. http://dx.doi.org/10.1016/S0009-2509(03)00097-6.

Senoner, J.-M., García, M., Mendez, S., Staffelbach, G., Vermorel, O., 2008. Growth of rounding errors and repetitivity of large-Eddy simulations. AIAA J. http://dx.doi.org/10.2514/1.34862.

Van den Akker, H.E., 2006. The details of turbulent mixing process and their simulation. Adv. Chem. Eng. 31, 151–229. http://dx.doi.org/10.1016/S0065-2377(06)31003-4.