

FlowFit3: Efficient data assimilation of LPT measurements

Philipp Godbersen^{1,*}, Sebastian Gesemann¹, Daniel Schanz¹, Andreas Schröder^{1,2}

1: German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, Göttingen, Germany

2: Brandenburgisch Technische Universität (BTU) Cottbus-Senftenberg, Germany

*Corresponding author: Philipp.Godbersen@dlr.de

ABSTRACT

Lagrangian particle tracking (LPT, (Schröder & Schanz, 2023)) techniques such as Shake-The-Box (Schanz et al., 2016) provide accurate flow measurements by following the movement of tracer particles within a flow. The technique gives measured values of position, velocity and acceleration at the locations of the scattered particles as opposed to a Eulerian grid like with window based techniques such as PIV. This is advantageous for e.g spatially highly resolved flow statistics. Often it is also desirable to have access to a grid-structured representation of the data, especially for the calculation of spatial gradients of flow field measures. While a simple interpolation onto a grid is possible, data assimilation techniques using physics based regularization such as FlowFit2 (Gesemann et al., 2016) or VIC+ (Schneiders & Scarano, 2016) and VIC# (Jeon et al., 2022) are frequently used nowadays. Through knowledge of the Navier-Stokes equations a higher spatial resolution can be obtained than from interpolation alone (see e.g. Zhou et al. (2024)). We present the new and highly efficient FlowFit3 approach and its evaluation on synthetic data.

1. Introduction

FlowFit3 is a new development within the existing FlowFit concept. Common to its predecessors is the representation of the instantaneous flow field using a 3D uniform B-spline grid of weighting coefficients. B-splines can be efficiently evaluated at arbitrary positions within the reconstructed volume and provide easy access to spatial gradients. The FlowFit3 approach thoroughly changes the way these coefficients are determined from the data. In this paper we will focus on the description of the method and will conduct comparisons with prior FlowFit versions. A comprehensive evaluation of the results obtained using FlowFit3 in comparison to other data assimilation techniques on synthetic and real world data can be found in Zhou et al. (2024).

Just as with prior versions of FlowFit the incompressible Navier-Stokes equations are used as underlying physics equations:

$$\vec{a} = \frac{D\vec{u}}{Dt} = -\nabla\bar{p} + \nu\Delta\vec{u} \tag{1}$$

$$\nabla \cdot \vec{u} = 0 \tag{2}$$

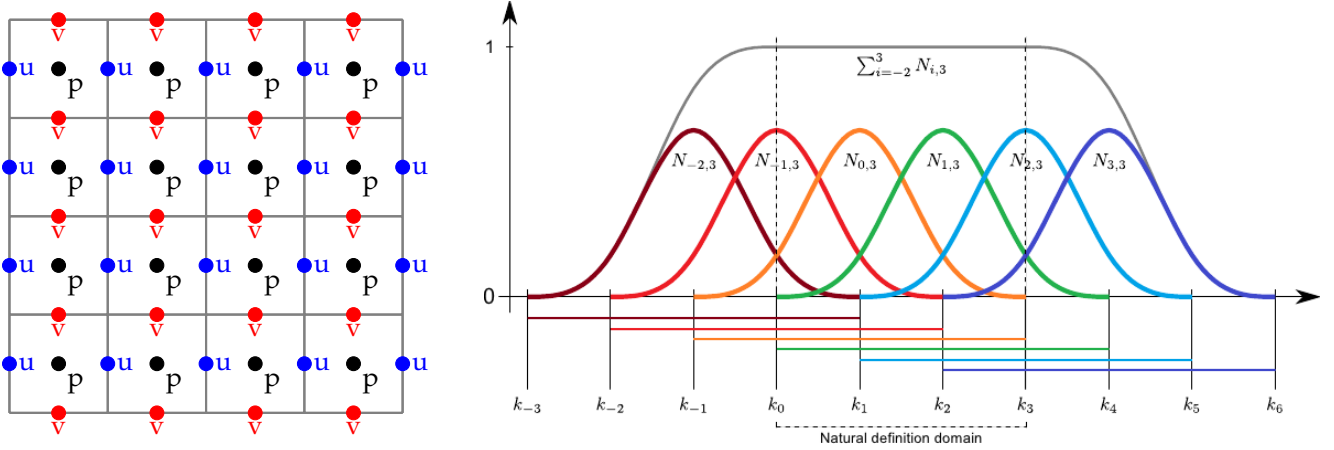


Figure 1. FlowFit3 basis: (left): Example of a 2D staggered grid. (right): Example of 1D B-spline basis functions and interpolation. Lines at the bottom highlight the limited support

with velocity $\vec{u}$, acceleration $\vec{a}$, kinematic pressure $\bar{p}$ and kinematic viscosity ν . With LPT we directly measure the material derivative so there is no need to expand it into its convective form. By rearranging these equations one also obtains an equation for pressure which is used by some variants of FlowFit:

$$\Delta \bar{p} + \nabla \cdot ((\vec{u} \cdot \nabla) \vec{u}) = 0 \quad (3)$$

2. Method

We introduce two variants of FlowFit3, similar to the div1 and div2 modes of FlowFit2. The first, constraining the divergence of velocity (1), will be called "linear" mode in the following as only linear terms are involved in the physics constraint. No measured acceleration data is needed for this mode. The second variant is the "nonlinear" mode using also the momentum equation (2) as an additional constraint. This mode uses measured accelerations to improve the reconstruction and is also able to recover pressure fields using (3). Both modes build on the same internal architecture so we begin with an explanation of the simpler linear mode.

2.1. B-Spline representation

Just as in previous iterations of FlowFit a B-spline representation is used as basis for the data assimilation. This takes the form of a grid of spline basis functions of e.g 3rd order, each with compact support. One can then select scaling weights for each of the basis functions. The basis functions overlap, but due to the compact support only a small amount of basis functions is nonzero for a given point in the domain. In order to evaluate such a B-spline representation at a certain point, one needs to evaluate the value of all local basis functions at this position and scale them by their respective weighting coefficients. The sum now represents the evaluated value at that point.

The basis functions itself are fixed for a given FlowFit grid so all that is needed during reconstruction is to compute appropriate weighting coefficients that best represent a given flow. Each of the components of the velocity vector (u,v,w) is considered a separate field. A finished FlowFit reconstruction corresponds to a grid of B-spline coefficients for each flow quantity which can be evaluated at arbitrary points within the domain. This setup also allows for easy evaluation of spatial gradients since the derivative of the spatial field can be obtained using the derivative of the basis functions which can be computed beforehand. In the following we will use the term *coefficients* to refer to the values of the B-spline coefficients themselves whereas the term *field* will be used for the resulting continuous interpolation (e.g, pressure coefficients vs. pressure field).

At its heart the data assimilation is accomplished by a gradient based optimization with a cost function that incorporates measured data as well as additional constraints. The measured data in the form of particles with positions and velocity values can simply be included by evaluating the current velocity field at the particles locations and examining the velocity difference, which is included as a term in the cost function. In order to move beyond simple interpolation knowledge of the underlying flow physics is included into the optimization. Assuming an incompressible fluid with constant density, the continuity equation transforms to a divergence constraint on the velocity field (1). Previous versions of FlowFit would incorporate such constraints as penalty terms into the cost function. Such a soft constraint has several disadvantages as not only this term has to be properly weighted with respect to the other cost function terms, the constraint is also not exactly enforced.

2.2. Hard constraint on divergence

One key improvement of FlowFit3 over prior versions is to move the divergence constraint into the design of the Ansatz instead. In this way the constraint is satisfied at any point in the domain and the optimizer is able to more efficiently reach an optimal solution as it is restricted from even considering infeasible solutions during intermediate steps. In the following we give a brief overview of how this is accomplished. FlowFit3 now utilizes a staggered grid (Harlow & Welch, 1965) for the velocity coefficients where the grids are shifted by half of the spacing. The shift is performed in the direction of the individual velocity component (e.g. in y-direction for v). A visual example of the resulting grid structure for the case of a 2D grid is given in figure 1. In a 3D grid the w component is shifted similarly in the z -direction.

Such a staggered grid is useful to avoid the problem of even/odd decoupling when solving the pressure Poisson equation which will be a topic when discussing the nonlinear mode. For the moment we will focus on the hard divergence constraint and how the staggered grid can be used for this. Originally such a grid was introduced for the application of finite differences used in the discretization of the spatial derivatives in the incompressible Navier-Stokes equations but the concept applies to the use of uniform B-splines in a comparable way. The derivative of an $(n+1)$ th order uniform B-spline is an n -th order uniform B-spline on a grid shifted by half a spacing unit. The

divergence of the velocity field requires the derivatives $\partial u/\partial x$, $\partial v/\partial y$ and $\partial w/\partial z$. If one constructs the 3D B-spline for u as (4th, 3rd, 3rd) order for (x, y, z) respectively then $\partial u/\partial x$ is a B-spline with (3rd, 3rd, 3rd) order. Moreover, since the derivative grid is shifted by half a spacing unit in the x -direction its location is now back at a grid position co-located with pressure. The same applies for the other staggered velocity components (e.g. (3rd, 4th, 3rd) order for v) giving a (3rd, 3rd, 3rd) order uniform B-spline for the divergence. If we wish to force this divergence to be zero everywhere the coefficients need to satisfy:

$$0 = (u_{x+1/2,y,z} - u_{x-1/2,y,z}) + (v_{x,y+1/2,z} - v_{x,y-1/2,z}) + (w_{x,y,z+1/2} - w_{x,y,z-1/2}). \quad (4)$$

Essentially the divergence of the coefficient grid needs to be zero, which then guarantees by construction that the divergence field is exactly zero in the entire domain. If we can ensure that the divergence of the coefficients is zero we successfully constrain divergence in the entire field. Instead of simply removing divergence from the final result given by the optimizer we more closely incorporate this constraint into the optimization procedure. For the gradient based optimization techniques utilized, the optimizer is provided with a starting solution and the current gradient of the cost function with respect to the state vector for each iteration step. Assuming for the moment the use of simple gradient descent. If we provide a starting solution that is divergence free as initial state and then subtract a scaled gradient for the descent this next state is no longer necessarily divergence free. We can enforce this by requiring the gradient to be divergence free as well. If we perform an orthogonal projection of the gradient into the divergence free subspace the optimizer is never able to leave this subspace as any movement away from the initial solution is at most a linear combination of vectors from within the subspace. This not only ensures that the final result must be divergence free, but also that the optimizer is constrained in every intermediate step of the way thus eliminating these unnecessary degrees of freedom. The optimizer is simply unable to explore physically infeasible areas of the solution space. In the actual implementation we do not use simple gradient descent but the quasi newton method L-BFGS (Liu & Nocedal, 1989). The above approach is still valid since L-BFGS uses a linear combination of previous gradients for its step so it is still unable to leave the divergence free subspace at any time.

The divergence free projection is based on the Helmholtz-Hodge decomposition of a vector field into an solenoidal (divergence free) part $\vec{u}_{\text{sol}}$ and an irrotational part $\vec{u}_{\text{irrot}}$:

$$\vec{u} = \vec{u}_{\text{sol}} + \vec{u}_{\text{irrot}} = \vec{u}_{\text{sol}} + \nabla\phi, \quad (5)$$

with some yet to be determined potential field ϕ . Taking the divergence of (5) gives a Poisson equation for ϕ since $\nabla \cdot \vec{u}_{\text{sol}} = 0$:

$$\nabla \cdot \vec{u} = \Delta\phi. \quad (6)$$

Once ϕ is known the divergence free part of $\vec{u}$ can be determined:

$$\vec{u}_{\text{sol}} = \vec{u} - \nabla\phi. \quad (7)$$

We are solving (6) efficiently under homogeneous Dirichlet boundary conditions using a Discrete Sine Transform. This process then forms an orthogonal projection of a vector field $\vec{u}$ into the divergence free subspace.

2.3. Linear mode

By this approach the only needed term in the cost function is the velocity discrepancy between field and particles, however a regularization term that penalizes high wavenumber oscillations of the coefficients is still added just as in prior FlowFit versions. The cost function for the linear mode FlowFit3 thus becomes:

$$\text{cost}_{\text{lin}}(\vec{s}) = \frac{1}{2} (\|\text{vel}_{\text{err}}(\vec{s})\|^2 + \alpha \|\text{hfpen}(\vec{s})\|^2) \quad (8)$$

with $\vec{s}$ as the optimizer state in form of the velocity coefficients, vel_{err} as the velocity discrepancy of the particles to the field, hfpen as the high wavenumber penalization with a weighting term α . This penalization is calculated just as in FlowFit2 as the L2 norm of the highpass-filtered version of the coefficient field (Gesemann et al., 2016).

This variant of FlowFit3 provides a very efficient way to perform data assimilation but does not yet use acceleration information in order to further improve assimilation quality or provide pressure reconstruction as well. This is realized in a second mode we call nonlinear mode FlowFit3.

2.4. Nonlinear mode

If acceleration measurements are available we can use them to further improve reconstruction quality as well as to recover pressure fields. The additional data allows for the use of the momentum equation from the incompressible Navier-Stokes equations in addition to the continuity equation used so far. Accelerations obtained from LPT measurements provide the material derivative of velocity, the left side of the momentum equation:

$$\vec{a} = \frac{D\vec{u}}{Dt} = \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + \nu \Delta \vec{u} \quad (9)$$

Just as the continuity equation requires zero velocity divergence, the divergence of the temporal derivative of velocity must vanish as well.

$$\nabla \cdot \frac{\partial \vec{u}}{\partial t} = \nabla \cdot (\vec{a} - \vec{u} \cdot \nabla \vec{u}) = 0 \quad (10)$$

After some transformations this leads to the Poisson equation for pressure:

$$\Delta p + \nabla \cdot (\vec{u} \cdot \nabla \vec{u}) = 0 \quad (11)$$

The nonlinear source term can be further simplified as:

$$\nabla \cdot (\vec{u} \cdot \nabla \vec{u}) = \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 + 2 \left(\frac{\partial u}{\partial y} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial z} \frac{\partial w}{\partial x} + \frac{\partial v}{\partial z} \frac{\partial w}{\partial y} \right) \quad (12)$$

and is the origin for the naming of this mode as nonlinear. Once the pressure field has been determined one can calculate the acceleration field as needed using (9) in order to be able to

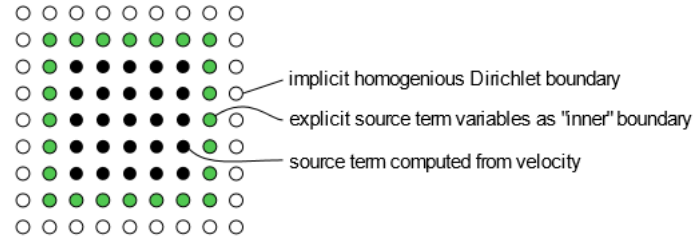


Figure 2. Approach for allowing arbitrary boundary conditions even when solving the Poisson equation using an efficient method implying homogeneous Dirichlet boundaries. The explicit source term variables are part of the optimizer state and are thus adapted as needed.

calculate an acceleration error at the particle positions. Whereas prior FlowFit versions would add an additional set of pressure coefficients to be optimized alongside the velocity the new FlowFit3 approach does not include them in the state vector anymore. Pressure is calculated from velocities by use of the pressure Poisson equation as needed which is solved efficiently using a FFT based approach. We utilize a Discrete Sine Transform (DST) which implies homogeneous Dirichlet boundary conditions. While this allows for a very efficient computation of the solution for the Poisson equation it is not necessarily the correct boundary for a given reconstruction. To allow for arbitrary boundary conditions while still using the DST we add an inner boundary layer to the source term as shown in figure 2. Due to the support region of the B-Splines the nonlinear term can only be computed from velocity in the interior region. By surrounding this region with the additional boundary layer we retain the correct size in order to reuse the DST used for the divergence free projection, now just with a different transfer function in Fourier space. By selecting appropriate values for this layer of source terms we can enforce arbitrary boundary conditions for the original problem. These explicit source term variables are simply included into the optimizer state in order to obtain suitable values for a given reconstruction problem.

With this Ansatz pressure is no longer part of the state of the optimizer (except for boundary conditions) but this approach introduces greater complexity in the implementation of the cost function and especially its gradient. Instead of just backpropagating cost function gradients to the pressure coefficients as in prior FlowFit versions they now need to be calculated further through the pressure reconstruction all the way to the velocity coefficients instead. But again, just as with the divergence constraint, the degrees of freedom for the optimizer are reduced since this way the dependent relationship between velocity and pressure is encoded explicitly. The cost function for the nonlinear mode is very similar to the linear mode (8):

$$\text{cost}_{\text{nonlin}}(\vec{s}) = \frac{1}{2} (\|\text{vel}_{\text{err}}(\vec{s})\|^2 + \alpha \|\text{hfpn}(\vec{s})\|^2 + \beta \|\text{acc}_{\text{err}}(\vec{s})\|^2) \quad (13)$$

now including an additional acceleration discrepancy term acc_{err} with a weighting factor between velocity and acceleration errors β . This factor can be determined from the data using the Trackfit (Gesemann et al., 2016) approach.

One could obtain the required cost function gradient with respect to the state required for the

optimizer using automatic differentiation on an implementation of (13). Our implementation uses a handwritten backpropagation approach for efficiency reasons. The initial two terms also present in the cost function for the linear mode in (8) are fairly straightforward to handle since they either directly or almost directly operate on the velocity coefficients. The acceleration term is more complex so we will discuss its gradient in more detail. The cost term for the acceleration is:

$$\frac{1}{2} \sum_i^N \left\| a_{\text{fitted}}|_{\text{pos}=\text{particlepos}[i]} - a_{\text{particle}[i]} \right\|^2, \quad (14)$$

a sum of acceleration errors over all particles. A gradient of sums is the sum of its gradients so in the following we look at the cost contribution of just a single particle within the sum with its measured acceleration $\vec{a}_p$ and the fitted acceleration $\vec{a}_f$ at its position:

$$C_a = \frac{1}{2} \|\vec{a}_f - \vec{a}_p\|^2 = \frac{1}{2} \|\text{err}\|^2. \quad (15)$$

We can only affect this by modifying the fitted acceleration:

$$\frac{\partial C_a}{\partial \vec{a}_f} = \text{err} \left(\frac{\partial \vec{a}_f}{\partial \vec{a}_f} - \frac{\partial \vec{a}_p}{\partial \vec{a}_f} \right) = \text{err} \quad (16)$$

Which can be related to the velocity coefficient field $\vec{u}_k$ by chaining the Jacobians of the intermediate operations:

$$\frac{\partial C_a}{\partial \vec{u}_k} = \frac{\partial C_a}{\partial \vec{a}_f} \frac{\partial \vec{a}_f}{\partial \Delta \vec{u}} \frac{\partial \Delta \vec{u}}{\partial \vec{u}_k} \quad \text{with} \quad \frac{\partial \vec{a}_f}{\partial \Delta \vec{u}} = \nu - \frac{\partial \nabla p}{\partial \Delta \vec{u}} \quad \text{since} \quad \vec{a}_f = \nu \Delta \vec{u} - \nabla p \quad (17)$$

The term $\frac{\partial \Delta \vec{u}}{\partial \vec{u}_k}$ is known from the B-spline construction but $\frac{\partial \nabla p}{\partial \Delta \vec{u}}$ must still be determined. Using the nonlinear term (12) as N and the Poisson solver including the mapping to the B-Spline coefficients as P we obtain:

$$\Delta p = N(\nabla \vec{u}) \quad \text{and} \quad p_k = P(\Delta p). \quad (18)$$

We can then express the missing term in terms of known operations after some chain rule wrangling of the derivatives:

$$\frac{\partial \nabla p}{\partial \Delta \vec{u}} = \frac{\partial \nabla p}{\partial p_k} \frac{\partial P}{\partial \Delta p} \frac{\partial N}{\partial \nabla \vec{u}} \frac{\partial \nabla \vec{u}}{\partial \Delta \vec{u}} \quad (19)$$

Substituting into (17):

$$\frac{\partial C_a}{\partial \vec{u}_k} = \frac{\partial C_a}{\partial \vec{a}_f} \left(\nu - \frac{\partial \nabla p}{\partial p_k} \frac{\partial P}{\partial \Delta p} \frac{\partial N}{\partial \nabla \vec{u}} \frac{\partial \nabla \vec{u}}{\partial \Delta \vec{u}} \right) \frac{\partial \Delta \vec{u}}{\partial \vec{u}_k}, \quad (20)$$

followed by simplification of some $\vec{u}, \vec{u}_k$ terms and substituting (16) one finally obtains:

$$\frac{\partial C_a}{\partial \vec{u}_k} = \text{err} \frac{\partial \Delta \vec{u}}{\partial \vec{u}_k} \nu - \text{err} \frac{\partial \nabla p}{\partial p_k} \frac{\partial P}{\partial \Delta p} \frac{\partial N}{\partial \nabla \vec{u}} \frac{\partial \nabla \vec{u}}{\partial \vec{u}_k}, \quad (21)$$

the desired gradient of the acceleration cost function with respect to the optimizer state in form of the velocity coefficients. One essentially multiplies this from left to right as vector Jacobi products (vJp) starting from the calculated acceleration error value at the particle location $e\vec{r}$. The Jacobi matrices in (21) are not supposed to be actually created as one can implement the result of the vJp directly. These adjoints generally can be obtained for a similar computational cost as the respective operations in the forward pass. This is especially the case for the Poisson solver step P as it is self-adjoint. At that stage of the backpropagation one would also extract the gradient for the source term used for the pressure boundary. As the value of the source term is jointly optimized together with the velocity coefficients one needs to scale its gradient to a similar norm as the gradient for the velocity coefficients.

3. Evaluation

To compare the new FlowFit3 variants to the prior versions we use synthetic data based on a subvolume of the isotropic turbulence data set from the Johns Hopkins Turbulence Database (Yeung et al., 2012) just as in the original FlowFit2 publication (Gesemann et al., 2016). Synthetic particle data is generated simply by sampling the database at random positions in the domain. Four different particle densities are investigated resulting in a mean inter particle distance of approx. 7.5 to 3.75 Kolmogorov length scales. We compare the performance of FlowFit3 with results obtained using the FlowFit2 approach on the same data. Figure 3 shows a comparison of reconstruction quality in a slice of the vorticity field for a particle sampling at 6 Kolmogorov length scales. The FlowFit2 div1 version, using only velocity information, yields the lowest quality of reconstruction. Interestingly, the linear version of FlowFit3 shows visibly better results, despite depending on the same restricted data. The div2 variant of FlowFit2, additionally utilizing acceleration data, markedly improves on its div1 counterpart, producing results quite alike to FlowFit3 linear, albeit with a slightly better rendition of the gradient. The FlowFit3 nonlinear mode yields a further increase of perceived sharpness.

This can also be seen in a visualization of isosurfaces of the Q-criterion of the same reconstructions in figure 4. Again the nonlinear mode FlowFit3 provides a slight improvement over FlowFit2 div2 whereas a significant gain can be observed for the linear mode versus the div1 result.

Reducing the particle density to a mean spacing of approx. 7.5 length scales highlights the benefit of including acceleration data using the momentum equation as shown in figure 5. Both FlowFit variants using accelerations are much less affected by the decrease of particle density than the variants using only velocity. The FlowFit2 div1 result has degraded significantly. Linear mode FlowFit3 now shows less detail than the nonlinear variant but still presents a relatively smooth solution without the noise visible for div1 FlowFit2.

A quantitative comparison of rms-errors of the Q-value compared to the ground truth is presented in figure 6 for a range of particle densities. The results match the visual interpretations of the prior figures with FlowFit3 nonlinear as an improvement over FlowFit2 div1. As the particle distance

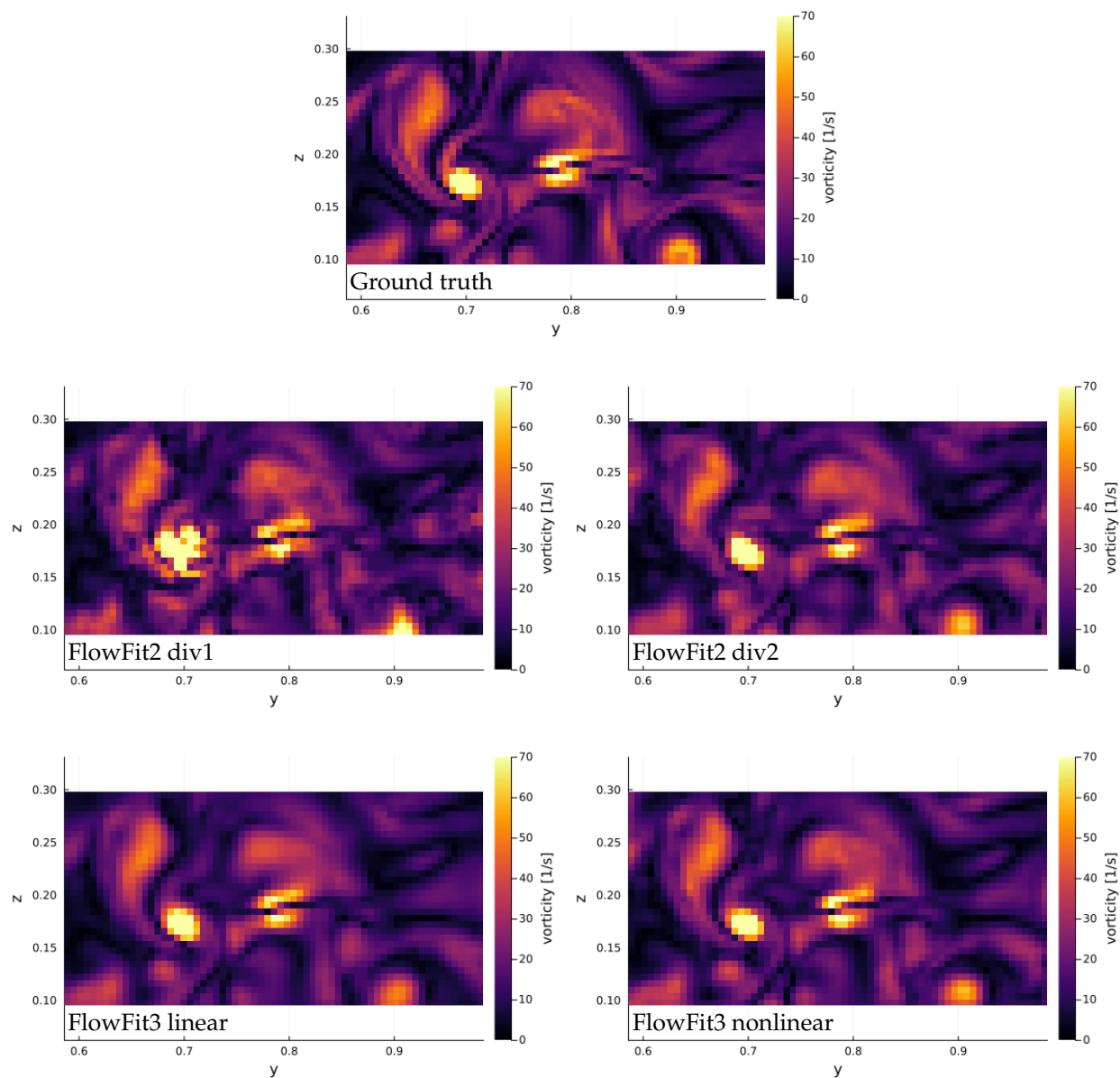


Figure 3. Slices of vorticity field reconstructed using the different methods for a particle density with a mean spacing of approx. 6 Kolmogorov length scales

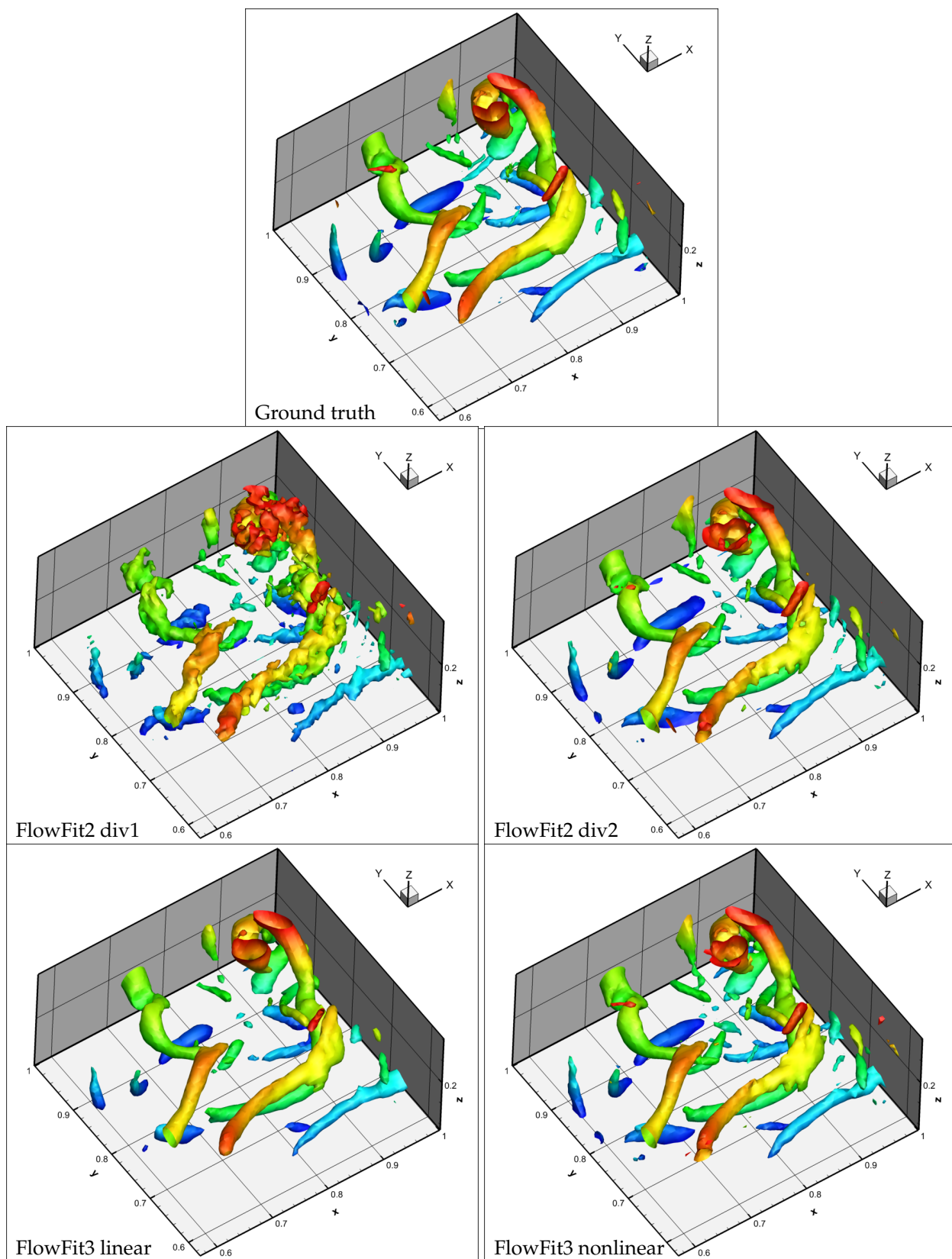


Figure 4. Isosurfaces of Q -criterion at 400 1/s^2 reconstructed using the different methods for a particle density with a mean spacing of approx. 6 Kolmogorov length scales

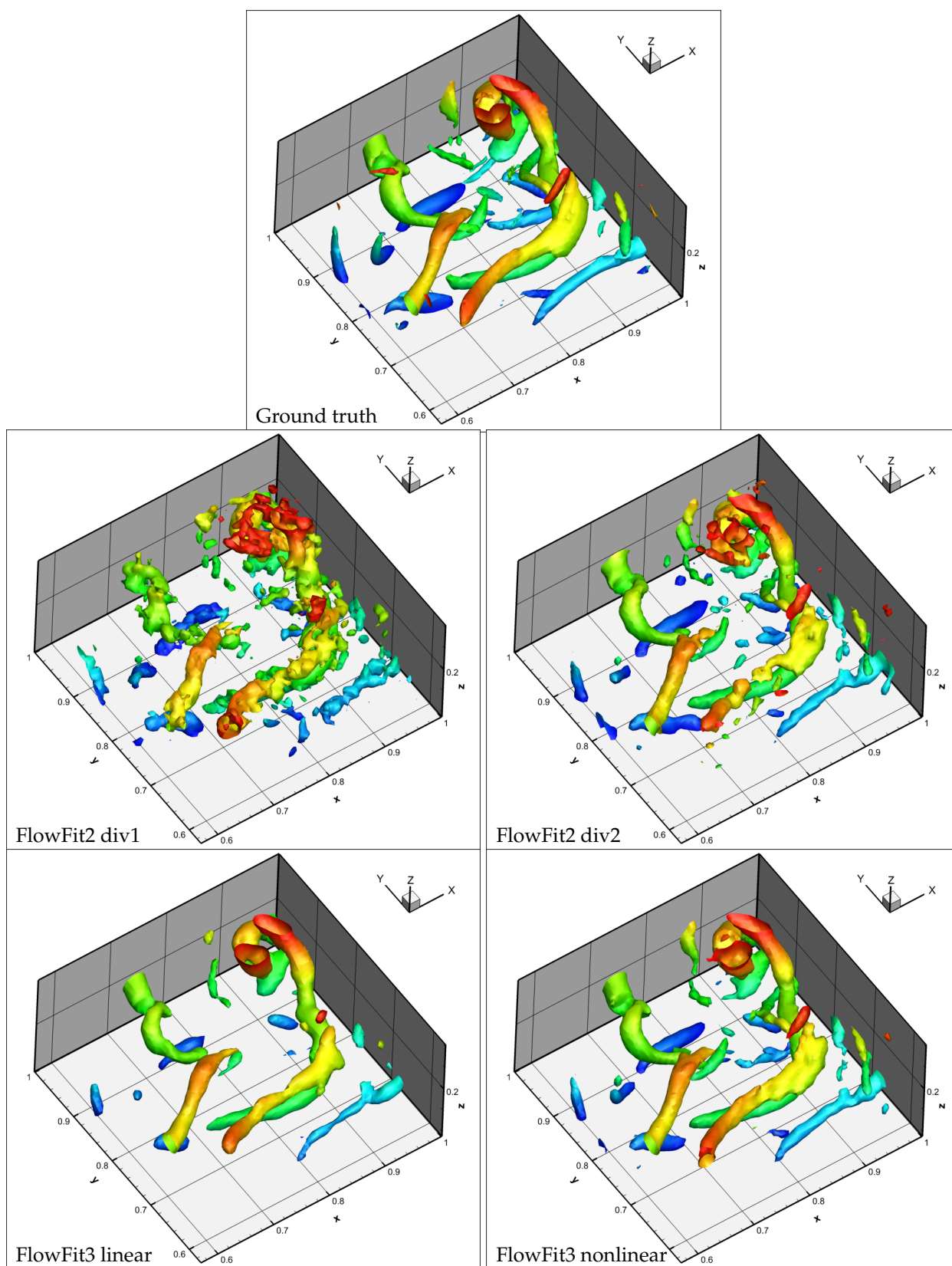


Figure 5. Isosurfaces of Q -criterion at 400 1/s^2 reconstructed using the different methods for a particle density with a mean spacing of approx. 7.5 Kolmogorov length scales

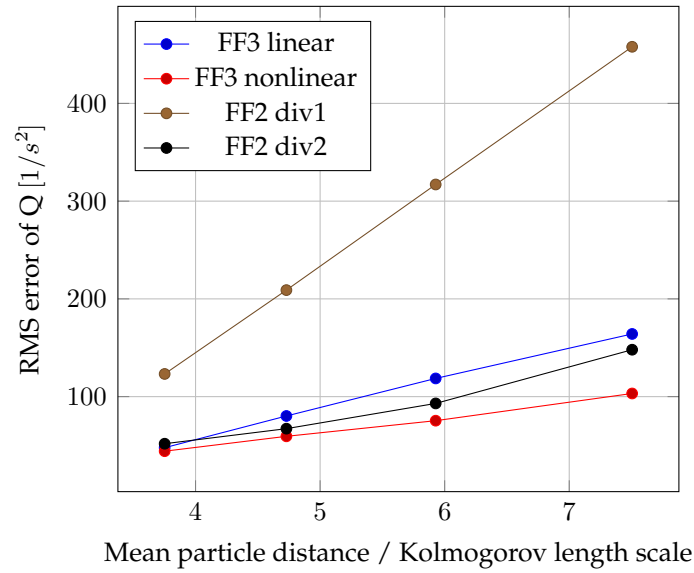


Figure 6. RMS error of the reconstructed Q field for the different methods at various particle densities

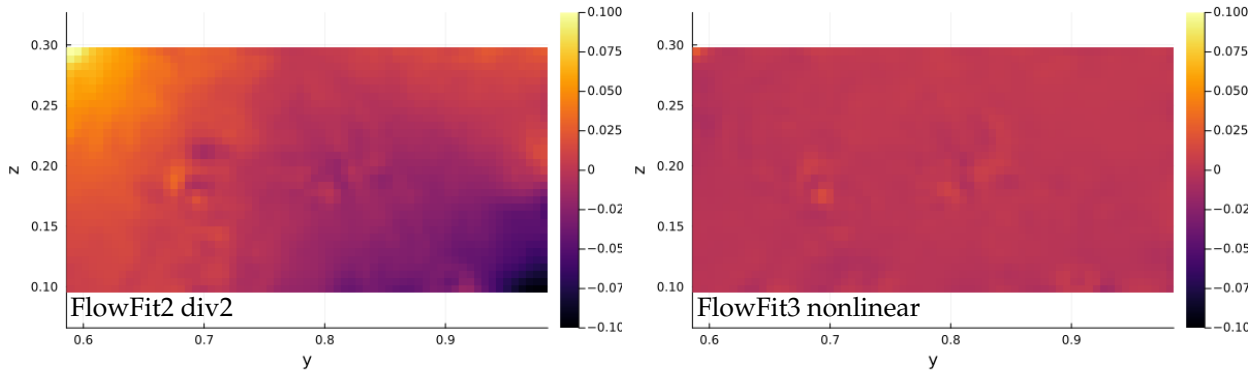


Figure 7. Pressure reconstruction error for the two methods for a particle density with a mean spacing of approx. 6 Kolmogorov length scales

increases the difference between the variants with and without use of the momentum equation increases. Remarkable is the comparatively low error of the FlowFit3 linear mode which almost approaches the performance of the prior FlowFit2 div2 version.

A brief evaluation of the reconstructed pressures shows the handling of arbitrary boundary conditions in FlowFit3 nonlinear mode is successful. Figure 7 shows the error of reconstructed pressure in a slice through the volume. The FlowFit2 div2 result shows a slight gradient in the error field which is not present in the FlowFit3 nonlinear case.

The architectural changes to the FlowFit internals in the handling of the divergence constraint appear to be highly successful in guiding the optimizer towards a physically correct solution. Such performance makes the linear mode very attractive for the processing of a large amount of snapshots as it is very efficient. On a reasonably modern CPU, a typical measurement domain can usually be reconstructed in less than 10 seconds, while the nonlinear mode requires around one order of

magnitude more processing resources. However, compared to other data assimilation methods, even this mode is highly competitive in terms of computational cost, as reported in Zhou et al. (2024). Some aspects of the nonlinear mode are still under development so we still expect some improvements to the above results.

4. Conclusion and future work

Based on the initial testing presented above the new FlowFit3 approach is a successful development that outperforms its predecessor in data quality and reconstruction time. The linear mode provides very fast reconstruction times which will be of interest for large data sets. The nonlinear mode results in even higher fidelity reconstructions and also allows for pressure reconstructions. For a more detailed evaluation of the FlowFit3 variants see Zhou et al. (2024).

For the current implementation of FlowFit3 only a limited number of parameters needs to be selected. A key value that needs to be adjusted is the penalization of higher wave numbers α . This parameter directly controls the smoothness of the solution and as such can be easily adjusted based on visual examination. A current topic for further development is to better determine this regularization based on physical properties. Different regions of a flow field likely require different strengths of this regularization which could be determined from Flow structure sizes obtained from two-point correlations. Currently the high-pass filter used for the penalty operates directly on the coefficient field. In principle this would still allow for spatially varying regularization since the coefficients directly correspond to regions in physical space. However it would be beneficial to instead define the penalization in world space in order to more precisely allow for spatially varying anisotropic curvature penalization.

Currently FlowFit conducts an independent reconstruction of individual time steps in an sequential measurement. This can be an advantage as no time resolved data is required but leaves some room for further improvement. The use of virtual particles to enhance temporal consistency has already shown some benefit (Ehlers et al., 2020) but future work will also consider more explicitly linking FlowFit reconstructions across multiple timesteps for a joint optimization.

References

- Ehlers, F., Schröder, A., & Gesemann, S. (2020, June). Enforcing temporal consistency in physically constrained flow field reconstruction with FlowFit by use of virtual tracer particles. *Measurement Science and Technology*, 31(9), 094013.
- Gesemann, S., Huhn, F., Schanz, D., & Schröder, A. (2016). From noisy particle tracks to velocity and acceleration and pressure fields and using b-splines and penalties. In *18th international symposium on the application of laser and imaging techniques to fluid mechanics* (pp. 2684–2700).

- Harlow, F. H., & Welch, J. E. (1965, 12). Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *The Physics of Fluids*, 8(12), 2182-2189.
- Jeon, Y. J., Müller, M., & Michaelis, D. (2022, April). Fine scale reconstruction (vic#) by implementing additional constraints and coarse-grid approximation into VIC+. *Experiments in Fluids*, 63(4).
- Liu, D. C., & Nocedal, J. (1989, August). On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1-3), 503-528.
- Schanz, D., Gesemann, S., & Schröder, A. (2016, Apr 27). Shake-The-Box: Lagrangian particle tracking at high particle image densities. *Experiments in Fluids*, 57(5), 70.
- Schneiders, J. F. G., & Scarano, F. (2016, August). Dense velocity reconstruction from tomographic ptv with material derivatives. *Experiments in Fluids*, 57(9).
- Schröder, A., & Schanz, D. (2023). 3D Lagrangian Particle Tracking in Fluid Mechanics. *Annual Review of Fluid Mechanics*, 55(1), 511-540.
- Yeung, P. K., Donzis, D. A., & Sreenivasan, K. R. (2012, February). Dissipation, enstrophy and pressure statistics in turbulence simulations at high reynolds numbers. *Journal of Fluid Mechanics*, 700, 5-15.
- Zhou, K., Grauer, S., Schanz, D., Godbersen, P., Schröder, A., Rockstroh, T., ... Wieneke, B. (2024). Benchmarking Data Assimilation Algorithms for 3D Lagrangian Particle Tracking. In *21st International Symposium on the Application of Laser and Imaging Techniques to Fluid Mechanics, Lisbon, Portugal*.