Trajectory integrated smoothening of exchange fields for discrete phase simulations

Citation for published version (APA):

Simonsen, A. S., Deen, N. G., Sørensen, K., Condra, T., & Rosendahl, L. (2019). Trajectory integrated smoothening of exchange fields for discrete phase simulations. *Computers & Fluids*, 186, 15-23. <https://doi.org/10.1016/j.compfluid.2019.03.015>

DOI:

[10.1016/j.compfluid.2019.03.015](https://doi.org/10.1016/j.compfluid.2019.03.015)

Document status and date:

Published: 30/05/2019

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Trajectory Integrated Smoothing of Exchange Fields for Discrete Phase Simulations

A. S. Simonsen, N. G. Deen, K. Sørensen, T. Condra, and L. Rosendahl

Abstract—Eulerian-Lagrangian simulations often require a two-way coupling, where the discrete phase affects the continuous phase and vice versa. The continuous phase in a CFD-simulation is typically represented on a computational mesh, which consists of a number of discrete cells. As the discrete phase is not directly associated with this mesh, a mapping from the discrete phase to the continuous phase is required in order to couple the two phases. This will distribute the exchange rates from the discrete to the continuous phase, where the exchanged properties can be thermal energy, momentum, species etc.

The mapping can be done in several ways, and different methods exist. Some of these include a smoothing step, which ensures that the exchange fields do not contain large spikes between any two neighbouring cells, which typically increases the convergence rate of the CFD simulation. Gaussian smoothing is often applied, where the discrete phase exchange rates are spread to the surrounding cells by a normal distribution with a specified standard deviation. This paper proposes a new weight for distributing the discrete phase exchange rates to the continuous phase cells. This is done by analytically integrating any distribution over the entirety of the particle trajectory, and only requires the probability- and cumulative density functions to be specified.

The proposed weight is used in combination with three different smoothing methods: node-based, kernel-based, and cell-based. Common existing weight distributions are also investigated, which are used in combination with the different methods. Each method and weight combination is compared with the analytical ground-truth exchange field, where the distributions are accurately integrated over the cell volumes.

The results show that the proposed weight has an average error of $\approx 2\%$ in the exchange field for the test case used. This is about three times more accurate compared to existing weights, where the exchange field is not integrated along the trajectory of the particle. Alternative distributions were investigated to decrease the computational requirements. It was found that a fourth-order polynomial distribution was about 20 times faster to evaluate, whilst the error induced by switching to this distribution was $\approx 4.5\%$ compared with the previous $\approx 2\%$.

Index Terms—Gaussian, Mapping, Smoothing, Weighting

I. INTRODUCTION

SMOOTHENING methods are commonly used for discrete phase simulations. These methods are used to distribute the rate of exchanged properties, ϕ , between the discrete and the continuous phase, where ϕ can be thermal energy, momentum, species etc. The exchange fields are used afterwards by the CFD solver as source terms in each cell. The advantage of smoothing out the exchange fields is to increase convergence rate, and to reduce the required number of discrete particles in the simulation. The reduction in the number of required particles is due to the fact that a single particle can cover a larger spatial volume when smoothing methods are used properly. Furthermore, the advantage is that the required

computational time to achieve a converged solution is reduced. The disadvantage is that the smoothing methods can cause the exchange rates to take place in unphysical locations, where the particles should not have any effect. This phenomenon was investigated by [1], where three different methods were compared, namely *diffusive smoothing*, *particle cloud model* and *darning socks*. Only the latter of these three methods did not smear the exchange fields to unphysical locations.

When the discrete phase represents a liquid, such as fine droplets, the required number of particles is usually very large and in the order of $10^6 - 10^9$. In order to reduce the computational requirements, the concept of parcels is commonly used. A parcel is a collection of droplets, which share properties such as diameter, velocity, temperature etc. [2]. When combining the droplets into grouped parcels, the position of each droplet within the parcel can be expressed using statistical methods. Thus, the position of each droplet is specified relative to the position of the parcel.

The spread of droplets can be modelled by assuming that the particles experience Brownian motion, where the velocity fluctuations are calculated based on the local turbulence kinetic energy [2]. Brownian motion can be expressed using the normal distribution, where the spread is defined using the standard deviation, σ [3] [4].

Based on the assumption that all droplets experience Brownian motion, and that parcel tracking is being used, the position of the droplets should follow the normal distribution relative to the position of the parcels. Thus, when the droplets are exchanging properties with the continuous phase, the exchange rates should be distributed using the normal distribution as well. This will act indirectly as a Gaussian smoothing weight.

A method employing a similar approach is the *Particle Cloud Tracking* algorithm. This method, which was investigated by [5] and [6], allows each parcel to grow in size as a function of time. This growth is governed by the turbulent fluctuations in the flow field, where statistical methods are used to integrate the spread. When the parcels span a large spatial volume, the exchange fields can be calculated by using the normal distribution. This will effectively smooth out the exchange fields. However, the computational requirements for analytically integrating the exchange fields on an arbitrary mesh cell can be large, so faster methods are desirable.

This paper assumes that the exchanged properties are being distributed following a normal distribution, which is defined in Eq. (1).

Fig. 1. The normal, truncated normal, and adjusted normal distributions shown in blue, red and black respectively. The upper and lower figures show the probability and cumulative density functions respectively.

$$\begin{aligned} y_N(x, \sigma) &= \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \\ Y_N(x, \sigma) &= \frac{1}{2} \cdot \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right)\right) \end{aligned} \quad (1)$$

Where $y_N(x, \sigma)$ is the probability density function, and $Y_N(x, \sigma)$ is the cumulative density function. The normal distribution spans from $-\infty < x < \infty$, but a particle cannot interact with the continuous phase at these distances, and as a result of this an alternative distribution is used, which spans a limited distance defined by $-\chi < x < \chi$. This cut-off distance, χ , is defined as $2 \cdot \sigma$ in this paper, which results in $\approx 95\%$ of the area of the normal distribution. The truncated normal distribution is spatially limited, which is shown in Eq. (2).

$$\begin{aligned} Z(\chi, \sigma) &= Y_N(\chi, \sigma) - Y_N(-\chi, \sigma) \\ y_{TN}(x, \chi, \sigma) &= \frac{y_N(x, \sigma)}{Z(\chi, \sigma)} \\ Y_{TN}(x, \chi, \sigma) &= \frac{Y_N(x, \sigma) - Y_N(-\chi, \sigma)}{Z(\chi, \sigma)} \end{aligned} \quad (2)$$

$$-\chi \leq x \leq \chi$$

One problem with the truncated normal distribution is that the value of y_{TN} at $x = \chi$ is non-zero, which is undesirable. Thus, the truncated normal distribution is adjusted by subtracting $y_{TN}(\chi, \chi, \sigma)$, and afterwards renormalizing such that the integral between $-\chi < x < \chi$ is equal to unity. The cumulative density function can be derived afterwards based on the adjusted probability density function. This is shown in Eq. (3).

Fig. 2. The exchange field using the adjusted normal distribution for a given particle trajectory. The dashed grey lines indicate contour lines of the underlying exchange field, which is coloured from white to black with increasing intensity. The exchange rate, ϕ , is constant along the trajectory in this example. The solid blue line is offset from the particle trajectory by χ .

$$\begin{aligned} f(x, \chi, \sigma) &= \sqrt{2}x - \sqrt{\pi}\sigma \cdot \exp\left(\frac{1}{2}\left(\frac{\chi}{\sigma}\right)^2\right) \cdot \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) \\ y_{AN}(x, \chi, \sigma) &= \frac{1 - \exp\left(\frac{\chi^2 - x^2}{2\sigma^2}\right)}{f(\chi, \chi, \sigma) \cdot \sqrt{2}} \\ Y_{AN}(x, \chi, \sigma) &= \frac{1}{2} \left(1 + \frac{f(x, \chi, \sigma)}{f(\chi, \chi, \sigma)}\right) \end{aligned} \quad (3)$$

The adjusted normal distribution is limited from $-\chi < x < \chi$ and the values of y_{AN} at $x = -\chi$ and $x = \chi$ are both zero. The normal, the truncated normal, and the adjusted normal distributions are shown in Fig. 1. The adjusted normal distribution is illustrated along a random trajectory in Fig. 2.

II. GROUND-TRUTH EVALUATION

The ground-truth is defined as the exchange field, where the adjusted normal distribution is analytically integrated over the volume of each cell along the entire particle trajectory. This is shown in Eq. (4), where the subscript iC indicates the i^{th} cell, and $\vec{x}_{iC}$ is the centroid of the cell. w'_{iC} is the volume integrated weight, whereas w_{iC} is the normalized weight, which ensures that the exchange field is conserved.

$$\begin{aligned} w'_{iC}(t) &= \int_V y_{AN}(|\vec{x}_P(t) - \vec{x}_{iC}|, 2 \cdot \sigma, \sigma) dV \\ w_{iC}(t) &= \frac{w'_{iC}(t)}{\sum_j w'_{jC}(t)} \\ \phi_{iC} &= \int_{t_0}^{t_1} w_{iC}(t) \cdot \phi_P(t) dt \end{aligned} \quad (4)$$

The integral shown in Eq. (4) is evaluated for all cells in Fig. 2. Each cell was subdivided a number of times until no

mesh nodes, the cells are used instead. After accumulating the exchange rates, a smoothening step is applied, which redistributes the exchange rates to the cells located within a distance of χ . This method is illustrated in Fig. 6.

Fig. 6. The first step of the kernel-based method is to accumulate the particle exchange rates on the current cell, in which the particle is located. The second step is to redistribute the accumulated exchange rates to the cells located within a distance of χ .

The kernel-based method is also divided into two steps. The first step is to accumulate the raw exchange rates on the mesh cells. No weights are required to do the first step, which is shown in Eq. (7).

$$\phi_{iC} = \phi_{iP} \quad (7)$$

The second step is to redistribute the exchange field. This is done by first identifying all cells located within a distance of χ relative to the current cell centre. When the cells are identified, weights are calculated based on the weighting function and the cell volumes. This is shown in Eq. (8).

$$\begin{aligned} w'_{jC} &= W(|\vec{x}_{iC} - \vec{x}_{jC}|) \cdot V_{jC} \\ w_{jC} &= \frac{w'_{jC}}{\sum_i w'_i} \\ \phi_{jC} &= w_{jC} \cdot \phi_{iC} \end{aligned} \quad (8)$$

Notice that the cell, in which the particle is located, is included in the subset of affected cells. In this cell, the distance input to the weight function, $W()$, is $|\vec{x}_{iC} - \vec{x}_{iC}| = 0$, which is likely to cause a *division by zero* error, which should be taken care of in the implementation (See section IV. Weights).

C. Cell-based method

The cell-based method only has a single step, as the exchange rate is directly distributed from the particle to the cells in a single step. This method is illustrated in Fig. 7.

The cell-based method starts by identifying all cells located within a distance of χ relative to the position of the particle. When the cells are identified, weights are calculated from the particle to each of the cells based on the distance and the volumes of the cells. When the weights are calculated, the

Fig. 7. The cell-based method only has a single step. All cells located within a distance of χ relative to the particle/parcel position are identified, and the exchange rate is distributed according to the weights to each of the cells.

particle exchange rate is distributed according to the weights. This process is shown in Eq. (9).

$$\begin{aligned} w'_{iC} &= W(|\vec{x}_{iC} - \vec{x}_P|) \cdot V_{iC} \\ w_{iC} &= \frac{w'_{iC}}{\sum_i w'_i} \\ \phi_{iC} &= w_{iC} \cdot \phi_P \end{aligned} \quad (9)$$

IV. WEIGHTS

The three different smoothening methods defined in the previous section all use weights to distribute the exchange rates. Different weights exist, where some of these are defined in this section [7]. All the weights are included and a comparison is made in section VI. Comparison.

1) *Even*: The even-weight simply weights all elements equally as defined in Eq. (10).

$$W_i = 1 \quad (10)$$

2) *Shortest-distance*: This weight uses the shortest-distance definition as shown in Eq. (11).

$$W_i = 1 - \frac{\Delta_i}{\sum_j \Delta_j} \quad (11)$$

3) *Inverse-distance*: This weight uses the inverse distance, as defined in Eq. (12). In order to avoid the singularity when $\Delta_i = 0$, which would cause a *division by zero* error, the denominator includes a $\max()$ function, where ϵ is the machine precision.

$$W_i = \frac{1}{\max(\Delta_i, \epsilon)} \quad (12)$$

4) *Probability density function*: This weight uses any probability density function to weight the exchange field. This can, for instance, be the normal distribution as defined in Eq. (1). In section VII different functions for $y(\Delta_i)$ will be compared.

$$W_i = y(\Delta_i) \quad (13)$$

Fig. 8. An overview of the proposed smoothing weight. The accumulation point is located at $\vec{x}_M$ and the particle/parcel moves from $\vec{x}_P^k$ and $\vec{x}_P^{k+1}$. The integrated probability density function is highlighted in the upper figure, which is evaluated using the notation in the lower image corresponding to Eq. (17).

5) *Continuous integrated*: The previous weights all use the location of the particle at a single instance in time, $\Delta_i = |(\vec{x}_P^k + \vec{x}_P^{k+1})/2 - \vec{x}_i|$, where the superscripts indicate the k^{th} and $k^{\text{th}}+1$ time-step, whereas the previous weights have been using the midpoints. A new weight is proposed in this paper, which uses the trajectory spanning between $\vec{x}_P^k$ and $\vec{x}_P^{k+1}$, and weights the exchanged property, ϕ , by evaluating the integral of the probability density function, $y(x)$, between these points. This is useful for large time-step values or when the flow properties have large spatial gradients. This is stated in Eq. (14), and shown in Fig. 8.

$$W_i = \int_{t^k}^{t^{k+1}} y(|\vec{x}_{iM} - \vec{x}_P(t)|, \sigma) dt \quad (14)$$

The accumulation point, which is either a node or cell centre based on the smoothing method, is defined as $\vec{x}_{iM}$. The particle moves from $\vec{x}_P^k$ to $\vec{x}_P^{k+1}$ as indicated by the arrow in Fig. 8. The distribution peaks, when the particle is closest to $\vec{x}_M$, where the distance between $\vec{x}_M$ and $\vec{x}_P(t)$ is δ . The probability density function at the peak point evaluates to $y(\delta)$, which is shown in Fig. 8.

The highlighted integral can be calculated using the cumulative density function, $Y(x)$. However, this function needs to

be scaled, as it is not going through $\vec{x}_M$. The scaling can be calculated as $y(\delta)/y(0)$, and, as such, the probability density function evaluated along the trajectory spanning between $\vec{x}_P^k$ and $\vec{x}_P^{k+1}$ can be defined as shown in Eq. (15).

$$y_{\text{Sub}}(x) = y(x) \cdot \frac{y(\delta)}{y(0)} \quad (15)$$

The highlighted area in Fig. 8 between $\vec{x}_P^k$ and $\vec{x}_P^{k+1}$ can be calculated using the cumulative density function.

$$W_i = \int_{x_0}^{x_1} y_{\text{Sub}}(x) dx = (Y(x_1) - Y(x_0)) \cdot \frac{y(\delta)}{y(0)} \quad (16)$$

The values required for evaluating Eq. (16), δ , x_0 , and x_1 , can be calculated using Eq. (17), where these values are shown in the lower part of Fig. 8.

$$\begin{aligned} \vec{a} &= \vec{x}_M - \vec{x}_P^k & \delta &= |\vec{a} - \vec{c}| \\ \vec{b} &= \vec{x}_P^{k+1} - \vec{x}_P^k & x_0 &= -\frac{\vec{c} \cdot \vec{b}}{|\vec{b}|} \\ \vec{c} &= \frac{(\vec{a} \cdot \vec{b}) \vec{b}}{|\vec{b}|^2} & x_1 &= x_0 + |\vec{b}| \end{aligned} \quad (17)$$

The weight is generalized, so any function, where the probability and the cumulative density functions, $y(x)$ and $Y(x)$, are known, can be used. An example is given in Eq. (18), where the normal distribution in Eq. (1) is substituted into Eq. (16).

$$W_i = \underbrace{\frac{1}{2} \left(\text{erf} \left(\frac{x_1}{\sigma\sqrt{2}} \right) - \text{erf} \left(\frac{x_0}{\sigma\sqrt{2}} \right) \right)}_{Y_N(x_1, \sigma) - Y_N(x_0, \sigma)} \cdot \underbrace{\exp \left(-\frac{1}{2} \left(\frac{\delta}{\sigma} \right)^2 \right)}_{y_N(\delta, \sigma) / y_N(0, \sigma)} \quad (18)$$

V. TEST CASE

In order to compare the results from the different smoothing methods and weights, a test case is defined. This test case simulates a point particle moving around in a meshed unit square. The test mesh consists of 3011 nodes and 5810 triangular cells and span between $(0 \leq x \leq 1 \text{ m})$ and $(0 \leq y \leq 1 \text{ m})$. The unit depth is defined as 1 m, and the cell volumes vary from $3.3 \cdot 10^{-5} \text{ m}^3$ to $4.7 \cdot 10^{-3} \text{ m}^3$. The test trajectory was defined as a circle with varying radius spanning two full rotations. This is given by Eq. (19).

$$\begin{aligned} R(\theta) &= \frac{1}{8} \cos(\theta \cdot 3.5) + 0.325 \\ \vec{x}_P(\theta) &= [\cos(\theta), \sin(\theta)] \cdot R(\theta) + \frac{1}{2} \\ 0 &\leq \theta < 4 \cdot \pi \end{aligned} \quad (19)$$

The analytical exchange field was calculated for the trajectory according to Eq. (4). The field, the mesh, and the trajectory are all shown in Fig. 9.

Fig. 9. The test mesh used for comparing the different methods and weights. The trajectory is defined using 150 points, which are indicated by the solid black dots. The mesh intersections are shown by the white dots in the lower part of the figure. The analytical exchange field per unit volume is shown in the background, where the upper part of the figure shows the continuous exchange field, and the lower part shows the exchange field, after integration, on the mesh cells.

VI. COMPARISON

In order to compare the different methods and weights previously described, two different evaluation parameters are defined in Eq. (20). These parameters compare the calculated exchange fields using the methods and weights described, ϕ_i , with the analytically integrated exchange field, $\phi_i^{\text{Analytical}}$.

$$E_1 = \sum_{i=1}^{n_{\text{Cells}}} \left| \phi_i^{\text{Analytical}} - \phi_i \right| / \left(\sum_{i=1}^{n_{\text{Cells}}} \left| \phi_i^{\text{Analytical}} \right| \right)$$

$$E_2 = \sum_{i=1}^{n_{\text{Cells}}} \left| \frac{\phi_i^{\text{Analytical}} - \phi_i}{V_i} \right| / \left(\sum_{i=1}^{n_{\text{Cells}}} \left| \frac{\phi_i^{\text{Analytical}}}{V_i} \right| \right) \quad (20)$$

The first evaluation parameter, E_1 , is evaluated by calculating the absolute error in the exchange rate between the analytical and used method/weight for all cells. The sum of these errors divided by the sum of the absolute analytical exchange rate is E_1 . This parameter is thus a percentage of the total exchange rate.

The second evaluation parameter, E_2 , is similar to E_1 , but instead of the absolute exchange rates, the volume specific exchange rates are used. The two error parameters will each approach zero, if the calculated exchange field is similar to the analytical exchange field.

The three methods, node-, kernel-, and cell-based, are each tested with the five different weights. Each of these 15 exchange fields are compared to the analytically integrated case, where the differences are shown in Fig. 10. It should be noted that each of the exchange fields have conserved ϕ , but the spatial locations of ϕ are different. σ was defined to be 0.05m, which results in a cut-off distance of $\chi = 2 \cdot \sigma = 0.1\text{m}$.

E_1 and E_2 are evaluated for all cases, where the results can be seen in Table I. The proposed weight is abbreviated CI for *Continuous Integrated*.

TABLE I
A COMPARISON OF THE ERRORS RELATIVE TO THE ANALYTICAL SOLUTION. THE VALUES IN THE CELLS ARE INDICATED AS E_1 [%] / E_2 [%].

Weight \ Method	Node-based	Kernel-based	Cell-based
Even	28.4 / 28.6	36.3 / 33.9	36.9 / 34.1
Shortest-distance	28.2 / 28.6	36.3 / 33.9	36.7 / 34
Inverse-distance	13.4 / 13.1	132 / 161	22.7 / 21.9
Adjusted normal dist.	5.32 / 3.75	6.9 / 4.84	5.99 / 4.88
(CI) Adjusted normal dist.	5.33 / 3.75	6.9 / 4.84	2.12 / 1.08

As seen in Table I, the *Even* and *Shortest-distance* weights have almost equal values of E_1 and E_2 . This is caused by the shortest-distance definition, which returns almost even weights, when the number of included cells increase.

The *Inverse-distance* weight have large values of E_1 and E_2 when the kernel-based method is used. This is caused by the *Inverse-distance* weight definition, where each cell is redistributing the accumulated exchange rates to all cells within a distance of χ , including itself. By the definition given in Eq. (12), this will return a weight of unity to its own cell. Thus, no redistribution occurs when the inverse-distance is coupled with the kernel-based method.

The *Adjusted normal distribution* weight is close to the ground-truth, where the closest match is found for the *Node-based* method. The *Continuous Integrated (CI) Adjusted normal distribution* weight has the overall lowest error. When combined with the cell-based method, the values of E_1 and E_2 are 2.12 and 1.08 % respectively, which is significantly less compared to the other methods and weights presented in this paper as seen in Table I.

The results are visually illustrated in Fig. 10, where the local errors in the specific exchange fields are shown. As seen in Fig. 10, the different methods combined with the different weights produce very different results. The *Even* weight generally has negative values near the trajectory and positive values far from the trajectory, which is caused by the weight, which does not take the distance into account at all. The same tendency can be seen for the *Shortest-distance* weight, which returns almost even values, when the number of cells increases.

The problem, previously described, with the *Inverse-distance* weight combined with the kernel-based method is apparent in the figure, where only the cells containing part of the trajectory contains any exchanged property. This is equivalent to a cell-marking weight, where only the cells hit by the trajectory receive the exchanged property. This has further been elaborated in the lower part of Fig. 10 to the left, where the values of E_1 and E_2 are identical to the *Inverse-distance* weight combined with the kernel-based method.

The fourth and fifth rows in Fig. 10 show the *Adjusted normal distribution* weights, where the bottom one is the continuous integrated version. It can be seen that both are close to the analytically integrated exchange field, but the cell-

based method with the *Continuous integrated adjusted normal distribution* weight has the lowest overall error.

A *Purely diffusive model* is shown in the last row to the right in Fig. 10. This model simply distributes the exchanged property evenly to all cells, which corresponds to the *Even* weight as $\chi \rightarrow \infty$. This model is included for comparison with the other methods and weights, where $E_1 = 74.6\%$ and $E_2 = 90.4\%$, which, for this test case, is lower compared to the *Cell-marking model*. However, the *Purely diffusive model* is unsuitable for most problems, as the location of the particle is not taken into account.

Fig. 10. The difference between the analytical exchange field and the different methods / weights. The rows indicate different weights, and the columns indicate the different methods. The error parameters are shown in Table I and in the lower left part of each graph. The blue colour indicates lower cell values compared with the analytical case, whereas red colours indicate greater values. All results are shown per unit volume as defined by the colour-bar.

VII. IMPLEMENTATION

A. Methods

The three methods described in this paper all require mesh information. The procedure to find all cells located within a distance of χ can be computationally expensive. Therefore, this information can be pre-calculated before the simulation is started. One way to do this is to create a Boolean sparse matrix, which would have a size of $[n_{\text{Nodes}} \times n_{\text{Cells}}]$ for node-based methods. The values of this sparse matrix is defined in Eq. (21). Notice that this requires a static mesh, which is not deforming over time.

$$\mathbf{K}_{iN,iC} = \begin{cases} \text{true} & \text{if } |\vec{x}_{iN} - \vec{x}_{iC}| \leq \chi \\ \text{false} & \text{otherwise} \end{cases} \quad (21)$$

When the matrix is pre-computed, the cells can quickly and efficiently be looked-up. The same method can be used for the kernel-based methods, where $\mathbf{K}$ would have a size of $[n_{\text{Cells}} \times n_{\text{Cells}}]$. For cell-based methods the particle is constantly moving and the cells located within a distance of χ is constantly changing, so a lookup-matrix cannot be pre-computed. In this case the cell, in which the particle is located, can be used in the kernel-based sparse matrix. This will approximate the cells located within a distance of χ .

Another way to speed up the methods is to pre-compute the node-cell weights or cell-cell weights assuming a constant value of σ . These weight matrices would be sparse, and can be used to distribute the exchange rates by simple matrix multiplication, as shown in Eq. (22).

$$\begin{aligned} \phi_C &= \mathbf{K}_{N \rightarrow C} \cdot \phi_N \\ \phi_C &= \mathbf{K}_{C \rightarrow C} \cdot \phi_{C,\text{Raw}} \end{aligned} \quad (22)$$

The upper equation in Eq. (22) is for the node-based method, whereas the lower is for the kernel-based method. $\mathbf{K}_{N \rightarrow C}$ has size $[n_{\text{Cells}} \times n_{\text{Nodes}}]$ and $\mathbf{K}_{C \rightarrow C}$ has size $[n_{\text{Cells}} \times n_{\text{Cells}}]$. When implementing the smoothening methods using these techniques, the procedure can be speeded up significantly.

B. Distributions

This paper was based on the assumption that the ground-truth exchange field follows the adjusted normal distribution defined in Eq. (3) along the trajectory of the particle/parcel. The proposed weight to continuously integrate the probability density function requires that the cumulative density function is evaluated twice for each particle-cell pair as shown in Eq. (16). The cumulative density function can be computationally expensive to evaluate, where the *Normal distribution* uses the error-function. However, alternative distributions can be used to approximate the *Adjusted normal distribution*, which are computationally less expensive to evaluate. This section lists some distributions, which are all limited to span between $-\chi < x < \chi$. These will all be compared to the ground-truth along with the computational requirements for each of them.

- 1) *Normal distribution*: Refer to Eq. (1).
- 2) *Truncated normal distribution*: Refer to Eq. (2).
- 3) *Adjusted normal distribution*: Refer to Eq. (3).

4) Triangular:

$$\begin{aligned} y_{\text{Tri}}(x, \chi) &= \frac{1}{\chi} - \frac{|x|}{\chi^2} \\ Y_{\text{Tri}}(x, \chi) &= \frac{-\text{signum}(x) x^2 + 2\chi x + \chi^2}{2\chi^2} \end{aligned} \quad (23)$$

5) *Deen polynomial*: The Deen polynomial approximates the normal distribution and is constructed from a fourth and a fifth order polynomial for the probability and cumulative density functions respectively [9].

$$\begin{aligned} y_{\text{Deen}}(x, \chi) &= \frac{15}{16\chi} \left(\left(\frac{x}{\chi} \right)^4 - 2 \left(\frac{x}{\chi} \right)^2 + 1 \right) \\ Y_{\text{Deen}}(x, \chi) &= \frac{1}{16} \left(3 \left(\frac{x}{\chi} \right)^5 - 10 \left(\frac{x}{\chi} \right)^3 + 15 \left(\frac{x}{\chi} \right) + 8 \right) \end{aligned} \quad (24)$$

6) Cosine distribution:

$$\begin{aligned} y_{\text{cos}}(x, \chi) &= \frac{1}{2\chi} \left(1 + \cos \left(\pi \frac{x}{\chi} \right) \right) \\ Y_{\text{cos}}(x, \chi) &= \frac{1}{2} \left(\frac{1}{\pi} \sin \left(\pi \frac{x}{\chi} \right) + \frac{x}{\chi} + 1 \right) \end{aligned} \quad (25)$$

7) Top-hat:

$$\begin{aligned} y_{\text{TopHat}}(x, \chi) &= \frac{1}{2\chi} \\ Y_{\text{TopHat}}(x, \chi) &= \frac{1}{2} + \frac{x}{2\chi} \end{aligned} \quad (26)$$

The latter five distributions can be seen in Fig. 11, whereas the first three were shown earlier in Fig. 1.

Fig. 11. The probability, $y(x)$, and cumulative, $Y(x)$, density functions plotted in the upper and lower figure respectively. Notice that all distributions shown in this figure are limited to $-\chi < x < \chi$, where $\chi = 2 \cdot \sigma$ in this paper.

C. Comparison

The different distributions are all compared to the analytical integrated exchange field. Furthermore, each of the distributions are timed to get the relative computational requirements.

The timing results are shown in Table II, where each distribution was evaluated 10^9 times, where x was varied between $-\chi$ and χ . The analysis was carried out in C++, where each distribution was optimized to run as fast as possible.

As seen in Table II, the *Top-hat distribution* is the fastest, which only requires 3.5 % of the time of the *Adjusted normal distribution*. the *Deen polynomial* is almost as fast as the *Top-hat* distribution, which requires ≈ 4.6 % of the *Adjusted normal distribution*.

The exchange fields for the different distributions were compared with the analytical integrated exchange field, where the values of E_1 and E_2 were evaluated. These results are shown in Table III.

As seen in Table III, the error is, in general, lowest for the cell-based methods. It can be seen that both the *Triangular distribution* and the *Deen polynomial* are close to the ground-truth, while only requiring a fraction of the computational time when compared with the adjusted normal distribution (See Table II).

Reference [10] compared different distributions as well, where the normal, top-hat and triangular distributions were compared. The results from [10] showed similar tendencies in terms of accuracy as this study, where the top-hat distribution had the largest error of the three compared distributions.

VIII. CONCLUSIONS

This paper investigated different two-way coupling methods. Each of these methods requires some weighting function. In this work, five different weights were investigated. In order to quantify the accuracy of the different methods and weights, a test case was defined, which contained different sized triangular cells. A trajectory was defined for a single particle, which spanned the whole trajectory. A semi analytical integrated exchange field was calculated, which served as the ground-truth, which all methods and weights were compared to.

Three different smoothening methods and five different weights were evaluated. Two error parameters were defined, which were related to the error between the calculated exchange field and the ground-truth. The results showed that

TABLE II
THE COMPUTATIONAL REQUIRED CLOCK TIME TO COMPLETE 10^9 FUNCTION EVALUATIONS. THE PERCENTAGE IS DEFINED RELATIVE TO THE ADJUSTED NORMAL DISTRIBUTION, WHICH IS HIGHLIGHTED IN BOLD FONT.

Name	Relative [%]	Time [s]
Normal distribution	51.40	126.63
Truncated normal distribution	68.17	167.95
Adjusted normal distribution	100.00	246.38
Cosine distribution	68.73	169.32
Triangular distribution	14.82	36.51
Deen polynomial	4.58	11.28
Top-hat distribution	3.52	8.67

TABLE III

A COMPARISON OF THE ERRORS RELATIVE TO THE ANALYTICAL SOLUTION. THE VALUES IN THE CELLS ARE INDICATED AS E_1 [%] / E_2 [%]. NOTICE THAT THE CONTINUOUS INTEGRATED WEIGHT IS APPLIED FOR ALL DISTRIBUTIONS.

Weight \ Method	Node-based	Kernel-based	Cell-based
Normal distribution	10.2 / 7.27	9.96 / 8.61	8.41 / 8.26
Truncated normal dist.	10.2 / 7.27	6.93 / 6.3	7.08 / 7.59
Adjusted normal dist.	5.33 / 3.75	6.9 / 4.84	2.12 / 1.08
Cosine distribution	8.05 / 9.53	12.7 / 11.4	8.09 / 6.99
Triangular distribution	6.16 / 2.82	5.03 / 2.54	3.99 / 4.1
Deen polynomial	6.43 / 6.46	9.58 / 7.99	4.45 / 3.29
Top-hat distribution	28.4 / 28.6	27.4 / 28.6	33.6 / 33

the cell-based method was in general more accurate when compared to the node-based and kernel-based methods. When comparing the different weights, it was found that the proposed weights to continuously integrate the exchange rates along the particle trajectory was the most accurate weight, where ≈ 2 % of the exchange rates were incorrectly distributed.

The proposed weight requires the probability and cumulative density functions to be specified. These can be computationally expensive to evaluate, so a number of alternative distributions were investigated. A C++ program was coded to compare the computational requirements, where it was found that the "*Deen polynomial*" distribution was about 20 times faster compared with the distribution used for the analytical exchange field. The error in the exchange rates was ≈ 4.5 % using this distribution. It serves as a good alternative, if the computational requirement are to be kept to a minimum.

This paper has described how to distribute exchange rates from particles to a mesh while assuming the flux from the particle to be constant. However, for most problems the flux will change over time. Assuming a small time-step, during which the flux does not change significantly, the methods in this paper still apply. If, however, the flux varies significantly, special care must be taken, which is a subject for another study. One approach would be to linearise the weights at the start and end of the trajectory, which allows for integrating the weights over time.

IX. ACKNOWLEDGEMENTS

This work was supported by *Danmarks Innovationsfond* under grant no. 5016-00061.

REFERENCES

- [1] S. Pirker, D. Kahrmanovic and C. Goniva, *Improving the applicability of discrete phase simulations by smoothening their exchange fields*, Applied Mathematical Modelling, Volume 35, Issue 5, May 2011, Pages 2479-2488
- [2] Q. Zhou and S. C. Yao, *Group Modeling of Impacting Spray Dynamics*, International Journal of Heat and Mass Transfer, Volume 35, Issue 1, January 1992, Pages 121-129
- [3] P. R. Jonas and J. T. Bartlett, *The Numerical Simulation of Particle Motion in a Homogeneous Field of Turbulence*, Journal of Computational Physics, Volume 9, Issue 2, April 1972, Pages 290-302
- [4] W. J. Guo, Y. X. Wang, M. X. Xie and Y. J. Cui, *Modeling oil spill trajectory in coastal waters based on fractional Brownian motion*, Marine Pollution Bulletin, Volume 58, Issue 9, September 2009, Pages 1339-1346

- [5] L. L. Baxter and P. J. Smith, *Turbulent dispersion of particles: the STP model*, Energy & Fuels , 7 (6) , November 1993 , Pages 852–859
- [6] R. J. Litchford and S. Jeng, *Efficient Statistical Transport Model for Turbulent Particle Dispersion in Sprays*, AIAA Journal , Volume 29 , No. 9 , September 1991 , Pages 1443-1451
- [7] ANSYS Inc., *ANSYS Fluent Theory Guide*, ANSYS Inc. 2018
- [8] ANSYS Inc., *ANSYS CFX Theory Guide*, ANSYS Inc. 2018
- [9] N. G. Deen, M. van Sint Annaland and J. A. M. Kuipers, *Multi-scale modeling of dispersed gas–liquid two-phase flow*, Chemical Engineering Science , Volume 59 , Issues 8–9, April–May 2004 , Pages 1853-1861
- [10] A. Kaufmann, M. Moreau, O. Simonin and J. Helie, *Comparison between Lagrangian and mesoscopic Eulerian modelling approaches for inertial particles suspended in decaying isotropic turbulence*, Journal of Computational Physics , Volume 227, Issue 13 , 20 June 2008 , Pages 6448-6472