In-situ flame particle tracking based on barycentric coordinates for studying local flame dynamics in pulsating Bunsen flames

Thorsten Zirwes [a, b, *], Feichi Zhang [b], Yiqing Wang [c], Peter Habisreuther [b], Jordan A. Denev [a], Zheng Chen [c], Henning Bockhorn [b], Dimosthenis Trimis [b]

[a] *Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, Eggenstein-Leopoldshafen 76344, Germany*

[b] *Engler-Bunte-Institute, Division of Combustion Technology, Karlsruhe Institute of Technology, Engler-Bunte-Ring 1, Karlsruhe 76131, Germany*

[c] *SKLTCS, CAPT, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China*

Abstract

Flame particles (FP) are massless, virtual particles which follow material points on the flame surface. This work presents a tracking algorithm for FPs which utilizes barycentric coordinates. The methodology can be used with any cell shape in the computational mesh and allows computationally fast spatial interpolation as well as efficient determination of the intersection of FP trajectories with iso-surfaces. In contrast to previous flame particle tracking (FPT) approaches, the code is fully parallelized and can therefore be used in-situ during the simulation. It also includes fully parallelized computation of flame consumption speed by integrating reaction rates along a line normal to the flame surface at each FP position. Direct numerical simulations of laminar pulsating premixed hydrogen–air Bunsen flames serve as validation cases and showcase the added value of tracking material points for studying local flame dynamics. Exciting the inlet flow harmonically with frequencies equal to the inverse flame time scale leads to a pulsating mode where the flame front is corrugated. Ten times higher frequencies nearly resemble the steady state solution. The FPs are seeded along the flame surface and are used to track the unsteady diffusive, convective and chemical contributions at arbitrary points on the flame front over time. Their trajectories reveal a phase shift between the unsteady flame stretch rate and local flame speed of the order of 0.1 flame time scales for rich hydrogen flames. This is caused by a time delay between straining and stretch due to curvature. The reason is that diffusive processes follow the time signal of curvature while chemical processes are most strongly affected by the straining rate, which

* Corresponding author at: Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany.

E-mail address: thorsten.zirwes@kit.edu (T. Zirwes).

dominates the high Lewis number hydrogen flames investigated. This time history effect may help to explain the large scattering in the correlation of local flame speed with flame stretch found in turbulent flames.

Keywords: Flame particle tracking; Barycentric coordinates; Oscillating flames; Laminar flames; Direct numerical simulation

1. Introduction

The interaction of flames with turbulent flows is a highly complex phenomenon due to the large number of relevant time and length scales of many physical processes. Therefore, it is often described by global quantities like integral length and time scales or turbulent flame speed. When it comes to local flame dynamics, however, statistical descriptions are usually utilized, like the correlation between local flame speed and flame stretch. These often show a large scattering, although the same correlation in laminar steady-state flames yields a unique relation between each value of flame stretch and flame speed [1].

In order to study local flame dynamics in more detail and gain more information from direct numerical simulation (DNS), a Lagrangian viewpoint can be employed. In this work, we apply the flame particle tracking (FPT) methodology [2]. FPT is a Lagrangian method, where infinitely small, massless, virtual particles, that do not interfere with the flame or flow, are tracked during the simulation. These particles, called flame particles (FP), follow material points on the flame surface [3]. Their trajectories give insight into the time evolution of local flame dynamics. The method known as FPT was introduced by Chauduri [2,4] and Uranakara et al. [5]. They used FPs to study the correlation of local displacement speed with curvature and strain rates on temperature iso-surfaces using data from DNS of turbulent flames. The same methodology was used by Dave et al. [6], but in their work, the particles are tracked backwards in time. In this way, the FPs can be used to track the evolution of flame surface area. Chauduri [7] studied Bachelor's pair dispersion law for FPs in turbulent flames. Uranakara et al. [8] used FPT to create ensemble statistics of displacement speed and curvature in ignition kernels in turbulent flows. These works all were limited in the number of FPs, because the FPT methodology was implemented as a post-processing step. For example, in [2], Matlab is utilized for the post-processing of the DNS data. This requires to store large amounts of data with time steps in high frequency and also limits the utilization of parallel computer hardware.

Other authors used Lagrangian methods for studying local flame dynamics recently as well: Hamlington et al. [9] studied *fluid* particles moving with the fluid flow to investigate residence times and path lengths of the particles in turbulent flames. Another study involving fluid particles was done by Towery et al. [10] who tracked chemical modes in highly turbulent flames. Scholtissek et al. [11] used Lagrangian tracking to recover unsteady flamelet histories from turbulent flames. Zheng et al. [12] tracked the evolution of flame surface area elements in turbulent flames to study the statistics of principal curvature and surface area ratio evolution. Similarly, Yang et al. [13] investigated the evolution of Lagrangian surfaces in turbulent flows and Oster et al. [14] tracked the deformation of flame surface micro-patches. In an even earlier work, Sripakagorn et al. [15] used flame element tracking to gain insight into reignition events.

Because all previous FPT approaches were implemented as a post-processing step, we present a fully parallelized FPT implementation in this work, which gains additional computational efficiency and flexibility due to the use of barycentric coordinates, and can be used in-situ during the simulation. In all aforementioned works, local flame speed has been evaluated in terms of the displacement speed. In some cases, the consumption speed represents a physically more meaningful quantity because it measures the consumption of fuel across the whole thickness of the flame front. The displacement speed can be calculated at each position in the flame front based on the local mixture composition. The consumption speed, however, requires the integration of reaction rate of e.g. the fuel species along a line normal to the flame front. Doing this on a large number of points is computationally expensive. Therefore, a fully parallelized in-situ computation of the consumption speed is presented as well. The implementation details are given in Section 2.

The main focus of this work lies on introducing the novel, computationally efficient flame particle tracking (FPT) algorithm along with its mathematical basics and implementation, which is then further validated by means of oscillating laminar flames similar to a previous work [1]. The flame dynamics of highly transient laminar flames can help to explain and interpret the more complex phenomena in turbulent flames. Therefore, DNS of pulsating laminar Bunsen flames are carried out in Section 4 to validate the implemented approach and study local time history effects. Additionally,

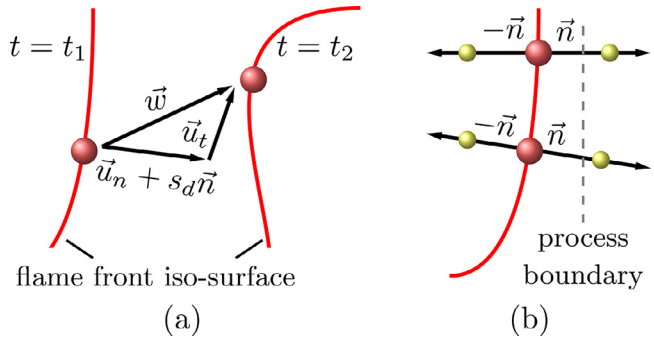

Fig. 1. (a) Iso-surfaces at two different times $t_1 < t_2$. A point on the iso-surface can be tracked by using a FP, which is shown as red sphere and moves with the velocity $\vec{w}$. (b) Consumption speed is computed by tracking two additional virtual particles in positive and negative normal direction, summing up the reaction rates along the line.

the usefulness of the FPT method for studying local flame dynamics is demonstrated by tracking chemical, diffusive and convective contributions along trajectories of material points on the flame surface together with the different components of flame stretch and displacement speed, which is not possible without the use of FPs. Choosing a laminar setup allows to study transiently stretched flames in a controlled manner with specific oscillation frequencies instead of the broad-band frequency spectrum present in turbulent flames. Nevertheless, first results for FPT on an iso-surface of a turbulent flame are given in S10 in the Supplementary materials.

2. Tracking algorithm

Flame particles (FPs) track material points on the flame surface. The flame surface is defined as an iso-surface of a scalar quantity, e.g. temperature or species mass fraction. Once a FP is seeded onto an iso-surface, it will stay on that iso-surface over time. By only looking at iso-surfaces representing the flame front at different times, it is not immediately clear which point on an iso-surface at time t_1 corresponds to which other point on an iso-surface at a later time t_2. The iso-surface of the flame front moves due to two effects: convection with the fluid velocity $\vec{u}$ and a velocity due to the burning of the fuel, the displacement speed s_d, which moves the iso-surface in its normal direction $\vec{n}$. The absolute movement velocity of the iso-surface $\vec{w}$ can therefore be expressed as

$$\vec{w} = \vec{u} + s_d\vec{n} = \vec{u}_t + \vec{u}_n + s_d\vec{n} \tag{1}$$

using the convention that $\vec{n}$ points towards the fresh gases. This method describes the movement of material points exactly and without assumptions and all quantities can be recovered from DNS data. The movement in normal direction is caused by $\vec{u}_n + s_d\vec{n}$, and the movement in tangential direction to the iso-surface is due to $\vec{u}_t$, where $\vec{u}_n = (\vec{u}\cdot\vec{n})\vec{n}$ is the fluid velocity in normal direction and $\vec{u}_t = \vec{u} - \vec{u}_n$ in tangential direction. This is illustrated in Fig. 1a. Therefore, *FPs* move with $\vec{w}$, whereas the more commonly known *fluid* particles move only with $\vec{u}$.

In some cases, it is advantageous to compute the unit normal vector $\vec{n}$ from

$$\vec{n} = \nabla\left(\frac{\Phi - \Phi_{\text{iso}}}{|\nabla\Phi|}\right)\Bigg|_{\Phi=\Phi_{\text{iso}}} \tag{2}$$

If evaluated on the iso-surface, this is equivalent to the more common definition $\vec{n} = \nabla\Phi/|\nabla\Phi|$. However, using Eq. (2) ensures that $\nabla\times\vec{n} = 0$. This makes computation of some stretch contributions easier, because it is guaranteed that $\nabla_t\cdot\vec{a}_t = \nabla\cdot\vec{a}_t$, where ∇_t is the tangential gradient and $\vec{a}_t$ an arbitrary vector tangential to the flame surface (see also the Supplementary material for the derivation).

In the most simple case, the position $\vec{p}$ of a material point on the new iso-surface can be found by $\vec{p}(t_1 + \Delta t) \approx \vec{p}(t_1) + \vec{w}\Delta t$, where Δt is the time step. This methodology is the same for laminar and turbulent flame cases. However, especially for turbulent flames, the particle movement should be computed using a predictor-corrector step, Runge–Kutta integration or special higher order interpolation methods to achieve sufficient accuracy. A more detailed discussion on this topic can be found in [5,16,17].

Because the FP stays on the iso-surface of an arbitrary scalar Φ it fulfills the conditions

$$\frac{\partial\Phi}{\partial t} + \vec{w}\cdot\nabla\Phi = 0,\ s_d = -\frac{1}{|\nabla\Phi|}\frac{\mathrm{D}\Phi}{\mathrm{D}t} \tag{3}$$

where $\mathrm{D}/\mathrm{D}t = \partial/\partial t + \vec{u}\cdot\nabla$. If a specie mass fraction $\Phi = Y_k$ is chosen for the iso-surface, the displacement speed can equivalently be computed as $s_d = (\nabla\cdot\vec{j}_k - \dot{\omega}_k)/|\rho\nabla Y_k|$, where $\vec{j}_k$ is the diffusive flux of specie k and $\dot{\omega}_k$ its reaction rate.

In addition to tracking the FPs, another particle type is introduced for the computation of the con-

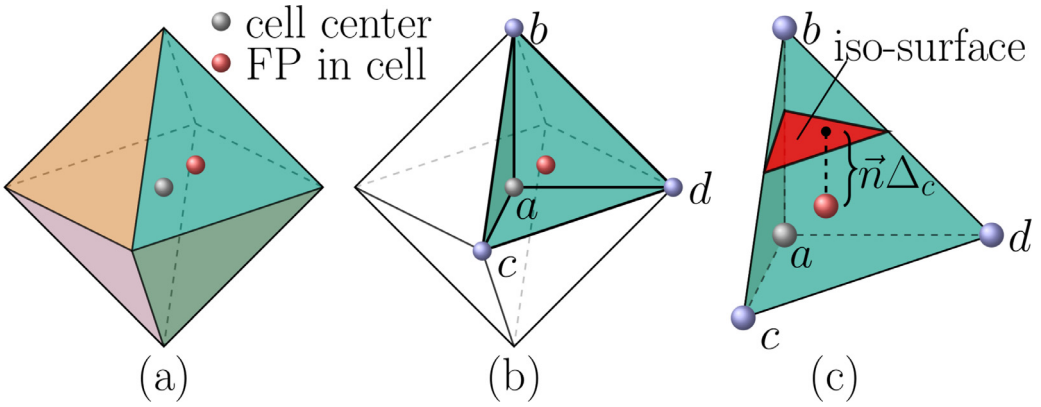

Fig. 2. (a) Arbitrarily shaped cell with cell center and FP inside. (b) Decomposition into tetrahedra. (c) Correction of the FP position onto the iso-surface.

sumption speed s_c in terms of the fuel species F:

$$s_c = \frac{1}{\rho_u(Y_{F,b} - Y_{F,u})} \int \dot{\omega}_F \, \mathrm{d}n \tag{4}$$

Here, u and b denote the fully unburnt and burnt state. At each position on the flame front, where a FP is located, two additional virtual particles are generated (yellow spheres in Fig. 1b). These are tracked in positive and negative normal direction to the flame front over a distance of the laminar flame unstretched thickness δ^0_{th} in each direction. Because this is done in-situ, these virtual particles can hit the boundary between processes in the decomposed computational domain, so that they have to be transferred from one process to another. The tracking and movement between processes is done concurrently for all virtual particles, enabling the computation of s_c on a large number of positions on the flame front without reducing the performance of the simulation significantly.

In this implementation, all virtual particles are tracked in barycentric coordinates. The position of each particle is given by the four-component vector $\hat{\lambda} = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)^T$. First, each cell in the computational domain is decomposed into tetrahedra, where one vertex of the tetrahedra is always the cell center, and the other three are cell corner points. One advantage is, that any polyhedral shape can be uniquely decomposed into tetrahedra, so that the methodology also works on arbitrary unstructured grids. Fig. 2 illustrates this: in Fig. 2a, a polyhedral cell is shown with the cell center in the middle and a FP in the cell. In Fig. 2b, the cell is decomposed into tetrahedra, highlighting the tetrahedron made up by the points *a, b, c, d* containing the FP. The barycentric coordinates $\hat{\lambda}$ represent the weights at the tetrahedron vertices, e.g. vertex a is represented by $(1, 0, 0, 0)^T$, b by $(0, 1, 0, 0)^T$ and so on. The barycentric weights fulfill the constraint

$$\sum_{i=0}^{3} \lambda_i = 1 \tag{5}$$

For an in depth discussion of barycentric coordinates, see [18]. Another advantage of the barycentric coordinate representation is, that interpolation of any quantity φ, e.g. temperature, at the FP position φ_p can be done in a computationally efficient way:

$$\varphi_p = \sum_{i=0}^{3} \lambda_{p,i} \varphi_i \tag{6}$$

where $\lambda_{p,i}$ are the barycentric coordinates of the FP and φ_i are the values of the scalar φ at the four tetrahedron vertices. Because the Lagrangian tracking of the FP is subject to numerical errors, the FP will not be moved exactly on the iso-surface at the next time step after tracking. In order to avoid accumulation of numerical errors, the position of the FP is corrected after each time step so that it is positioned exactly on the iso-surface. In previous works [2,5], this was achieved by computing the iso-surface globally and finding the intersection of the iso-surface with the FP trajectory via ray tracing, which is computationally demanding. In this work, the iso-surface is only determined within the tetrahedron, where the FP is located in. The intersection of the iso-surface with the FP trajectory can be determined analytically without the need for ray tracing. The FP position is corrected by $\vec{n}\Delta_c$ from Eq. (7), which is the shortest distance between the iso-surface and the FP.

$$\Delta_c = (\mathbf{T}\hat{\lambda}_p) \cdot \vec{n} - (\mathbf{T}\hat{\lambda}_{\text{edge}}) \cdot \vec{n} \tag{7}$$

$\mathbf{T}$ is the 3×4 matrix $(\vec{a}\,\vec{b}\,\vec{c}\,\vec{d})$ containing the Cartesian position vectors of the tetrahedron corners and $\hat{\lambda}_p$ is the position of the FP in barycentric coordinates. $\hat{\lambda}_{\text{edge}}$ is found by combining Eq. (5) with the description of the iso-surface $\Phi_{\text{iso}} = \sum_i \lambda_i \Phi_i$ according to Eq. (6). By identifying an edge of the tetrahedron that intersects the iso-surface (e.g. edge b–c in Fig. 2c) and setting the two components of λ_{edge} to zero corresponding to the tetrahedron corners not part of that edge (e.g. $\hat{\lambda}_{\text{edge}} = (0, \lambda_1, 0, \lambda_3)^T$), the remaining two unknowns λ_1, λ_3

can be found with the help of the two equations (Eqs. (5) and (6)) and the correction vector Δ_c can be computed. If the intersection point lies in another tetrahedron, the particle is iteratively moved by $\Delta_c = \varepsilon(\Phi_{\text{iso}} - \Phi)\nabla\Phi|\nabla\Phi|^{-2}$ evaluated at the particle position until it enters the correct tetrahedron. ε is an under-relaxation factor set to 0.9.

The algorithm as described here is implemented in the open-source tool OpenFOAM [19], which provides a framework for particle tracking in barycentric coordinates. In a medium-scale DNS of a turbulent methane/air flame on a grid with 64 million cells using 500 CPU cores, 6 million FPs were added and tracked during the simulation. The overall computation time was increased by about 20 %. An animation of the turbulent flame case is included in the Supplementary materials. More information about the in-situ tracking of the FPs, calculating $\vec{n}$ and $\vec{w}$, position correction onto the iso-surface, interpolation of arbitrary quantities at the FP position and parallel computation of the consumption speed can be found at `vbt.ebi.kit.edu/index.pl/specialtopic/DNS-Links`. Note that the code is written in a general way to perform the particle tracking for any kind of iso-surface, not just for reactive flows.

3. Physical interpretation of FP movement

Surface flame stretch K is usually expressed as

$$K = \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}t} = \nabla_t \cdot \vec{u} + s_d \nabla \cdot \vec{n} \tag{8}$$

where A is the area of an infinitesimal surface element on the flame front. An equivalent expression is [20]:

$$K = \frac{1}{A}\frac{\mathrm{d}A}{\mathrm{d}t} = \nabla_t \cdot \vec{w} \tag{9}$$

This is the same $\vec{w}$ from Eq. (1). Therefore, the movement of material points and by definition the trajectories of FPs are directly linked to flame stretch. FPs moving closer to each other signifies negative flame stretch and FPs moving away from each other positive stretch. Therefore, FPs tend to move toward regions with negative flame stretch which act as sink for flame surface area. This means that regions with strong negative stretch act as attractors for FPs. This effect can be observed in the animation of the turbulent flame case in the Supplementary materials. In [6], this effect is shown by starting out with equally spaced FPs, which after some time collect in regions with negative curvature or negative flame stretch, respectively. When taking ensemble probability density functions (pdf), this effect should be considered because otherwise the pdf will be skewed toward the negative stretch range.

4. Pulsating Bunsen flames

The implemented FPT methodology is employed in DNS of pulsating premixed hydrogen–air Bunsen flames. The aim is to show the added value of employing FPT by studying time history effects on the local flame dynamics, even in the context of laminar flames.

Pulsating laminar flames have been investigated by many authors in the past [21–28]. Many of the earlier works, however, used oscillating counterflow flames, which are not stretched due to curvature. In this work, all stretch components are included. Both lean ($\phi = 0.5$) and rich ($\phi = 4$) hydrogen–air flames are considered and the oscillation frequency is set to $f = 0.1/\tau_c$, $f = 1/\tau_c$ and $f = 10/\tau_c$, where τ_c is the time scale of the flame, determined from $\tau_c = \delta^0_{th}/s^0_l$.

The domain is 2D with rotational symmetry (Fig. 3). A center inlet for fresh gas with $T = 300$ K, $p = 1$ bar and prescribed equivalence ratio ϕ, and a pilot with the burnt gas at the same ϕ in chemical equilibrium is used. Both inlets have parabolic velocity profiles. All dimensions of the domain are set as multiples of the flame thickness $\delta^0_{th} = (\max(T) - \min(T))/\max(\partial T/\partial n)$ and flame speed s^0_l, both of the unstretched freely propagating flame at the inlet conditions. Therefore, as the equivalence ratio changes in different simulations, the physical dimensions of the domain change as well. The maximum velocity is chosen to be $u_{\max} = 6s^0_l$, resulting in Reynolds numbers between 500 and 1000 with respect to the fresh gas inlet for the investigated cases. The height of the domain is approximately 3 flame heights h_f and the radius of the inlets are multiples of δ^0_{th} as well. This has several advantages: flames with different equivalence ratio experience stretch rates of comparable Karlovitz number $Ka = K\,\delta^0_{th}/s^0_l$; the resolution of the mesh is equidistant $\Delta x = \Delta y = \delta^0_{th}/20$, ensuring that the flame is resolved with about 20 cells. Compared to the setup employed in a previous work [1], this setup has several advantages: only half a flame is included due to the symmetry, making the simulation more computationally efficient; radial symmetry instead of Cartesian symmetry causes higher stretch rate due to the curvature in circumferential direction; the hot pilot stabilizes the flame to allow for higher stretch rates.

This setup has several advantages over common laminar flame setups: the flame is stretched both positively and negatively and is also curved both positively and negatively. In the oscillating case, the flame is also transiently stretched by one frequency f. This is achieved by letting the velocity profiles at the inlet oscillate with $u_y = u_{y,inlet}(1 + 0.5\sin(2\pi f t))$. Hydrogen has been chosen as fuel because it exhibits a strong Lewis number effect and therefore high sensitivity to flame stretch for lean mixtures ($Le \ll 1$) and rich mixtures ($Le \gg 1$).

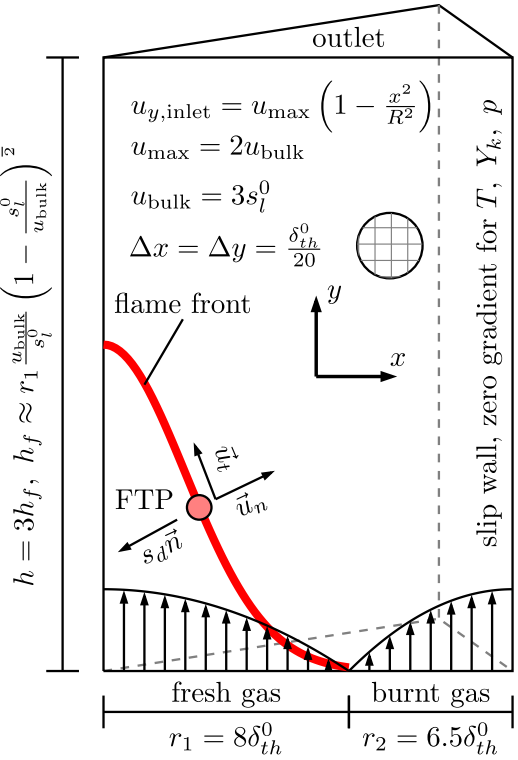

Fig. 3. Computational domain with boundary conditions, dimensions, inlet profiles, grid resolution and the flame front as red line.

During the simulation, FPs are seeded onto the flame surface, shown in Fig. 3 as red circle. As explained in the previous section, material points on the flame surface will generally move from positive stretch regions to negative stretch regions. In this configuration, the tip is negatively stretched and the flame base positively, so that the FP move toward the flame tip. This can also be seen from the components of $\vec{w}$, shown in Fig. 3: in the stationary case, the FP does not move in the normal direction, because for any stationary flame, $\vec{u}_n = -s_d\vec{n}$. The remaining part of $\vec{w}$ is purely tangential, which is the tangential fluid velocity component pointing toward the flame tip, causing a movement of the FP upward.

The simulation is performed with an in-house DNS extension for OpenFOAM [19], which includes detailed chemical reaction mechanisms, in this case a hydrogen mechanism by Li et al. [29], and uses Cantera [30] to compute detailed diffusion coefficients. For more information, see [1,31,32]. The code uses the finite volume method, second order implicit time discretization and fourth order interpolation methods for spatial discretization. The total number of cells in the computational mesh is approximately 350 000.

Fig. 4 shows snapshots from the simulation of the $\phi = 4$ case. At $f = 0$ (left), the flame is in steady state. At the flame tip, the heat release rate $\dot{Q}$ is largest due to the tip being negatively stretched and $Le > 1$. At $f = 0.1/\tau_c$, the flame oscillates as a whole, here depicted at the time with the largest flame height as 3D reconstructed hydrogen mass fraction iso-surface in the foreground and the heat release field at the highest and lowest point during the oscillation in the background (see also the animation in the Supplementary materials). At $f = 1/\tau_c$, the flame front becomes corrugated, as shown by the varying temperature values on the hydrogen iso-surface. For $f = 10/\tau_c$, the flame cannot follow the excitation anymore and the flame resembles the steady-state solution. This behavior is the same for all ϕ.

For the lean case ($\phi = 0.5$) and $f = 1/\tau_c$ in Fig. 5, the wrinkles become more pronounced due to $Le \ll 1$. The flame shows the typical tip opening and quenching regions where the flame is strongly negatively curved. There is even island formation at the flame tip (animation in the Supplementary materials). Yellow spheres depict the FP which track material points on the iso-surface. The black line on the right shows the iso-surface that is tracked by

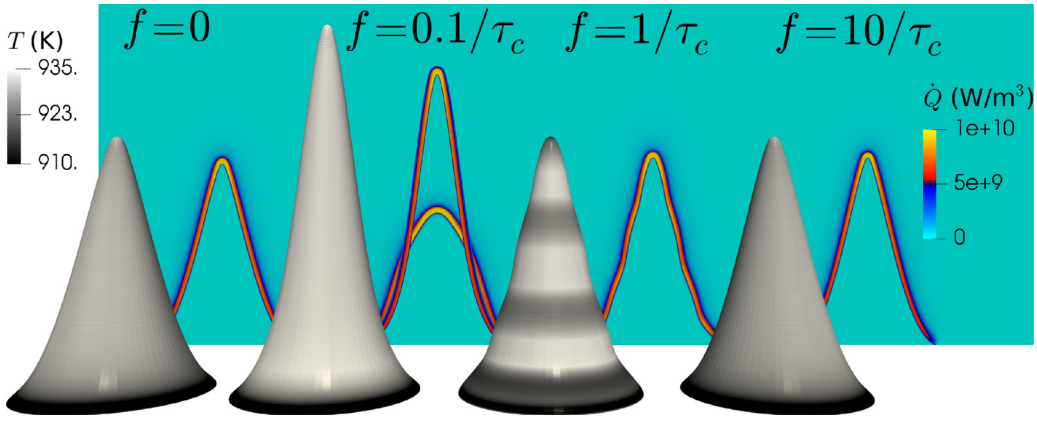

Fig. 4. Cutting plane in the back shows heat release rate $\dot{Q}$ of the $\phi = 4$ flames excited by $f = 0$, $f = 0.1/\tau_c$, $f = 1/\tau_c$ and $f = 10/\tau_c$ (left to right). In the foreground, 3D reconstructed iso-surfaces of $Y_{H_2} = Y_{H_2,iso}$ are depicted colored by temperature. For the $f = 0.1/\tau_c$, heat release contours in the background show the flame in its highest and lowest position during the oscillation. See also the animation in the Supplementary materials.

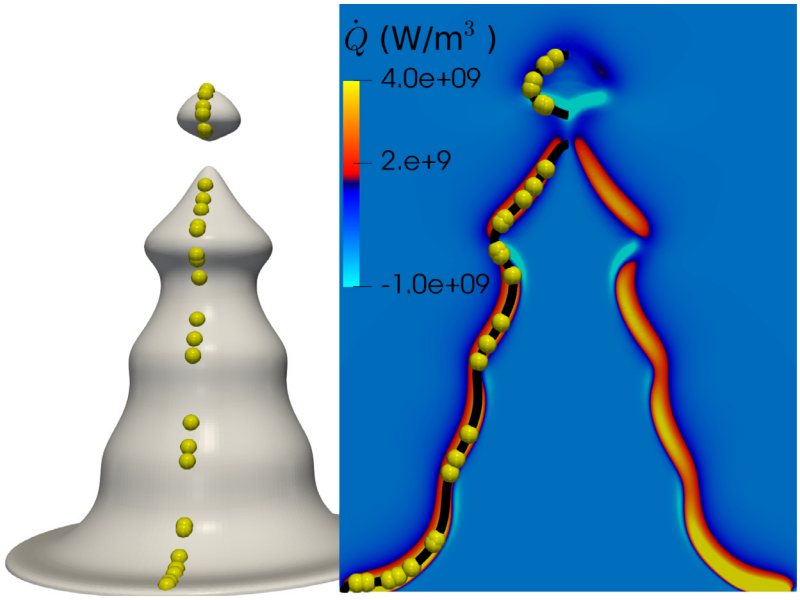

Fig. 5. Case $\phi = 0.5$ and $f = 1/\tau_c$. Left: snapshot of the 3D reconstructed iso-surface of $Y_{H_2} = Y_{H_2,iso}$. Right: Black line shows the $Y_{H_2} = Y_{H_2,iso}$ iso-line. The FP are illustrated by yellow spheres. See also the animation in the Supplementary materials.

the FP, which in this work is always defined as the value of the hydrogen mass fraction $Y_{H_2,iso}$, where the heat release rate has its maximum in an unstretched flame at the same ϕ.

An example of the recorded trajectory of a FP tracking the $Y_{H_2,iso}$ iso-surface is shown in Fig. 6. The sub-figures in the lower right corner show the height (y) of the FP. It is seeded near the root of the flame and moves along the iso-surface to the tip over time. The contribution terms to the mass fraction balance equation of hydrogen $\partial(\rho Y_{H_2})/\partial t + \nabla \cdot (\rho \vec{u} Y_{H_2}) = -\nabla \cdot \vec{j}_{H_2} + \dot{\omega}_{H_2}$ are recorded over time for one material point, where $\nabla \cdot (\rho \vec{u} Y_{H_2})$ is the convective term (conv) and the two terms on the rhs are diffusion (diff) and chemical source term (RR). At steady-state (left of Fig. 6), the flame stretch becomes more negative as the FP approaches the flame tip, leading to a reduced chemical reaction rate until the flame extinguishes because $Le < 1$ at $\phi = 0.5$. Since the

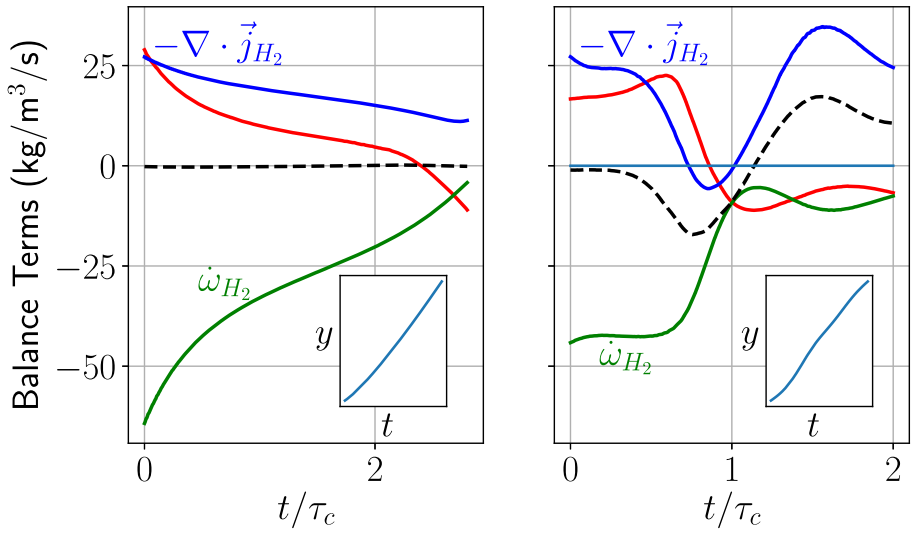

Fig. 6. Time signal on the steady-state flame (left) and $f = 1/\tau_c$ (right) for $\phi = 0.5$. Sub-figures in the lower right corner show the y position of the FP over time, which moves upward from the flame base to the tip. Along its trajectory, the contribution of the chemical source term (▬), (negative) convection (▬) and diffusion (▬) are depicted over time. At steady-state (left), their sum (▬) is zero.

case is in steady-state, the sum of the three contributions is zero. For the oscillating case (right of Fig. 6), the sum of the three contributions is not zero.

Fig. 7 shows time signals of different quantities from a single FP moving along the flame surface of an oscillating H_2/air Bunsen flame at $\phi = 4$, $f = 1/\tau_c$ (third from the left in Fig. 4). Fig. 7a) shows time histories of the flame speed (consumption speed s_c and displacement speed s_d^*) as well as the normalized total flame stretch Ka. Blue dashed lines mark the times where peaks in Ka occur and the red dashed line shows peaks of the flame speed. It can be clearly seen that there is a time delay of about $0.25\tau_c$ to $0.3\tau_c$ between the flame stretch and the response of the flame in terms of s_c and s_d^* while staying in a moderately low stretch regime. In a previous work [1], this analysis was limited to the movement of the flame tip because it is the only point on the flame surface that can be tracked without FP. There, phase shifts between the local flow and the flame movement are of the order of $\mathcal{O}(0.1/\tau_c)$ for rich hydrogen flames, which is consistent with the new findings.

In Fig. 7b), the normalized flame stretch Ka is split into its components $Ka_c = s_d \nabla \cdot \vec{n}\tau_c$ and $Ka_s = \nabla_t \cdot \vec{u}\tau_c$. Peaks of $s_d^* = s_d \rho/\rho_u$ and s_c coincide with peaks of the aerodynamic straining Ka_s in time, while peaks of the flame stretch due to curvature Ka_c are shifted in time. This explains the time delay between the total flame stretch Ka and the flame speed s_d^* and s_c from Fig. 7a). Because the flame's curvature is negative for $t/\tau_c > 1$, the black dashed lines mark the *negative* peaks of Ka_c, although they are strictly speaking half an oscillation period ahead compared to the other dashed lines. s_d^* can be further split into $s_{d,RR}^* = -\dot{\omega}_{\mathrm{H_2}}/(\rho_u |\nabla Y_{\mathrm{H_2}}|)$ and $s_{d,\mathrm{diff}}^* = s_d^* - s_{d,\mathrm{RR}}^*$, where $s_{d,\mathrm{RR}}^*$ accounts for chemical reactions and $s_{d,\mathrm{diff}}^*$ for diffusion. Fig. 7c) shows that the time signal of $s_{d,\mathrm{RR}}^*$ follows most closely the time signal of Ka_s, while $s_{d,\mathrm{diff}}^*$ follows Ka_c (this is also true for the tangential and normal components, not shown here). Because $s_{d,\mathrm{RR}}^*$ is the largest component of s_d^* in this flame case, the time shift of the flame speed in Fig. 7a) can be explained. A similar result is found in Fig. 7d), where time signals of the convective, diffusive and chemical terms of the oxygen mass fraction balance equation are depicted. Here, the convective and chemical term follow the time signal of Ka_s, while the diffusion term follows Ka_c more closely. The diffusion decreases with increasing negative curvature due to the defocusing of diffusive reactant fluxes. Because the reactive term is much larger than the diffusive contribution in this flame at the chosen iso-surface, convection mostly balances the chemical source terms and therefore shares the same time history. In conclusion, the time delay between total stretch rate and observed local flame speed (both in terms of s_c and s_d^*) is caused by a time delay between Ka_s and Ka_c due to the different time scales of diffusion and chemistry. The flame's curvature most strongly affects diffusion while the chemical source term follows the aerodynamic straining, which dominates this flame.

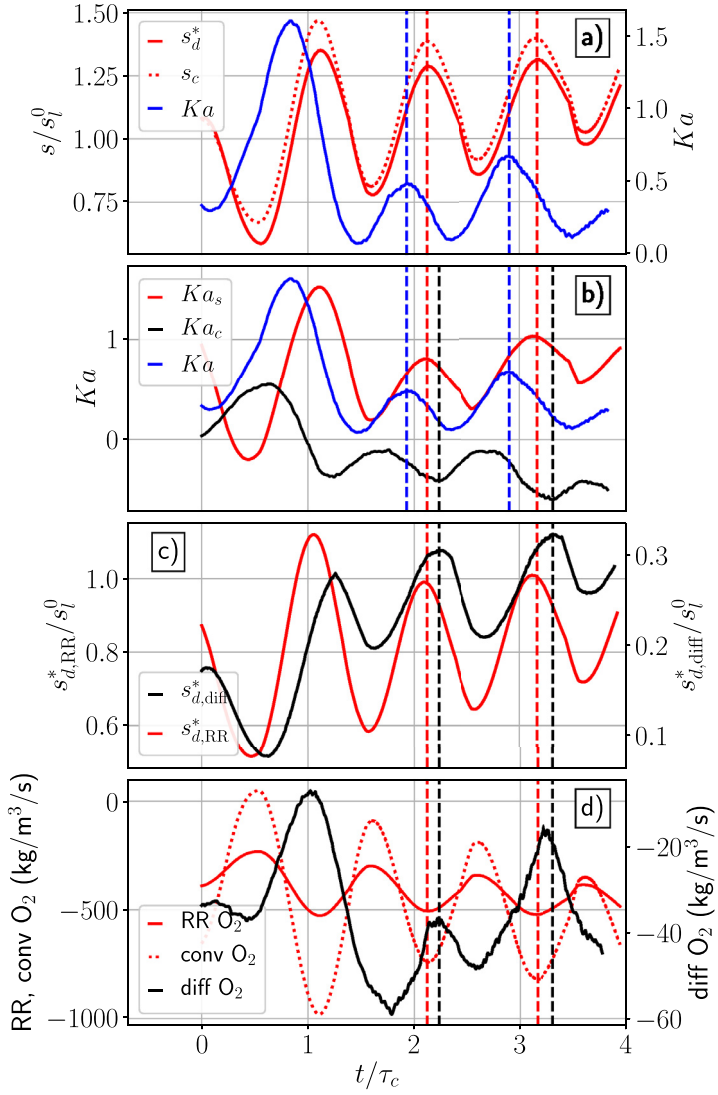

Fig. 7. Time signals of one single FP moving along the flame to the tip for the case $\phi = 4$ and $f = 1/\tau_c$ showing: a) $s_d^* = s_d \frac{\rho}{\rho_u}$, s_c and Ka. b) Total flame stretch and its components. c) Reactive and diffusive contribution to s_d. d) Balance terms for the transport equation of Y_{O_2}.

5. Conclusions

A computationally efficient flame particle tracking (FPT) implementation based on barycentric coordinates is introduced for tracking flame particles (FP) which follow material points on the flame surface. This allows fast interpolation of arbitrary quantities at the FP location and efficient position correction of the FP to the iso-surface. The methodology can also be used on unstructured meshes due to the decomposition of computational cells into tetrahedra. In contrast to previous works, the implementation is fully parallelized and can be used in-situ. An in-situ implementation of consumption speed calculation is included as well, which has not been considered in the previous works related to FP.

DNS of pulsating laminar Bunsen flames, including both positive and negative stretch, serve as validation for the method and demonstrate its ability to gain additional information by evaluating time-resolved data along trajectories of material points on the flame. Tracking material points in this case reveals that flame stretch and flame speed show a time shift of the order of $0.1\tau_c$ for the considered case, so that the history of previously encountered flame stretch determines the current value of flame speed. This is caused by a time delay between the

aerodynamic straining and flame stretch due to curvature. It has been shown that the reason for this delay is that time signals of chemical processes follow the aerodynamic straining while diffusive processes have the same time evolution as the flame's curvature for the investigated H_2/air case with $Le > 1$. Applying the FPT method to turbulent flames has revealed similar time history effects but their quantification is left for future work. These results can help to shed light on what the major influences on the local flame dynamics are and lead to a better understanding of global quantities such as the turbulent flame speed. The presented implementation has been shown to efficiently use the resources of high performance computers and can be applied to large-scale DNS.

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.

Acknowledgments

This work utilized the computational resource ForHLR at KIT funded by the Ministry of Science, Research and the Arts Baden-Württemberg.

References

[1] F. Zhang, T. Zirwes, P. Habisreuther, H. Bockhorn, *Combust. Flame* 175 (2017) 170–179.
[2] S. Chaudhuri, *Proc. Combust. Inst.* 35 (2) (2015) 1305–1312.
[3] S. Pope, *Int. J. Eng. Sci.* 26 (5) (1988) 445–469.
[4] S. Chaudhuri, in: Combustion for Power Generation and Transportation, Springer, 2017, pp. 101–123.
[5] H.A. Uranakara, S. Chaudhuri, H.L. Dave, P.G. Arias, H.G. Im, *Combust. Flame* 163 (2016) 220–240.
[6] H.L. Dave, A. Mohan, S. Chaudhuri, *Combust. Flame* 196 (2018) 386–399.
[7] S. Chaudhuri, *Phys. Rev. E* 91 (2) (2015) 021001.
[8] H.A. Uranakara, S. Chaudhuri, K. Lakshmisha, *Symp. (Int.) Combust.* 36 (2) (2017) 1793–1800.
[9] P.E. Hamlington, R. Darragh, C.A. Briner, C.A. Towery, B.D. Taylor, A.Y. Poludnenko, *Combust. Flame* 186 (2017) 193–207.
[10] C.A. Towery, P.E. Hamlington, X. Zhao, C. Xu, T. Lu, A. Poludnenko, in: Proceedings of the AIAA Scitech 2019 Forum, 2019, p. 1643.
[11] A. Scholtissek, F. Dietzsch, M. Gauding, C. Hasse, *Combust. Flame* 175 (2017) 243–258.
[12] T. Zheng, J. You, Y. Yang, *Phys. Rev. F.* 2 (10) (2017) 103201.
[13] Y. Yang, D. Pullin, I. Bermejo-Moreno, *J. Fluid Mech.* 654 (2010) 233–270.
[14] T. Oster, A. Abdelsamie, M. Motejat, T. Gerrits, C. Rössl, D. Thévenin, H. Theisel, in: Proceedings of the 2018 Computer Graphics Forum, 37, Wiley Online Library, 2018, pp. 358–369.
[15] P. Sripakagorn, S. Mitarai, G. Kosály, H. Pitsch, *J. Fluid Mech.* 518 (2004) 231–259.
[16] Y. Kozak, S. Dammati, L. Bravo, P. Hamlington, A. Poludnenko, *J. Comput. Phys.* (2019) 109054.
[17] Y. Kozak, S.S. Dammati, L.G. Bravo, P.E. Hamlington, A. Poludnenko, in: Proceedings of the AIAA Scitech 2019 Forum, 2019, p. 1642.
[18] K. Hormann, N. Sukumar, *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*, CRC, 2017.
[19] H. Weller, G. Tabor, H. Jasak, C. Fureby, OpenFOAM, openCFD Ltd., 2017, https://openfoam.org.
[20] R. Prud'homme, *Flows of Reactive Fluids*, Springer, 2010.
[21] M. Sahafzadeh, S.B. Dworkin, L.W. Kostiuk, *Combust. Flame* 196 (2018) 237–248.
[22] C. Sung, A. Makino, C.K. Law, *Combust. Flame* 128 (4) (2002) 422–434.
[23] F.N. Egolfopoulos, *Symp. (Int.) Combust.* 25 (1994) 1365–1373. Elsevier
[24] J. Kistler, C. Sung, T. Kreut, C.K. Law, M. Nishioka, *Symp. (Int.) Combust.* 26 (1996) 113–120. Elsevier
[25] F.N. Egolfopoulos, C.S. Campbell, *J. Fluid Mech.* 318 (1996) 1–29.
[26] H. Wang, C.K. Law, T. Lieuwen, *Combust. Flame* 156 (4) (2009) 889–895.
[27] H.G. Im, J.H. Chen, *Proc. Combust. Inst.* 28 (2) (2000) 1833–1840.
[28] C. Sung, C.K. Law, *Combust. Flame* 123 (3) (2000) 375–388.
[29] J. Li, Z. Zhao, A. Kazakov, F.L. Dryer, *Int. J. Chem. Kinet.* 36 (10) (2004) 566–575.
[30] D. Goodwin, H. Moffat, R. Speth, Cantera: An Object-Oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes, 2017. http://www.cantera.org.
[31] T. Zirwes, F. Zhang, P. Habisreuther, M. Hansinger, H. Bockhorn, M. Pfitzner, D. Trimis, *Flow Turbul. Combust.* 104 (2019) 997–1027, doi:10.1007/s10494-019-00081-5.
[32] T. Zirwes, F. Zhang, T. Häber, H. Bockhorn, *Combust. Sci. Technol.* 191 (2019) 178–195.