CONTROLLING SHAPE OF FLAME WITH DIFFERENTIABLE PHYSICS

Laura Leja[1], Karlis Freivalds[2]
[1]University of Latvia, Latvia; [2]Institute of Electronics and Computer Science, Latvia
l13172@students.lu.lv, karlis.freivalds@edi.lv

Abstract. The study focuses on creating symbol-shaped flame designs by optimizing flame dynamics, using machine learning techniques integrated with advanced combustion modelling. This research addresses the limitations of traditional pyrotechnic systems, which lack precise control over flame shapes, by introducing a method that uses differentiable physics simulations for shaping flames. The method uses the optimization framework PhiFlow and includes radiative heat loss and chemical kinetics for improved accuracy. Machine learning is used not only to train and adapt the model but also to determine the physical parameters that correspond to the actual observed behaviour of the flames. However, it is also important to mathematically and analytically evaluate the results obtained to ensure their accuracy and relevance. Results demonstrate the successful formation of custom flame shapes in a 2D simulated environment, with promising alignment to real-world flame stage projector behaviour. This approach enables control over flame shapes and dynamics, expanding creative possibilities in stage design and interactive art while proving valuable for applications requiring detailed control of dynamic systems, such as pyrotechnics, performance art, and educational demonstrations.

Keywords: flame shaping, differentiable physics, machine learning.

Introduction

We suggest a device in which several flame emitting nozzles are placed side by side and carefully chosen fuel amount and timings enable us to create customizable flame shapes. Machine learning incorporated differentiable physics [1] simulations are used for automatically calculating the fuel timings in order to create the required flame shape. This type of simulation could potentially be applied to real-stage flame equipment. Flame throwers create customizable flame shapes, such as company logos, words, symbols, or other designs, which can be showcased during art performances, thus expanding the possibilities of artistic expression. Machine learning embedded in the model helps ensure shape accuracy. At the same time, differentiable physics provides gradient-based training to accurately model and control flame dynamics in real time, allowing fire effects to be created with high accuracy and adaptability. This approach allows for the real-time generation of intricate flame patterns that can be adapted to a wide range of performance scenarios.

Earlier research indicates that employing differential physics can enable real-time control of simulations [2] and animate smoke through control forces [3]. Other studies have illustrated that manipulating fluid flows can produce dynamic shapes [4] while a more recent method demonstrates that differential programming permits real-time 3D object motion control solely by adjusting their positions within the fluid based on observations [5]. In our work, we develop control policies using differentiable physics (DP) simulations. And recent progress in DP has revealed its robust ability to tackle simulated inverse problems and complex control tasks [2; 6-8]. Nonetheless, simulating flames remains particularly challenging because accurately modelling cooling and fuel dispersion is crucial for combustion processes. Although various approaches exist for representing fire and flames [6; 9], not all tools consider ignition and radiation factors, limiting their applicability in realistic scenarios.

Materials and methods

Shaping flame concept and aim. This work builds upon our previous study [10], which established a proof of concept demonstrating that it is possible to form symbolic shapes using hot air flows and flames. To evaluate the concept viability, we began by developing a simplified model of hot air flow. This model provides insights into the dynamics of heated air and serves as a foundation for further exploration of shape formation. It allows us to study how flames can be manipulated to create structured visually impressive creations, paving the way for more advanced simulations that incorporate complex combustion processes. The current stage extends that idea by exploring the feasibility of controlled flame shaping and its potential applications in real-world scenarios. Figure 1 presents a schematic depiction of the envisioned setup, where four flame projectors operate in synchrony to generate a circular flame pattern.

Fig. 1. **Schematic illustration of the envisioned setup, where four flame projectors produce a circle of flames**

In Figure 2, we demonstrate the generation of glowing letters “E-D-I” by applying differentiable physics-based optimization from previous research. The simulation output (left) is compared to the target shape (right), highlighting the model’s ability to reproduce desired forms with high accuracy. The current research extends this idea by exploring the feasibility of controlled flame shaping and its potential applications in real-world settings.

Fig. 2. **Creating glowing letters “E-D-I”:** left side – simulation-generated letters; right side – target letters

Governing equations of combustion and flame dynamics convection simulation and control. To move beyond simple hot air flow and simulate combustion more realistically, we extended our initial simulation model by integrating additional terms specific to fire dynamics. For this purpose, we adopted the combustion modelling framework by [9] implemented using the PhiFlow [11] simulation library, which serves as the basis for constructing our realistic fireball simulation.

The use of the PhiFlow framework allows for flexible and efficient simulation of coupled PDEs with support for differentiable programming, making it well-suited for learning-based control and optimization in combustion scenarios.

The equations presented below represent the fundamental mathematical formulation for modelling flame combustion processes. These partial differential equations describe incompressible fluid flow and incorporate the essential physical principles needed to simulate flame behaviour: mass conservation, momentum exchange, and heat transport.

Mass conservation (continuity equation for incompressible flow):

$$\nabla \cdot u = 0, \tag{1}$$

where $u = (x, y, t)$ – velocity field.

Momentum conservation (Navier–Stokes equation):

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{1}{\rho}\nabla p + \nu \nabla^2 u + g, \tag{2}$$

where ρ – fluid density;
p – pressure;
ν – kinematic viscosity;
g – includes gravity and buoyancy.

Energy conservation (heat equation):

$$\frac{\partial T}{\partial t} + (u \cdot \nabla)T = \alpha \nabla^2 T + Q, \tag{3}$$

where T – temperature;
α is the thermal diffusivity $\alpha = \frac{k}{\rho\, c_p}$;
Q is the heat source due to combustion.

Mass fraction equation (mass fraction transport):

$$\frac{\partial Y}{\partial t} + (u \cdot \nabla)Y = D\nabla^2 Y + \omega, \tag{4}$$

where Y – mass fraction of a species (fuel, etc.),
D – diffusion coefficient;
ω – accounts for chemical reactions.

To enhance the physical realism of the model, a radiative heat transfer term was introduced to account for thermal radiation. The resulting heat equation captures the balance between convective and conductive heat transport, heat generation from combustion, and radiative heat loss, ensuring a more accurate representation of energy flow within the system. Heat change is sum of conduction, combustion and radiation:

$$\rho C_p \left(\frac{\partial T}{\partial t} + u \cdot \nabla T\right) = \lambda \Delta T + q - \sigma_\varepsilon (T^4 - T_a^4), \tag{5}$$

where ρ – fluid density,
C_p – specific heat,
T – temperature,
u – velocity,
λ – thermal conductivity;
q – heat from combustion.

The final term models radiative heat loss using the Stefan–Boltzmann law, where ε is emissivity, and σ_ε is effective radiation coefficient, but T_a is ambient temperature.

For the simulation to remain physically accurate and stable, appropriate boundary and initial conditions were defined alongside the governing equations, playing a key role in maintaining physical and computational consistency. Boundary conditions on walls ($u = 0$) or ($u \cdot n = 0, \nabla \times u = 0$), open boundaries (simulation above and below the domain): e.g., $p = 1013.25$ Pa, $\frac{\partial u}{\partial n} = 0$, initial conditions: $u(x,0) = u_0, T(x,0) = T_0, Y(x,0) = Y_0$.

Results and discussion

Model training and simulation process. Our simulation is created in a two-dimensional rectangular domain with an aspect ratio of 1:2 concerning x and y, respectively, where the lower boundary is lined with 32 nozzles emitting upward fuel streams. A 2D model was chosen to strike a balance between physical realism and computational efficiency. While it simplifies the full complexity of real flame behaviour, it allows for rapid experimentation, intuitive visualization, and effective gradient-based optimization of flame shapes using differentiable physics. The vertical dimension is suitable for the natural growth and development of flame-like formations. Our simulation is performed in a 2D environment where ignited fuel is discharged through the nozzles as controlled vertical jets to create spatial configurations of different shapes and appearances. For the simulation, we use the PhiFlow system [6], which facilitates physics-based modelling and machine learning. This system provides differentiated fluid dynamics, allowing accurate optimisation of flow behaviour based on gradients.

We use the training method from [12] on rigid object manipulation, we extend the same training strategy to flame-based tasks (see Figure 3).

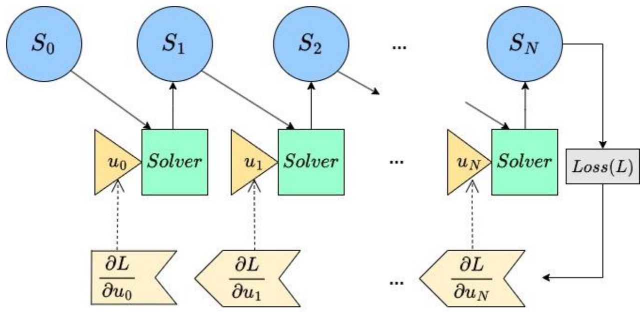

Fig. 3. **Integrated training process. Sarting from the initial state S_0 (temperature and speed):** at each step, the current state S_i and control input u_i are sent into the solver to get the next state; after the final step, the loss is calculated and used to improve the control inputs by sending the error backwards through all steps

The top diagram represents the simulation process, where the PhiFlow solver iteratively updates the system by using the current control inputs u_i and state S_i to determine the next state S_{i+1}.

The solver adjusts temperature and velocity fields according to advection dynamics and incompressibility conditions. The control variables learned nozzle speeds and temperature are optimized to shape the flow. The environmental state tracks temperature and velocity throughout the domain. The loss is calculated as the summed squared error between simulated flame intensity and grayscale target images (see Figure 4). The bottom part of the training scheme shows a gradient-based optimization process using the Adam optimizer [13] with a learning rate of 0.05 over 1000 iterations.

After evaluating the results and analysing the loss function behaviour of the differentiable physics (DP) method, we observe that DP performs very good, demonstrating faster convergence and a smoother loss reduction throughout the training process, see Figure 4.

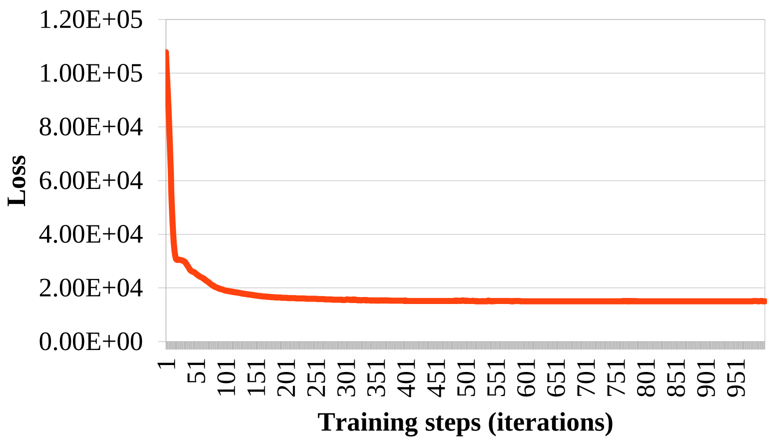

Fig. 4. **Differentiable Physics (DP) training loss curve over training iterations.** The x-axis indicates the number of training steps, but the y-axis shows the loss values

Simulation of real fireball behaviour. We aim to develop a two-dimensional simulation that accurately reproduces the behaviour of flames generated by stage equipment. For model training, slow-motion recordings of actual propane-butane flame devices were captured, clearly documenting the ignition process, flame growth, and cooling phases. Multiple blower setups including single, dual, sequential, diagonal arrangements, and combinations of up to four blowers were systematically evaluated, generating flames exceeding 2.5 meters in height, with recordings taken from 5.40 meters.

To ensure the reliability of flame shape simulation, we developed a model of a single flame projector, aligned with the behaviour of its real-world flame. The simulation is grounded in physically accurate combustion equations, enabling precise replication of flame dynamics. The flame projector was trained to emit fuel in 50 millisecond intervals, capturing the transient characteristics of combustion.

The flame emitter (blowing profile) is modeled as a horizontal line segment emitting fuel upward with learnable velocity $V(x)$, and fuel $F(x)$ profiles along the line are 1D learnable variables, discretized over the simulation grid. Activations occur over $(t_{1\,,}t_2)$ with smooth transitions controlled by learnable slopes: s_1 (activation slope) for how quickly the flame starts at t_1, and s_2 (deactivation slope) for how quickly it fades at t_2:

$$R(t) = \sigma(s_1(t - t_{1\,,}))[1 - \sigma(s_2(t - t_{2\,,}))], \tag{6}$$

Then the blower velocity is $u(x,t) = (0, V(x), R(t)))$ and fuel injection is $F(x)R(t)$.

From this training process, we extracted key physical parameters, which were subsequently applied to test the model's performance across a range of scenarios. As a result, we created a reliable simulation model that closely matches how a real flame projector works and can be used to generate and study different flame shapes.

The following physical parameters were obtained through training to accurately simulate the behaviour of a real flame projector. The radiation threshold was determined to be 1249.76, with a radiation epsilon of 0.72, indicating the sensitivity of the model to heat radiation. The diffuse constant, which influences the spread of the flame, was estimated at 0.00069. The buoyancy factor, reflecting the upward force due to heated air, was set at 0.61, while the velocity decay, determining how quickly the flame movement slows down, was 0.0064. The model also defined a temperature range between –1.68 °C and 28.25 °C, and the fuel emission duration (blowing time) was established as 29.93 milliseconds. These parameters enable realistic and stable flame behaviour within the simulation environment. The trained single-blow simulation is illustrated in Figure 5.

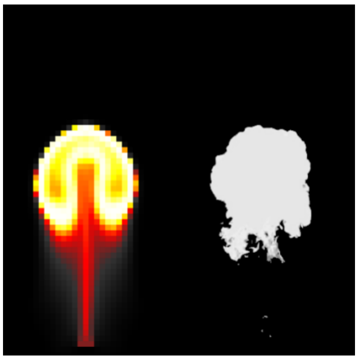

Fig. 5. **Fireball simulation. Left side - trained simulation with one blow (32x64). Right side -real flame frames with specified intensity (grayscale image)**

Our single-flame simulation model was adapted to replicate a two-blow flame projector by reusing the previously trained physical parameters (see Figure 6). The sequence includes 50ms of fuel, a 50ms pause, and a second 50ms burst. The results show a good match between the simulation and the real-world behaviour, with both producing an elongated second flame, suggesting the model is performing accurately.

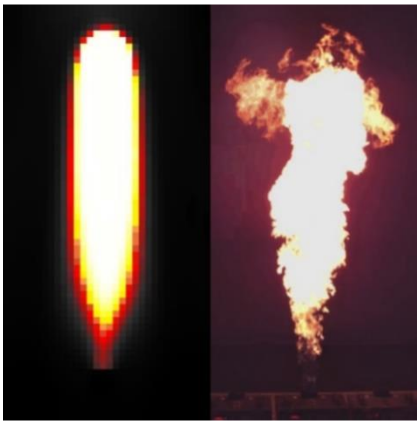

Fig. 6. **Fireball simulation:** left side – trained simulation with two-blow (32x64); right side – real flame frames with specified intensity.

We extended the single-flame simulation model to represent four flame projectors (in Figure 7), each performing a burst simultaneously. The same trained physical parameters were used, with fuel injection lasting 25ms per projector. In the real-world setup, ignition was inconsistent – two of the projectors only ignited after a second spark. The simulation also showed limitations, as the flames merged in the center, drawing in air from the sides. To achieve more realistic results, a transition to 3D simulation is needed.

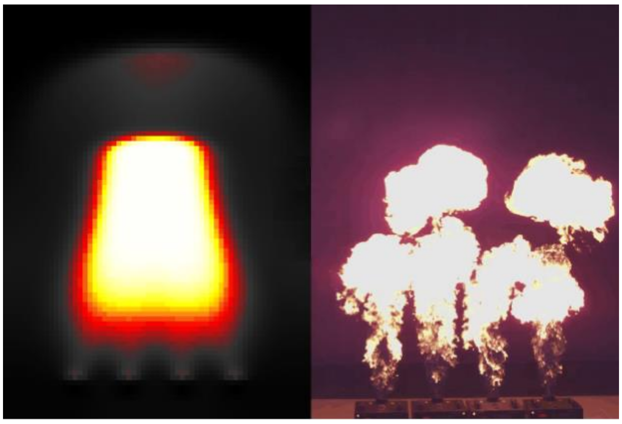

Fig. 7. **4-stage flame simulation:** left side – trained simulation with 4 fire blowers; right side – real flame frames with 4 blowers and a certain intensity

Conclusions

This work demonstrates that a data-driven simulation model trained on single flame bursts can successfully generalize new scenarios, such as handling dual-flame emitters, indicating good model robustness. The 2D simulation framework effectively captures certain structural aspects of flame behaviour and provides a useful platform for initial exploration and optimization. However, the limitations of two-dimensional modelling become evident in more complex configurations – such as with four flame sources – where air flow and oxygen distribution require full spatial representation. To accurately capture such interactions and achieve greater physical fidelity, future work must have transition to 3D simulation environments, enabling more realistic modelling and enhanced control of flame-based phenome. The results demonstrate that it is possible to successfully model the thermal effects and flame plume formation in this way, making it useful for both animations and practical applications. Exploring multiple flame emitters and alternative fuel types, such as alcohol-based burners, offers promising insights for advancing precise flame control in future applications.

Acknowledgements

Authors thank the University of Latvia for the financial support for this article and thus the opportunity to participate in the conference as part of doctoral studies.

Author contributions

Conceptualization, L.L, K.F.; methodology, L.L, K.F.; software, L.L, K.F.; validation, K.F.; formal analysis, L.L writing – original draft preparation, L.L., K.F.; writing – review and editing, L.L., F.F.; visualization, L.L. Both authors have read and agreed to the published version of the manuscript.

References

[1] Thuerey N., Holl P., Mueller M., Schnell P., Trost F., Um K. Physics-based deep learning. arXiv preprint arXiv:2109.05237, 2021, p. 461.
[2] Hu Y., Anderson L., Li T. M., Sun Q., Carr N., Ragan-Kelley J., Durand F. (2019). Difftaichi: Differentiable programming for physical simulation. arXiv preprint arXiv:1910.00935, (p. 20).
[3] Fattal R., Lischinski D. (2004). Target-driven smoke animation. In ACM SIGGRAPH 2004 Papers (pp. 44-448).
[4] Shi L., Yu Y. (2005 July). Taming liquids for rapidly changing targets. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (pp. 229-236).

[5] Freivalds K., Leja L., Teikmanis O. (2024, November). Learning to Move Objects with Fluid Streams in a Differentiable Simulation. In 2024 7th Iberian Robotics Conference (ROBOT) (pp. 1-7). IEEE.
[6] Holl P., Koltun V., Thuerey N. (2020). Learning to control pdes with differentiable physics. arXiv preprint arXiv:2001.07457, (p. 28)
[7] Du T., Wu K., Ma P., Wah S., Spielberg A., Rus D., Matusik W. (2021). Diffpd:
[8] Differentiable projective dynamics. ACM Transactions on Graphics (ToG), 41(2), (pp.1-21.). Ramos B., Trost F., Thuerey N. (2022). Control of two-way coupled fluid systems with differentiable solvers. arXiv preprint arXiv:2206.00342, (p. 24).
[9] Tathawadekar, N. N., Doan, N. A. K., Silva, C. F., Thuerey, N. (2023). Incomplete to complete multiphysics forecasting: a hybrid approach for learning unknown phenomena. Data-Centric Engineering, 4, (p. 27).
[10] Leja L., Teikmanis O., Freivalds K. (2024). Shaping Flames with Differentiable Physics Simulations. Machine Learning and the Physical Sciences" at NeurIPS, 202, (pp. 1-6).
[11] P. Holl, V. Koltun, and N. Thuerey, "Learning to Control PDEs with Differentiable Physics," in Proceedings of the 8th International Conference on Learning Representations (ICLR), 2020, (p. 28).
[12] Teikmanis, O., Leja, L., Freivalds, K. (2023). Applying a Differentiable Physics Simulation to Move Objects with Fluid Streams. In International Workshop on Embedded Digital Intelligence (IWoEDI), (p. 2).
[13] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, (p. 15).