Shaping Flames with Differentiable Physics Simulations

Laura Leja, Oskars Teikmanis, Kārlis Freivalds
Institute of Electronics and Computer Science
Latvia
{laura.leja, oskars.teikmanis, karlis.freivalds}@edi.lv

Abstract

We develop a method for shaping simulated flames into customisable forms, which is a step forward compared to the existing pyrotechnic systems which produce fire balls or fire columns with minimal control over the resulting shape. Our approach integrates differentiable physics with combustion simulations to generate letter-shaped flames. By employing differentiable physics-based training, we successfully produce simulated flame shapes and take initial steps toward practical implementation by comparing the simulation results with real-world flame projectors. The ability to control flame shape would significantly expand the possibilities of stage pyrotechnics and creative applications in performance art.

1 Introduction

Fire effects have long been a central element in the performing arts, delivering a strong visual impact at concerts, artistic performances and other live events. Despite their dramatic appeal, traditional stage flames are constrained in their versatility, typically limited to producing basic shapes controlled by manual or rudimentary systems. These restrictions hinder the creative potential of artists and directors aiming to push the boundaries of visual storytelling. As a result, there is growing demand for advanced fire effects that offer visually impressive, customisable, and interactive designs—such as the concept illustrated in Fig. 1.

Figure 1: Schematic depiction of the envisaged scenario where 4 flame projectors work in unison to produce a circle of flames.

We explore the possibilities of generating flame shapes by optimizing flame projection timings, using machine learning (ML) and flame simulations to control the intricate dynamics of the combustion process. Our method is based on differentiable physics (DP) simulations, which have proven to be a powerful tool for controlling dynamic systems. DP enables gradient-based optimization, allowing the training of neural networks for complex and highly dynamic tasks. It has become particularly influential in fluid flow control, facilitating precise manipulation and optimization of liquid and air flow characteristics. Hu et al. (2019) introduced a differentiable programming language enabling precise, real-time control, and optimization of physical simulations, including fluid dynamics. (Fattal and Lischinski, 2004) introduced a method to control smoke dynamics using guiding forces, allowing animators to create realistic smoke effects efficiently. Shi and Yu (2005) focuses on controlling fluid behaviour to adapt rapidly changing targets shapes while preserving natural motion. A more recent work by (Freivalds et al., 2024) extended the application of DP by developing a method for controlling objects in 3D space through fluid flows. This technique allows for real-time positioning, movement, and simultaneous control of multiple objects, relying solely on object-state observations.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

Despite these advances, many current methods are hard to apply for flame manipulation due to the inherent complexity of combustion processes. The integration of ML into physical simulations has been studied across various domains, but its application to fire simulation research is still relatively new. While advancements like PhiFlow (Holl et al., 2020; Tathawadekar et al., 2023) have improved flame modeling, they neglect crucial factors like cooling during combustion, limiting their ability to accurately simulate fuel dispersion or ignition in the air.

Our contribution includes the development of a simulation method and a simulated model of physical devices that closely replicate the properties of real-world flames. We demonstrate that it is possible to create shapes using hot air in simulation, and we simulate the burning process of a flame to closely match the behavior of observed flame plumes from a stage-flame device.

2 Forced Convection Simulation and Control

To evaluate the proof of concept and assess the feasibility of forming shapes from flames, we first of all developed a simplified hot air model. Our aims is to understand the behavior of hot air flows and their potential to form distinct shapes, providing a foundation for more advanced simulations that incorporate complex combustion dynamics.

Our model is a rectangular domain with an aspect ratio of 1:2 (x:y), where the bottom row consists of 32 (or 64) nozzles releasing hot air, while the y-axis, twice the length, allows space for flame propagation. The experiment is conducted in a 2D simulation, where bottom-positioned nozzles emit controlled vertical jets of hot air to form the desired structure. The hot air simulation uses the PhiFlow framework (Holl et al., 2020), which enables physics-based optimization and machine learning. This framework offers a differentiable flow dynamics simulation based on the Navier-Stokes equations, discretized and solved using finite difference methods.

The training process is illustrated in Fig. 2. Its upper part depicts the simulation process.At each timestep, the PhiFlow-based solver reads the current control variables u_i and the current environment state S_i and produces the next environment state S_{i+1}, $i \in \{0, 1, 2, \ldots$,N$\}$, where N represents the total number of time steps,(120 in our case). The simulation iteratively updates temperature and velocity fields through advection and incompressibility constraints, incorporating dynamic inputs and applying spatial masks to control propagation. The control variables (learnable parameters) contain the velocities and blower temperature at each time step. The environment state contains the velocity and the temperature for each point of the discretized simulation domain. Once the simulation process is completed, the loss function is calculated based on the final state S_N. The loss is calculated as the mean squared error between the simulation results and the target shape in the upper region of the domain (e.g.,Fig. 3a). The lower part of the scheme depicts the training process, in which the loss gradient is backpropagated across all the training steps. Gradients are used to optimise the input variables. We use the Adam optimizer Kingma and Ba (2015) with a learning rate of 0.05 to update the control variables in 1000 training steps.

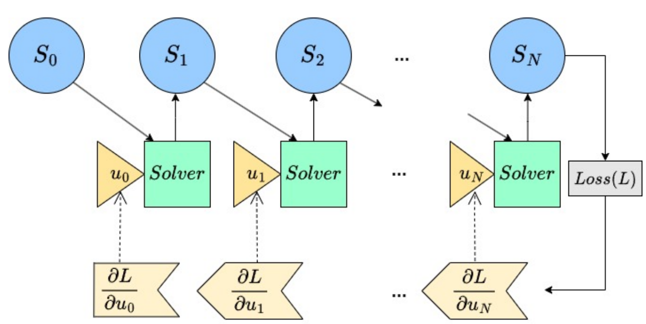

Figure 2: The training process is integrated with the physics simulation over multiple timesteps. Starting with the initial state S_0 (velocity and heat), at each step, the current state S_i and control variable u_i are input into the solver to generate the next state. The loss function is computed at the final state S_N, and its gradient is backpropagated to update the control variables.

(a) (b)

Figure 3: Creating glowing letters 'E-D-I' using differentiable physics-based optimization to replicate the given shape. The figure shows simulation result (a) and the given shape (b).

The proposed method was evaluated across multiple scenarios, including the formation of letters such as "E-D-I" (see Fig. 3), which appear as glowing letter shapes in the upper part of the simulation domain. The results demonstrate that the desired shapes are accurately reproduced. Therefore it is possible to create the desired shape at the top by only controlling the blowers distantly placed at the bottom, at least in a simplified simulation. The supplementary materials include a video demonstrating the formation of the blown "EDI" letters.

3 Real Fireball Replication

The hot air simulation suggests promising potential for achieving the desired shape. However, further refinement is needed to ensure the simulation accurately reflects the behavior of real stage-flame equipment. For this, we analyze the behavior of actual fire blowers and improve the simulation to incorporate realistic combustion dynamics, focusing on gas burning rather than merely modeling hot air flow.

Building on the simulation model initially developed for hot air flow, additional terms were incorporated to develop a realistic fireball combustion model. We adopted the combustion model from (Tathawadekar et al., 2023) as our foundational framework developed on Phi-flow. Considering that this model is not entirely comprehensive, we incorporated an additional term to account for heat release due to thermal radiation. As a result, the governing equation for the combustion process is obtained as follows:

$$\rho C_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot (\lambda \nabla T) + \dot{\omega}_T + \sigma \epsilon \left(T^4 - T_\infty^4 \right)$$

where ρ is the fluid density, C_p is the specific heat capacity, T is the temperature, and $\mathbf{u}$ is the velocity field of the flow. The left-hand side of the equation represents the total heat transport due to convection and temperature changes over time. On the right-hand side, $\nabla \cdot (\lambda \nabla T)$ describes the heat conduction within the medium, with λ representing the thermal conductivity. The term $\dot{\omega}_T$ accounts for the heat released due to combustion, while $\sigma \epsilon \left(T^4 - T_\infty^4 \right)$ represents radiative heat loss, where σ is the Stefan-Boltzmann constant, ϵ is the emissivity, and T_∞ is the ambient temperature. Including radiation and combustion heat release based on chemical kinetics enables a more realistic simulation of flame dynamics, improving accuracy in temperature gradients, flame spreading, and energy dissipation, thus enhancing the model's fidelity compared to the previous hot air flow simulations.

Our goal is to come up with the simulated model of the particular stage-flame device where the simulated flame in 2D replicates the observed flame behavior. To observe the actual evolution of flames and train our own model that represents real fireball replication, we recorded videos of real stage-flame blowers fueled by a propane-butane mixture using slow-motion cameras, see Fig. 4a. These recordings captured the ignition of the dispersed fuel mass, the flame's expansion, and its subsequent cooling. A wide range of configurations was systematically tested, including single and up to four blowers arranged in setups such as single-acting, double-acting, sequential, diagonal configurations, etc. The expelled fireballs attained heights slightly exceeding 2.5 meters, and the camera was positioned at a distance of 5.40 meters.

(a) (b) (c)

Figure 4: Matching simulation to a real fire blower; (a) a frame from real stage-flame equipment video where four blowers are activated for a short time interval one-by-one; (b) one fireball cut out; (c) simulated fireball to match the real one.

Setting up a simulation environment the flame blower of the device is modeled as a small horizontal line segment in the simulation domain emitting a controllable amount of fuel with a controllable speed vertically upwards. The velocity profile $V(x)$ and fuel profile $F(x)$ of the emitting line are learnable variables with 1D shape when discretized with the simulation grid resolution. The time-activation of the device is described as an interval (t_1, t_2) where the device starts emitting flames at t_1 and ends at t_2. To enable differentiable learning of t_1 and t_2, we smooth out the time-activation using activation slope s_1 and deactivation slope s_2, which also are learnable, so the blowing rate at a given time t is given by:

$$R(t) = \sigma(s_1(t - t_1))(1 - \sigma(s_2(t - t_2)))$$

where σ is the sigmoid function. The blower velocity and fuel at location x and time t are $V(x)R(t)$ and $F(x)R(t)$, respectively.

In addition to the device parameters, the key combustion parameters – thermal flame emissivity ϵ, diffusion rate λ, buoyancy factor, and camera sensitivity are also learnable. The captured videos were processed, and we cut them into frames and cropped the specific region surrounding the target fireball. To prepare these target images for training, we created segmentation masks incorporating grayscale intensity to enhance the training process.

The training was done to identify all parameters the same way as in the hot air. But in this case, observed radiation in each frame is compared to the results produced by the simulation-trained model in the whole frame, not only the domain's upper part. The loss is the mean squared error between the observed radiation and the filmed one and is summed for all video frames. In Fig. 4 we can see one frame of the real fireball and the corresponding one from the simulation. We can see a good match and, overall, the results demonstrate good correspondence of the simulation to real-world flame dynamics. It should be noted that our combustion equation enables the simulation of parameters to develop a model applicable to general scenarios without capturing the intricate details of the flames. This is achieved by the model producing images that illustrate the general formation of an average fireball. We have developed simulation model of a real stage-flame device.

We have compared our Differentiable Physics based training (DP) with the Simultaneous Perturbation method (SPSA) (Spall, 1992) which uses gradient approximation from function values at carefully chosen points, the training loss is depicted in Fig. 5. DP performs significantly better showing faster convergence and smoother loss reduction while SPSA shows slower convergence and more fluctuations.

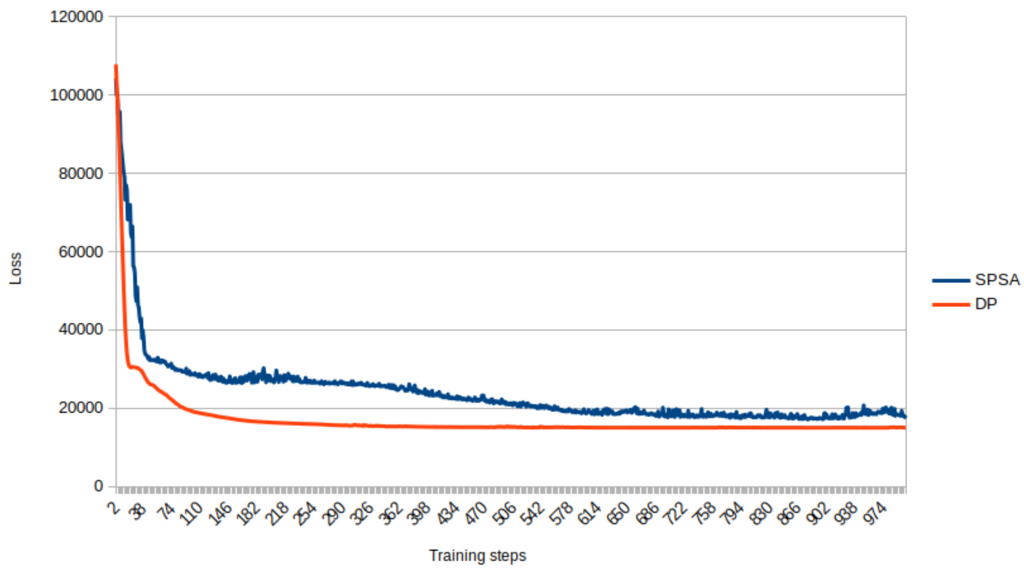

Figure 5: Comparison of Differentiable Physics (DP) and Simultaneos Perturbation (SPSA) based training. The y-axis represents loss, while the x-axis shows training steps.

4 Conclusion

We have developed a tage-flame simulation technology that models flame behavior based on real-world parameters and properties. Through differentiable physics-based training, we demonstrated the ability to create glowing letters of the desired shape in simulation. We have performed steps towards implementing the method on real flame blowers by matching the simulation to a real device. To simulate realistic flame shapes, we included terms for heat release and radiation effects. These modifications improved the accuracy of both the visual and physical representation of fire behavior.

The present work is limited to 2D simulations and, while the results show reasonably good correspondence to the 2D projections of the observed 3D flames, flame dynamics only partially correspond to real behavior. To achieve higher fidelity, switching to the more realistic 3D modeling is necessary. Looking ahead, the integration of multiple flame sources (as shown schematically in Fig. 1) and the exploration of alternative fuel types, such as alcohol-based blowers, will further enhance our understanding of flame dynamics and allow for even more refined control over fire properties. This ongoing research promises to unlock new possibilities for creative expression in a variety of fields, from stage effects to interactive art.

References

R. Fattal and D. Lischinski. Target-driven smoke animation. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, page 441–448, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 9781450378239. doi: 10.1145/1186562.1015743. URL `https://doi.org/10.1145/1186562.1015743`.

K. Freivalds, L. Leja, and O. Teikmanis. Learning to move objects with fluid streams in a differentiable simulation. *arXiv preprint arXiv:2404.18181*, 2024.

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.

Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

L. Shi and Y. Yu. Taming liquids for rapidly changing targets. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 229–236, 2005.

J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

N. N. Tathawadekar, N. A. K. Doan, C. F. Silva, and N. Thuerey. Incomplete to complete multiphysics forecasting: a hybrid approach for learning unknown phenomena. *Data-Centric Engineering*, 4: e27, 2023.