Avengers: Endgame, A new approach for combustion simulations

Gerardo Aguilera
gaguilera@wetafx.co.nz
Weta Digital
Wellington, New Zealand

John Johansson
jjohansson@wetafx.co.nz
Weta Digital
Wellington, New Zealand

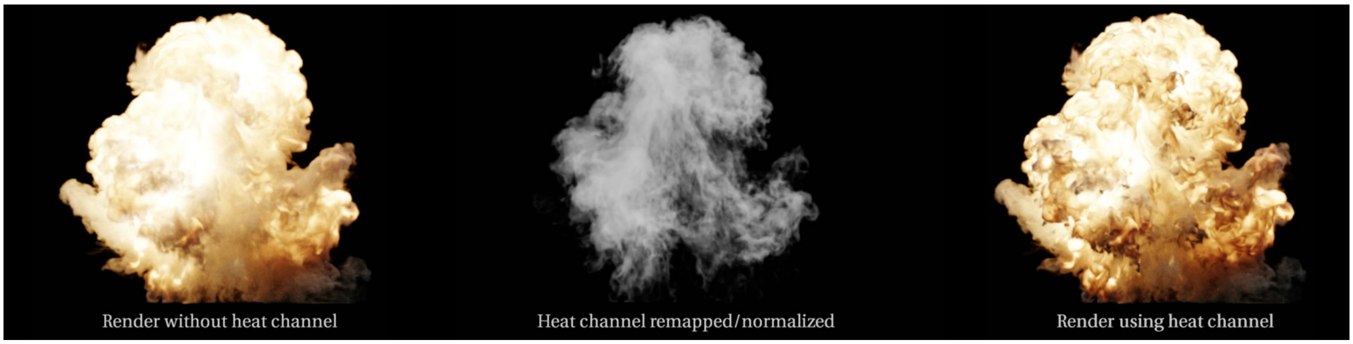

Figure 1: Explosion render demonstrating the result of utilizing the heat channel.

ABSTRACT

In Avengers: Endgame we wanted to improve the physical accuracy of our simulations and level of artistic control for the look of our combustion simulations, i.e. explosions, grenades and fire. Our in-house Simulation RnD department developed an improved combustion solver, internally referred as Combustion 2. This paper will focus on some of the technical aspects of the fluid solver, however the presentation will pivot more towards the production work related to its adoption on Avengers: Endgame.

CCS CONCEPTS

• **Computing methodologies** → *Simulation theory*;

KEYWORDS

FX, explosions, volumetric, volumes, combustion, fire, simulation

ACM Reference format:
Gerardo Aguilera and John Johansson. 2019. Avengers: Endgame, A new approach for combustion simulations. In *Proceedings of SIGGRAPH '19 Talks, Los Angeles, CA, USA, July 28 - August 01, 2019,* 2 pages.
https://doi.org/10.1145/3306307.3328203

SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328203

1 INTRODUCTION

The Combustion 2 solver is implemented in our proprietary software Synapse. It is a component of a modular gas solver framework. The framework solves the Navier-Stokes equation for gas using Eulerian operator splitting with pressure projection and semi-Lagrangian integration for the self-advection term. These are all standard algorithms for fast but approximate simulation of gaseous phenomena. We are able to achieve physically plausible simulations with extra added controls in the solver for artistic manipulation.

The new Combustion 2 algorithm models the chemical process of fuel and oxidizer reacting under high temperature to release energy, which leads to increased temperature and the composition of new output products. Our previous combustion solver had three main scalar quantities to represent the process: "`temperature`", "`soot`" and "`fuel`". The new model extends this with three extra quantities: "`oxidizer`", "`heat`" and "`restProduct`".

2 PRODUCTION

In Avengers: Endgame we used the new Combustion 2 solver for all the explosion and grenades that we see throughout our sequence of the film. In addition, we worked with our Shaders department to modify the existing explosion palette to be able to utilize the "`heat`" quantity to bring out more detail in the higher temperature areas of our renders, resulting in more detailed flames.

Combustion 2 introduces new features for how we approach our volumetric simulations, making us think in a more physically accurate way - having to consider how fuel will be burnt, utilizing the oxidizer in the solve and how efficient the combustion will be. It required a slight change on our workflows and "`re-training`" our TDs to use Combustion 2 vs Combustion 1, since the new solver handles fuel differently and incorporates extra controls such as "`combustion efficiency`" that we previously did not have. The amount of generated temperature and the output products are determined by the combustion efficiency. This is a simplification of the stoichiometric relation between the fuel and oxidizer. The efficiency is dependent on the ratio of oxidizer and fuel. To allow maximum user control, this efficiency mapping function is implemented as a ramp control in our Synapse interface for the artists. An excess of fuel would lead to an incomplete combustion, with a low temperature increase and lots of soot. Excess oxidizer would result in only part of it reacting with the smaller amount of fuel. The amount of soot generated is based on the consumed fuel and the combustion efficiency. If there is high combustion efficiency, there will be less soot and the fuel will instead become restProduct. A certain amount of the consumed oxidiser will also become restProduct.

The quantities "`fuel`", "`soot`", "`restProduct`" and "oxidizer" are represented as volume fractions. These volume fractions are normalised at each simulation step to satisfy the condition: fuel + oxidizer + soot + restProduct = 1.0

"`fuel`", "`soot`" and "`restProduct`" are stored as volume channels and advected with the velocity, the "oxidizer" volume fraction is computed when needed. The reason to not store the oxidizer volume fraction is that it is assumed to exist everywhere in the ambient regions of the simulation. That makes it impractical to track as an advected scalar quantity. Combustion 2 uses volume fractions to allow tracking the concentration of different materials and varying the efficiency and output of the combustion based on the mix of these materials. Without it the combustion would only be dependent on the amount of fuel and temperature.

One of the big additions to Combustion 2 is the output quantity "`heat`". The "`heat`" quantity represents the energy released in the combustion process. Heat is an auxiliary output of the combustion process and is used to mask emission regions in the explosion palette. It gives an idea of the regions that are currently burning. The heat is advected by the velocity, but is not intended to persist and is decayed over a few simulation steps. Thanks to the added controls to the solver, the TDs are able to manipulate the output of the heat data. Furthermore, we have collaborated with the Shaders department to utilize "`heat`" in our palette. In our palette for explosion and fire, emission is defined by a blackbody function. The palette was updated to read an extra quantity, heat. Before sending this information to the palette, we can remap how much of the heat value we want to use, with the goal to focus the data more in the higher temperature areas. The palette uses this information in two ways. First, the data is used as a volume mask to define what is flame and smoke. With this volume mask we can give these two regions different treatments. Then for areas defined as smoke we use "`soot`" data for smoke density and for areas defined as flame we use "`heat`" data for flame density. This gives us more detail in flame regions because "`heat`" data contains sharper detail relative to "`soot`" data. The result in the renders is that we get "`crispier`" and higher detailed fire and explosions. This is why it is favorable to isolate the data of the "`heat`" in the higher temperature areas.

3 FUTURE WORK

The Simulation RnD team is now working on incorporating the model into our new simulation platform Loki. Loki is built as a standalone interface that is not linked to Maya (which Synapse is).

Figure 2: Explosion render using heat channel.

This gives TDs the power to utilize the gas solver and Combustion 2 in their preferred package such as Maya or Houdini. The FX department at Weta Digital has now moved almost all of our workflows into Houdini. Having Loki in Houdini gives us the advantage of combining the power of our in-house solvers with the procedural power of Houdini.

ACKNOWLEDGMENTS

We like to acknowledge the FX team, Simulation RnD and Shaders in Weta Digital for their work, development, testing and contributions to the solver.