TECHNISCHE UNIVERSITÄT DRESDEN

Title of Bachelor thesis

Bachelor-Arbeit
zur Erlangung des Hochschulgrades
Bachelor of Science
im Bachelor-Studiengang Physik

vorgelegt von

EDGAR MARQUARDT
geboren am 07.09.2002 in Leipzig

Institut für Kern- und Teilchenphysik
Fakultät Physik
Bereich Mathematik und Naturwissenschaften
Technische Universität Dresden
2025

Eingereicht am 05. Januar 2026

1. Gutachter: Prof. Dr. Ulrich Schramm
2. Gutachter: Prof. Dr. Thomas Cowan

Summary

Abstract
English:

Abstract
Deutsch

Contents

Introduction

1 Laser wakefield acceleration (LWFA)

2 The problem of Dephasing

3 The Axiparabola

In 2019 Smartsev et al proposed and tested a new optical device called the axiparabola [2]. It is a mirror shaped similarly to an off-axis parabola

$$s(r) = a_2 \cdot r^2 + a_4 \cdot r^4 + a_6 \cdot r^6 + O(r^8)[2] \tag{3.1}$$

4 Spatio-temporal control

While The axiparabola allows for spacial control over where the focus point moves it can not steer, when it arrives at the place. However, because the focus point moves as $z = z(r)$ this behavior can be modified using a radial group delay echelon. By delaying the laser pulse depending on on r by $\Delta t = \tau_D(r)$ it is possible to adjust $z = z(r(t)) = z(t)$ as needed by setting the function $\tau_D(r)$ accordingly.

5 Simulation tools

5.1 PIConGPU

[1]

5.2 Lasy

Lasy is a python library designed to simulate laser pulses on basis of the axiprop library. It uses an angular spectrum propagator and applies optical elements by adding a phase to a semi-Fourier-space.

New Simulations

1 generating the pulse

The Lasy library offers some tools to generate, modify and propagate a laser pulse as a simulation.

1.1 my additions

```
print
```

2 saving the pulse

The goal of this section was to save the laser pulse in such a way that it could be fed as an incident field into PIConGPU simulations. This would happen via the file format openPMD and the pulse profile FromOpenPMDPulse in PIConGPU.

2.1 calculating the full field

Because the laser pulse is in its lasy representation only the complex envelope is directly available. Lasy does offer a function to help here:

```
field = lasy.utils.laser_utils.get_full_field(laser)
```

This even offers the option to specify, how many time points to sample. That is neccessary to be able to resolve the wavelength of the laser properly. However, whatever you do, this function will only ever give you a 2D cut though the field. We want to have the full 3D field for our simulations in PIConGPU. Therefore I wrote a new function, based on the function that lasy provides:

```
field = full_field.get_full_field(laser)
```

It offers a range of options the lasy function does not have. First of all it will generate a 3D field if the laser field is already 3D. Secondly one can specify a number of transversal points beyond which the field will be cut. This is useful when a laser pulse is being focused: we are of course only interested in the focal region, there is no need to calculate out to the edge of the focusing mirror. It is also possible to force the time step to a specific value, if, for example, you want your wavelength to be resolved by a specific number of points or if you need a resolution that works with your PIC simulation.

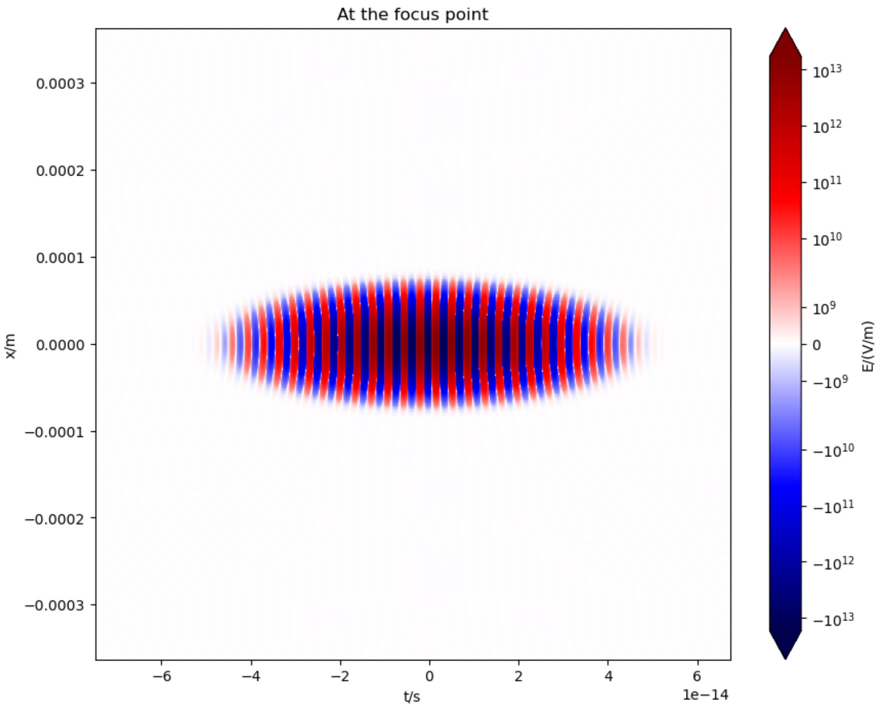

Figure 2.1: Full electric field of a gaussian laser pulse at its focus

This function now enables another function I wrote:

```
full_field.show_field(laser)
```

This function first generates the full field (or a region of it) and displays it in a sym-log plot using `matplotlib`. An example of this can be seen in figure 2.1.

2.2 PIConGPU friendly openPMD

Lasy offers a function in its laser class to write an openPMD file already:

```
laser.write_to_file()
```

This function writes a file containing the envelope data that lasy itself uses. PIConGPU can not work with this so I wrote my own function:

```
full_field.write_to_openpmd_file(directory, filename, array, extent)
```

This function saves the content of the given array of field values to the file at the location directory/filename. It also adds all the metadata PIConGPU needs. Of note may be, that the field is stored in single precision float values to save some memory space. This is especially important when loading the file onto the GPU memory for a simulation. To further save space this function therefore also offers the option to just save every n-th field value into the file. The use of this function may be considered somewhat cumbersome because the array containing the field values feeds to be calculated first so I wrote a function combining the `get_full_field` and the `write_to_openpmd_file` functions:

```
full_field.laser_to_openPMD(laser, filename)
```

It is relatively simple: the field of `laser` is saved to `filename`. However, there are a lot of additional options. These are all listed in the documentation in the appendix 1.1. With the `show=True` option the saved field like in figure 2.2 is displayed. Importantly, this function is

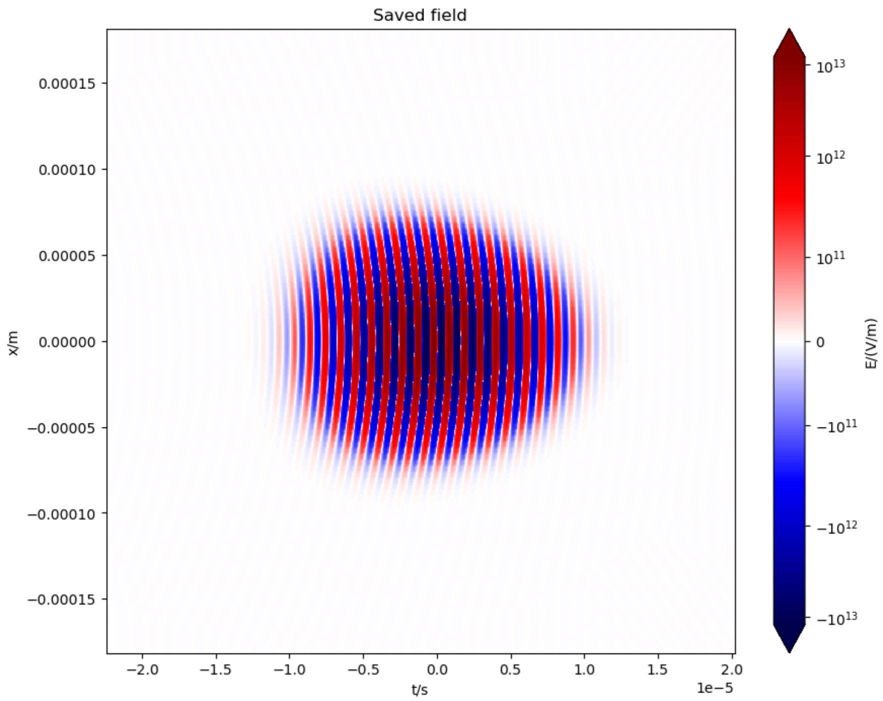

Figure 2.2: The field that has been saved as an openPMD file, one Rayleigh length before the focus

also able to generate and then save the field of a laser that has been simulated just radially.

2.3 PIConGPU simulation

Courant-Friedrich-Levy condition

2.4 displaying the openPMD data

3 the axiparabola laser

3.1 RGD

shape as expected?

3.2 axiparabola

3.3 RGD + axiparabola

all the tests

4 Zwischenfazit

Summary and Outlook

Bibliography

[1] M. Bussmann, H. Burau, T. E. Cowan, A. Debus, A. Huebl, G. Juckeland, T. Kluge, W. E. Nagel, R. Pausch, F. Schmitt, U. Schramm, J. Schuchart, and R. Widera. Radiative signatures of the relativistic kelvin-helmholtz instability. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 5:1–5:12, New York, NY, USA, 2013. ACM.

[2] Slava Smartsev, Clement Caizergues, Kosta Oubrerie, Julien Gautier, Jean-Philippe Goddet, Amar Tafzi, Kim Ta Phuoc, Victor Malka, and Cedric Thaury. Axiparabola: a long-focal-depth, high-resolution mirror for broadband high-intensity lasers. *Optics Letters*, 44, 2019.

Appendix

1 Documentation of my code

2 Code

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit im Rahmen der Betreuung am Institut für ??? Physik ohne unzulässige Hilfe Dritter verfasst und alle Quellen als solche gekennzeichnet habe.

Vorname Nachname
Dresden, Monat 2019