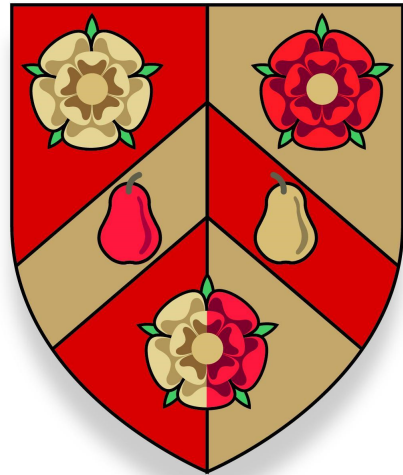


Quantitative optical probing of plasma accelerators

Muhammad Firmansyah Kasim

Wolfson College, Oxford



Thesis submitted in fulfilment of the requirements for the degree of Doctor
of Philosophy at the University of Oxford

Trinity Term, 2017

Abstract

Four novel methods to diagnose plasma wakefield accelerators using optical probes are presented in this thesis. The first method involves sending an optical probe pulse to cross the wakefield at an oblique angle of incidence. The wakefield then imprints a phase modulation onto the probe which is read using a spectral interferometry technique. At first, the method was developed analytically and verified against multi-dimensional Particle-in-Cell (PIC) simulations. These allowed the relation between the phase modulation of the probe and the electron density in the wakefield to be extracted. An experiment employing this technique was also performed and modulations with similar wavelength to those expected in the wakefield accelerator were detected.

The second method — which I have named three-dimensional (3D) spectrometer for brevity — is based on the concept of compressed sensing. It involves the retrieval of multiple two-dimensional (2D) spectral profiles just from a single 2D image captured by a planar detector. Numerical tests show that it can retrieve up to ten sets of 2D spectral profile just from a single image. The retrieved signal is also robust for further post-processing.

The third method is based on a numerical technique to retrieve information from shadowgraphy or proton radiography. The information is the self-generated magnetic and/or electric fields for proton radiography and the variation of refractive index for shadowgraphy. The technique is adapted from a computational graphics algorithm. Numerical simulations show that the retrieved information is accurate with an error of 10% using the method, even if caustics appear. The algorithm is also applied to retrieve the modulation of the refractive index from a real experimental result of plasma wakefield using shadowgraphy.

The fourth and the final method is a novel optimisation algorithm and software based on neuromorphic computing to optimise systems via simulations. The software was employed to optimise the performance of a laser-driven plasma wakefield system by testing it using PIC simulations on a computer cluster. It is demonstrated that by running it and letting it learn laser-driven plasma wakefield parameters for a number of days, the software can find optimal parameters in a laser-plasma system without explicitly being taught.

For my father and my late mother, for their love and continuous support.

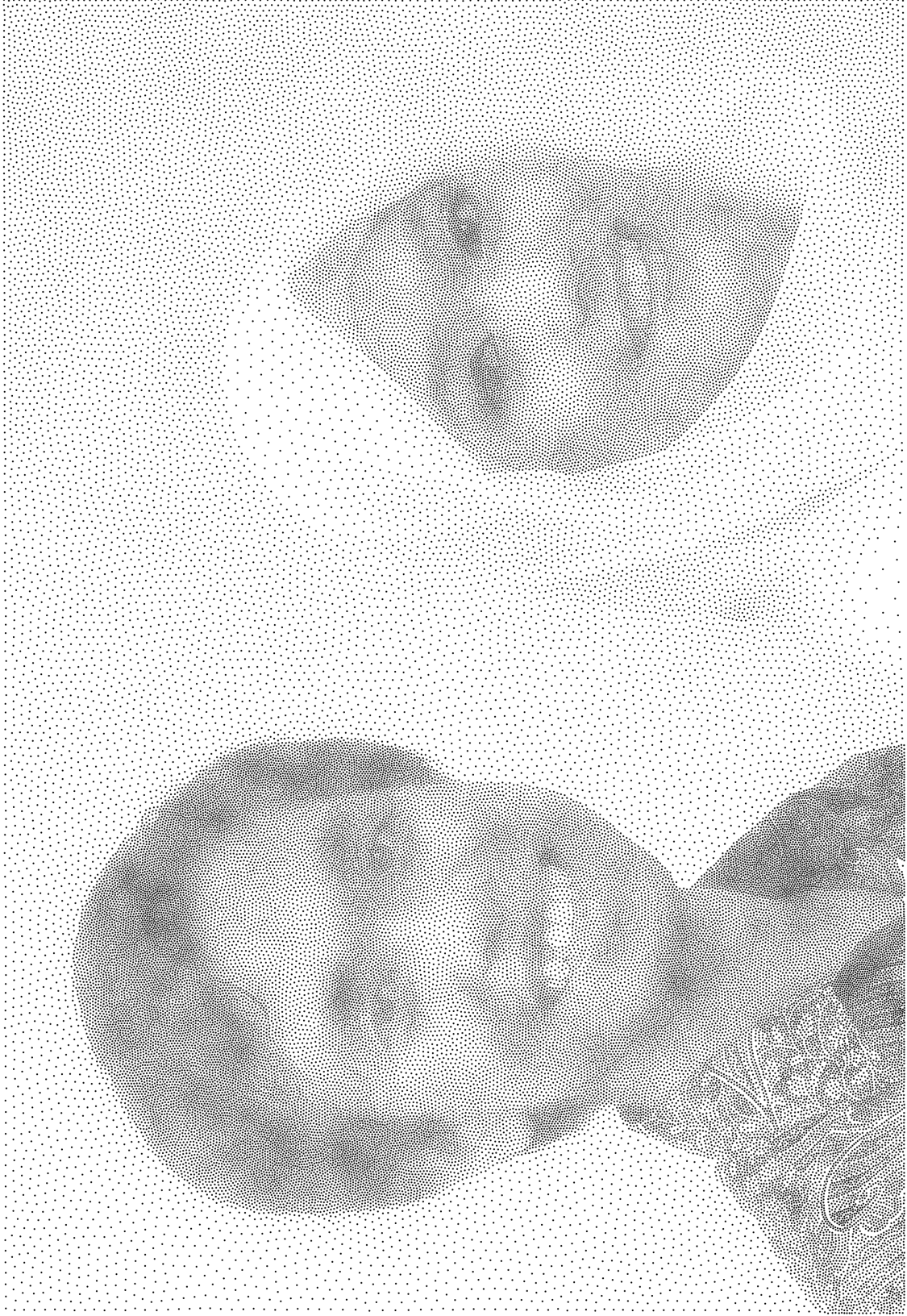


Figure 1: A stippled picture produced as a side product of my work in chapter 6.

Acknowledgements

First, I would like to thank my primary supervisor, Prof. Peter A. Norreys for supervising and giving direction to me whenever I needed advice on plasma physics. Also thank to my co-supervisor, Prof. Philip N. Burrows for his time making sure everything went well for me. I would give the acknowledgements to Dr. Raoul Trines and Dr. James Holloway who always helped me when I had any questions about plasma wakefield accelerators or physics in general. I would also thank the AWAKE collaboration for their helpful feedback to our project: Prof. Allen Caldwell, Prof. Matthew Wing, Dr. Patric Muggli, and Edda Gschwendtner.

Special thanks to those who were involved in the TA2 experiment. Luke Ceurvorst for helping me realise the experiment and giving many feedbacks in our weekly group meetings. Kevin Glize for his amazing experimental skill and his cheerful stories. Nicolas Bourgeois and Dan Symes who gave amazing support from STFC to our experiment. James Sadler whose enthusiasm always made me want to work harder. Naren Ratan for his continuous curiosity that helped me understand the intricacies of the topic. Fearghus Keeble, Alex Savin, and Aimee Ross for their incredible support and insights during the experiment.

I would like to give acknowledgements the people behind the administration who helped me a lot during my DPhil degree: Sue Geddes, Kim Proudfoot, Francesca Oliver, and Jennifer Matthews.

Finally, I would like to thank my parents, my brothers, my sister for their support during my DPhil. Separation by thousands of miles did not stop them from keeping in touch with me. Special thanks to my wife, Ayesha Chairannisa, who always cheers my days during everyday ups and downs. I cannot be where I am now without all of them.

Contents

Introduction	1
1 Theory of plasma wakefield	5
1.1 History	5
1.2 Plasma frequency	7
1.3 Light in plasma	9
1.4 Ponderomotive force	12
1.4.1 Longitudinal oscillating force	12
1.4.2 Electromagnetic wave	14
1.5 Linear plasma wakefield	16
1.6 Dephasing length	17
1.7 Pulse depletion	17
1.8 Plasma channel and relativistic self focusing	18
1.9 Modulational instability	22
1.10 Cluster plasma model	24
1.10.1 Dispersion relation in cluster plasma	25
1.10.2 Ponderomotive force on cluster plasma	27
1.10.3 Plasma wakefield in cluster plasma	28
2 Photon acceleration simulations	29
2.1 Theory of photon acceleration	30
2.2 Simulation method	31
2.2.1 Particle-in-Cell	31

2.2.2	Parameters	33
2.2.3	Obtaining the local frequency	34
2.3	Simulation results	35
2.4	Analysis and limitations	36
2.4.1	Photon Trapping	37
2.4.2	Diffraction	39
2.5	Conclusion	39
3	Oblique crossing angle probe	40
3.1	Modified Abel transformation	42
3.2	3D simulations of measurement	44
3.3	Simulation results and analysis	46
3.4	Limitations	49
3.4.1	Diffraction	49
3.4.2	Large crossing angle	50
3.4.3	Method to acquire the phase modulation	51
3.5	Fast moving object tomography	52
3.6	Conclusions	53
4	Experiment on plasma wakefield diagnostic	54
4.1	Single-shot supercontinuum spectral interferometry	55
4.2	Experimental design	57
4.2.1	Schematic in the target chamber	57
4.2.2	Schematic on the probe table	60
4.3	Experimental setup	63
4.3.1	Probe table	64
4.3.2	Target chamber	65
4.3.3	West table	65
4.3.4	South table	65
4.3.5	Diagnostics	66
4.4	Experimental results	67
4.4.1	Calibrations	69

4.4.2	Helium and nitrogen gas	72
4.4.3	Methane gas	74
4.5	Conclusion	78
5	3D spectrometer	79
5.1	Compressed sensing	81
5.2	ANOVA: A differentiable programming	84
5.3	3D Spectrometer	86
5.4	Numerical tests	90
5.5	Conclusion	93
6	Quantitative shadowgraphy and proton radiography	95
6.1	The theory of shadowgraphy and proton radiography	97
6.1.1	Deflections with electric and/or magnetic fields	98
6.1.2	Deflection with varied refractive indices	99
6.1.3	Intensity on screen	101
6.2	Computational graphics tools	104
6.2.1	Voronoi and power diagram	104
6.2.2	Lloyd's algorithm	105
6.3	Quantitative shadowgraphy algorithms	106
6.3.1	Forward transform	106
6.3.2	Inverse transform with simple gradient descent	108
6.3.3	Inverse transform with power diagram	113
6.3.4	Implementation	118
6.4	Benchmark with simulations	121
6.4.1	Gaussian magnetic field proton radiography	121
6.4.2	Arbitrary shape	122
6.5	Test with experimental results	125
6.6	Conclusions	126
7	Optimisations in plasma physics	128
7.1	Infinite dimensional optimisation	130

7.1.1	Simultaneous Optimistic Optimisation	131
7.1.2	Multi-level SOO	132
7.1.3	Technical implementation details	136
7.1.4	Analysis	138
7.1.5	Numerical experiments	142
7.2	Optimisation via simulations	144
7.2.1	Software architecture	145
7.2.2	Inputs from users	147
7.3	Optimising laser plasma wakefield	152
7.3.1	Longitudinally uniform density	152
7.3.2	Varied plasma density	155
7.4	Conclusions	158
	Summary and future works	159
	A Derivations	162
A.1	Oblique angle Abel transformation and inverse	162
A.1.1	Forward transform	162
A.1.2	Inverse transform	165
A.2	Oblique angle tomography	167
A.2.1	Forward transformation	168
A.2.2	Inverse transformation	171
	Bibliography	172

List of Figures

1	Stippled picture	ii
1.1	Gaussian beam.	20
1.2	Illustration of cluster plasma.	25
2.1	An illustration of the simulation set up	33
2.2	Comparison between the retrieved and calculated profiles	36
2.3	Relative error vs propagation distance plot	37
2.4	Laser intensity profile while propagating in the plasma	38
3.1	An illustration of a probe crossing wakefield in an oblique angle	41
3.2	An illustration of cases where Abel transformation does and does not apply	42
3.3	An illustration of the 3D simulation cases	45
3.4	Comparison between the retrieved and actual density profiles	46
3.5	Peaks and troughs values comparison	47
3.6	Amplitudes from the retrieved and actual wakefield profiles for different simulation parameters	48
3.7	The measured density profiles evolution caused by diffraction	49
3.8	Tomography set up for a moving object diagnostic	52
4.1	Steps to retrieve the phase modulation in SSSI.	56
4.2	Schematic inside the target chamber	58
4.3	Density profiles of the wakefield with $n = 5 \times 10^{18} \text{ cm}^{-3}$	60
4.4	Phase modulation of the probe pulse with plasma density $n = 5 \times 10^{18} \text{ cm}^{-3}$	61
4.5	Density profiles of the wakefield with $n = 2 \times 10^{19} \text{ cm}^{-3}$	62

4.6	Phase modulation of the probe pulse with plasma density $n = 2 \times 10^{19} \text{ cm}^{-3}$	62
4.7	Schematic of the probe table	63
4.8	DAQ map	68
4.9	Examples of modulation at the ionisation front	69
4.10	Position of the modulation as a function of the pulses' delay	70
4.11	Examples of Mach-Zehnder interferometer with different delays	70
4.12	Position of the plasma front as a function of Mach-Zehnder probe's delay path	71
4.13	Main laser calibration with focal spot camera	72
4.14	Spectral interferometry results from shots 160818-0141 and 160818-0143 . . .	73
4.15	Spectral interferometry results from shot 160818-0146	74
4.16	Plasma density diagnostic results with methane	75
4.17	Spectral interferometry results with methane.	76
4.18	Plot of the modulation wavelengths vs the plasma density.	77
4.19	The third modulation wavelength against the corrected average plasma density with p/f value equals to (a) 1×10^{-2} , (b) 3×10^{-2} , and (c) 1×10^{-1} . . .	78
5.1	Illustration of the plasma wakefield diagnostic experiment.	80
5.2	Sparse image example in DCT domain.	82
5.3	Computational graph for back-propagation	85
5.4	A code example of doing compressed sensing retrieval with ANOA.	86
5.5	The result of the compressed sensing example using ANOA.	87
5.6	Schematic designs of the 3D spectrometer.	88
5.7	The datacube transformation of the light in the 3D spectrometer.	89
5.8	The simulated spectrum retrieval in the 3D spectrometer.	91
5.9	Illustration of the datacube to be inverted and simulated pattern on the screen.	93
5.10	The retrieved spectrum results with the simulated interferometry signal. . . .	94
6.1	An example set up for proton radiography	96
6.2	A case for light refraction	99
6.3	An example of radiogram with caustics	103
6.4	Examples of Voronoi and power diagrams, and Lloyd's algorithm	105

6.5	Projections of 4 protons from object plane to the source plane	108
6.6	Results of various forward transform algorithms of proton radiography	109
6.7	1D proton radiography inversion via gradient descent	113
6.8	Failed example of 1D proton radiography inversion via gradient descent	114
6.9	Failed results of 2D proton radiography inversion via gradient descent	115
6.10	Examples of source and target profiles	116
6.11	Illustration of the quantitative shadowgraphy algorithm	119
6.12	Examples of radiograms generated with different strengths of magnetic field	122
6.13	Benchmark results on toroidal magnetic field	123
6.14	Benchmark results on arbitrary structures magnetic field	124
6.15	Shadowgraph and its inversion from a plasma wakefield experiment	127
7.1	Illustration of the notations used in ML-SOO	134
7.2	Reduced search space illustration	135
7.3	ML-SOO cell divisions	137
7.4	Results of ML-SOO algorithm	140
7.5	Optimisation test results benchmarked against SOO and IMGPO	143
7.6	Software architecture of the optimisation framework.	146
7.7	An example of <code>config.json</code> file.	148
7.8	An example of input template file (<code>os-stdin.template</code>).	150
7.9	Preprocess file example (<code>preprocess.py</code>).	150
7.10	Submission file example (<code>submission</code>).	151
7.11	A simple example of the postprocess code (<code>postprocess.py</code>).	151
7.12	The electron energy gain plotted against the simulation ID and time for uniform density.	154
7.13	The electric field profiles from the uniform plasma optimisation result.	155
7.14	The electron energy gain plotted against the simulation ID and time for non-uniform density.	156
7.15	The electric field and plasma density vs the propagation distance from the non-uniform optimisation result.	157

A.1	The case illustration in non-perpendicular probe.	163
A.2	The case illustration in the tomography.	168

Introduction

If a sufficiently short duration, intense laser pulse [1–6], electron bunch [7, 8], or proton bunch [9–12] is fired into a plasma, it will drive an electron plasma wave, now commonly known as a wakefield accelerator. The interesting thing about wakefield accelerators are that the electric fields can reach up to 100 GV/m [1, 13–15], about three orders of magnitude higher than the electric field conventional accelerators can provide. Therefore, this can be used to accelerate electrons in a much shorter interaction length, opening the potential for many new areas of research to emerge in the coming decades.

Plasma wakefield diagnostic development is an active research topic in the plasma accelerator community because it can provide insight on what actually happens inside the plasma accelerator itself. One of the most elegant methods to diagnose wakefields is called Frequency Domain Holography (FDH) first developed by N. H. Matlis and colleagues, led by Prof. M. Downer in reference [16]. In FDH, two chirped probe pulses with durations longer than plasma wakefield period co-propagate with the wakefield and then into a spectrometer for spectral interferometry. Only one of the pulses interacts with the wakefield while the other one is used as a reference pulse. Using this method, it is possible to capture a 2D picture of plasma wakefield in a single shot. However, this method is only able to work when the amplitude of the plasma wave is relatively small. It also produces an integrated profile of the plasma wakefield along the propagation distance, hence, it cannot detect any temporal evolution of the plasma wakefield.

In chapter 3, I present a technique to diagnose the plasma wakefield with a probe crossing with an oblique angle to solve this problem. By crossing the probe at an oblique angle, one

can choose the diagnosed position along the propagation distance of the plasma wakefield, allowing, for the first time, the plasma wakefield evolution to be observed. The quantitative information of the plasma wakefield amplitude, assuming the wakefield is cylindrically symmetric, can be retrieved using an inversion I derived in chapter 3. The results were confirmed using particle-in-cell (PIC) simulations [17–19]. The technique complements with Frequency Domain Streak Camera (FDSC) [20] that detects the wakefield evolution in a different plane.

An experiment to support the idea in chapter 3 was performed by our group in June - November 2016 in Target Area 2, Central Laser Facility, STFC Rutherford Appleton Laboratory. The experiment set-up, results, and analysis of the experiment are presented in chapter 4.

The idea of extending the work from chapters 3 and 4 is presented in the chapter 5 using a new 3D spectrometer concept that I developed using compressed sensing [21, 22]. Using the 3D spectrometer and supporting simulations, I show that it is possible to acquire a trillion-frames-per-second video of plasma wakefield in a single shot.

Up to this point, I have only mentioned spectral interferometry techniques to diagnose plasma wakefields. There is also another technique commonly used to investigate plasma wakefields, i.e. that of optical shadowgraphy. In 2015, a group in Jena, Germany, led by Prof. Dr. Malte Kaluza, and Imperial College London, led by Dr. Stuart Mangles and Prof. Zulfikar Najmudin, captured pictures of plasma wakefields using the shadowgraphy method and a 6-fs optical probe pulse [23]. However, retrieving quantitative information from shadowgrams is a challenging task due to its non-linear nature, as explained in terms of proton radiography in [24]. In chapter 6 of this thesis, I present my new method of using computational graphics techniques to retrieve quantitative information from both shadowgrams and proton radiography images [25].

I present a different approach in the last chapter (chapter 7). In the recent blossoming of machine learning techniques, I tried to think about possible applications of machine learning to plasma accelerator research. One subtopic of machine learning that I found suitable is Bayesian Optimisation or optimisation in general. In the plasma accelerator field, it is inter-

esting to optimise the output of the plasma accelerators, e.g. electron energy and electron energy distribution. However, doing simulations and experiments is costly. Therefore, the number of simulations and experiments performed needs to be minimised. In chapter 7, I developed a new method to optimise a mathematical functional over a 1D function with the minimum number of functional evaluations. Following on the work presented in chapter 7 is a novel software framework to perform the optimisation using simulations as means to perform the function evaluation. In chapter 7, it is deployed to optimise the output of the plasma wakefield system and it found the optimum parameters required to generate the highest energy gain. By the time this thesis was written and submitted for examination, the software is being used to fit 10 parameters to experimental spectral results from the National Ignition Facility (NIF).

As a summary, my contributions to the plasma accelerator field as well as other fields are:

1. deriving a novel transformation and its inversion of cases where the probe is crossing at a non-perpendicular angle to the diagnosed object as well as the code implementation (chapter 3) [26,27];
2. presenting the idea of plasma wakefield diagnostics in a spatially-resolved manner and also providing the first experimental validation of the technique (chapters 3 and 4) [27,28];
3. developing a novel concept of 3D spectrometer and its applications to plasma wakefield diagnostics (chapter 5); (UK patent application no. GB1712357.1 [29])
4. introducing a method from computational graphics to retrieve quantitative information from non-linear shadowgrams and proton radiography images (chapter 6) [25];
5. developing a novel machine learning algorithm to optimise a functional over a 1D function, developed the software and applied it to optimise the output of plasma wakefield accelerators via simulations (chapter 7) [30].

Selected publications and patents

1. M. F. Kasim, N. Ratan, L. Ceurvorst, J. Sadler, P. N. Burrows, *et al.*, *Simulation of density measurements in plasma wakefields using photon acceleration*, Phys. Rev. ST Accel. Beams **18**, 032801, 2015 [26].
2. M. F. Kasim, J. Holloway, L. Ceurvorst, M. Levy, N. Ratan, *et al.*, *Quantitative single shot and spatially resolved plasma wakefield diagnostics*, Phys. Rev. ST Accel. Beams **18**, 081302, 2015 [27].
3. M. F. Kasim and P. A. Norreys, *Infinite dimensional optimistic optimisation with applications on physical systems*, Bayesian Optimization Workshop at NIPS and arXiv: 1611.05845, 2016 [30].
4. M. F. Kasim, L. Ceurvorst, N. Ratan, J. Sadler, A. Savert, *et al.*, *Quantitative shadowgraphy and proton radiography for large intensity modulations*, Phys. Rev. E **95**, 023306, 2017 [25].
5. N. F. Y. Chen, M. F. Kasim, L. Ceurvorst, N. Ratan, J. Sadler, *et al.*, *Machine learning applied to proton radiography*, Phys. Rev. E **95**, 043305, 2017 [31].
6. M. F. Kasim, P. A. Norreys, and P. N. Burrows, *3D Laser Spectrometer*, UK patent application no. GB1712357.1 [29].
7. M. F. Kasim, K. Glize, L. Ceurvorst, F. Keeble, J. Sadler, N. Ratan, *et al.*, *Diagnostics of plasma wakefield in clustered gas*, in preparation for Phys. Rev. Lett. [28].

Chapter 1

Theory of plasma wakefield

In this chapter, I present a brief history of plasma accelerators and some aspects of the theory of light interaction with plasma, especially in the case of plasma wakefield, starting from the derivation of the plasma frequency. As work in this thesis involves a wide range of techniques, the specific theory related to each chapter is included in each corresponding chapter for completeness. Following the derivation of the plasma frequency is that of the dispersion relationship of electromagnetic radiation in plasma. This is followed by a discussion of the role of the ponderomotive force and its application to the laser wakefield accelerators. Current-day limitations are highlighted, including: the pump depletion length; the de-phasing length; as well as the modulational instability for both laser and beam driven wakefields. Finally, as one of the experiments described in this thesis involved cluster plasma, some theories on cluster plasma are also derived and presented.

1.1 History

The history of the plasma accelerator dates back to 1979 when Tajima and Dawson published a paper [1] describing how electrons can be accelerated using electric field generated due to the plasma's electron density modulation, now known as the wakefield accelerator. The

interesting thing about the electric field in plasma wakefield is that it can reach 100 GeV/m, or 3 orders of magnitude higher than in conventional accelerators.

As the technology required to produce high power short laser pulses (~ 100 TW and \sim tens fs) had not been invented, in the early days of plasma accelerators, beatwaves [13,32,33] was proposed as the method of driving laser-driven plasma wakefield accelerators. With the beatwave, two laser beams with different frequencies were fired and formed the beatwave in the plasma. The frequency difference was tuned so that it matched the plasma frequency. A later method used self modulation [34–36], where just one laser beam was fired. The laser beam pulse duration was long enough to cover several plasma periods. Some modulation in the plasma density was caused by stimulated Raman forward scattering growing from noise. This modulation then grew rapidly to form the wakefield. The problem with the beatwave method is that ion motion, induced by the ponderomotive force associated with the modulational instability, rapidly detunes the density from the beatwave envelop, thereby limiting the amplitude of the plasma wave. Equally, the problem with the self-modulation method is the on-set of wavebreaking and the associated thermal spread of energetic electrons observed by Modena *et al.* [13].

When the broad-bandwidth Ti:Sapphire laser technology became available for optical wavelengths around the turn of the millenium [37], it made ultrashort optical lasers with multi-terawatt, and even petawatt, peak powers possible. The few-ten-femtosecond duration laser pulse (shorter than one plasma duration) could drive non-linear wakefields for the first time [38]. The results from three groups in 2004 [2–4] showed the high quality mono-energetic electron beams with hundreds of MeV energy can be achieved using ultrashort Ti:Sapphire laser pulses. These milestone experiments provided great impetus to investigation of laser wakefield accelerator world-wide.

Since then, many techniques have been proposed to enhance the energy and quality of the electron beam. These include the use of plasma channels to guide the laser beam [5,39], the use of petawatt-scale lasers and relatively low plasma density [39,40], optimising the electron injection mechanism [41], multiple stages of accelerators [42,43], or combinations

of them all. These techniques can produce electron beams with energy of GeV in only a centimetre acceleration length. An interesting application of laser-driven wakefields is that the X-ray generated from the electron beam from the accelerator [44, 45] can be used for tomography imaging on biological tissues [46, 47]. Another application is from a group from SLAC who successfully accelerate positron beams using hollow-profile laser beams [48].

Besides driving the wakefield using intense laser pulses, there has been a number of experiments that have used electron [7, 8] or proton beams [9–12] to drive the wakefield. With the electron beam as the driver, it can accelerate the trailing electron beam to have the twice the energy of the driver beam [7]. It has also been recently shown via simulations that combining optical and electron beam drivers it is possible to generate high flux X-rays from a plasma wakefield [49]. The problem with electron beams as the driver is that it can quickly lose its energy due to the mass of the electrons. To solve this, Caldwell, *et al.* proposed a proton beam as a driver [9]. It was shown using simulations in reference [9] that it might be possible to reach electron energy of 0.62 TeV in 450 m, which is clearly a very interesting case for a future lepton collider. A proof-of-concept experiment is being performed at CERN using the 450 GeV Super Proton Synchrotron (SPS) beam. The experiment is known as AWAKE [10, 12].

1.2 Plasma frequency

A plasma is a collection of free electrons and ions. In a stable condition, the negative charge density from the electrons equals the positive charge density from the ions (the net density is equal to 0). If there is a disturbance so that the net charge density is not 0, then that produces an electric field to bring back charge to neutralise it. To start calculating charge motion in the plasma, first make the following two assumptions: (1) the ions are not moving because usually they are much heavier than the electrons, (2) the disturbance is small, so one can neglect the second order terms of the disturbance.

Then, given that the initial (undisturbed) electron plasma density is n_0 , and there is

disturbance so that the electron density at position \mathbf{r} is $n(\mathbf{r})$ (while the ion charge density is still n_0e), Poisson's equation of the electric potential, Φ , is

$$\nabla^2\Phi = \frac{(n - n_0)e}{\varepsilon_0}, \quad (1.1)$$

where e is the elementary charge and ε_0 is the vacuum permittivity. The gradient of the electric potential induces the force to the electron, so we can write the momentum equation for an electron as

$$m\frac{d\mathbf{v}}{dt} = e\nabla\Phi, \quad (1.2)$$

where m and \mathbf{v} is the electron mass and velocity, respectively. The momentum equation is not in the relativistic regime because we assume small movement. The d/dt derivative can be expanded to $d/dt = \partial/\partial t + \mathbf{v} \cdot \nabla$. However, the second term is neglected here because of the disturbance (hence the velocity) is assumed to be small. Thus, the equation above is

$$m\frac{\partial\mathbf{v}}{\partial t} = e\nabla\Phi, \quad (1.3)$$

As the electron's movement can change the density, the continuity equation is

$$\frac{\partial n}{\partial t} + \mathbf{v} \cdot \nabla n + n\nabla \cdot \mathbf{v} = 0. \quad (1.4)$$

The second term of the equation above can be neglected as the movement is assumed to be small. For the third term, the electron density can be substituted as $n = n_0 + \delta n$. In that terms, the δn is the small disturbance and v is also small. Therefore, by ignoring the second order terms of the disturbance, we can simplify the equation (1.4) into

$$\frac{\partial n}{\partial t} + n_0\nabla \cdot \mathbf{v} = 0. \quad (1.5)$$

Applying the div operator on equation (1.3), and substitute $\nabla^2\Phi$ from equation (1.1), yields

$$\frac{(n - n_0)e^2}{\varepsilon_0} = m \frac{\partial}{\partial t} (\nabla \cdot \mathbf{v}). \quad (1.6)$$

Next, substitute $\nabla \cdot \mathbf{v}$ from equation (1.5) to the equation above. It gives us

$$\frac{(n - n_0)e^2}{\varepsilon_0} = m \frac{\partial}{\partial t} \left[-\frac{1}{n_0} \frac{\partial n}{\partial t} \right]. \quad (1.7)$$

Rearranging the equation above gives us a harmonic oscillator equation,

$$\frac{\partial^2 n}{\partial t^2} + \omega_p^2 (n - n_0) = 0, \quad (1.8)$$

where $\omega_p^2 = n_0 e^2 / m \varepsilon_0$ is the square of the plasma frequency.

1.3 Light in plasma

In this section, the plasma density is assumed to be uniform and the light is a plane wave with uniform intensity. Only the transverse mode of the wave is considered in this section. Even though the electric and magnetic fields move an electron away from its origin, another electron will immediately replace it because the velocity on the same transverse plane is uniform. This makes no charge separation in this case. It is also assumed that the light intensity is small enough so that it does not increase the electron's mass due to the effect of special relativity. The ions are assumed to be stationary. Last, the derivation in this section is treated as in linear case, so all the second order terms are neglected.

Let us start by writing the Maxwell's equations for electric field \mathbf{E} , magnetic field, \mathbf{B} ,

and the current density, \mathbf{J} ,

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} = 0 \quad (1.9)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.10)$$

$$\nabla \times \mathbf{B} = \mu_0 \left(\varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J} \right), \quad (1.11)$$

where ε_0 and μ_0 are the vacuum permittivity and permeability, respectively. The divergence of the electric field in the first equation is zero because no charge separation due to the transverse mode plane wave. The current density, \mathbf{J} , can be written as

$$\mathbf{J} = -n_0 e \mathbf{v}, \quad (1.12)$$

where n_0 is the electron density, e is the electron charge, and \mathbf{v} is the electron velocity. The electric field and the magnetic field are contributing to the electrons' movements. We can write the momentum equation from the electric force and the Lorentz force of the electron as,

$$m \frac{\partial \mathbf{v}}{\partial t} = -e \mathbf{E} - e \mathbf{v} \times \mathbf{B}. \quad (1.13)$$

The derivative on the left hand side should be d/dt . However, as we are doing derivation in the linear regime, the derivative d/dt can be approximated as $d/dt = \partial/\partial t + \mathbf{v} \cdot \nabla \approx \partial/\partial t$. The second term on the right hand side of the equation multiplies the electron velocity and the magnetic field of the light which produces a second order term, which can be neglected. Thus,

$$m \frac{\partial \mathbf{v}}{\partial t} = -e \mathbf{E}. \quad (1.14)$$

Substituting \mathbf{J} from equation (1.12) to equation (1.11) and apply the partial derivative

of time to that equation, $\partial/\partial t$, gives

$$\nabla \times \frac{\partial \mathbf{B}}{\partial t} = \mu_0 \varepsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} - \mu_0 n_0 e \frac{\partial \mathbf{v}}{\partial t}. \quad (1.15)$$

Substituting $\partial \mathbf{B}/\partial t$ from equation (1.10) and $\partial \mathbf{v}/\partial t$ from equation (1.14) to the equation above and writing the speed of light, $c = 1/\sqrt{\mu_0 \varepsilon_0}$, gives

$$-\nabla \times (\nabla \times \mathbf{E}) = \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} + \frac{\omega_p^2}{c^2} \mathbf{E}, \quad (1.16)$$

where $\omega_p^2 = n_0 e^2 / m \varepsilon_0$ is the square of the plasma frequency. The term on the left hand side of the equation can be substituted using the identity,

$$\nabla \times (\nabla \times \mathbf{E}) = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E}. \quad (1.17)$$

As the term $\nabla \cdot \mathbf{E} = 0$ from equation (1.9), equation (1.16) becomes

$$c^2 \nabla^2 \mathbf{E} = \frac{\partial^2 \mathbf{E}}{\partial t^2} + \omega_p^2 \mathbf{E}. \quad (1.18)$$

For a monochromatic wave, the second derivatives can be written as $\nabla^2 \mathbf{E} = -k^2 \mathbf{E}$ and $\partial^2 \mathbf{E}/\partial t^2 = -\omega^2 \mathbf{E}$, where k and ω are the wavenumber and the frequency of the wave, respectively. Therefore, substituting the second derivatives to the equation above yields the dispersion relation of light in plasma,

$$\omega^2 = \omega_p^2 + c^2 k^2. \quad (1.19)$$

From the dispersion relation, the phase velocity and group velocity of an electromagnetic

wave are respectively,

$$v_p = \frac{\omega}{k} = c \sqrt{1 - \frac{\omega_p^2}{\omega^2}} \quad (1.20)$$

$$v_g = \frac{\partial \omega}{\partial k} = c \sqrt{1 - \frac{\omega_p^2}{\omega^2}}. \quad (1.21)$$

1.4 Ponderomotive force

The spatial gradient of amplitude of an oscillating force can exert force to an object. This is known as the *ponderomotive* force. It can be exerted by electromagnetic wave where the wave intensity is not uniform. To understand the origin of the force, I will explain the ponderomotive force in longitudinal oscillation first and then move to the electromagnetic wave.

1.4.1 Longitudinal oscillating force

Suppose there is a longitudinal oscillating force in the z direction,

$$F(z, t) = F_0(z) \cos(\omega t), \quad (1.22)$$

where F_0 is the amplitude of the force which varies in the z direction, ω is the frequency, and t is the time. For simplicity, regard all the vector parameters to have a direction only along the z axis. An object with mass m with zero initial velocity has the velocity, v , and displacement, δz , as functions of time as below,

$$v = \frac{F_0(z)}{m\omega} \sin(\omega t) \quad (1.23)$$

$$z - z_0 = -\frac{F_0(z)}{m\omega^2} \cos(\omega t). \quad (1.24)$$

The equations above are obtained by assuming the amplitude of the force, $F_0(z)$, does not vary much within the object's oscillation position.

When the object is oscillating, it experiences different amplitude of the force. Although the difference is minuscule, but when it experiences it many times, it experiences a drift. From equation (1.22) and (1.24), we can obtain the force exerted to the object as a function of time,

$$\begin{aligned}
 F(t) &= F_0[z(t)] \cos(\omega t) \\
 &\approx \left[F_0 + \frac{\partial F_0}{\partial z} (z - z_0) \right] \cos(\omega t) \\
 &\approx F_0 \cos(\omega t) - \frac{\partial F_0}{\partial z} \frac{F_0}{m\omega^2} \cos^2(\omega t).
 \end{aligned} \tag{1.25}$$

In this equation, $F_0(z_0)$ and $\partial F_0(z_0)/\partial z$ are written without the bracketed (z_0) , i.e. F_0 and $\partial F_0/\partial z$, to avoid confusion between multiplication and function. On the second line, the term $F_0(z)$ is expanded until the second order term of F_0 , while on the third line, the $F_0(z)$ term from equation (1.24) is approximated to be $F_0(z_0)$ to neglect the third and higher order terms of F_0 . Taking the time average of the force in equation (1.25) gives us the ponderomotive force,

$$F_p = \langle F(t) \rangle \approx -\frac{1}{4m\omega^2} \frac{\partial}{\partial z} F_0^2. \tag{1.26}$$

The negative term tells us that the ponderomotive force tends to bring the object to the place where the amplitude is the smallest.

1.4.2 Electromagnetic wave

Consider an oscillating electric, \mathbf{E} , and magnetic, \mathbf{B} , fields with frequency ω ,

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_1(\mathbf{r}) \cos(\omega t) + \mathbf{E}_2(\mathbf{r}) \sin(\omega t) \quad (1.27)$$

$$\mathbf{B}(\mathbf{r}, t) = \mathbf{B}_1(\mathbf{r}) \cos(\omega t) + \mathbf{B}_2(\mathbf{r}) \sin(\omega t). \quad (1.28)$$

From Maxwell's equation, $\nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t$, we obtain,

$$\mathbf{B}_1(\mathbf{r}) = \frac{1}{\omega} \nabla \times \mathbf{E}_2(\mathbf{r}) \quad (1.29)$$

$$\mathbf{B}_2(\mathbf{r}) = -\frac{1}{\omega} \nabla \times \mathbf{E}_1(\mathbf{r}). \quad (1.30)$$

An object with electric charge $-e$ and mass m will experience a force,

$$\mathbf{F} = m \frac{d\mathbf{v}}{dt} = -e\mathbf{E} - e\mathbf{v} \times \mathbf{B}. \quad (1.31)$$

Now assume the electric force is weak enough so that the object does not move at relativistic speed. Also assume that the object displacement is small enough so the electric and magnetic fields do not vary much. In this case, the velocity and the displacement of the object are dominated by the effect from the electric field. With a simple calculus, one obtains the velocity, \mathbf{v} , and the displacement, \mathbf{s} , of the object as,

$$\mathbf{v}(t) \approx \frac{e}{m\omega} [\mathbf{E}_2(\mathbf{r}_0) \cos(\omega t) - \mathbf{E}_1(\mathbf{r}_0) \sin(\omega t)] \quad (1.32)$$

$$\mathbf{s}(t) \approx \frac{e}{m\omega^2} [\mathbf{E}_2(\mathbf{r}_0) \sin(\omega t) + \mathbf{E}_1(\mathbf{r}_0) \cos(\omega t)], \quad (1.33)$$

where \mathbf{r}_0 is the object average position.

Having the velocity and the displacement of the object, we can follow the similar steps from the previous subsection to obtain the ponderomotive force. Using Taylor's expansion to the first order terms, the electric field can be approximated into

$$\mathbf{E} \approx \mathbf{E}(\mathbf{r}_0) + (\mathbf{s} \cdot \nabla)\mathbf{E}(\mathbf{r}_0). \quad (1.34)$$

Substituting the electric field from equation (1.34), magnetic fields from equations (1.29) and (1.30) into the equation (1.31) gives

$$\begin{aligned} \mathbf{F}(t) \approx & -e \{ \mathbf{E}(\mathbf{r}_0) + [\mathbf{s}(t) \cdot \nabla] [\mathbf{E}_1(\mathbf{r}_0) \cos(\omega t) + \mathbf{E}_2(\mathbf{r}_0) \sin(\omega t)] \} + \\ & - \frac{e}{\omega} \mathbf{v}(t) \times \{ [\nabla \times \mathbf{E}_2(\mathbf{r}_0)] \cos(\omega t) - [\nabla \times \mathbf{E}_1(\mathbf{r}_0)] \sin(\omega t) \}. \end{aligned} \quad (1.35)$$

Substituting the velocity and the displacement from equations (1.32) and (1.33) to the equation above and take the time average, yields the ponderomotive force,

$$\langle \mathbf{F}(t) \rangle = -\frac{e^2}{2m\omega^2} [(\mathbf{E}_1 \cdot \nabla)\mathbf{E}_1 + (\mathbf{E}_2 \cdot \nabla)\mathbf{E}_2 + \mathbf{E}_1 \times (\nabla \times \mathbf{E}_1) + \mathbf{E}_2 \times (\nabla \times \mathbf{E}_2)]. \quad (1.36)$$

The triple cross product identity gives

$$\mathbf{E} \times (\nabla \times \mathbf{E}) = \nabla(\mathbf{E} \cdot \mathbf{E}) - (\mathbf{E} \cdot \nabla)\mathbf{E}. \quad (1.37)$$

Therefore, the equation (1.36) can be simplified to

$$\langle \mathbf{F}(t) \rangle = -\frac{e^2}{2m\omega^2} \nabla [E_1^2 + E_2^2] = -\frac{e^2}{2m\omega^2} \nabla |E^2|. \quad (1.38)$$

To express the ponderomotive force in terms of the normalised vector potential, a , substitute $|E^2| = kmc^2a/e$. Thus,

$$\langle \mathbf{F}(t) \rangle = -\frac{1}{2}mc^2\nabla(a^2). \quad (1.39)$$

1.5 Linear plasma wakefield

When a laser pulse is fired into a plasma, the electrons are repelled by the action of the ponderomotive force, leaving a positive charge in its original position. This charge separation then produces a strong electric field that results in the return of the expelled electrons. The kinetic energy acquired by the electrons means that they overshoot, and this is repeated over a number of oscillations, resulting in a formation of a longitudinal electric field structure behind the laser pulse. To calculate the electron density profile, one starts by writing the equations from section 1.2 that are still valid in this case. Those are,

$$\nabla^2\Phi = \frac{(n - n_0)e}{\epsilon_0} \quad (1.40)$$

$$\frac{\partial n}{\partial t} + n_0\nabla \cdot \mathbf{v} = 0 \quad (1.41)$$

$$m\frac{\partial \mathbf{v}}{\partial t} = e\nabla\Phi + \mathbf{F}_p, \quad (1.42)$$

where \mathbf{F}_p is the ponderomotive force. All the equations above are the same as equations in section 1.2, except the last equation where we add the ponderomotive force term.

Following the similar steps in section 1.2 and substituting the ponderomotive force $\mathbf{F}_p = -mc^2\nabla a^2/2$ leads to the equation

$$\frac{\partial^2 n}{\partial t^2} + \omega_p^2(n - n_0) = \frac{1}{2}n_0c^2\nabla^2(a^2), \quad (1.43)$$

where $\omega_p^2 = n_0 e^2 / m \varepsilon_0$ is the square of the plasma frequency and $\mathbf{a} = e\mathbf{A}/mc$ is the normalised intensity of the laser pulse with \mathbf{A} is the vector potential.

1.6 Dephasing length

If the driver of the wakefield travels with slower speed than the beam it accelerates, then dephasing can occur: the beam from accelerating region can reach the decelerating region. If the wakefield's wavelength is λ_p , the dephasing occurs when the distance between the beam and the driver is reduced by $\lambda_p/2$ in the lab frame. The distance where the dephasing occurs can be estimated by assuming the beam travels with speed very close to the speed of light, c , compared to the driver. Given the driver's velocity, v_d , the distance the driver can travel before the dephasing occurs is [50]

$$s_d \approx \frac{\lambda_p}{2} \frac{1}{1 - v_d/c} \quad (1.44)$$

$$\approx (16.72 \text{ } \mu\text{m}) [n(\times 10^{18} \text{ cm}^{-3})]^{-1/2} \frac{1}{1 - v_d/c}, \quad (1.45)$$

where the second line equation is the approximate equation in terms of electron density (in $\times 10^{18} \text{ cm}^{-3}$).

If an electromagnetic wave is used as the driver, the driver velocity equals to the group velocity of the wave, given by equation (1.21).

1.7 Pulse depletion

In the laser-driven plasma accelerator, the laser pulse transfers the energy to the wakefield. Due to this energy transfer, the front of the driving pulse etches back and it slowly depletes.

The etching velocity of the driving pulse in the bubble regime is given by [50, 51],

$$v_{etch} \approx \frac{\omega_p^2}{\omega_0^2} c, \quad (1.46)$$

where ω_p is the plasma frequency, ω_0 is the laser pulse's frequency, and c is the speed of light. The equation above has been shown to agree with Particle-in-Cell (PIC) simulations [17–19, 52, 53] in the reference [51].

One factor that causes the etch is the velocity difference between the front and back of the pulse. In the bubble regime, the back of the pulse is at the region where there is no electron, so the speed is c , while the front of the pulse still propagates with group velocity $v_d < c$. Another factor is the photon acceleration process where the front of the pulse is decelerating due to positive gradient of the electron density. The photon acceleration is discussed in chapter 2.

If most of the laser pulse is etched, then it depletes. The depletion length can be approximated by calculating the length required by v_{etch} to etch the full-width-half-maximum period of the pulse,

$$L_{etch} \approx \frac{\omega_0^2}{\omega_p^2} c \tau_{FWHM}. \quad (1.47)$$

1.8 Plasma channel and relativistic self focusing

One of the problems in laser wakefield experiments is the defocusing of the laser pulse. If a laser pulse is focused, then it can drive a relatively large wakefield. However, once it defocuses, the intensity of the laser pulse, and with it the wakefield amplitude, also decreases. In order to solve that, one can make use of a plasma channel or utilise the effect of relativistic self-focusing.

A plasma channel [54] can be made by increasing the density of the plasma radially, so

it decreases the refractive index and confines the laser pulse to be at the centre. It acts like an optical guiding structure.

For relativistic self-focusing [55, 56], the principle is similar, by decreasing the refractive index radially. However, it is done not by varying the plasma density, but by increasing the amplitude of the laser pulse so that it reaches the relativistic regime (i.e. the normalised intensity, $a_0 \sim 1$), where the refractive index depends on the amplitude.

In this section, the criteria for both plasma channeling and relativistic self-focusing are derived. The derivation follows closely reference [55].

Consider a Gaussian light pulse with focused beam waist size w_0 with frequency and wavenumber are ω_0 and k_0 , respectively. The spot size of the pulse will vary with time in a uniform medium in the lab frame as [57],

$$w = w_0 \sqrt{1 + \frac{t^2}{t_r^2}}, \quad (1.48)$$

where $t_r = k_0 w_0^2 / 2c$ is the Rayleigh or diffraction time. The time $t = 0$ is chosen when the beam is at focus. Differentiating the equation above twice with respect to the time with $t = 0$ gives us,

$$\frac{\partial^2 w}{\partial t^2} = \frac{4c^2}{k_0^2 w_0^3}. \quad (1.49)$$

By taking a line perpendicular to the wavefront, see Figure 1.1(a), the outward acceleration of the wave at distance r is given by,

$$\frac{\partial^2 r}{\partial t^2} = \frac{4c^2}{k_0^2 w_0^4} r. \quad (1.50)$$

If the refractive index of the medium is varied radially, $\eta(r)$, the phase velocities of the

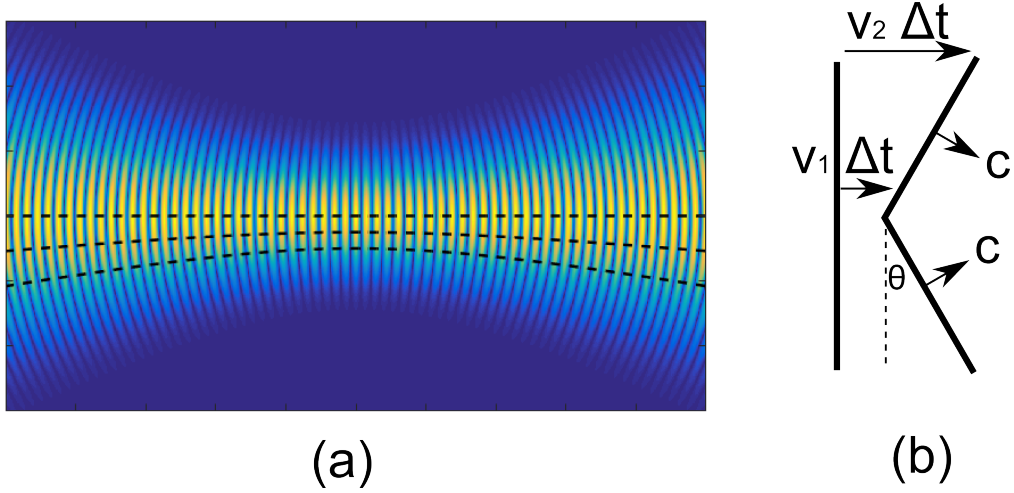


Figure 1.1: (a) Gaussian beam and lines drawn perpendicular to the wavefront. (b) Illustration of wavefront bending due to difference in phase velocity.

light at the centre and off centre are different. When the off-centre phase velocity of the light is greater than the phase velocity of the light at the centre, then the wavefront forms an angle towards the centre. This is illustrated in Figure 1.1(b). The angle is given by,

$$\theta = \frac{[\eta(r)^{-1} - \eta(0)^{-1}]c\Delta t}{r}, \quad (1.51)$$

where $\eta(r)$ is the refractive index of the medium, Δt is the time interval, and r is the position of the off-centre part of the light.

With the angle, the wavefront travels inward with velocity approximately $c\theta$, thus, the inward velocity of the part at r is given by,

$$\frac{\partial r}{\partial t} = -\frac{[\eta(r)^{-1} - \eta(0)^{-1}]c^2\Delta t}{r}. \quad (1.52)$$

The inward acceleration is then given by,

$$\frac{\partial^2 r}{\partial t^2} = -\frac{[\eta(r)^{-1} - \eta(0)^{-1}]c^2}{r}. \quad (1.53)$$

To keep the wave from defocusing, the total acceleration terms in equations (1.50) and (1.53) must be less than or equal to zero. Therefore,

$$\eta(r)^{-1} - \eta(0)^{-1} \geq \frac{4}{k_0^2 w_0^4} r^2. \quad (1.54)$$

For a wave in the relativistic regime in the varying density plasma, the refractive index in a plasma is given by [55]

$$\eta^{-1} \approx 1 + \frac{\omega_p^2}{2\omega_0^2} \left(1 + \frac{\delta n}{n_0} - \frac{a^2}{4} \right), \quad (1.55)$$

with plasma central density and frequency are n_0 and ω_p , respectively, and a^2 is the peak normalised intensity over the time at the given point. The δn is the density variation. For a Gaussian beam, $a^2 = a_0^2 \exp(-2r^2/w_0^2)$. Therefore, to keep the wave focused, the requirement below must be fulfilled,

$$\frac{\delta n}{n_0} + \frac{a_0^2}{4} \gtrsim \frac{8c^2}{w_0^4 \omega_p^2} r^2, \quad (1.56)$$

where $\omega_0/k_0 \approx c$. The critical intensity for relativistic self-focusing in a uniform plasma can be obtained by substituting $r = w_0$, therefore,

$$a_0^2 \gtrsim \frac{32c^2}{w_0^2 \omega_p^2}. \quad (1.57)$$

The critical intensity above still depends on the focused waist size, w_0 , which changing the waist size will change the intensity. Therefore, it will be more convenient to express the equation above in terms of the critical power,

$$P_c \approx \frac{8\pi m^2 c^5 \varepsilon_0 \omega^2}{e^2 \omega_p^2} \approx 17.5 \left(\frac{\omega}{\omega_p} \right)^2 \text{ GW}. \quad (1.58)$$

1.9 Modulational instability

Suppose that there is a very long laser pulse with normalised intensity a_0 and frequency ω_0 propagate in a cold plasma with initial electron charge density n_0 and stationary ions. The pulse duration is sufficient to cover several plasma wave's periods. If there are a small modulations in the electron density, it changes the group velocity of some parts of the laser pulse. The variation on the group velocity of the pulse causes modulation on photons distribution along the pulse, hence the intensity. The modulation on intensity then causes ponderomotive force applied to the plasma's electrons, therefore amplifying the electron's density modulation. This is the scenario of the modulational instability with long laser pulse in a plasma.

First, let us rewrite the group velocity of the laser pulse as a function of electron density, n ,

$$v_g = c \sqrt{1 - \frac{\omega_p^2}{\omega_0^2} \frac{n}{n_0}} \approx v_{g0} \left(1 - \frac{\omega_p^2}{2\omega_0^2} \frac{c^2}{v_{g0}^2} \frac{\delta n}{n_0} \right), \quad (1.59)$$

where c is the speed of light, ω_p and v_{g0} are the plasma frequency and the laser pulse's group velocity with electron density $n = n_0$, respectively. The variable $\delta n = n - n_0$ is the modulation of the electron density. The approximation was made using Taylor's expansion up to the first order term only.

If δn is not uniform, the group velocity of the laser pulse is varied and thus makes the photons distribution changes. We can use the fluid equation to express the equation of number of photon as function of position and time, $N_{phot}(z, t)$. As the squared of normalised intensity is proportional to the number of photon, we can substitute it with the normalised intensity, instead,

$$\frac{\partial(a^2)}{\partial t} + v_{g0} \frac{\partial(a^2)}{\partial z} + a_0^2 \frac{\partial v_g}{\partial z} = 0. \quad (1.60)$$

This is the continuity equation for photons. The equation above only keeps the zero and first order terms and ignore higher terms.

To simplify the equation above, we can introduce a new coordinate, $\xi = z - v_{g0}t$ and $\tau = t$. Therefore, the relation between the differential equations of two coordinates are,

$$\frac{\partial}{\partial z} = \frac{\partial}{\partial \xi} \qquad \frac{\partial}{\partial \xi} = \frac{\partial}{\partial z} \qquad (1.61)$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\partial}{\partial \xi} v_{g0} \qquad \frac{\partial}{\partial \tau} = \frac{\partial}{\partial t} + \frac{\partial}{\partial z} v_{g0}. \qquad (1.62)$$

Applying the coordinate transformation above and substituting v_g from equation (1.59) to equation (1.60) yield,

$$\frac{\partial(a^2)}{\partial \tau} = b \frac{\partial n}{\partial \xi}, \qquad (1.63)$$

where $b \equiv (a_0^2 \omega_p^2 c^2) / (2\omega_0^2 n_0 v_{g0})$.

The equation above shows that the normalised intensity can be changed if there is electron density modulation. The variation of normalised intensity then can exert ponderomotive force on the plasma. To express this mathematically, write the equation (1.43), but substituting ∇^2 with $\partial^2 / \partial \xi^2$,

$$\frac{\partial^2 n}{\partial t^2} + \omega_p^2 (n - n_0) = \frac{1}{2} n_0 c^2 \frac{\partial^2 (a^2)}{\partial \xi^2}. \qquad (1.64)$$

Applying $\partial / \partial \tau$ operator on the equation above and substitute a^2 from equation (1.63) gives

$$\frac{\partial^3 n}{\partial t^2 \partial \tau} + \omega_p^2 \frac{\partial n}{\partial \tau} = \frac{1}{2} n_0 c^2 b \frac{\partial^3 n}{\partial \xi^3}. \qquad (1.65)$$

Now, transform back the ξ and τ to z and t using equations (1.61) and (1.62) to obtain,

$$\frac{\partial^3 n}{\partial t^3} + \frac{\partial^3 n}{\partial t^2 \partial z} v_{g0} + \omega_p^2 \frac{\partial n}{\partial t} + \omega_p^2 v_{g0} \frac{\partial n}{\partial z} - \frac{1}{2} n_0 c^2 b \frac{\partial^3 n}{\partial z^3} = 0. \qquad (1.66)$$

Without loss of generality, substitute $n - n_0 = \delta n_0 e^{i\omega t - ikz}$ into the equation above. It gives

$$y(\omega) = (\omega^2 - \omega_p^2)(\omega - v_{g0}k) + \frac{1}{2}n_0 c^2 b k^3 = 0. \quad (1.67)$$

The function $y(\omega)$ is a cubic function with local minimum at $\omega_{min} = (v_{g0}k + \sqrt{v_{g0}^2 k^2 + 3\omega_p^2})/3$.

The function $y(\omega)$ will have 2 imaginary roots if $y(\omega_{min}) > 0$, or

$$y(\omega_{min}) = \frac{2}{9}v_{g0}^3 k^3 \left(1 + \sqrt{1 + \frac{3\omega_p^2}{v_{g0}^2 k^2}}\right) \left(-2 + \sqrt{1 + \frac{3\omega_p^2}{v_{g0}^2 k^2}}\right) + \frac{1}{2}n_0 c^2 b k^3 > 0. \quad (1.68)$$

If at the beginning of the laser pulse it drives a small wakefield ($v_{g0}k \approx \omega_p$), then it is clear from the equation above that $y(\omega)$ has 2 imaginary roots, no matter how small the a_0^2 is, as long as it is non-zero. The imaginary roots are conjugate to each other, one imaginary root suppresses the modulation, but another imaginary root grows the modulation.

1.10 Cluster plasma model

In a cluster plasma, electrons and ions form clusters or blobs with higher density than the average. An illustration of cluster plasma is shown in Figure 1.2(a).

In order to avoid confusion, in this section only, the electron density **at** the cluster is denoted as n_0 and the oscillation frequency of the cluster is $\omega_p = (n_0 e^2 / m \epsilon_0)^{1/2}$. As not all the space is filled with electrons and ions with density n_0 , the space-averaged density is denoted by $\langle n_0 \rangle = p n_0$, where p is the fill factor.

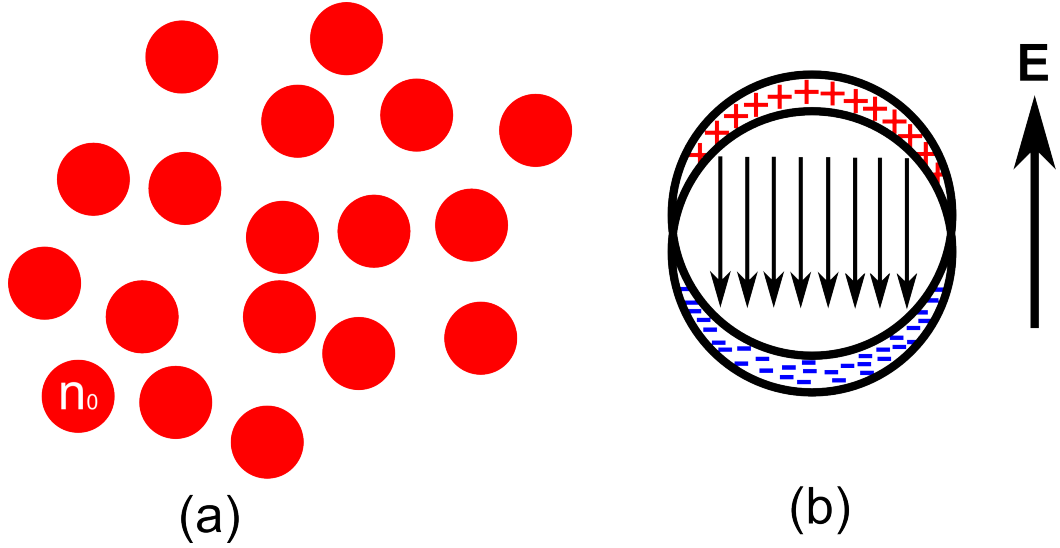


Figure 1.2: (a) Illustration of cluster plasma, where the red blobs have density of n_0 and other regions have density less than n_0 . (b) Charge separation in a cluster produces the electrostatic field (smaller arrows) inside the cluster.

1.10.1 Dispersion relation in cluster plasma

The derivation of light's dispersion relation in cluster plasma in this subsection follows closely the derivation on reference [58]. Consider a case where a plane electromagnetic wave propagates through the plasma. Assume the size of the wave is much larger than the cluster's size and the wave's amplitude is not large (i.e. so that the cluster displacement is much less than the cluster size), so a linear analysis can be performed. Unlike uniform plasma, in cluster plasma, the electrons motion's are affected by the electrostatic field resulting from the charge separation in a cluster. This is illustrated in Figure 1.2(b). Therefore, the electron's equation of motion from equation (1.14) can be modified to be

$$m \frac{\partial \mathbf{v}}{\partial t} = -e \mathbf{E} - f m \omega_p^2 \mathbf{x}, \quad (1.69)$$

where f is the geometric factor of the cluster (e.g. $1/3$ for sphere), ω_p is the oscillation frequency of the cluster. The last term in the equation above captures the electrostatic force due to charge separation exerted to the electron. In equation (1.69), the effect of magnetic field is ignored and the electric field is not strong enough to displace the electron far from

its original position (i.e. the displacement is much shorter than the cluster size).

Without loss of generality, the electric field from the electromagnetic wave can be written as

$$\mathbf{E} = \mathbf{E}_1 e^{i\omega t - ikz} + \mathbf{E}_2 e^{-i\omega t + ikz}. \quad (1.70)$$

Substituting the equation above to equation (1.69) and solving the forced harmonic oscillator equation give the solution to the electron's displacement and the velocity,

$$\mathbf{x} = \frac{e}{m(\omega^2 - f\omega_p^2)} \mathbf{E} \quad (1.71)$$

$$\mathbf{v} = \frac{e}{m(\omega^2 - f\omega_p^2)} \frac{\partial \mathbf{E}}{\partial t}. \quad (1.72)$$

Knowing the velocity of an electron, it is relatively straightforward to get the expression for the current density, which is

$$\mathbf{J} = -\langle n_0 \rangle e \mathbf{v} = -\frac{pn_0 e^2}{m(\omega^2 - f\omega_p^2)} \frac{\partial \mathbf{E}}{\partial t}. \quad (1.73)$$

From Maxwell's equations (1.9), (1.10), and (1.11),

$$\nabla^2 \mathbf{E} = \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} + \mu_0 \frac{\partial \mathbf{J}}{\partial t}. \quad (1.74)$$

Substituting equations (1.70) and (1.73) to the equation above gives the light's dispersion relation in cluster plasma,

$$k^2 c^2 = \omega^2 \left[1 - \frac{p\omega_p^2}{\omega_0^2 - f\omega_p^2} \right]. \quad (1.75)$$

If one substitutes $p = 1$ and $f = 0$, it reduces to the dispersion relation of uniform plasma.

The refractive index, η , can be easily obtained from the equation above,

$$\eta^2 = \frac{k^2 c^2}{\omega^2} = 1 - \frac{p\omega_p^2}{\omega_0^2 - f\omega_p^2}. \quad (1.76)$$

1.10.2 Ponderomotive force on cluster plasma

As the equation of motion of the electron in the cluster plasma is different from uniform plasma, the ponderomotive force exerted on an electron is also modified. The displacements of electrons with electromagnetic wave, with frequency ω and wavenumber k , in a uniform plasma and cluster plasma are given below, respectively,

$$\mathbf{x}_u = \frac{e}{m\omega^2} \mathbf{E} \quad (1.77)$$

$$\mathbf{x}_c = \frac{e}{m(\omega^2 - f\omega_p^2)} \mathbf{E}. \quad (1.78)$$

The \mathbf{x}_c can be obtained by simply replacing $\omega^2 \rightarrow (\omega^2 - f\omega_p^2)$ from \mathbf{x}_u . The velocities are just the time derivative of the displacement above.

The derivation steps in section 1.4 also applies in the cluster plasma case. Even though there is $f m \omega_p^2 \mathbf{x}$ term in the electron's equation of motion (see equation (1.69)), it will be reduced to zero when taking the time average. Therefore, by simply replacing $\omega^2 \rightarrow (\omega^2 - f\omega_p^2)$, we obtain the ponderomotive force on cluster plasma,

$$\langle \mathbf{F} \rangle_t = -\frac{1}{2} \frac{\omega^2}{\omega^2 - f\omega_p^2} m c^2 \nabla (a^2), \quad (1.79)$$

where $a = e|E^2|/kmc^2$ is the normalised intensity. From the equation above, the ponderomotive force on cluster plasma is greater than the ponderomotive force on uniform plasma

with the same electromagnetic wave's intensity, if $|\omega^2 - f\omega_p^2| < \omega^2$.

1.10.3 Plasma wakefield in cluster plasma

The motion of electrons in cluster plasma differs quite significantly from the motion of electrons in uniform plasma when the displacement is relatively small due to their oscillation around the clusters. However, if the electrons' displacements are relatively large and produce electrostatic field that is much larger than the electric field on the cluster's surface, then the electrons' motions become similar to the uniform plasma. Therefore, in this limiting case, the plasma wakefield in cluster plasma is not much different from the uniform plasma.

Chapter 2

Photon acceleration simulations

In 1989, S. Wilks and colleagues [59] discovered the effect of “photon acceleration”, i.e. a short pulse co-propagates along with plasma waves undergoes frequency change. The amount of frequency-shift of the pulse is proportional to the gradient of the electron density at the pulse’s position. If a long pulse that covers several plasma wakefield is fired and co-propagates with the plasma wakefield, the frequency in some parts of it will change. By measuring the frequency change at different parts of the pulse, one can obtain the gradient of electron density profile, and integrating it to obtain the electron density profile [60–62].

In this chapter, photon acceleration in plasma will be discussed, along with supporting simulations. They show that it can be used to diagnose a plasma wakefield’s electron density quantitatively. Before getting into the main idea and results, the theory of photon acceleration is briefly presented in the section 2.1. The method and results of simulations I performed are presented in the sections 2.2 and 2.3, respectively. Analysis and limitations of the technique are presented in the section 2.4. Last, section 2.5 summarise this chapter.

2.1 Theory of photon acceleration

In a plasma wakefield, the electron density profile varies in space and time. As a result, the refractive index also varies. For plasma electron density profile, $n(\mathbf{r}, t)$ at position \mathbf{r} and time t , the refractive index of light with frequency ω_0 is given by [63]

$$\eta(\mathbf{r}, t) = \left[1 - \frac{n(\mathbf{r}, t) \omega_p^2}{n_0 \omega_0^2} \right]^{1/2} \approx 1 - \frac{n(\mathbf{r}, t) \omega_p^2}{2n_0 \omega_0^2}, \quad (2.1)$$

where n_0 is the unmodulated electron density and ω_p is the plasma frequency. The approximation is taken with an assumption that $\omega_p^2 \ll \omega_0^2$.

If a laser pulse co-propagates with the wakefield with approximately the same velocity, the laser pulse will remain in the same relative position in the wakefield rest frame, $\mathbf{r}' = \mathbf{r} - \mathbf{v}_p t$, where \mathbf{v}_p is the plasma wakefield phase velocity. Because the refractive index in the wakefield varies, the phase of the laser pulse also varies as

$$\phi(\mathbf{r}', t) = \frac{\omega_0}{c} \int_0^s \eta(\mathbf{r}', s') ds', \quad (2.2)$$

with $s = v_p t$ denotes the distance co-propagated by the probe pulse and the wakefield.

From the phase, it is straightforward to obtain the frequency modulation of the probe pulse, as $\omega = c \partial \phi / \partial \zeta$ with ζ is the longitudinal axis in \mathbf{r}' . Substituting η from equation (2.1) to (2.2), the frequency modulation is obtained as below,

$$\frac{\omega(\mathbf{r}', s) - \omega_0}{\omega_0} = -\frac{\omega_p^2}{2\omega_0} \frac{1}{n_0} \int_0^s \frac{\partial n(\mathbf{r}', s')}{\partial \zeta} ds'. \quad (2.3)$$

The equation above is also confirmed by [64, 65]. In the remaining sections of this chapter, the frequency modulation of the probe pulse and the integrated density modulation of the wakefield along the propagation distance are compared. They confirm the relations in

equation (2.3).

2.2 Simulation method

Simulations are performed to confirm that the frequency modulation or phase modulation can be employed to quantitatively measure the plasma wakefield profile.

2.2.1 Particle-in-Cell

In the simulation, Particle-in-Cell (PIC) code was employed [17–19, 52, 53, 66]. PIC is a method to simulate the motion of huge numbers of particles in an electro-magnetic field efficiently by grouping the particles into super-particles and simulating the motion of the super-particles. Each super-particle has a weight that corresponds linearly to the number of particles it represents.

Initially, the super-particles are deployed randomly in the simulation window according to the particle's initial density distribution, $n_0(\mathbf{x})$. The simulation window also contains a grid where the density, electric and magnetic fields are calculated. They are deployed uniformly at random and the weight is assigned to be proportional to the initial density ($w_p \propto n_0(\mathbf{x})$). The super-particles are also initialised with the initial velocity, if specified by the user. Once the super-particles are distributed, the electric field from the super-particles can be obtained by solving the Poisson's equation on the grid. The electric and magnetic fields from electro-magnetic field are then added to initialise the electric and magnetic fields.

Once initialised, the iteration in the PIC simulations goes as follows:

1. calculating the density distribution, $n(\mathbf{x})$, and the current density distribution, $\mathbf{J}(\mathbf{x})$ of the plasma on the grid,
2. updating the electric and magnetic fields on the grid,
3. calculating the force acting on each super-particles based on the updated electric and

magnetic fields (interpolated from the grid), and

4. update the position and velocity of each super-particles based on the calculated force, the previous position, and the previous velocity.

The density distribution, $n(\mathbf{x})$, and the current density, $\mathbf{J}(\mathbf{x})$, are obtained by spreading the position of each super-particle, i.e.

$$n(\mathbf{x}) = \sum_p q_p w_p S(\mathbf{x} - \mathbf{x}_p) \quad (2.4)$$

$$\mathbf{J}(\mathbf{x}) = \sum_p q_p \mathbf{v}_p w_p S(\mathbf{x} - \mathbf{x}_p), \quad (2.5)$$

where q_p , w_p , x_p are respectively charge, weight, and position of the super-particle and S is the spreading function. The spreading function can be polynomial of order $R = 1, 2, 3$, or higher, i.e. $S(\mathbf{y}) = \sum_{r=0}^R \alpha_r \|\mathbf{y}\|^r$ with some coefficients α_r .

Once the density distribution and the current density distribution are calculated, the electric and magnetic fields for the next time step are solved using Maxwell's equations,

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\mu_0 \varepsilon_0} \nabla \times \mathbf{B} - \frac{1}{\varepsilon_0} \mathbf{J} \quad (2.6)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}. \quad (2.7)$$

The electric and magnetic fields acting on a super-particle are obtained by interpolating \mathbf{E} and \mathbf{B} using the spreading function of the super-particle. After that, the positions and velocities of super-particles are updated based on the previous positions, velocities, and the electromagnetic forces acting on them.

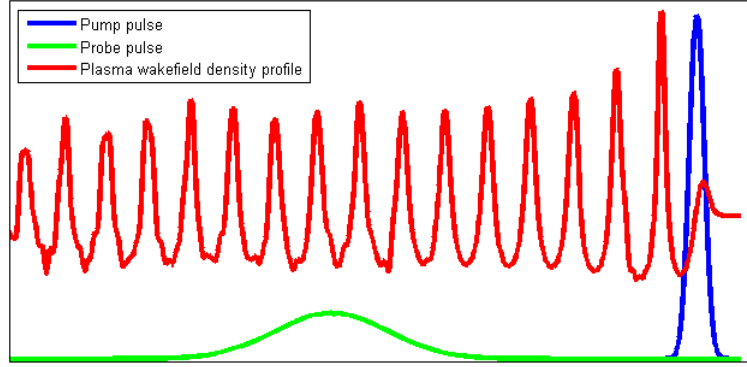


Figure 2.1: An illustration of the simulation set up. The figure is not to scale.

2.2.2 Parameters

The simulations were performed using OSIRIS 1D [17] in SCARF-LEXICON computer cluster at STFC Rutherford Appleton Laboratory. OSIRIS is a fully-relativistic code using particle-in-cell (PIC) to simulate the movement of particles in the plasma. In the baseline set up of the simulations, a short laser pulse with wavelength 800 nm and duration 39 fs to drive plasma wakefield is fired to a plasma with density of $2 \times 10^{18} \text{ cm}^{-3}$. The intensity of the driver pulse is $4.7 \times 10^{18} \text{ W/cm}^{-2}$ or in terms of normalised intensity, $a_0 = 1.5$. The simulations employed 10 particle-per-cell, Lindmann open boundary condition for the electromagnetic field [67], and absorbing boundaries for the electrons and ions. The laser pulse amplitude and short duration are sufficient to drive non-linear wakefields.

Behind the driver pulse, a long probe pulse with duration 300 fs and normalised intensity of $a_0 = 0.01$ is fired to diagnose the electron density modulation profile. The intensity of the probe is chosen to be much smaller than the driver pulse so that the probe pulse does not disturb the wakefield generated by the driver pulse. The probe and driver pulses are transform-limited pulses. These pulses co-propagate along 7 mm distance. The simulation set up is shown in the Figure 2.1.

2.2.3 Obtaining the local frequency

The simulations produced the electric field of the probe pulse during and after the interaction with the wakefield. Thus, to extract the local frequency at every longitudinal position of the probe, the Wigner transformation was employed [68].

First, denote the complex profile of electric field of a laser pulse in vacuum as $E(\xi)$ as function of $\xi = z - ct$, where z , t are position and time, and c is the speed of light. Then the Wigner transformation of the electric field profile is

$$W_E(\xi, k) = \int_{-\infty}^{\infty} E(\xi + \xi'/2) E^*(\xi - \xi'/2) e^{-ik\xi'} d\xi', \quad (2.8)$$

where $k = 2\pi f/c$ is the wavenumber of the laser pulse and $E^*(\xi)$ is the complex conjugate of $E(\xi)$.

From the Wigner distribution, one can obtain the longitudinal intensity profile, $|E(\xi)|^2$, and the longitudinal frequency profile, $f(\xi)$, of the pulse. This is done by performing an integration and weighted integration of the Wigner distribution over the wavenumber, k , as shown below,

$$|E(\xi)|^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_E(\xi, k) dk \quad (2.9)$$

$$f(\xi) = \frac{c}{2\pi} \frac{1}{|E(\xi)|^2} \int_{-\infty}^{\infty} kW_E(\xi, k) dk.$$

After obtaining the local frequency, $f(\xi)$, equation (2.3) is used to calculate the integrated electron density profile from the frequency modulation. This is compared with the integrated electron density profile from the simulation.

2.3 Simulation results

From the simulation, data points were taken along the propagation direction to obtain both the profile of electric field of the pulse and the density profile of the plasma. The first data point was taken at the centre of the laser pulse and from the electric field of the pulse, equation (2.9) was applied to obtain the longitudinal frequency profile of the pulse. Then equation (2.3) was applied to get the integrated density profile. This profile is labelled the ‘actual’ profile. From equation (2.9), the uncertainty in frequency is larger where the intensity is smaller. To maintain the accuracy in frequency retrieval of the pulse, only the frequency where the intensity is at least 0.5% of the maximum intensity was retrieved. At the other data points, the frequency was assigned to be constant.

The retrieved profile is then compared to the integrated density profile along the propagation direction and this is labelled the ‘retrieved’ profile. The comparisons between the actual profile and the retrieved profile at various propagation distance are shown in Figure 2.2.

In order to quantify the error in the retrieved profile with respect to the actual profile, the normalised root mean square error (NRMSE) between those profiles were calculated. The NRMSE is defined as

$$\text{NRMSE} = \frac{\text{RMSE}}{\max[n_c(\xi)] - \min[n_c(\xi)]} \quad (2.10)$$

$$\text{RMSE} = \sqrt{\frac{1}{\xi_f - \xi_0} \int_{\xi_0}^{\xi_f} [n_r(\xi) - n_c(\xi)]^2 d\xi},$$

where $n_r(\xi)$ and $n_c(\xi)$ are the actual and retrieved profiles, respectively, as function of $\xi = z - ct$. The error is integrated from ξ_0 until ξ_f . In this case, $\xi_f - \xi_0 = 0.2$ mm and centred at the centre of the probe pulse.

The values of NRMSE against the propagation distance is plotted in Figure 2.3. As seen in the plot, the relative error is below 10% for short propagation distances. However, as the

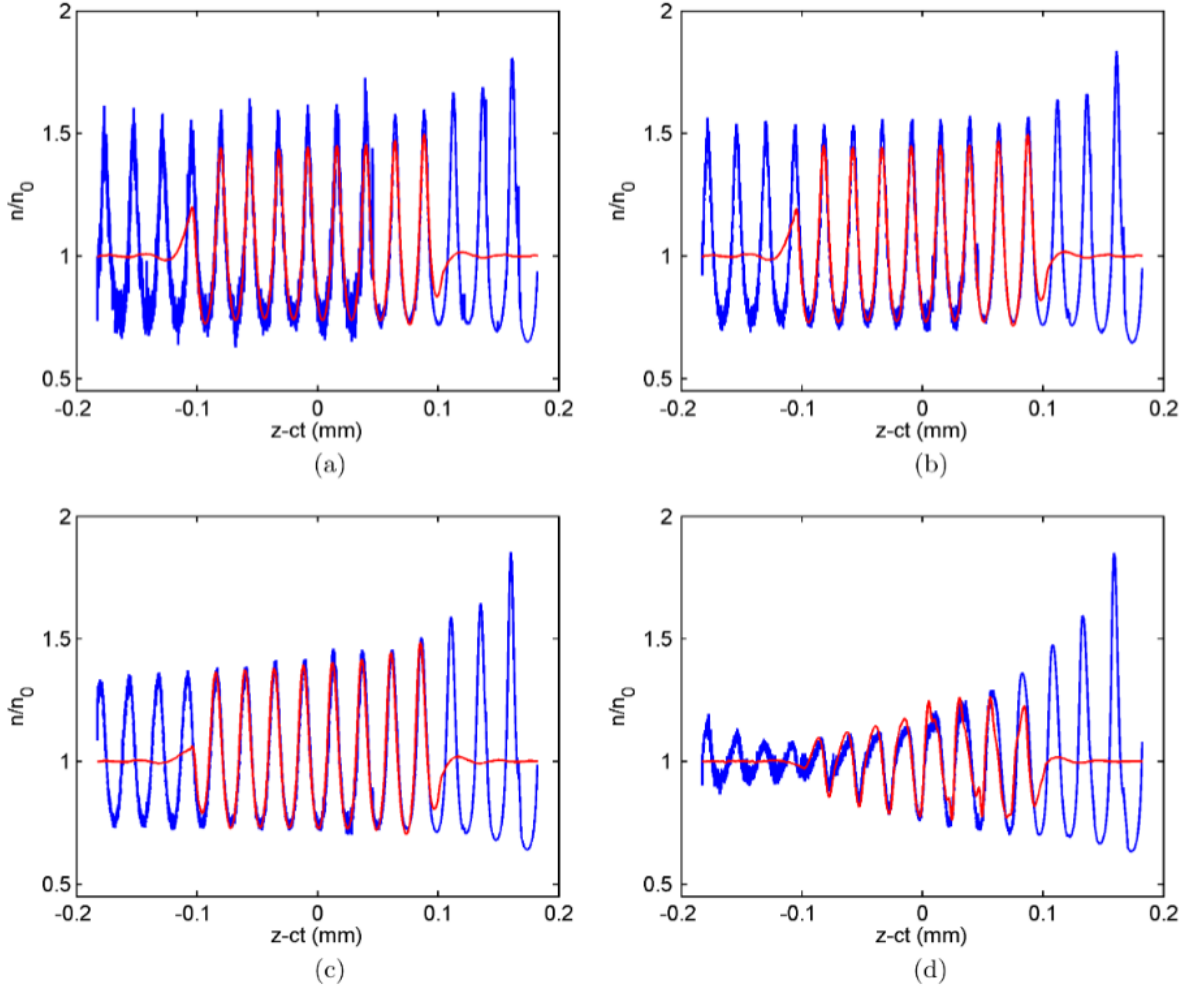


Figure 2.2: Comparison between the actual (red) and the retrieved (blue) profiles when the laser pulses have propagated (a) 0.8 mm, (b) 1.9 mm, (c) 3.8 mm, and (d) 6.3 mm.

probe pulse co-propagates for 6 mm, the relative error increases above 10%. This is due to photon trapping, an effect that is explained later in the next section.

2.4 Analysis and limitations

There are some factors that needs to be considered when doing any photon acceleration measurement. Those factors can spoil and reduce the accuracy of the results. However, these can be minimised by choosing the right parameters.

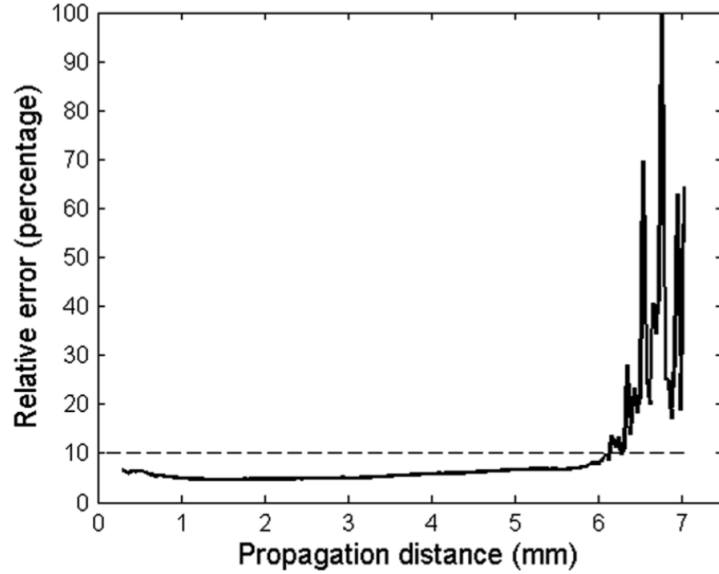


Figure 2.3: Plot of the relative error as function of the propagation distance of the laser probe pulse.

2.4.1 Photon Trapping

When a laser probe pulse propagates inside a plasma wakefield and covers at least one wakefield, some part of it will be blue-shifted and some part of it will be red-shifted. The blue-shifted part acquires higher group velocity and the red-shifted part gets lower group velocity. If the laser pulse propagates for a long enough time inside the wakefield, many photons in the pulse will gather in the troughs of the wakefield and away from the peaks of the wakefield. This reduces the accuracy of the measurement as there are some positions where the laser probe pulse intensity diminishes. This mechanism is called photon trapping [65], a form of modulation instabilities, described in chapter 1 [63, 69, 70].

The photon trapping mechanism is intrinsic to the process and starts from the beginning of the interaction of the probe pulse with the wakefield. However, for small interacting distances, this effect is still negligible. As shown in the Figure 2.4, the intensity of the laser probe pulse does not change much at the beginning of the interaction. As the pulse propagates further, the intensity profile gets disturbed and eventually some parts approach zero. The measurement should be done before this point is reached.

The photon trapping mechanism starts when the laser enters the plasma. However, this

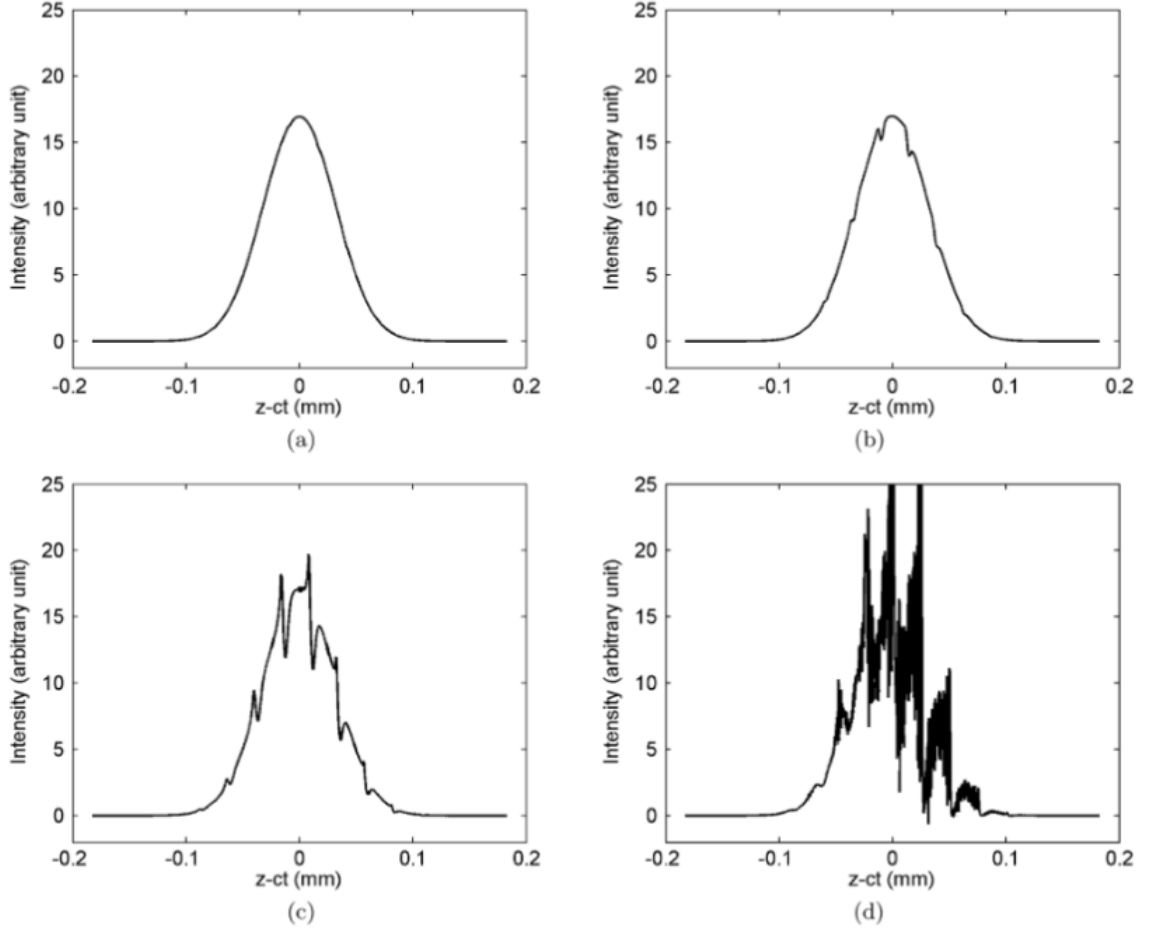


Figure 2.4: Distribution of the laser intensity when the laser has propagated (a) 0.8 mm, (b) 1.9 mm, (c) 3.8 mm, and (d) 6.3 mm. The measurement only takes place from about -0.1 mm to 0.1 mm relative to the center of the pulse. The last picture shows that the photon trapping occurs and causes intensity at some points go to zero.

effect is negligible at the beginning and would become significant after propagating some distance. One can estimate the propagation scale length in which the photon trapping would be significant. The propagation scale length is approximated to be

$$s_{trap} \approx \eta \lambda_p (\omega_0 / \omega_p)^2 (\delta n / n_0)^{-0.5}, \quad (2.11)$$

where λ_p is the plasma wavelength, $\delta n / n_0$ is the relative perturbation of the field, and η is the scaling factor with order less than unity. A safe approximation is to take $\eta = 0.1$.

2.4.2 Diffraction

All the calculations and simulations were done by assuming that the probe pulse is collimated during the interaction. However, for a finite beam size, diffraction will occur leading to an increase in measurement errors. This effect is apparent in the simulations presented on the chapter 3 subsection 3.4.1.

To minimise this effect, one should make the probe beam size large enough such that the Rayleigh length is longer than the interaction length by a factor of two. A simple calculation suggests that the minimum beam waist size should be

$$w_0 > \sqrt{sc/\omega_0}, \quad (2.12)$$

where w_0 is the beam waist size, s the interaction distance between the probe pulse and the wakefield, and ω_0 is the probe's frequency.

2.5 Conclusion

In this chapter, the first simulated measurements of plasma wakefield density profiles with the inclusion of photon acceleration have been presented. The simulations were performed in 1D using particle-in-cell (PIC) code, OSIRIS. In the simulations, the line-integrated density modulation of the wakefield was obtained from the frequency modulation of the probe pulse. When compared to the calculated line-integrated density modulation, the measured quantities agree with error smaller than 10%.

Some effects that can spoil the measurement with photon acceleration have also been discussed in this chapter. They are photon trapping and diffraction effects. It has been shown that, by choosing the right parameters for the probe pulse, one can minimise both sources of error.

Chapter 3

Oblique crossing angle probe

The first snapshots of the wakefield were published in 2006 by Matlis, *et al.* [16], based on Single Shot Spectral Interferometry by Kim, *et al.* [71]. In the experiment, they sent two chirp probe pulses to co-propagate with the wakefield. One of the probe pulse interacts with the wakefield while the second provides the reference pulse. The interaction between the probe pulse and the wakefield imprints a phase modulation on the probe pulse. The phase modulation is related linearly with the electron density modulation of the plasma. After the interaction, both pulses are sent to a spectrometer to perform spectral interferometry to retrieve the spectral modulation. From the retrieved phase modulation, one can interpret the density modulation of the wakefield. This method is called Frequency Domain Holography (FDH). However, this method yields an integrated plasma wakefield density profile and cannot detect any evolution of the wakefield.

Since then, other methods for diagnosing wakefield accelerators have been demonstrated. One of them used the shadowgraphy technique to capture the evolution of the wakefield [23,72]. An ultra-fast probe pulse is fired to cross the wakefield in a perpendicular direction. In a shadowgram, the differences in light intensity seen at the detector plane are proportional to the second spatial derivative of the refractive index. So in the experiments by the Jena / Imperial College groups the density gradients induced by the wakefield bent the path of

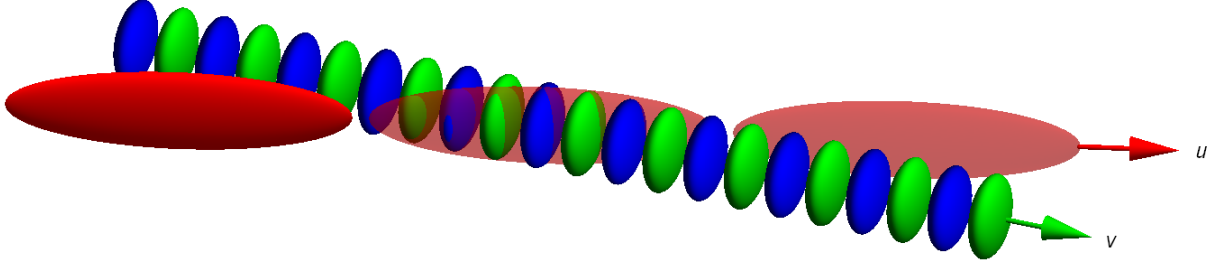


Figure 3.1: An illustration of the proposed method where a median plane slice of a probe pulse crossing wakefield in an oblique angle.

the light that resulted in the observation of the dark and bright pattern on a screen. The advantage of this method is that the temporal and spatial evolution of the wakefield can be obtained simply by delaying the probe pulse. The disadvantage is that it has a limited field of view, that is set principally by the required spatial resolution. It is particularly problematic when one wants to interrogate the evolution of beam-driven plasma wakefields, such as the 10-m long Rb plasma in the AWAKE experiment at CERN [10].

Another technique, Frequency Domain Streak Camera (FDSC) [20] employs spectral interferometry as in FDH, but the probe pulses are fired to cross the wakefield in an oblique angle, instead of co-propagating with the wakefield. After the interaction, a slice on the transverse plane of the probe pulse is captured on a spectrometer. The results of this method can be interpreted to obtain the evolution of the wakefield during the propagation. However, this method cannot capture the transversal structure of the wakefield at a certain point in the plasma.

The new method investigated in this chapter is similar to FDSC, but instead of capturing the transverse plane of the probe pulse, one simply captures the sagittal or median plane of the probe pulse. With this method, one can obtain a snapshot of the wakefield at a certain point in the plasma. To capture the evolution of the wakefield, all one needs to do is firing multiple shots by shifting the position of the probe pulse. Figure 3.1 illustrates this method. This is a complementary method to FDSC in diagnosing a wakefield with a long column and evolving wakefield. The content of this chapter have been published in [27].

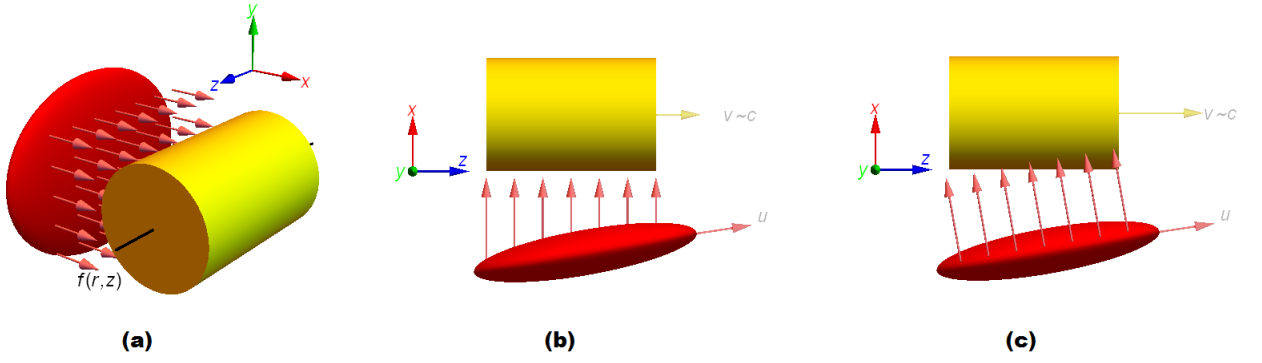


Figure 3.2: (a) An illustration of a probe crossing a cylindrically-symmetric object where one can apply Abel transformation and inversion to retrieve the object's properties. (b) If the longitudinal velocity of the probe equals to the object velocity, then the probe crosses perpendicularly in the object's frame, and the Abel transformation and inversion still apply. (c) Otherwise, the probe does not cross perpendicularly and the original Abel transformation does not work.

Another proposed method to capture the evolution of the wakefield by obtaining several pictures of the wakefield in a single shot is explained in the chapter 5.

3.1 Modified Abel transformation

Denote the group velocity of the probe pulse as v_g , the wakefield phase velocity as u_p , and the crossing angle is θ . If the longitudinal velocity of the probe pulse equals to the wakefield velocity, i.e. $v_g \cos \theta = u_p$, the probe pulse crosses the wakefield perpendicularly in the wakefield frame, as shown in Figure 3.2(b). If the wakefield has a cylindrical symmetry, then the electron density can be obtained by applying the inverse Abel transformation to the probe phase modulation, because the phase modulation is proportional to the integration of the electron density modulation over the interaction distance as given in (2.2).

However, if the longitudinal velocity of the probe pulse does not equal to the wakefield velocity, the probe pulse does not cross perpendicularly in the wakefield frame, and the inverse Abel transformation does not work any more, as in Figure 3.2(c). Motivated by this, a novel transformation was derived from Abel transformation where the probe does not cross perpendicularly with respect to the diagnosed object.

Assume an object with cylindrical symmetry as in Figure 3.2 has a property of $f(r, z)$, where r is the distance from the symmetry axis. The property is diagnosed with a probe and the probe reads $g = \int_p f d\mathbf{l}$ with p is the path of the probe and $d\mathbf{l}$ is the small part of the path. If the probe crosses the object perpendicularly, it can be obtained by Abel transformation,

$$g(y, z) = 2 \int_y^\infty \frac{f(r, z)r dr}{\sqrt{r^2 - y^2}}. \quad (3.1)$$

In most cases, people obtain $g(y, z)$, but interested in $f(r, z)$. To obtain $f(r, z)$, one can apply the inverse Abel transformation,

$$f(r, z) = -\frac{1}{\pi} \int_r^\infty \frac{\partial g}{\partial y}(y, z) \frac{dy}{\sqrt{y^2 - r^2}}. \quad (3.2)$$

In cases like Figure 3.2(b), one can employ the inversion above as the z -slices are independent to each other. However, for cases like in Figure 3.2(c), one part of the probe crosses multiple z -slices. Therefore, the equations above were modified to handle this case. For $a \equiv [(\cos \theta - u_p/v_g)/\sin \theta]$, the forward and inverse transformation can be written as,

$$\tilde{g}(y, k) = 2 \int_y^\infty \cos \left(ka \sqrt{r^2 - y^2} \right) \frac{r \tilde{f}(r, k)}{\sqrt{r^2 - y^2}} dr \quad (3.3)$$

$$\tilde{f}(r, k) = -\frac{1}{\pi} \int_r^\infty \frac{\partial \tilde{g}}{\partial y}(y, k) \frac{\cosh \left(ka \sqrt{y^2 - r^2} \right)}{\sqrt{y^2 - r^2}} dy, \quad (3.4)$$

where the tilde hat denotes the Fourier-transformed signal from the spatial domain z to the spatial frequency domain k , i.e. $\tilde{h}(k) = \int_{-\infty}^\infty h(z)e^{-ikz} dz$. Setting $a = 0$ brings the equations above back to the original Abel transformation. The complete derivations of the transformation and its inversion as well as its numerical implementation are presented in Appendix A.1.

For our case with a wakefield and an oblique-angle optical probe, the variables f and g need to be replaced by the corresponding parameters in the wakefield and the probe. Equations (2.1) and (2.2) and the relation $g = \int_p f dl$ yield

$$f(r, z) = \Delta n(r, z) \quad (3.5)$$

$$g(y, z) = -\frac{2\omega_0 c n_0}{\omega_p^2} \Delta\phi(y, z), \quad (3.6)$$

where Δn is the electron density modulation, ω_0 and ω_p is the probe and plasma frequency, respectively, n_0 is the unmodulated electron density, c is the speed of light, and $\Delta\phi$ is the phase modulation of the probe pulse.

3.2 3D simulations of measurement

The main objective of this chapter is to confirm that the oblique crossing angle with the modified Abel transform can be utilised to retrieve the plasma wakefield density profile using simulations. To achieve that, 3D simulations were performed using OSIRIS [17].

A baseline set up of the simulation uses a plasma with density of $n_0 = 2 \times 10^{19} \text{ cm}^{-3}$, divided into 3D grids with size $(12 \times 120 \times 120) \text{ nm}^3$ each, and a moving window that contains $9750 \times 400 \times 300$ cells. Periodic boundary condition were used on the transverse boundaries. One particle per cell was found sufficient for this purpose. Repeated simulations with a larger number of particle per cell were performed and the results were similar.

To drive the wakefield, a spherical Gaussian electron beam with radius $\sigma_r = 4.4 \text{ }\mu\text{m}$ and peak density of $n_e = 0.33n_0$ was used. Each electron in the beam was given an initial momentum of $p_e/m_e c = 45 \times 10^3$, with m_e is the mass of electron. As this is a simulation to measure the wakefield, the type of the driver is not important. An electron beam was chosen to reduce the simulation time.

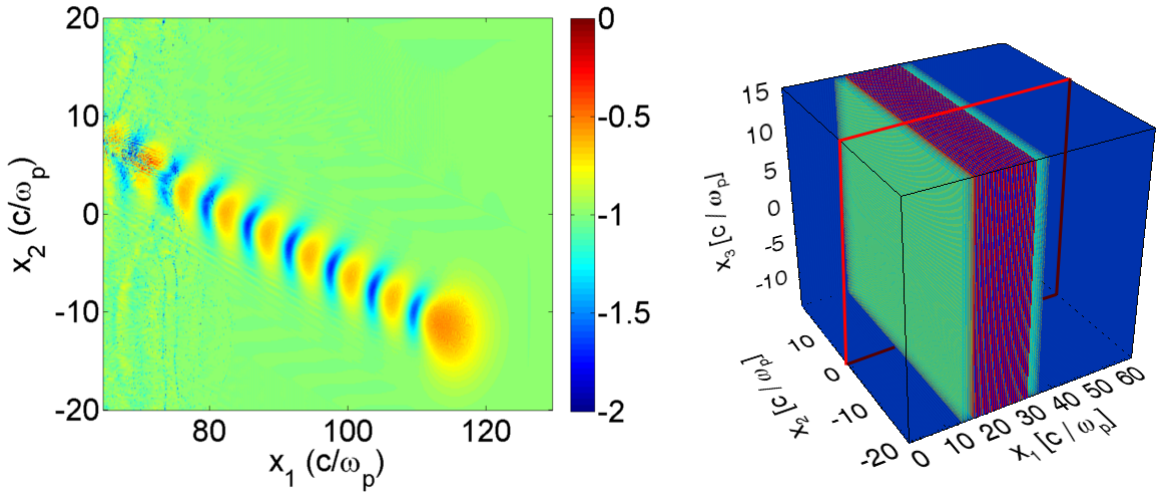


Figure 3.3: (Left) An x_2 - x_1 -plane slice of the wakefield, where the driver comes at an angle with respect to the x_1 axis. (Right) Electric field profile of the probe pulse where it propagates along the x_1 axis. The red lines shows the plane where the slices were taken.

The probe pulse was a plane wave with wavelength of 400 nm, duration 53 fs, and normalised intensity $a_0 = 0.01$, coming with an angle $\theta = 20^\circ$ to the wakefield longitudinal axis. As a parameter scan, the peak density of the driver beam was varied from $0.1n_0$ to $0.35n_0$, the crossing angle from 25° down to 5° , and the probe's wavelength from 260 to 800 nm.

In each simulation, five slices of the probe pulse were taken for the analysis. These slices cross the centre of the wakefield at $s = \{28, 39, 49, 60, 67\}$ μm measured from the beginning of the plasma. Each simulation took approximately 8000 CPU hours on average to finish. Figure 3.3 illustrates the simulation and the slicing of the probe pulse.

When a simulation finished, the plasma density profile, $n(r, z)$, was taken and labelled as the “actual” density profile. Also, from each taken slice of the probe pulse, the frequency modulation profile, $\Delta\omega(y, z)$ was calculated using the Wigner transform, as explained in section 2.2.3. From the frequency modulation, the phase modulation profile, $\Delta\phi(y, z)$ was calculated, as $\Delta\omega \approx c\partial\Delta\phi/\partial z$. Thus, using the calculated frequency modulation from the probe, equations (3.3)-(3.6), the measured density profile of the plasma wakefield was calculated. This was labelled as the “retrieved” density profile. Both profiles, “actual” and “retrieved” density profiles were then compared.

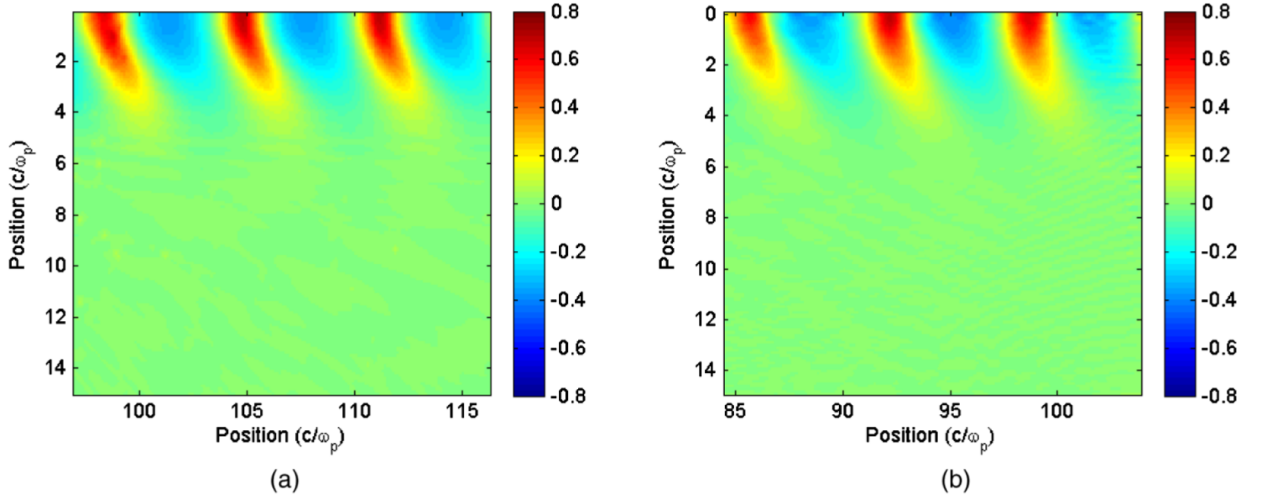


Figure 3.4: The electron density profile from the baseline simulation (a) obtained directly from the simulation and (b) retrieved by calculating from the frequency modulation profile of the probe pulse's slice that crosses the wakefield.

3.3 Simulation results and analysis

As a first step in this comparison, a slice with $s = 28 \mu\text{m}$ in the baseline simulation was taken just after the probe slice leaves the plasma wakefield. The retrieved density profile of the plasma wakefield was then compared to the actual density profile obtained directly from the simulation. Figure 3.4 shows a good qualitative agreement between the two density profiles.

A quantitative comparison between the retrieved and actual density profiles was done by comparing the peaks and troughs of the wakefield for every slice in the baseline simulation. There are five slices of the probe pulse taken from the simulation that cross the wakefield at $s = \{28, 39, 49, 60, 67\} \mu\text{m}$ after the wakefield driver enters the plasma. Each slice was taken just after it left the wakefield. A similar comparison was also done for a simulation with smaller driver density to drive smaller wakefield amplitude with other parameters remaining the same. The comparisons of the peaks and troughs values between the retrieved and the actual density profiles are shown in Figure 3.5. The measured peaks and troughs values agrees well with the actual value, taken directly from the simulation.

In order to get a bigger picture on the regime where this diagnostic works well, a com-

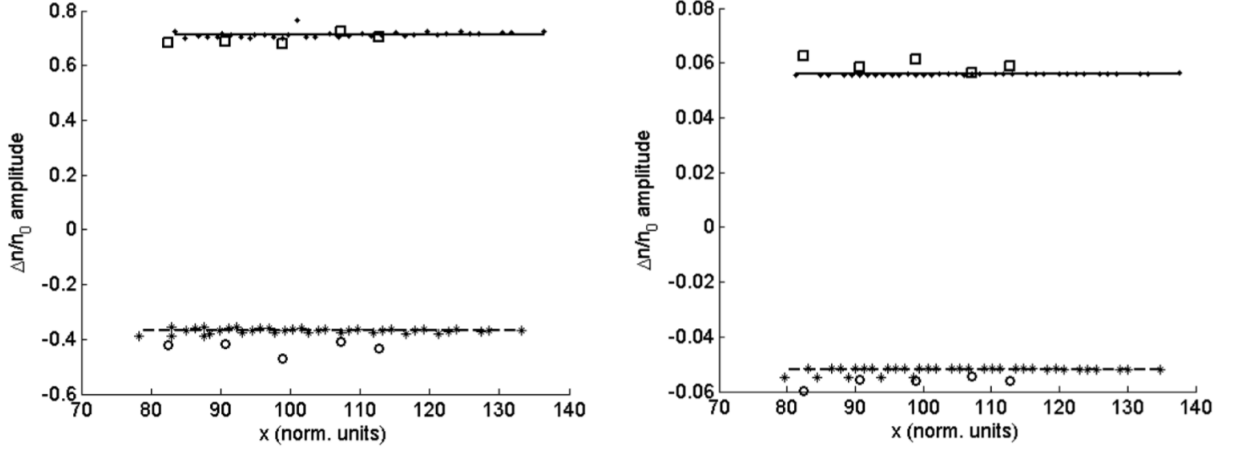


Figure 3.5: Comparison of the peaks and troughs values of the retrieved and actual density profiles for (left) non-linear wakefield amplitude and (right) small wakefield amplitude.

parison was made by simulating different values of peak driver density, crossing angle, and the probe's frequency. From the retrieved and the actual density profile, the amplitude of the wakefield was taken and compared between these two profiles. An amplitude is defined as the mean value of the peaks and the troughs of the central lineout of the wakefield. The slice that crosses the wakefield at $s = 28 \mu\text{m}$ was taken. The comparisons of the amplitude values for different driver beam densities, probe's frequencies, and crossing angles are shown in Figure 3.6.

As seen in Figure 3.6, varying the driver density to generate a non-linear wakefield for $\Delta n/n_0 < 0.7$ does not reduce the accuracy of the diagnostic method. Changing the probe's frequency from $10\omega_p$ to $30\omega_p$ also does not affect the measurement's accuracy. However, reducing the crossing angle tends to make the measurement less accurate. The measured values for small angles underestimate the actual amplitude of the wakefield. This is caused by the diffraction effect, discussed earlier in Chapter 2.

When the probe pulse interacts with the wakefield, it adds the phase modulation over the interaction distance. However, if the transverse size of the imparted phase modulation is small enough, the additional modulation gets diffracted before the probe pulse leaves the wakefield. An example of this diffraction effect is shown in Figure 3.7. Here it shows the measured densities evolution of a slice from the probe pulse as it propagates further from the

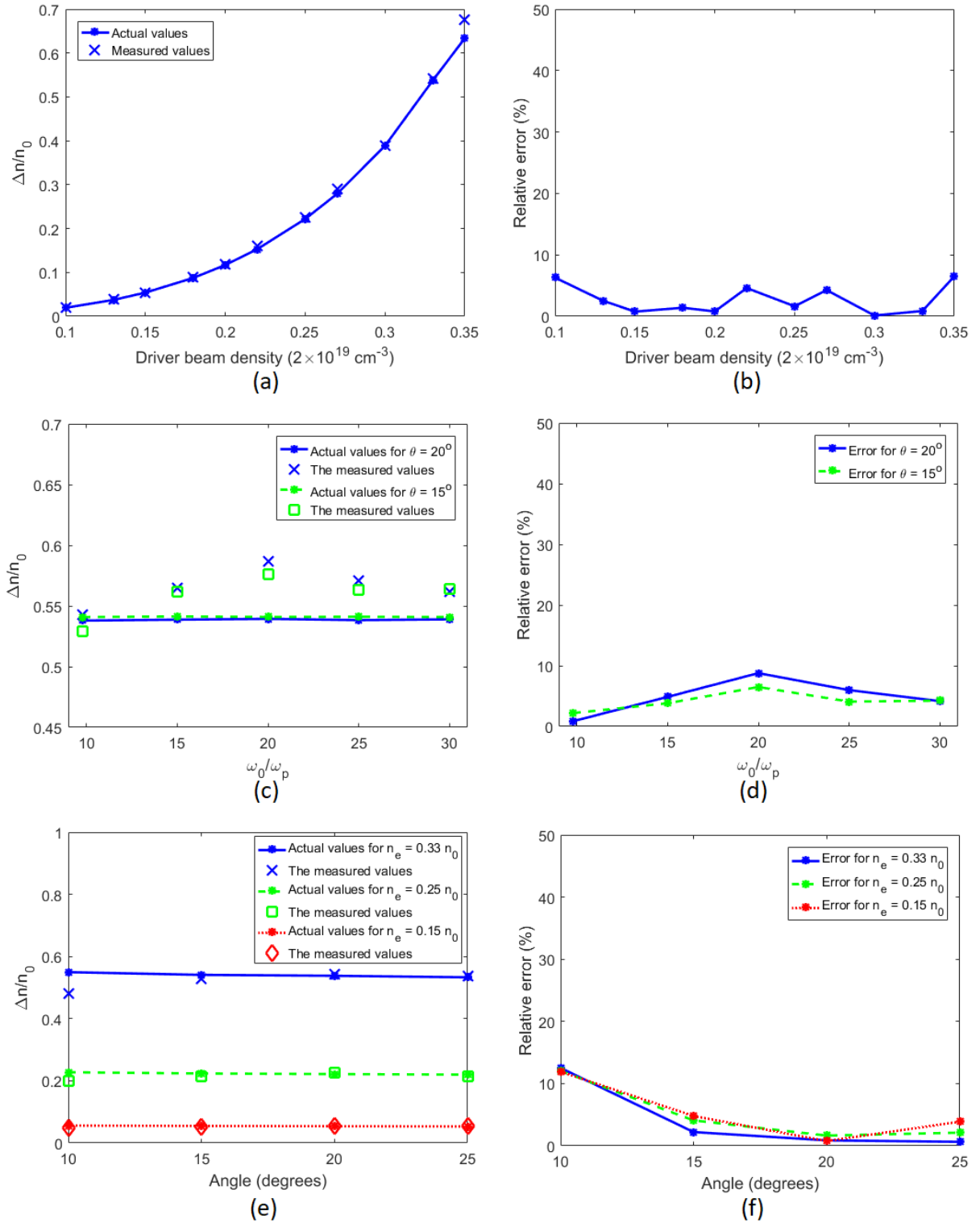


Figure 3.6: Comparison of the amplitude of the wakefield from the actual and the measured values for simulations with different (a-b) driver peak densities and wakefield amplitudes, (c-d) probe's frequencies, and (e-f) crossing angles. The left pictures plot the values and the right pictures show the relative error between the actual and measured values.

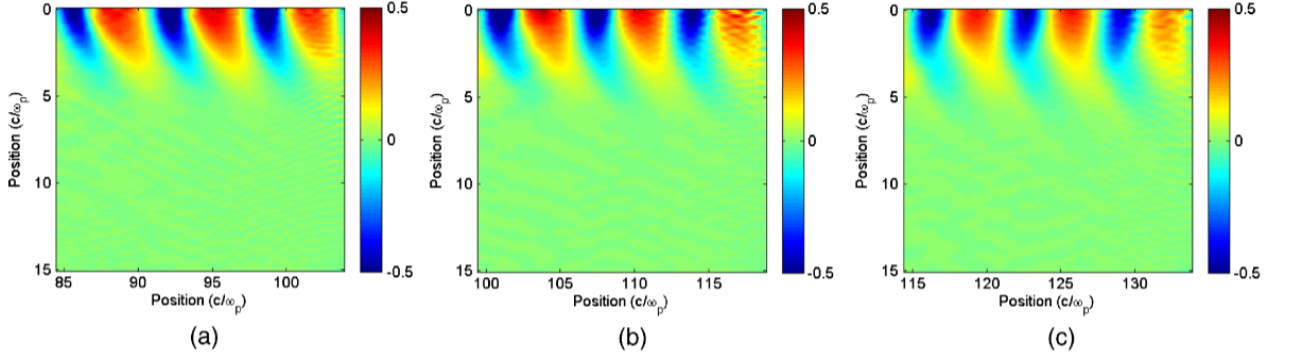


Figure 3.7: The measured density profiles at different positions as the probe pulse propagates away from the wakefield. Figure (a) is when the probe pulse just leave the wakefield, (b) and (c) is when the probe pulse already propagates for 18 and 36 μm from the figure (a), respectively.

wakefield after the interaction. If the crossing angle decreases, then the interaction distance increases, thus the diffraction can occur during the interaction and reduces the value of density profile it measured.

3.4 Limitations

In order to perform a good measurement, it is clear that correct parameters need to be chosen. The constraints on the chosen parameters relies on several factors that limits the method, which are diffraction and large crossing angles.

3.4.1 Diffraction

This effect is already apparent in Figure 3.7. Let the electric field profile of the probe be $U_0(\mathbf{r}, t)$. If the probe undergoes a slight phase modulation, $\delta\phi(\mathbf{r}, t)$, the electric field profile can be written as $U_0(\mathbf{r}, t)e^{i\delta\phi(\mathbf{r}, t)} \approx U_0(\mathbf{r}, t) + iU_0(\mathbf{r}, t)\delta\phi(\mathbf{r}, t)$. The right hand side of the equation shows that there is another term that can be regarded as a separate electromagnetic wave with the same frequency and amplitude profile of $|U_0(\mathbf{r}, t)||\delta\phi(\mathbf{r}, t)|$. Even though the probe transverse size is large and collimated, if the phase modulation size is small, it will be

diffracted much earlier than the probe itself.

To prevent the diffraction effect to be considerable during the probe-wakefield interaction, the Rayleigh length of the phase modulated part needs to be longer than the interaction length. If the probe has wavelength of λ_0 and the wakefield radius has a size parameter of r_p , the minimum angle is approximately

$$\theta \gtrsim \arcsin \left(\frac{\lambda_0}{r_p \sqrt{\pi}} \right). \quad (3.7)$$

In the simulated case, the minimum crossing angle is approximated to be 8° . This explains why the relative error on Figure 3.6 (f) increases when the angle is down to 10° and exceeds 15% when it is down to 5° .

3.4.2 Large crossing angle

In the previous subsection, the minimum crossing angle was derived based on the diffraction effect. In this subsection, the upper limit of the crossing angle is determined by limiting the error generated by the modified Abel inversion in equation (3.4). As the modified Abel inversion has a $\cosh(ka \dots)$ term inside the integral, the error in obtaining the density modulation rises exponentially as $|a| = |(\cos \theta - u_p/v_g)/\sin \theta|$ increases.

To quantify the error in obtaining the density modulation, assume the wakefield takes a Gaussian profile with radius r_p for its transverse profile and sinusoidal on its longitudinal profile. Although this assumption does not hold for non-linear wakefields, it is simple enough to quantify the error propagation. In this case, the phase modulation is given by

$$\Delta\phi = \Delta\phi_{max} \sin(\omega_p z/c) e^{-y^2/r_p^2}, \quad (3.8)$$

where $\Delta\phi_{max}$ is the amplitude of the phase modulation. From equation (3.4), the density

modulation is obtained as

$$\frac{\Delta n}{n_0} = \left(\frac{\Delta n}{n_0} \right)_{max} \sin(\omega_p z/c) e^{-r^2/r_p^2}, \quad (3.9)$$

where

$$\left(\frac{\Delta n}{n_0} \right)_{max} = -\frac{4}{\pi} \frac{\omega_0 c}{\omega_p^2 r_p} e^{\omega_p^2 a^2 r_p^2 / 4c^2} \Delta \phi_{max}. \quad (3.10)$$

As can be seen on the equation above, the error or noise in reading the phase modulation is amplified by an exponential factor of a^2 . If the crossing angle is larger than $\arccos(u_p/v_g)$, increasing the angle increases the absolute value of a . This sets an upper limit on the crossing angle. For example, if there is a 0.01 rad noise in the phase modulation and one wants to get the error of $\Delta n/n_0$ below 20%, then the upper limit (with the baseline parameters above) is around 70° .

3.4.3 Method to acquire the phase modulation

Another limits on the oblique angle probe pulse might come from the method to obtain the phase modulation profile. For example, in chapter 4, a method called Single-shot Supercontinuum Spectral Interferometry (SSSI) [71] is employed to get the phase modulation profile. This method is good for obtaining moderate number of phase modulation (~ 0.1 rad, depends on the noise), but it is quite hard to use if the phase modulation is small. The sensitivity of the method to acquire the phase modulation sets the upper limit of the crossing angle.

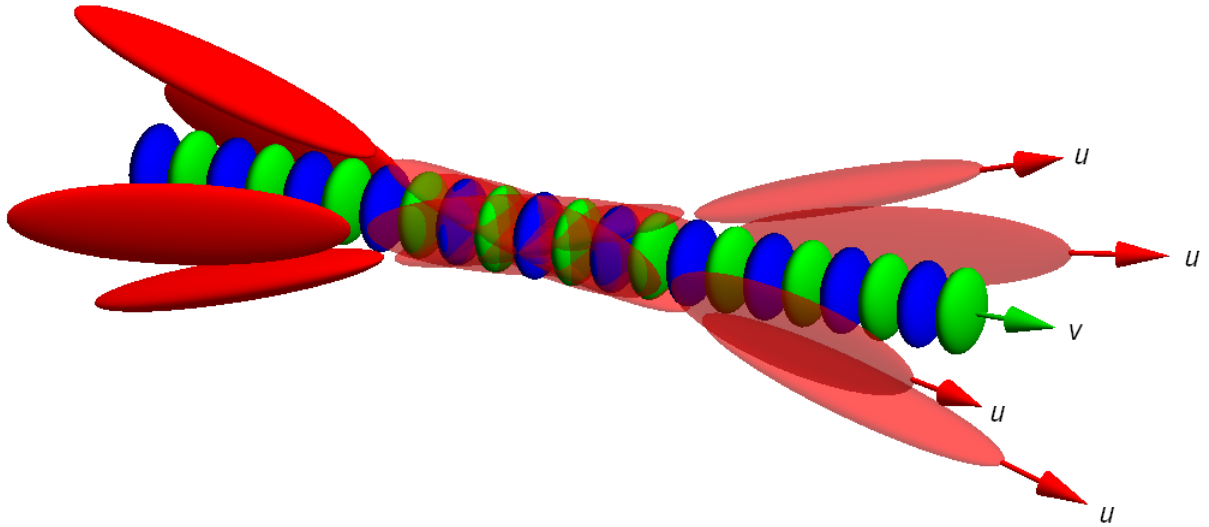


Figure 3.8: Tomography set up for a moving object.

3.5 Fast moving object tomography

The expressions of modified Abel transformations in equation (3.3) and (3.4) are only valid if the object has cylindrical symmetry. However, if the object does not have cylindrical symmetry (or is not approximately symmetry), a tomography method needs to be employed.

Current tomography technique is usually employed by sending multiple probes to the object with all the probes are perpendicular to an axis. However, if the object is moving, it poses another challenge to match the longitudinal velocity of the probe with the object's velocity to make the probe crosses perpendicularly relative to the object (see Figure 3.2(b) and (c)). Therefore, an analytical expression of the transformation was developed where the object is not cylindrically symmetric and the probes are crossing with an angle. Figure 3.8 illustrates the case considered in this section.

A general profile in cylindrical coordinate can be expressed by an infinite sum of profiles with a cylindrical symmetry. This is known as Fourier series where a function $h(r, z, \theta)$ can be expressed as [73]

$$h(r, z, \theta) = \sum_{m=-\infty}^{\infty} h_m(r, z) e^{im\theta}, \quad (3.11)$$

where

$$h_m(r, z) = \frac{1}{2\pi} \int_0^{2\pi} h(r, z, \theta) e^{-im\theta} d\theta. \quad (3.12)$$

An object with cylindrical symmetry has a non-zero element only at $m = 0$ and $h(r, z, \theta) = h_0(r, z)$.

Denoting the measurable profile from the probe as f and the diagnosed object's profile as g , the generalised Abel transformation for non-symmetric object and non-perpendicular probe can be expressed as,

$$\tilde{g}_m(y, k) = 2 \int_{|y|}^{\infty} \tilde{f}_m(r, k) \cos \left[ka\sqrt{r^2 - y^2} - m \arccos(y/r) \right] \frac{r dr}{\sqrt{r^2 - y^2}} \quad (3.13)$$

$$\tilde{f}_m(r, k) = -\frac{1}{\pi} \int_r^{\infty} \frac{\partial \tilde{g}_m(y, k)}{\partial y} \cosh \left[ka\sqrt{y^2 - r^2} - m \operatorname{arccosh}(y/r) \right] \frac{dy}{\sqrt{y^2 - r^2}}. \quad (3.14)$$

This is a combination of the transformations (3.3)-(3.4) and the equations 10 and 18 in [73]. The derivations of the transformations above are given in Appendix A.2.

3.6 Conclusions

In this chapter, a novel method for diagnosing a wakefield was developed, one where an oblique angle probe takes snapshots of the wakefield at the different positions in a plasma column. A novel transformation was also derived analytically for the case where a probe crosses the object in a non-perpendicular angle. The method was confirmed using 3D PIC simulations and shows that measurements can be done with error less than 15%.

Additionally, new transformations were derived relating the probe parameter and the diagnosed object's parameter where the object does not have a cylindrical symmetry and the probes are not sent perpendicularly to the axis. In short, the first transformations have been presented for tomography of a fast-moving object.

Chapter 4

Experiment on plasma wakefield diagnostic

An experiment to diagnose plasma wakefields and to qualify the experimental platform described in chapters 2 and 3 using an oblique-angle probe was performed in the Gemini laser facility's Target Area 2 in the Central Laser Facility, STFC Rutherford Appleton Laboratory. The experiment used helium, nitrogen, and methane gases as the target. An 800 nm / 45 fs laser pulse containing 0.5 J of energy was used to both ionise the gas and drive the wakefield. At this point, two second harmonic pulses crossed the wakefield at an oblique angle. As the primary diagnostic of the probe pulses, the single-shot supercontinuum spectral interferometry (SSSI) method was employed to read the phase modulation imprinted onto the probe. It is demonstrated here in this chapter that modulations with wavelength similar to the wakefield from the experiment was observed for the first time. A new model of wakefield acceleration in a cluster plasma is presented to explain the observations.

4.1 Single-shot supercontinuum spectral interferometry

In FDH, FDSC, as well as the proposed method, the phase modulation of a chirped probe pulse is obtained by sending a reference unmodulated chirped probe pulse with suitable delays to a spectrometer. This method of combining two chirped probe pulses into a spectrometer was first introduced by Kim, *et al.* as single-shot supercontinuum spectral interferometry [71].

Let two identical chirped pulses co-propagate with delay τ to each other. One of the pulses has some phase modulation, $\Delta\phi(t - \tau)$. Also, let the unmodulated pulse electric field be represented as $E_0(t)$. Then the total electric field of the pulses can be written as

$$E(t) = E_0(t) + E_0(t - \tau)e^{i\Delta\phi(t-\tau)}. \quad (4.1)$$

When both pulses enter the spectrometer, the modulus of the electric field in the frequency domain is generated. Putting a tilde hat on variables on the frequency domain, it is expressed as,

$$|\tilde{E}(\omega)|^2 = 2|\tilde{E}_0(\omega)|^2 \left\{ 1 + \cos \left[\omega\tau + \Delta\tilde{\phi}(\omega) \right] \right\}. \quad (4.2)$$

The equation above assumes that the second pulse does not have any modulation in terms of intensity, just the phase. Now the task is to obtain the phase modulation, $\Delta\tilde{\phi}(\omega)$, from the signal above.

There are two terms on the right hand side of the equation above, the constant part and the oscillating part. When a Fourier transform is applied to the term above, there will be three significant peaks. The central one corresponds to the constant part, and the others correspond to the positive and negative frequency of the oscillating part. Taking the positive frequency of the oscillating part can be done by zeroing the area around the central peak

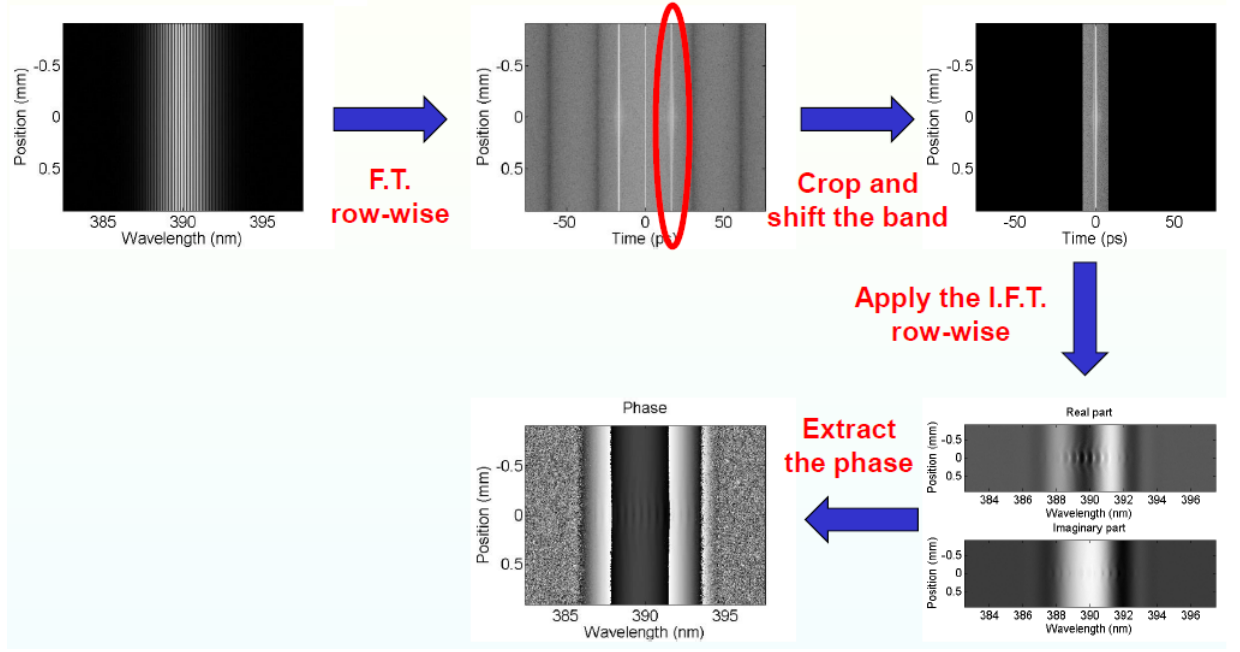


Figure 4.1: Steps to retrieve the phase modulation in SSSI.

and the negative peak. By doing that, it leave us with

$$\tilde{F}(\omega) = 2|\tilde{E}_0(\omega)|^2 \exp \left[i\omega\tau + i\Delta\tilde{\phi}(\omega) \right]. \quad (4.3)$$

In order to remove the $\omega\tau$ term, one can shift the Fourier transformed signal so that the positive peak is located at the centre. It yields

$$\tilde{F}_1(\omega) = 2|\tilde{E}_0(\omega)|^2 \exp \left[i\Delta\tilde{\phi}(\omega) \right]. \quad (4.4)$$

Now the phase modulation in frequency domain is relatively straightforward to obtain, i.e. $\Delta\tilde{\phi}(\omega) = \arg \left[\tilde{F}_1(\omega) \right]$. The procedure to obtain the phase modulation is illustrated in figure 4.1.

The next step in obtaining the important information is to convert the phase modulation from the frequency domain to the temporal domain. The relation between the phase

modulation in the temporal domain and in the frequency domain is expressed as

$$E_0(t)e^{i\Delta\phi(t)} = \mathcal{F}^{-1} \left[\tilde{E}_0(\omega)e^{i\Delta\tilde{\phi}(\omega)} \right], \quad (4.5)$$

with \mathcal{F} represents the Fourier transform and \mathcal{F}^{-1} the inverse Fourier transform.

For a Gaussian chirp pulse with width τ_0 , frequency ω_0 , bandwidth $\Delta\omega$, linear chirp coefficient α , and assumptions $\alpha^2\tau_0^4 \gg 1$ and $\alpha \ll (\Delta\omega)^2$, one can obtain

$$\Delta\phi(t) \approx \Delta\tilde{\phi}(\omega_0 + \alpha t). \quad (4.6)$$

4.2 Experimental design

4.2.1 Schematic in the target chamber

We were awarded 12 weeks of facility access time from June to November 2016 at the Gemini laser facility's Target Area 2 (TA2). This state of the art Ti:Sapphire laser facility is managed by the Central Laser Facility (CLF), STFC Rutherford Appleton Laboratory and is provided on a "free at the point of access" basis to academic institutions on application to STFC's Facility Access Panel. The time was awarded for an experimental proof-of-concept on diagnosing wakefields using an oblique angle probe, as explained in chapter 3. The detailed planning, for which I was principally responsible (under Professor Norreys' guidance) involved the conceptual design of the optical layout and the development of the diagnostic instruments. In this task, I was greatly assisted by Dan Symes, Chris Hooker, Steve Hawkes, Chris Gregory, Luke Ceurvorst, Kevin Glize, James Holloway, and other colleagues.

The target chamber in TA2 has a size of (1000×800) mm². For simplicity of diagnostic access, it was decided to use a gas jet as the target, an off-axis parabolic mirror as the focusing optic for the wakefield driver pulse, and frequency-doubled laser pulse as the probe.

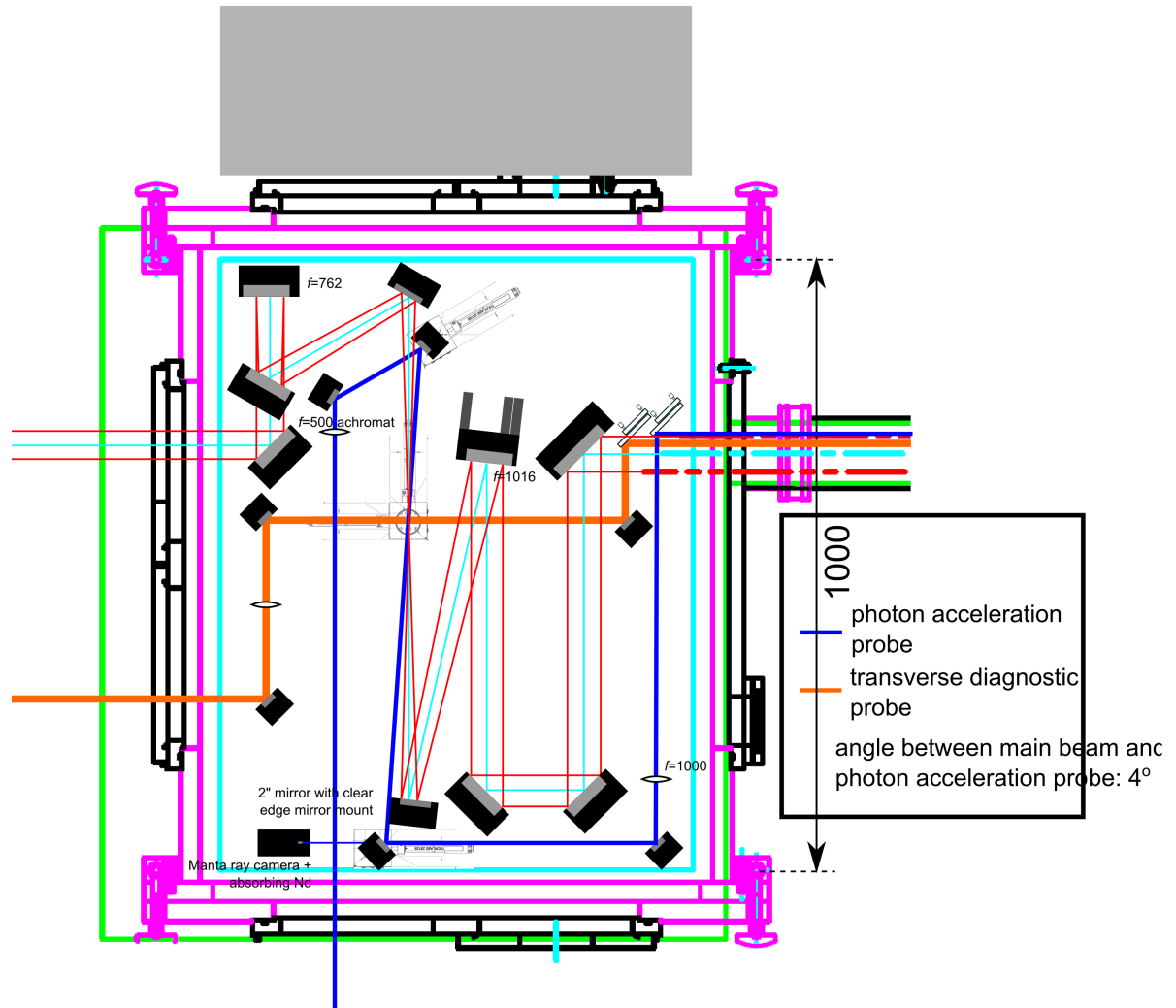


Figure 4.2: Schematic inside the target chamber. The red and cyan lines show the main laser beam line. The blue line is for the frequency-doubled probe pulses for the wakefield diagnostic and the orange line show the line for the perpendicular probe for plasma density diagnostic.

For the diagnostics, the principal one was to use an oblique crossing probe as the main diagnostic of the wakefield, supplemented by a perpendicular crossing probe to diagnose the plasma density profile, an electron spectrometer, main laser pulse exit mode, and other diagnostics around the target and before the target. The schematic of the target chamber is shown in Figure 4.2.

After exiting the target chamber, the blue line was directed into an imaging system with 40-50 times magnification and a spectrometer. The perpendicular probe (orange line) was directed into a Mach-Zehnder interferometer and a camera to image the fringes.

The $f = 1016$ mm off-axis parabolic mirror with $F\# = 17$ was chosen because it can focus down a 800 nm laser pulse down to around $2w_0 \approx 30 \mu\text{m}$ diameter (for an ideal Gaussian beam, $2w_0 \approx 17 \mu\text{m}$, assuming $M^2 \sim 2$) and it is in the same order with the predicted wakefield wavelength for the chosen density, i.e. $n \approx (5 - 20) \times 10^{18} \text{ cm}^{-3}$ and $\lambda_p \approx (7.5 - 15) \mu\text{m}$. Also, it was the most common set-up used in CLF experiments, so obtaining the off-axis parabolic mirror with the corresponding parameters was relatively straightforward for the facility staff.

Please note that all beam diameters are expressed in terms of twice the beam waist size, i.e. $2w_0$. The beam waist size is defined as $w_0 = d_{FWHM}/\sqrt{2\ln 2}$ where d_{FWHM} is the full-width-half-maximum width of the beam.

Laser probe pulse spot size at the target was designed to be larger than the main driver laser pulse spot size to make it easier to find the wakefield structure within the probe pulse. As a result, the probe pulse was designed to enter the target chamber with diameter 9 mm and focused using $f = 1000$ mm focusing lens. With that parameter and with 400 nm wavelength, it is predicted that the probe pulse would have diameter of $110 \mu\text{m}$ ($2w_0 \approx 57 \mu\text{m}$ with $M^2 \sim 2$). It is around 3 times larger, in diameter, than the main driver laser pulse spot size.

To optimise the crossing angle, simulations were performed using those parameters with $n = 5 \times 10^{18} \text{ cm}^{-3}$ and $n = 2 \times 10^{19} \text{ cm}^{-3}$ for 3 mm propagation length, then calculated the predicted phase modulation on the probe. The density profile from the simulation with $n = 5 \times 10^{18} \text{ cm}^{-3}$ is shown in Figure 4.3. The predicted phase modulation for 400 nm probe pulse crossing with an angle of 4° is shown in Figure 4.4. The phase modulation was calculated using the equations (3.3), (3.5), and (3.6).

From the images in Figure 4.4, it can be seen that the phase modulation is around ~ 60 mrad. However, based on my conversation with Stephen Dann (a DPhil student in Oxford who did FDH and SSSI in Dr. Laura Corner's and Prof. Roman Walczak's group in Oxford), the minimum phase modulation that FDH can detect is around 80 mrad. As a result, the 4° crossing angle determines the limit for the angle's order of magnitude, because increasing

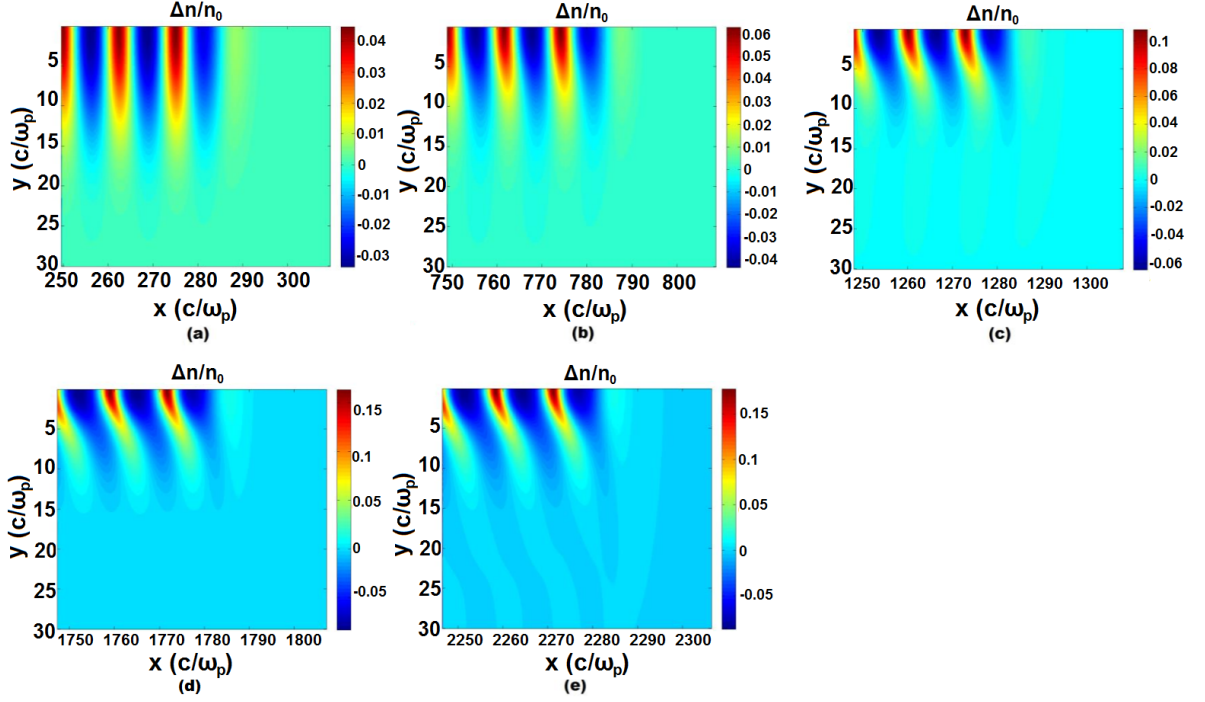


Figure 4.3: Density profiles of the wakefield with $n = 5 \times 10^{18} \text{ cm}^{-3}$ after (a) 0.2 mm, (b) 0.8 mm, (c) 1.4 mm, (d) 2.0 mm, and (e) 2.6 mm propagation.

the crossing angle decreases the interaction length and, hence, the phase modulation. That sensitivity limit forms the basis of the signal-to-noise ratio of the measurement and, although the parameters might be somewhat different in our case, it formed a useful estimate for the experiment design.

It was found that increasing the plasma density significantly helped. The density profile and the phase modulation from the simulation with plasma density $n = 2 \times 10^{19} \text{ cm}^{-3}$ is shown in Figure 4.5 and 4.6, respectively. The main driver laser pulse is depleted after travelling 1.5 mm inside the plasma, based on the simulation. In this case, the phase modulation at the beginning is around 120 mrad and then increases up to more than 1 rad before the pulse is depleted. Thus, this case does not have any problem with the sensitivity of FDH.

4.2.2 Schematic on the probe table

In order to do SSSI, two co-propagating probe pulses were needed. In this case, the pulses were designed to be generated using a Michelson interferometer. The frequency-doubling

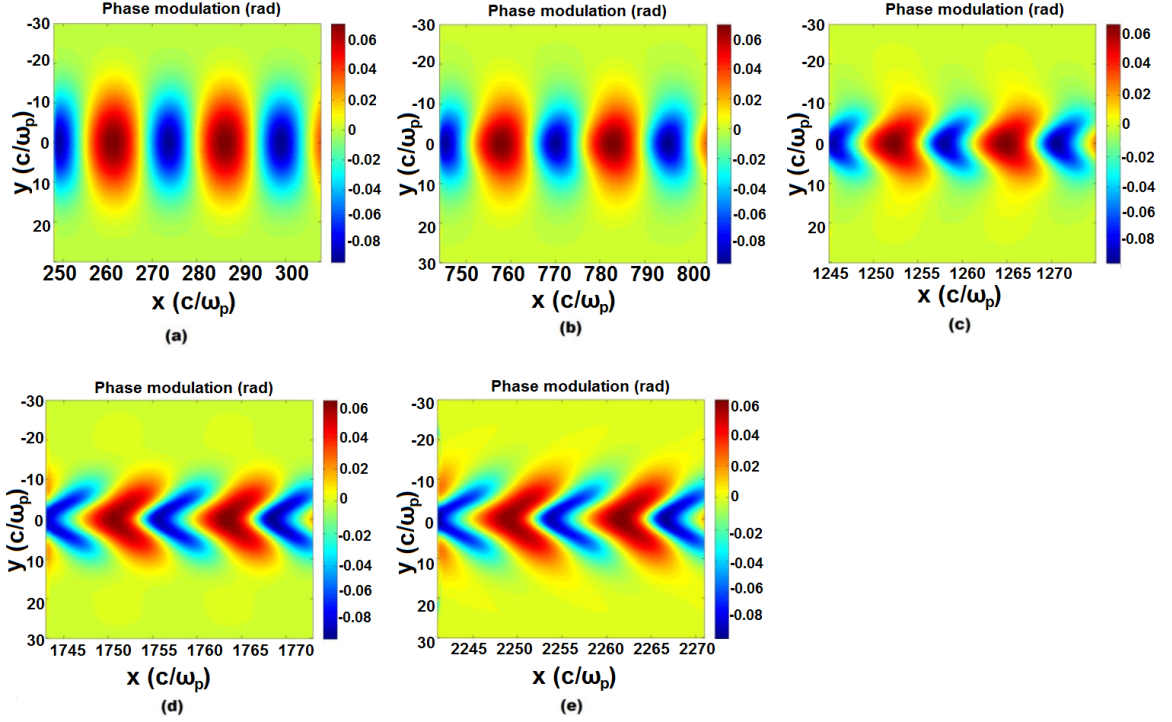


Figure 4.4: Phase modulation of the probe pulse with plasma density $n = 5 \times 10^{18} \text{ cm}^{-3}$ after (a) 0.2 mm, (b) 0.8 mm, (c) 1.4 mm, (d) 2.0 mm, and (e) 2.6 mm propagation.

was also done from 800 nm to 400 nm probe pulse using a BBO crystal. All of the set up for the probe pulses was done on a separate optical table. The schematic for the optical table is shown in Figure 4.7.

At the entrance of the optical table, the probe beam had a size of around 18 mm. The beam was reduced in size by half of its original diameter before being directed into the frequency-doubling crystal (BBO crystal) to increase the conversion efficiency. The frequency doubled pulse (blue) and the unconverted pulse (red) were then separated by a dichroic mirror. The blue pulse was then split into two parts by a Michelson interferometer set up with one variable-length arm to tune the delay between the pulses. After the Michelson interferometer, fine and coarse delay lines were installed to tune the delay between the blue probe pulses and the main laser pulse. Then, a 50-mm glass block was put in the blue line to stretch the blue pulses to have 400 fs duration each. All mirrors in the blue line are dielectric mirrors to prevent energy loss during the transport.

After the pulses exited the probe table, they came to a compressor table, where the main

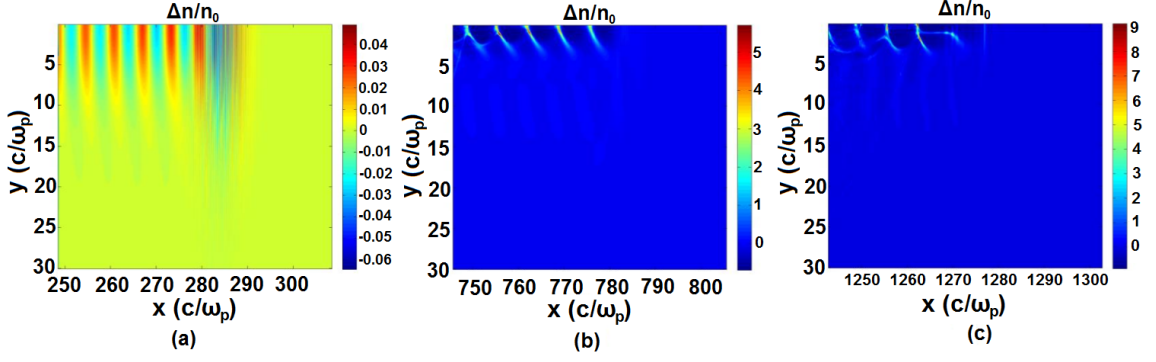


Figure 4.5: Density profiles of the wakefield with $n = 2 \times 10^{19} \text{ cm}^{-3}$ after (a) 0.2 mm, (b) 0.8 mm, (c) 1.4 mm propagation.

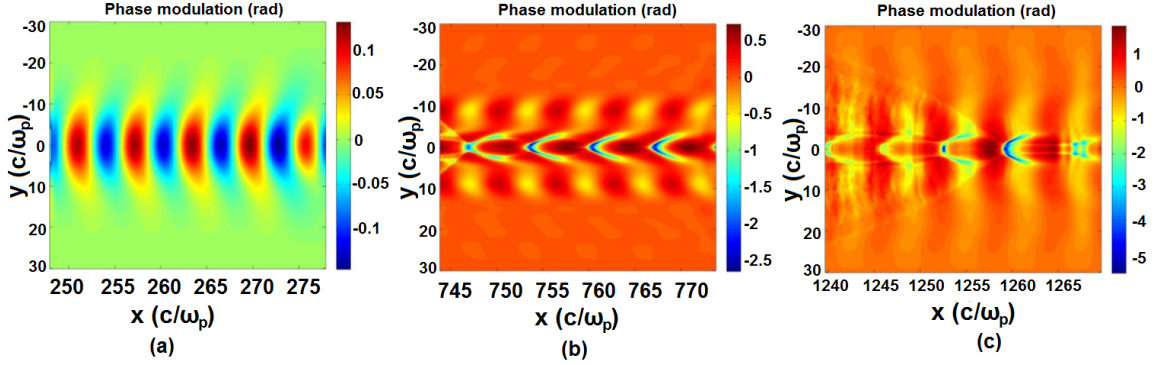


Figure 4.6: Phase modulation of the probe pulse with plasma density $n = 2 \times 10^{19} \text{ cm}^{-3}$ after (a) 0.2 mm, (b) 0.8 mm, (c) 1.4 mm propagation.

laser pulse was compressed. However, on this table, the probe pulses (blue and red) were simply transported to the target chamber.

The thickness of the glass block was calculated by assuming that the FWHM bandwidth of the probe pulse after frequency doubling was around $\Delta\lambda_{FWHM} = 6 \text{ nm}$. The group velocity dispersion (GVD) for N-BK7 glass block is $122 \text{ fs}^2/\text{mm}$. The thickness of the glass block is calculated by

$$t_g = \frac{1}{\text{GVD}} \tau_0^2 \sqrt{\left(\frac{\tau_1}{\tau_0}\right)^2 - 1}, \quad (4.7)$$

where τ_0 is the bandwidth-limited pulse duration and τ_1 is the intended pulse duration. We needed the pulse to have a pulse duration of $\tau_{FWHM1} = 400 \text{ fs}$ to cover a few wakefield periods $\tau_p \sim 25 - 50 \text{ fs}$. Therefore, $\tau_1 = \tau_{FWHM1}/2\sqrt{\ln 2} = 240 \text{ fs}$. The bandwidth-limited

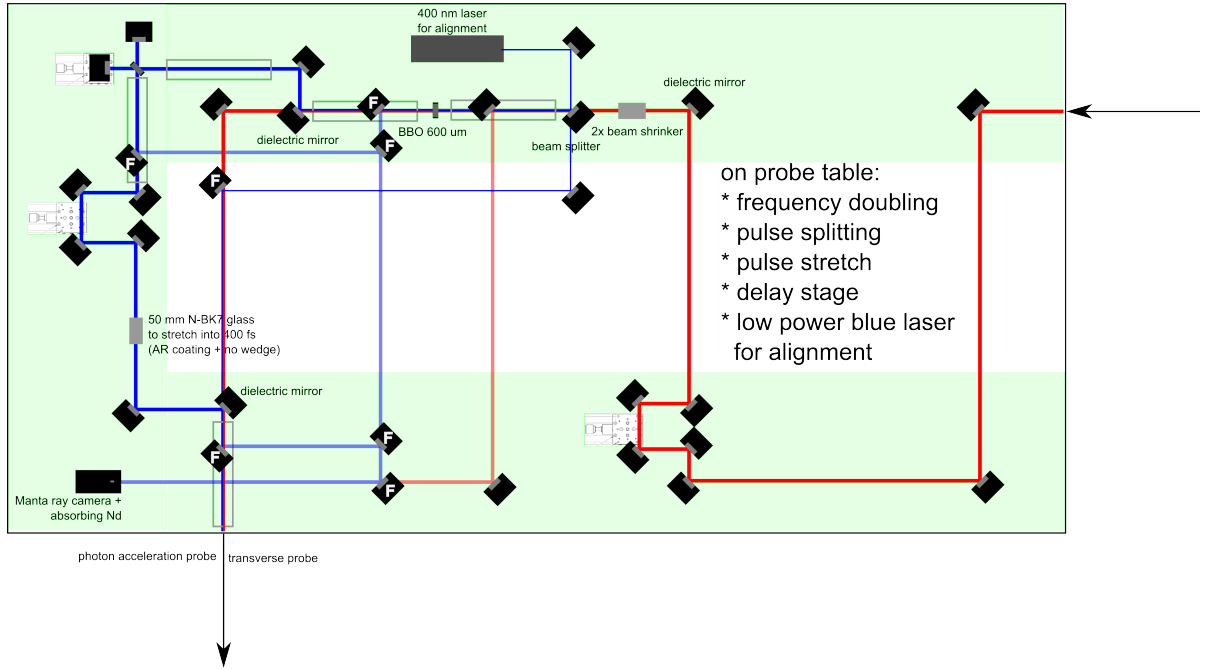


Figure 4.7: Schematic of the probe table. The blue line shows the line for main probe pulses for the wakefield diagnostic and the red line is for the perpendicular probe pulse to diagnose the plasma density. The mirrors denoted by white capital "F" letter are meant to be installed in flipping mounts.

pulse duration was also obtained from

$$\tau_0 = \frac{2\sqrt{\ln 2}}{\Delta\omega_{FWHM}}, \quad (4.8)$$

where $\Delta\omega_{FWHM} \approx (2\pi c/\lambda)(\Delta\lambda_{FWHM}/\lambda)$. Putting $\lambda = 400$ nm, $\Delta\lambda_{FWHM} = 6$ nm, yields $\tau_0 = 23.6$ fs. Using the value of τ_0 , τ_1 , and GVD for N-BK7 glass on equation (4.7) produces the glass thickness to be around $t_g = 46$ mm.

4.3 Experimental setup

Luke Ceurvorst and Kevin Glize were the designated Target Area Officer of the experiment. We were helped by Nicolas Bourgeois as the link scientist from the CLF. The group was

split into two teams during the experiment: an optics team and a diagnostics and DAQ team. The optics team consisted of Luke Ceurvorst, Kevin Glize, and Fearghus Keeble. The diagnostics and DAQ team comprised myself, Naren Ratan, and James Sadler. We were also helped by an undergraduate and two master students, Aimee Ross, Alexander Savin, and Nicholas Chen respectively. Professor Norreys oversaw the experiment and liaised with facility staff in his role of Principal Investigator.

When the experiment started, a set up from the previous group (by Lucy Wilson and colleagues) was left in the target area. As the optical layout of their experiment and our experiment were very similar, we used their design as a starting point. The new elements were to add the blue line on the probe table, position it correctly inside the target chamber and also build a diagnostic table for the blue probe pulses.

4.3.1 Probe table

The beam diameter reduction on the probe table was implemented by a concave silver mirror with $f = -1500$ mm and a convex silver mirror with $f = 750$ mm separated by a 750 mm distance. There was a problem when the experiment started: the glass block arrived late, so it was not possible to use the glass block to stretch the blue pulses. We overcame this obstacle by inserting a non-polarising cube beam splitter in the Michelson interferometer set up with 1 inch (25.4 mm) thickness. As a beam propagates inside the beam splitter twice in the set up, the cube beam splitter can also act as a stretcher glass block with thickness 50.8 mm. This solved the problem and we ended up using the cube beam splitter as the pulse stretcher.

For the low-hazard alignment laser, we set up a 400 nm class II laser for the blue line and a green laser to follow the red line. In summary, while some minor details of the optical layout changed between the conceptual design and the implementation phase, the operations performed on the probe table did not change (i.e. frequency doubling, pulse splitting, pulse stretching, delay stage, etc.).

4.3.2 Target chamber

The target chamber set up was left by the previous group (Lucy Wilson, *et al.*). The set up was similar to the schematic shown in Figure 4.2, except there was no optics to transport the blue probe.

A turning mirror with flippable mount between the gas jet and the last turning mirror was installed. This removable mirror reflected the light into a focal spot camera in low power laser mode. In high power/full power laser mode, this flippable mirror was removed to avoid damage on the focal spot camera and on the mirror itself.

The chamber was more packed than shown in the schematic mostly because of the off-axis parabolic mirror's mount and its motors and also the driver of the gas jet, which consisted of a three axis motor driver and a goniometer. Because of the geometry and the lack of space, we could not make the crossing angle to be 4° as planned, but had to contend with the crossing angle being 8° . This physical limitation in positioning the probe beam limited the signal to noise ratio, as discussed in the results section 4.2.1.

4.3.3 West table

On the west table (on the left side of Figure 4.2), a Mach-Zehnder interferometer was installed for the transverse probe diagnostics. A spectrometer for the post-interaction main laser beam was also installed as well as two cameras (exit mode cameras) to diagnose the main laser beam profile after the interaction. A camera to observe the longitudinal position of the gas jet was also placed on this table.

4.3.4 South table

On the south table, a spectrometer for the oblique crossing angle (the “blue” probe) probe pulse was positioned. Before getting into the spectrometer, the blue probe was magnified around 50 times larger using a $50\times$ infinity corrected microscope objective. A camera was

also installed to see the blue probe's transverse spot profile after the interaction with the plasma.

4.3.5 Diagnostics

During the experiment, numerous diagnostics before the target, after the target, and at the target were employed. Below is the complete list of the diagnostics employed in the experiment.

1. **Main beam near field monitor** on the probe table and before the target, to make sure that the main beam shape and alignment did not change much.
2. **Main beam far field monitor** on the probe table and before the target, with the same purpose on the near field diagnostic.
3. **Probe beam near field monitor** on the probe table and before the target, with the same purpose on the previous diagnostics, but this was applied to the blue probe pulses.
4. **Probe beam far field monitor** on the probe table and before the target.
5. **Focal spot monitor** in the target chamber and before the target, to see the main laser pulse spot size at the target and to overlap the main laser pulse and the blue probe pulses at the target.
6. **Top view** on top of the target chamber and looking at the target, to align the position of the target gas jet.
7. **Side view** on the west table and looking at the target, with the same purpose of the top view.
8. **Flippable camera monitor** on top of the target chamber to look at a flip mirror for the focal spot camera, to make sure that the flip mirror does not obstruct the beam path when full power shots were taken.
9. **Electron beam profile monitor** inside the lead box, to get the electron beam profile produced by the wakefield.

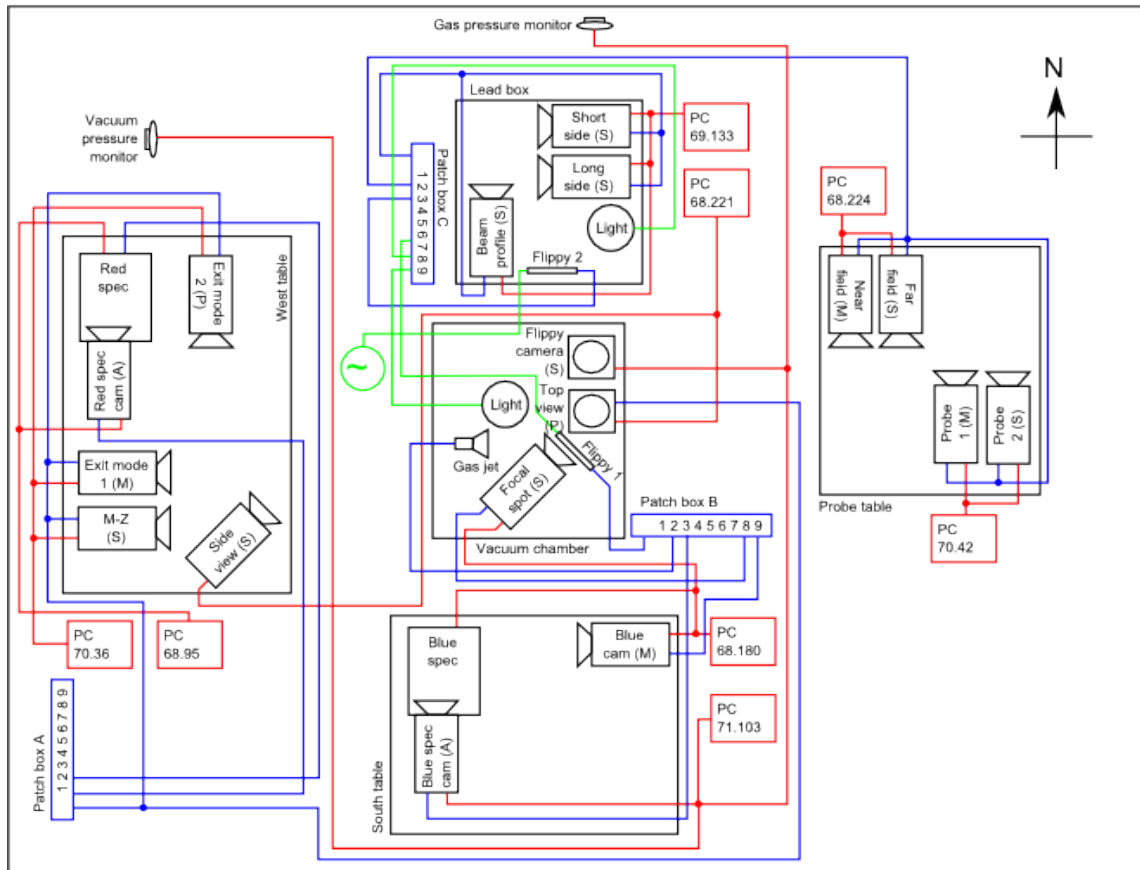
10. **E-spec short side** inside the lead box, to monitor the short side of the Lanex screen of the electron spectrometer.
11. **E-spec long side** inside the lead box, to monitor the long side of the Lanex screen of the electron spectrometer.
12. **Mach-Zehnder camera** on the west table and after the target, to capture the fringes caused by the perpendicular probe and Mach-Zehnder interferometer to get the plasma density profile.
13. **Exit mode 1** on the west table and after the target, to capture the transverse beam profile of the main laser pulse after the interaction with the plasma.
14. **Exit mode 2** on the west table and after the target, with the same purpose as exit mode 1, but with different distances.
15. **Red spectrometer** on the west table and after the target, to capture the spectrum of the main laser pulse after the interaction.
16. **Blue exit mode camera** on the south table and after the target, to capture the blue probe's profile after the interaction.
17. **Blue spectrometer** on the south table and after the target, to capture the blue probe's spectrum for the SSSI post-processing analysis to capture the wakefield.

The map of the DAQ system of our experiment is shown in Figure 4.8. The map turned out to be very helpful when we encountered problems with the DAQ system.

4.4 Experimental results

The experiment was performed using (484 ± 44) mJ main driver beam energy with central wavelength 800 nm and compressed to 45 fs. It is focused to $29 \mu\text{m}$ full width half maximum (FWHM) beam diameter. This gives the normalised intensity, $a_0 \approx (0.853 \times 10^{-9})\lambda(\mu\text{m})\sqrt{I(\text{W cm}^{-2})} \approx 0.72$. A gas jet with a 3 mm nozzle diameter was used with

Target area



Control room

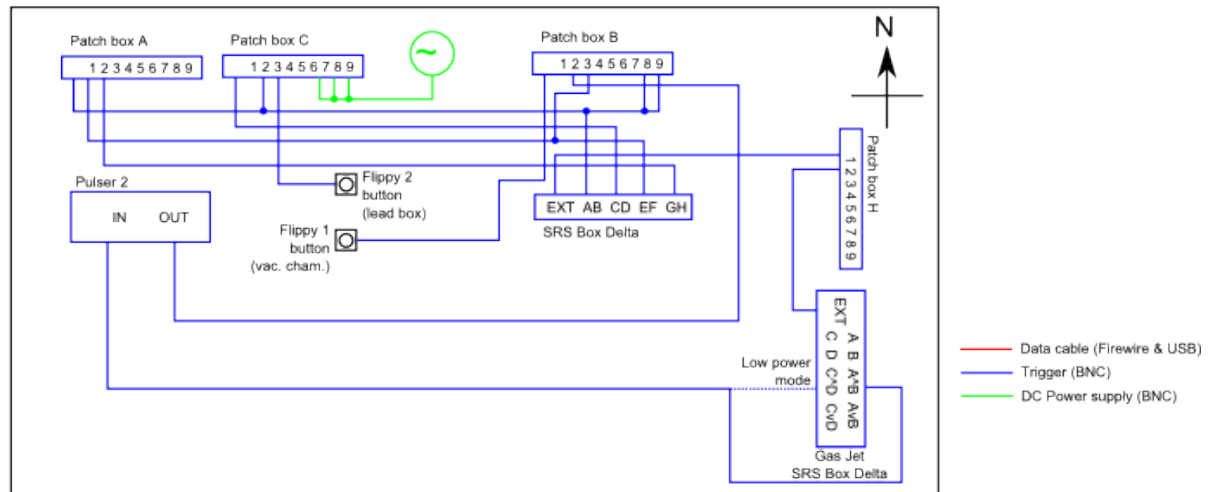


Figure 4.8: DAQ map of the experiment.

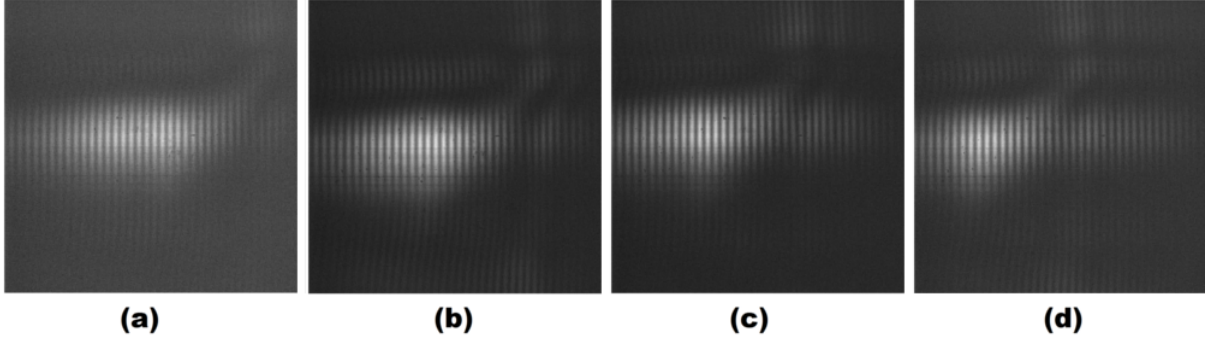


Figure 4.9: Modulations seen when placing the blue probe pulse at the ionisation front and the blue reference pulse in front of it. The pictures show where the relative delays between the blue pulses and the main driver pulse are (a) 0 fs, (b) 59.7 fs, (c) 119.4 fs, and (d) 179.1 fs. Right is the front of the pulse.

helium (He), nitrogen (N₂) and methane (CH₄). The backing pressure of the gas was varied from 7 bar to 15 bar.

4.4.1 Calibrations

The first thing we did for the SSSI method was to obtain the relation in the frequency domain in the spectrometer pictures with the time domain in the plasma. This was done by placing the blue probe pulse at the ionisation front and the blue reference pulse outside the plasma (in front of the blue probe pulse). This caused some modulation to appear on the spectrometer images. The relative delay between the blue pulses with the main pulse was then changed to see where the modulation shifts in the spectrometer images. Example pictures of the modulation shift are shown in Figure 4.9.

By plotting the modulation position as a function of the delay and doing the linear regression, the chirp coefficient was obtained. The plot is shown in Figure 4.10. From the linear regression $y = a + bx$, the gradient is obtained to be $b = (1.81 \pm 0.12)$ pix/fs. The coefficient of determination of the regression is $r^2 = 0.9773$. The spectrometer had a resolution of 0.012 nm/pixel, hence the chirp coefficient is $\alpha_\lambda = 0.022$ nm/fs or $\alpha = 2.5 \times 10^{-26}$ s⁻². The 1024 pixel of the image corresponded to a duration of $\tau = 566$ fs. For further analysis, the interesting number was the gradient, $b = (1.81 \pm 0.12)$ pixels/fs, because

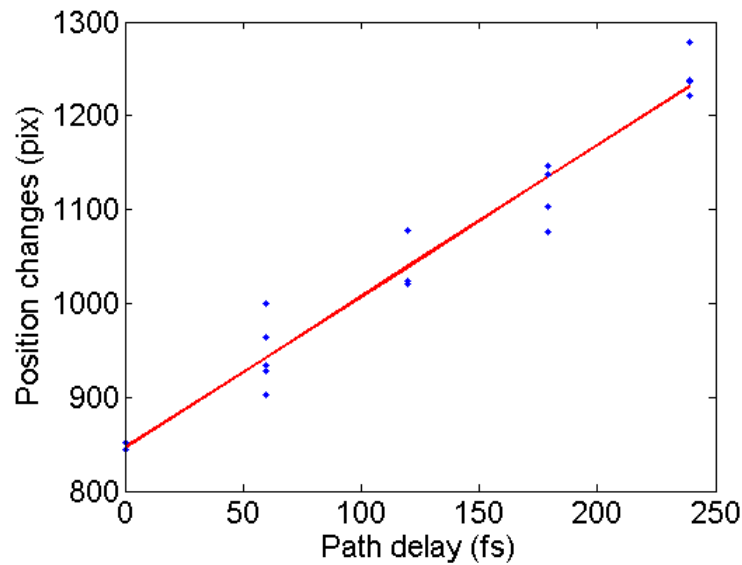


Figure 4.10: Position of the modulation as a function of the pulses' delay. The blue dots are the measurements from the experiment and the red line is the line fitted to the data. The errorbars for each measurement are much smaller than the variation of positions for the same path delay.

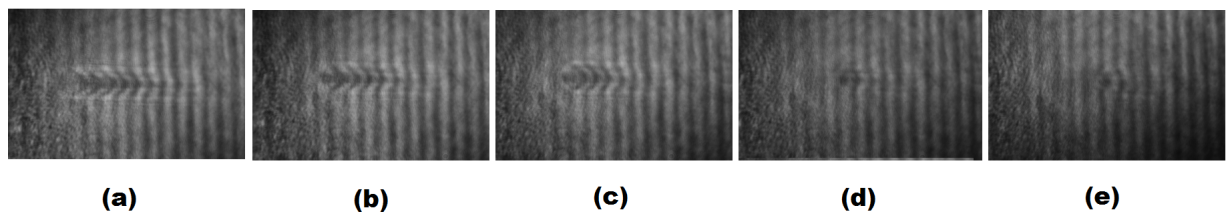


Figure 4.11: Mach-Zehnder (MZ) interferometer picture with relative delay paths between the MZ probe and the main driver pulse are (a) $0 \mu\text{m}$, (b) $200 \mu\text{m}$, (c) $400 \mu\text{m}$, (d) $600 \mu\text{m}$, and (e) $800 \mu\text{m}$.

it offered an end-to-end conversion from the spectrometer images into time domain.

The next calibration was the size calibration for the Mach-Zehnder (MZ) interferometry. This was done by setting the delay between the Mach-Zehnder probe pulse with the main driver pulse so that the Mach-Zehnder probe diagnoses the plasma when the driver is still in the plasma. By shifting the delay, the length of plasma diagnosed by Mach-Zehnder interferometer was changed. The delay path of the Mach-Zehnder probe was shortened by $200 \mu\text{m}$ for every 3-4 shots. The results of this calibration is a 5-frames movie with rate 1.5 trillion frame per second. Examples of the pictures are shown in Figure 4.11.

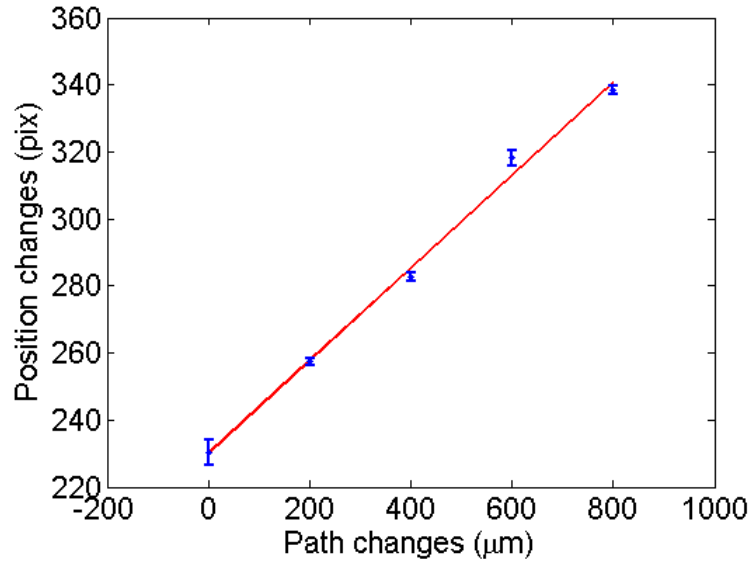


Figure 4.12: Position of the plasma front as a function of Mach-Zehnder probe's delay path.

As in the blue spectrometer calibration, the calibration was done by plotting the position of the plasma front as function of the delay path and applying a linear regression to it. The plot is shown in Figure 4.12. The linear regression $y = a + bx$ has $r^2 = 0.9947$ and gradient $b = 0.138 \text{ pix}/\mu\text{m}$. It means that one pixel in the camera corresponds to $0.138 \mu\text{m}$ on the focal plane, i.e. at the plasma.

The calibration for the main laser spot size was done by the focal spot camera. By shifting the focal spot camera by a known amount and observe the shift in the CCD, we could obtain the relation between the CCD pixel shift and the real length. The picture is shown in Figure 4.13. From the pictures, we obtain the length calibration is $0.97 \mu\text{m}/\text{pixel}$ and the Full Width Half Maximum (FWHM) of the focal spot is $29 \mu\text{m}$.

The energy calibration of the main laser beam was performed by measuring the energy of the full power main laser beam before compression, energy of the medium power laser beam before compression, and after compression. The measurement of medium power laser beam before and after the compression was done to obtain the compression efficiency. These measurements yield $7.06 \text{ mJ} \pm 5.5\%$ and $4.15 \text{ mJ} \pm 5.5\%$ for the medium power before and after the compression, respectively, and $823 \text{ mJ} \pm 4.6\%$ for the high power before the compression. Using a simple math and assuming the compression efficiency remains the

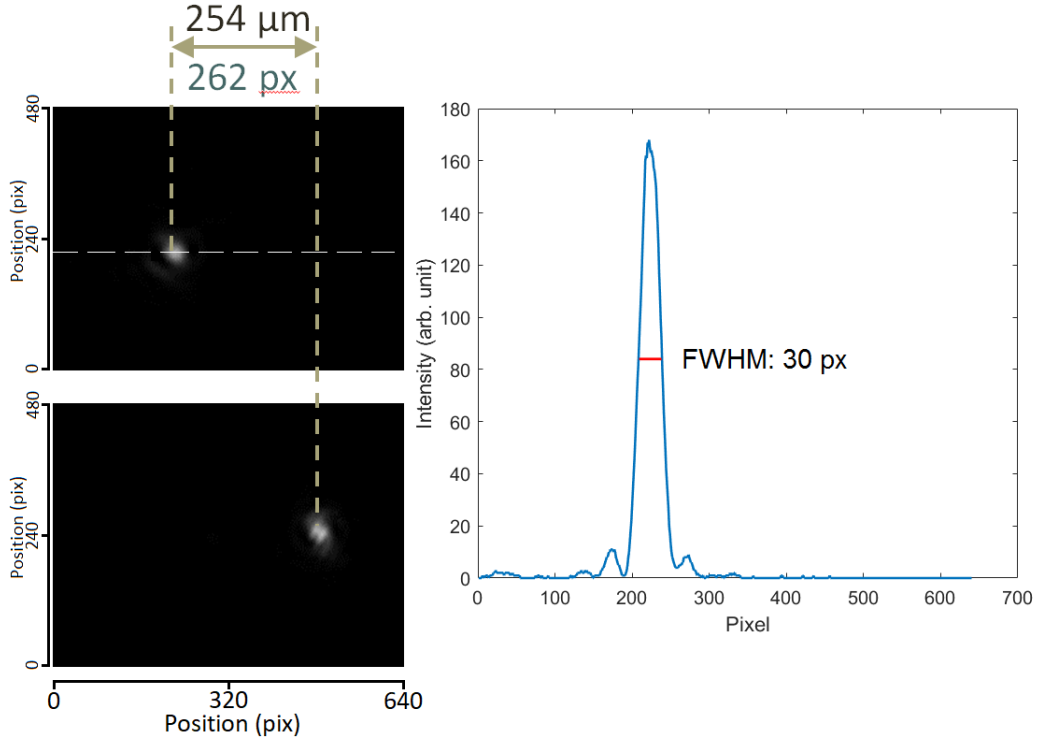


Figure 4.13: Main laser calibration with focal spot camera. Left pictures show the focal spot picture when the camera is shifted by $254 \mu\text{m}$ and the right picture is the lineout of the top focal spot picture (taken along the white dash line) to obtain the Full Width Half Maximum (FWHM) of the spot.

same for the medium and high power, we obtained the full power main laser beam after the compression was $484 \text{ mJ} \pm 9.0\%$.

4.4.2 Helium and nitrogen gas

In the experiment, we tried firing the laser with different kinds of gases. Among them were pure helium and nitrogen gas taken in different shots runs.

With helium gas, there was no visible modulation neither on the spectrometer fringes nor the phase map. With nitrogen gas, the fringes on spectrometer pictures did not produce visible modulations. However, when the SSSI inversion technique was applied, as described in section 4.1, the modulations were visible. This is shown in Figure 4.14. The modulations in the phase map were visible in the other two shots with plasma density around $n = 1.4 \times 10^{19} \text{ cm}^{-3}$. However, the modulations were also found in shot 160818-0146 where

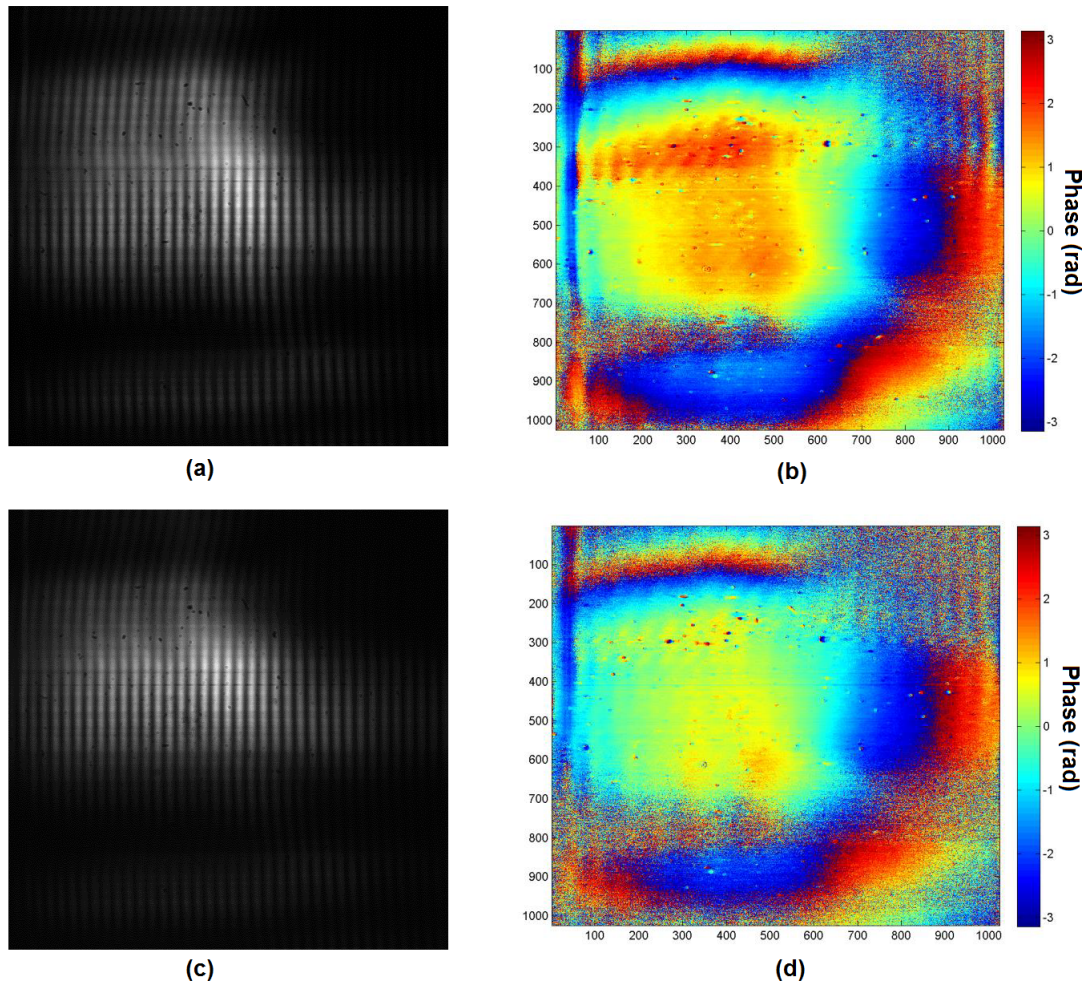


Figure 4.14: (Left) Fringes on the spectrometer picture and (right) the phase map retrieved from the fringes on the spectrometer using the method from section 4.1. Top pictures are pictures from shot 160818-0141 and the bottom ones are from shot 160818-0143. The modulations are visible at the upper half of the phase map pictures.

there was no gas as shown in Figure 4.15, indicating that this was an artefact of the signal processing and not a real signal.

Another proof that the modulation was not being generated by a wakefield was that the modulation is stronger on the left side and getting weaker to the right. The front of the pulse was on the right side, therefore the modulation should be stronger on the right side than the left side.

A likely explanation of why no modulations by the wakefield were observed is that the wakefield modulation was too small to be detected. Figure 4.4 shows the wakefield modulation with $n = 5 \times 10^{18} \text{ cm}^{-3}$ and 4° crossing angle has the amplitude around 60 mrad. If the

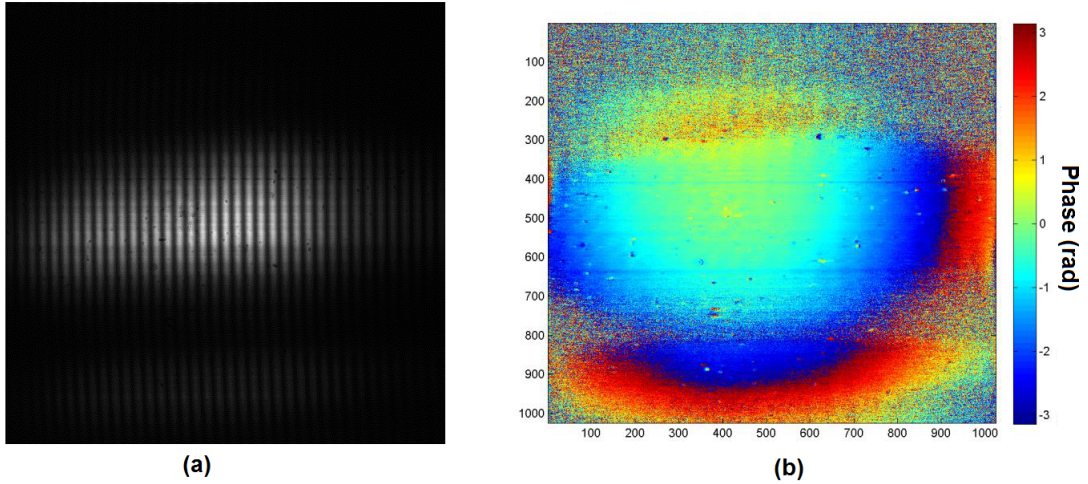


Figure 4.15: (Left) Fringes on the spectrometer picture and (right) the phase map retrieved from the fringes on the spectrometer for shot 160818-0146. The modulations are visible at the upper half of the phase map pictures.

crossing angle is increased to 8° , then the amplitude of the modulation decreased to around 30 mrad, a value that can be disrupted heavily by noise. The phase modulation amplitude with $n = 2 \times 10^{19} \text{ cm}^{-3}$ at the beginning of the plasma was around 150 mrad with 4° crossing angle (see Figure 4.6 (a)), or around 75 mrad with 8° crossing angle. This value is barely visible in the spectrometer.

4.4.3 Methane gas

The plasma density with methane gas was varied from $5 \times 10^{18} \text{ cm}^{-3}$ to $1.1 \times 10^{19} \text{ cm}^{-3}$. This was calculated by applying an Abel inversion routine [74, 75] from the Mach-Zehnder interferometry results. The symmetry axis for the Abel inversion was chosen by maximising the convolution value along the vertical axis of the image. One of the retrieved plasma profile is shown in Figure 4.16.

Unlike the previous results, the phase modulation of the crossing angle probe (the “blue” probe) seen in methane gas was very clear and it was repeatable between shots, unlike the case with nitrogen. Figure 4.17 shows one of the modulation captured during the experiment as well as the wrapped and unwrapped phase map. The phase map unwrapping was done

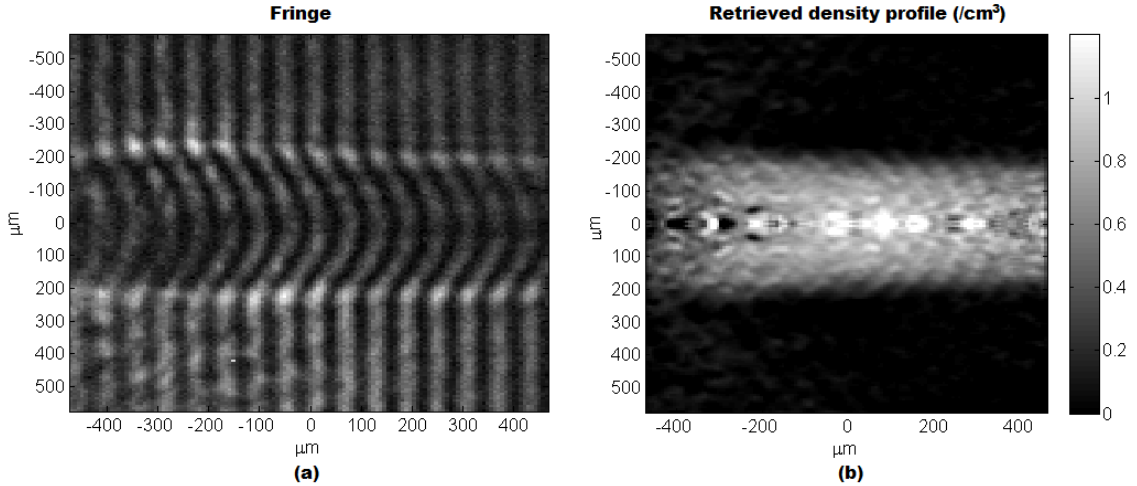


Figure 4.16: (a) Mach-Zehnder interferometer picture on the plasma density diagnostic. (b) The retrieved density profile of the plasma using Abel inversion. The density modulation on the axis is due to the singularity in Abel inversion that produce numerical instability on the axis.

using `unwrap_phase` function from `skimage.restoration` library in Python. The function utilised the phase unwrapping technique from [76].

As seen in Figure 4.17, the wavelength of the modulation at the front is large and the wavelength decreases towards the back of the wakefield, as expected. Figure 4.18 shows the comparison of the wavelength of the first, second, and third modulations with the expected wakefield wavelengths as a function of the plasma density. The expected wakefield wavelength is given by $\lambda_p = 2\pi c/\omega_p$, where ω_p is the plasma frequency.

The decreasing wavelength from the first modulation to the third modulation is due to non-linearity of the wakefield at the front. As explained in [15], the wakefield's wavelength increases as the wakefield's amplitude increases.

There are two remarkable points from the results of using cluster plasma. The first one is the amplitude of the modulation is larger in cluster plasma than using uniform plasma. This is probably due to the ponderomotive force exerted on electrons in cluster plasma can be larger than the ponderomotive force in uniform plasma as expressed in equation (1.79) in section 1.10.2.

Another point is that the modulation wavelength is on average longer than the expected

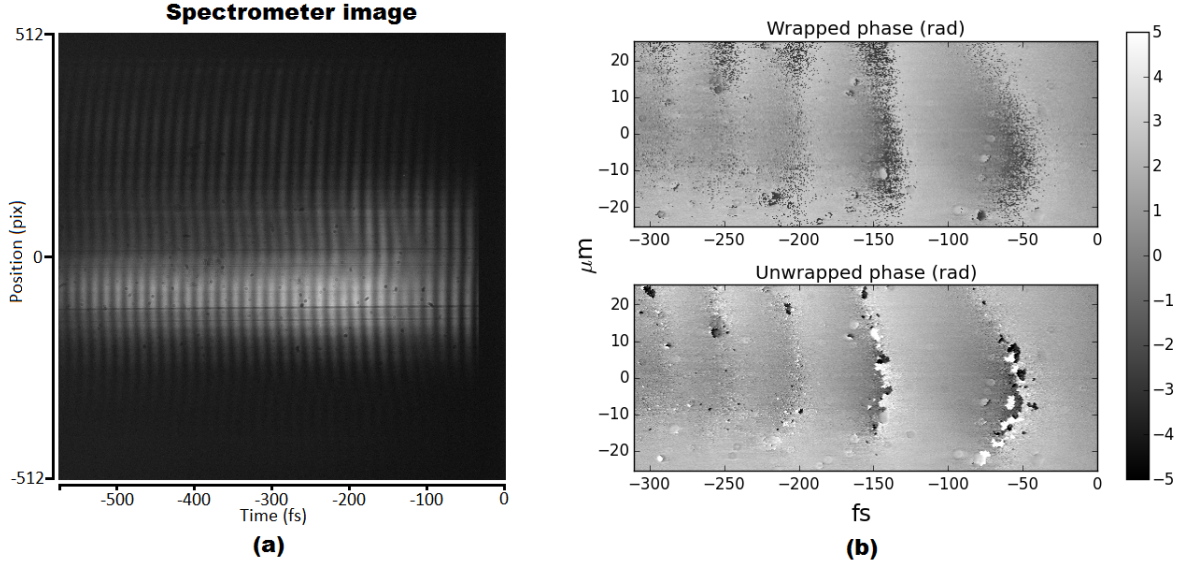


Figure 4.17: (Left) The spectrometer image of the spectral interferometry by the blue probe. (Right) The wrapped and unwrapped phase map of the blue probe.

wakefield's wavelength of the given density. This is probably because in retrieving the plasma density, the electromagnetic wave dispersion relation in uniform plasma was used to get the relationship between the refractive index and the plasma frequency or density. The relationship between the refractive index, η , and the plasma frequency in uniform and cluster plasma are respectively given by,

$$\omega_{pu}^2 = (1 - \eta^2)\omega_0^2 \quad (4.9)$$

$$\langle \omega_{pc}^2 \rangle = \frac{p/f}{p/f + (1 - \eta^2)}(1 - \eta^2)\omega_0^2. \quad (4.10)$$

The relations were obtained from equations (1.19) and (1.76), respectively. Here, the angle brackets is the spatial averaging operator and it is applied to the cluster plasma's frequency, $\langle \omega_{pc}^2 \rangle = p\omega_{pc}^2$, with p is the filled ratio and ω_{pc} is the oscillation frequency of the cluster. The variable f is the geometry factor of the cluster, where $f = 1/3$ for sphere.

As seen from the equations above, the average plasma frequency, $\langle \omega_{pc}^2 \rangle$, in cluster plasma is lower than the plasma frequency in uniform plasma, ω_{pu} , for the same values of ω_0 and

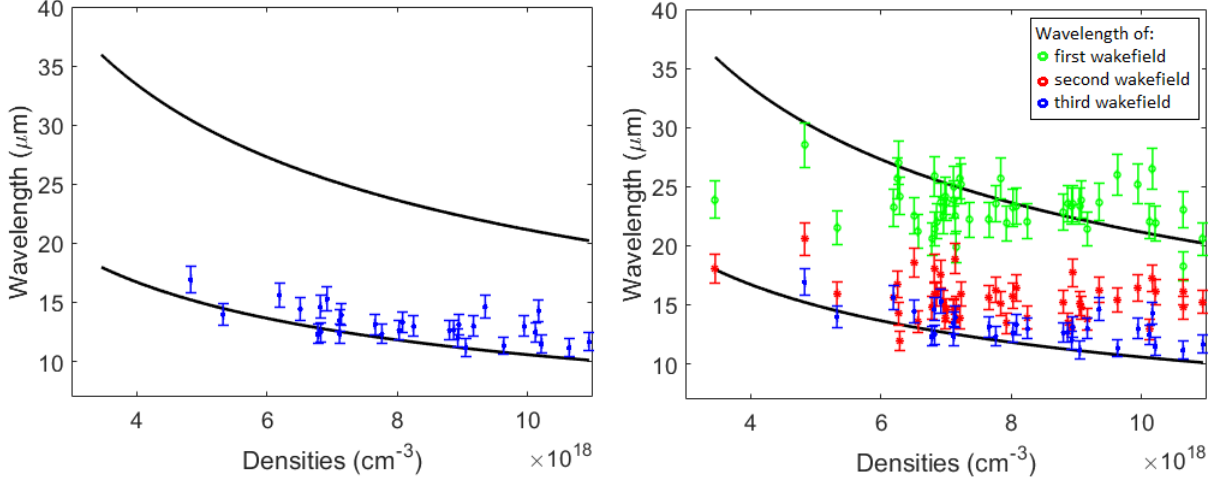


Figure 4.18: (Left) Plot of the third modulation wavelength vs the plasma density obtained from the Mach-Zehnder interferometry. The lines show the expected wakefield wavelength and twice the wavelength for the given plasma density (i.e. λ_p and $2\lambda_p$). (Right) Plot of the first (green circles), the second (red squares), and the third (blue points) modulation wavelength vs the plasma density.

η . Therefore, if the density is retrieved using the dispersion relation in uniform plasma, the calculated density will be over-estimated. The relations between the average cluster plasma density, $\langle n_c \rangle$, and the uniform plasma density with the same ω_0 and η can be retrieved from the equations above,

$$\langle n_c \rangle = \frac{p/f}{p/f + \omega_{pu}^2/\omega_0^2} n_u, \quad (4.11)$$

with n_u is the calculated plasma density using the dispersion relation in uniform plasma and $\omega_{pu} = n_u e^2 / m \epsilon_0$.

By trying several values of p/f to match the third modulation's wavelength with the expected wakefield's wavelength, it is found that $p/f \sim 3 \times 10^{-2}$. The plot of the third modulation's wavelength against the corrected density is given in Figure 4.19. By assuming that the cluster is sphere, $f = 1/3$, the cluster density is estimated to be around $5 - 10 \times 10^{20} \text{ cm}^{-3}$.

The estimated value of p/f also produces $f\omega_{pc} \approx (4 - 6) \times 10^{14} \text{ s}^{-1}$, which is less than the frequency of the light, $\omega_0 = 2.3 \times 10^{15} \text{ s}^{-1}$. As stated in section 1.10.2, the ponderomotive

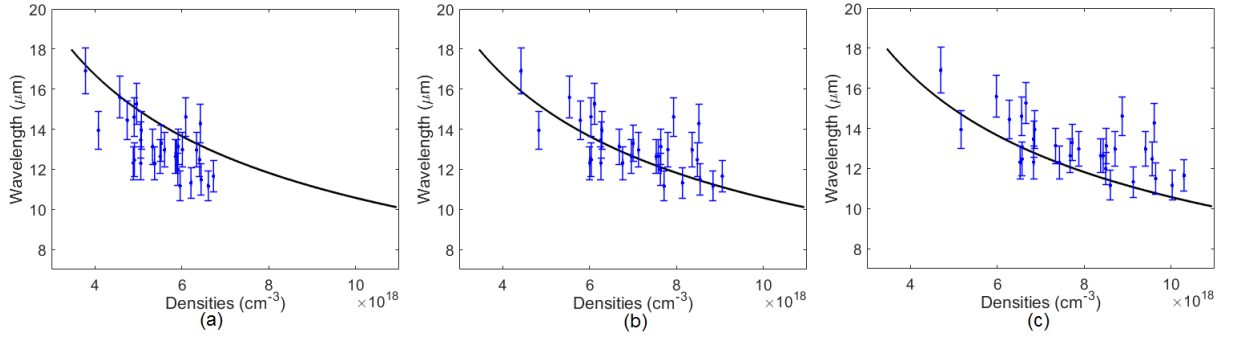


Figure 4.19: The third modulation wavelength against the corrected average plasma density with p/f value equals to (a) 1×10^{-2} , (b) 3×10^{-2} , and (c) 1×10^{-1} .

force increases if $|\omega_0^2 - f\omega_{pc}^2| < \omega_0^2$, which is valid in this case with the estimated value of p/f .

The longer first modulation wavelength could also be explained by elongation due to beam loading [77] in a bubble. As the electrons in the wakefield traverse along the bubble, the electron beam loaded in the wakefield repels the traversing electrons. With the repulsion, the wakefield bubble size, formed by the plasma's electrons, is longer than its size without any loaded beam.

To confirm the exact cause of the modulations in the cluster plasma, more Particle-in-Cell (PIC) simulations are needed. This is subject of future study.

4.5 Conclusion

In this chapter, the experimental set up and results for diagnosing plasma wakefields have been presented. The plasma wakefield was diagnosed using a second harmonic probe pulse crossing at an oblique angle. Variation of refractive index in the plasma imprints phase modulation in the probe pulse. The phase modulation in the probe pulse is then captured using spectral interferometry technique with another identical pulse without phase modulation. As for the results, modulations with similar length to plasma wakefield were successfully captured, especially in methane cluster target. A plausible analysis has also been presented to

explain the differences in plasma periods observed within the wakefield structure, although the precise cause requires further theoretical and simulation studies to confirm.

Chapter 5

3D spectrometer

Compressed sensing is an emerging field in signal processing. In compressed sensing, a measurement can obtain more information than is immediately apparent by assuming the measured signal is sparse in an arbitrary domain, which is the case for most natural signals. In this chapter, I present an idea of a new instrument based on compressed sensing, one that I have called “3D laser spectrometer”, where one can retrieve the spectral profile as a function of 2D spatial profile, i.e. $\lambda(x, y)$, in a single measurement. It has been numerically tested on expected signals from wakefield accelerator measurements. It is demonstrated here that the spectrometer can retrieve up to 10 times more information than an ordinary spectrometer from a single picture.

The initial idea of this 3D spectrometer arose after a google search with keywords ”high speed camera” and it led me to a Nature article describing a one-hundred billion frame per second camera [78]. In that article, the authors used the *compressed sensing* technique to extract multiple images from a single image of a streak camera. This new method apparently can be applied to extend the idea of plasma wakefield diagnostics with oblique crossing angle probe by deploying the 3D spectrometer.

Usually a spectrometer consists of a slit that is narrowly open and it gives the spectral profile of light coming through the spectrum. This slit needs to be as narrow as possible to

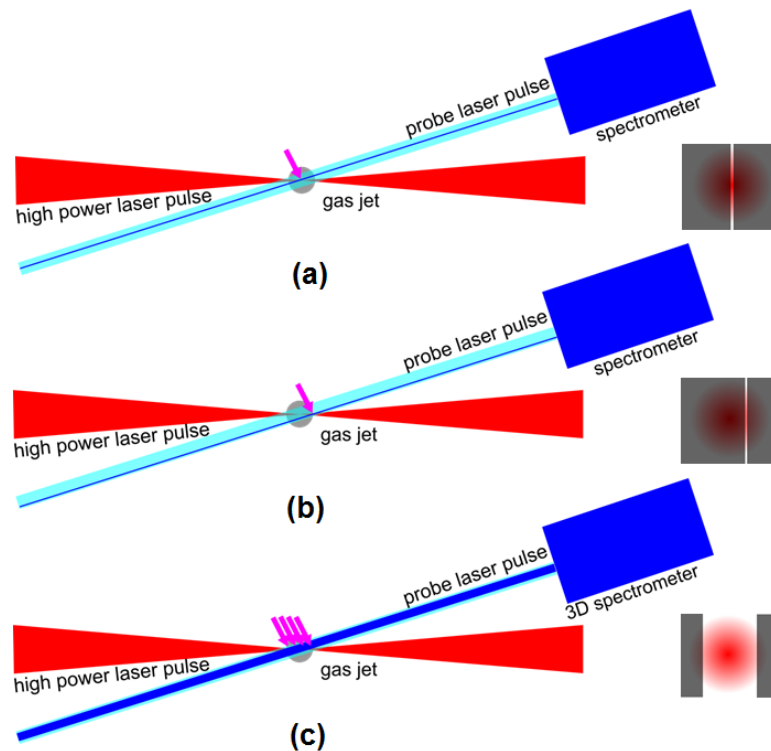


Figure 5.1: Illustration of the previous experiment. The pink arrows show the positions of the wakefield to be imaged. The blue lines show the slice of the probe pulse that are coming to the spectrometers.

get higher accuracy. If the slit is wide open, then multiple spectra from a large number of slices of the slit overlap with each other and that obviously decreases the spectral resolution.

Utilising a narrow-slit spectrometer in the previous three chapters provided us with an image where the slice of the probe crosses the wakefield. Moving the slit to a different position moves the diagnosed position of the wakefield. If there is a 3D spectrometer where one can open the slit as wide as possible and get multiple spectra images from a single shot, that could give us a video of wakefield evolution along the propagation distance in a single shot! Figure 5.1 illustrates this case.

Before I explain the schematic and how the 3D spectrometer works in detail, I will elucidate the concept of compressed sensing.

5.1 Compressed sensing

One of the main ideas behind compressed sensing is that most of natural signals are *sparse* in some representation. Sparse means it can be represented using only a few non-zeros elements. An example of this is shown in Figure 5.2. The top pictures show the original picture with its Discrete Cosine Transform (DCT) coefficients. DCT is a real-valued transformation that is similar to Discrete Fourier Transform (DFT). It uses $\cos(kx)$ as its kernel instead of e^{ikx} in DFT. After transforming the image to the DCT domain, a thresholding value is applied to the DCT coefficients, where the coefficients with lower values than the threshold are set to zero and the coefficients with higher values are left as they are. The bottom left picture shows the image after the reconstruction with thresholded DCT coefficients. The bottom left picture does not differ significantly with the original image. However, the number of non-zero coefficients for the top picture in the DCT domain is about 2.5 millions and the bottom picture only has around 250,000 non-zero elements. This is an example of an image that is sparse in the DCT domain.

Now let us consider a linear measurement expressed in matrix multiplication,

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (5.1)$$

where \mathbf{y} is the observation vector with size of $M \times 1$, \mathbf{A} is the measurement matrix with size of $M \times N$, and \mathbf{x} is the $N \times 1$ parameter vector that one is usually interested in. If the interesting signal and the observation are 2D or n -dimensional signals (e.g. images), then they are flattened into 1D vectors.

In most cases, \mathbf{y} and \mathbf{A} are known and \mathbf{x} is unknown. If $M < N$, then it is called as an underdetermined system or *ill-posed* inverse problem. There is more information to be retrieved than the measurement. It is like determining the values of 100 variables only from 10 equations.

In an underdetermined system, there are infinitely many possible solutions. However, using the knowledge that most of the natural signal is sparse, the problem above can be

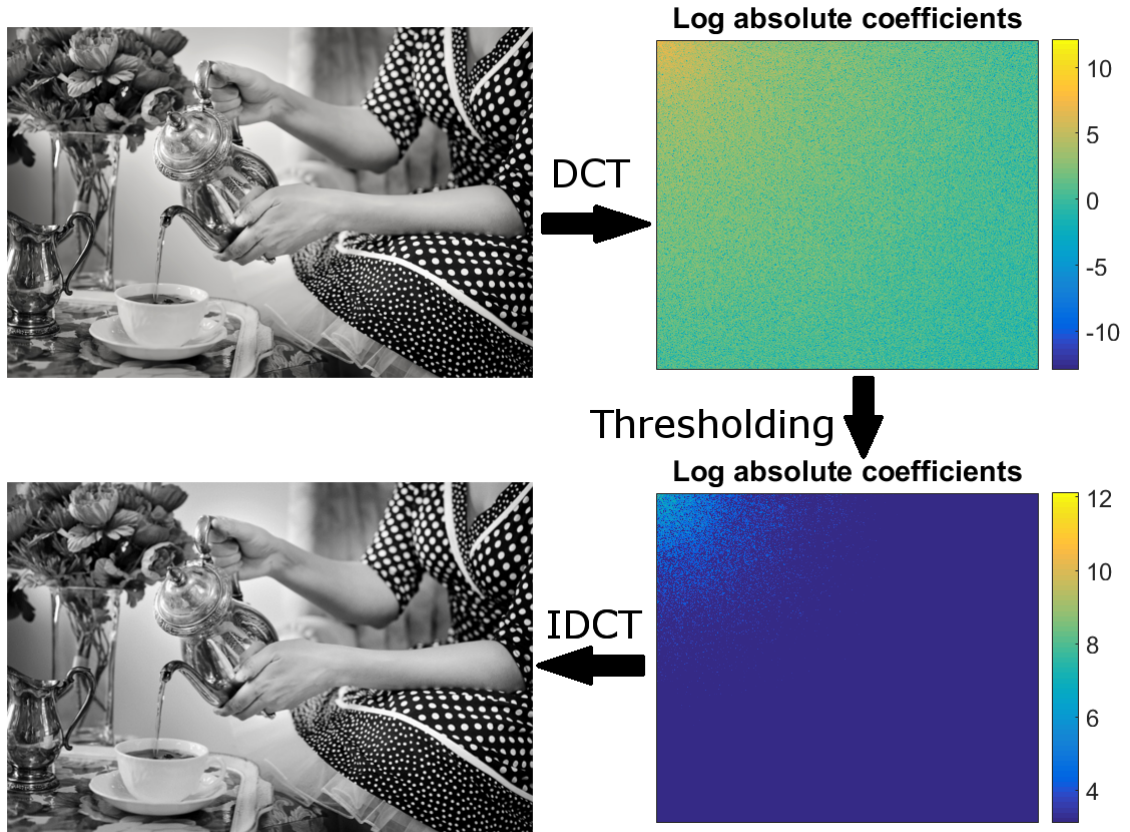


Figure 5.2: An example of image that is sparse in the Discrete Cosine Transform (DCT) domain. The top left picture is the original picture with 1920×1280 pixels and one color channel. The DCT coefficients of the picture is shown in the top right picture. In the bottom right, 10% of the largest absolute of the DCT coefficients are kept while the rest are zeroed. The bottom left picture is the picture transformed back from the sparse coefficients. The figure was taken from Pixabay and it is licensed for the public domain.

reduced to a solvable system. If \mathbf{x} is an $N \times 1$ vector that only has K non-zeros elements ($K \ll N$ and $K < M$), then the equation above can be solved by,

$$\mathbf{x} = \arg \min_{\mathbf{x}'} \|\mathbf{x}'\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}', \quad (5.2)$$

where $\|\mathbf{x}\|_0$ is the L_0 norm of the vector \mathbf{x} , i.e. the non-zero elements of the vector. L_p norm is defined as

$$\|\mathbf{x}\|_p = \left(\sum_i x_i^p \right)^{1/p}. \quad (5.3)$$

Solving the optimisation problem (5.2) is not trivial because the L_0 norm is non-convex. However, papers from Candes, *et al.* [21] and Donoho [22] shows that solving the optimisation problem above is equivalent to solving an optimisation problem with L_1 norm,

$$\mathbf{x} = \arg \min_{\mathbf{x}'} \|\mathbf{x}'\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}', \quad (5.4)$$

which can be solved using linear programming [21].

Equation (5.4) above only works if the measurement is noiseless. If there is some noise, the optimisation problem can be written as,

$$\mathbf{x} = \arg \min_{\mathbf{x}'} \|\mathbf{x}'\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}'\|_2 \leq \epsilon, \quad (5.5)$$

for some value $\epsilon \geq 0$. An equivalent form of equation (5.5) is

$$\mathbf{x} = \arg \min_{\mathbf{x}'} \mathcal{L}(\mathbf{x}') = \arg \min_{\mathbf{x}'} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}'\|_2^2 + \lambda \|\mathbf{x}'\|_1, \quad (5.6)$$

where \mathcal{L} is the loss function given by the right hand side of the equation and $\lambda > 0$ is the regulariser coefficient.

Most of the time the interesting signal is not sparse in its original representation, but it is sparse in some domain, e.g. Discrete Cosine Transform [79] or wavelet [80]. The picture in Figure 5.2 is an example of a signal which is not sparse in its original representation but sparse in the DCT domain. If this is the case, then the signal \mathbf{x} can be written as $\mathbf{x} = \mathbf{\Phi}\mathbf{c}$, where $\mathbf{\Phi}$ is the inverse sparsifying transformation and \mathbf{c} is the signal's coefficients in the sparse domain. In Figure 5.2 example, the sparsifying transformation, $\mathbf{\Phi}$, is the inverse Discrete Cosine Transformation. Thus, the loss function from equation (5.6) can be written as

$$\mathcal{L} = \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{\Phi}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1. \quad (5.7)$$

Compressed sensing does not only say that the sparsity can be achieved by L_1 norm, it also says that the measurement matrix needs to fulfil a Restricted Isometry Property (RIP) [81]. Candes, *et al.* [21, 82] say that a Gaussian random matrix fulfils RIP with high probability. In practice, introducing some randomisation (either Gaussian or binary) into the measurement matrix also works. For example, see single pixel camera [83] and 100 billion fps camera [78].

Besides L_1 norm, another form of regulariser that is commonly used is Total Variation (TV) regulariser. For a 2D image, isotropic and non-isotropic TV regulariser are respectively defined as [84]

$$\text{TV}_{\text{iso}}(\mathbf{x}) = \sum_i [(\Delta_h x_i)^2 + (\Delta_v x_i)^2]^{1/2} \quad (5.8)$$

$$\text{TV}_{\text{niso}}(\mathbf{x}) = \sum_i (|\Delta_h x_i| + |\Delta_v x_i|), \quad (5.9)$$

where Δ_h and Δ_v respectively denote the gradient in horizontal and vertical directions.

Various algorithms are already available in solving the optimisation problem encapsulated by equation (5.6). Among them are Two Steps Iterative Shrinkage Thresholding algorithm (TwIST) [84], Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [85], Orthant-Wise Limited-memory Quasi Newton (OWL-QN) [86], and Iterative Hard Thresholding (IHT) [87]. They are basically gradient-descent based algorithms where a thresholding is applied after each descending step.

5.2 ANOA: A differentiable programming

As most of the available algorithm to minimise the loss function in equation (5.7) are gradient-descent based, I wrote a library in Python to determine the gradient of a variable automatically. It is called ANOA [88]. The library is an automatic differentiation

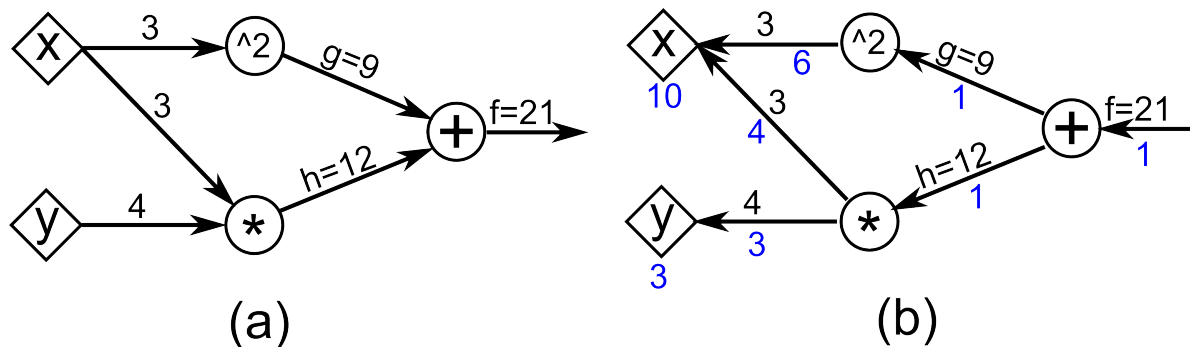


Figure 5.3: (a) The forward computational graph to compute the value $f = x^2 + x * y$ with intermediate variables $g = x^2$ and $h = x * y$. (b) The backward computational graph to compute the derivative of f with respect to each variables in the graph. The derivative values are shown in blue and they are below the arrows.

library based on the back-propagation algorithm [89].

The key to the back-propagation algorithm is the chain rule on differential. To understand how it works, let us take an example of differentiating $f(x, y) = x^2 + x * y$ with respect to x and y at $x = 3$ and $y = 4$. The variable f is written as $f = g + h$ where $g = x^2$ and $h = x * y$. Thus, the value of f is obtained by calculating the values of g and h first. This is straightforward. The computational graph can be seen in Figure 5.3 (a).

To obtain the differentiation of f with respect to x and y , one performs a back-propagation. We can start from $\partial f / \partial f = 1$. Then, moving backward from f , once it passes an operation node, the differential value is multiplied by the differential of that operation. If there are more than one edge going into the same node, the differential values from those edges are summed. The computational graph for this back-propagation can be seen in Figure 5.3 (b).

There has been a lot of automatic differentiation libraries available as they are used extensively in the machine learning community. The examples are Theano [90], Tensorflow [91], and Torch [92]. The difference between ANOA and those libraries is that ANOA is designed for compressed sensing and thus algorithms for compressed sensing and L_1 and TV minimisation are available. There are also lot more matrix transformations available in ANOA compared to the libraries used in the machine learning community.

Figure 5.4 shows an example of performing compressed sensing with the ANOA library

```

1  import anoa as an
2  import numpy as np
3  from scipy.ndimage import imread
4  from scipy.misc import imsave
5
6  # load the image and set the randomised picture
7  img = imread("stonehenge.jpg")
8  sample = 10./100 # take only 10% of the sample
9  randomPos = (np.random.random(img.shape) < sample)
10 observation = img[randomPos]
11
12 # set the variables
13 img_var = an.Variable(img.shape)
14 idx = img_var[randomPos]
15 loss = an.sum(an.square(observation - idx))
16 regulariser = an.iso_tv_2d_regulariser(img_var, weights=5)
17 img_val = an.minimise(loss, an.fista, regulariser)
18
19 # save the result
20 imsave("retrieved-stonehenge.jpg", img_val.astype(np.uint8))

```

Figure 5.4: An example of doing a compressed sensing retrieval from a picture that lost 90% of the information.

in just 20 lines. The first part of the code is (line 1-4) to import the required libraries. The second part (line 7-10) loads the sample image and take only 10% of the pixel-channel values and drop the other 90%. The third part (line 13-17) sets the loss as a function of an ANOA variable, `img_var`, and retrieve the value using FISTA algorithm [85] and the isotropic TV regularisation [84]. The last part is to save the retrieved picture. The result of this example is shown in Figure 5.5.

5.3 3D Spectrometer

The main objective of the 3D spectrometer is to obtain 3D spectral datacube of the light coming through a slit. The datacube consists of 2D spatial and 1D spectral dimensions. The schematics of the 3D spectrometer are shown in Figure 5.6.

The design consists of two Czerny-Turner type of imaging spectrometers [93] with a beam splitter to connect these two parts. On one side, there is a random aperture with a mirror to block some part of the light and reflect it back. This can be replaced by the Digital

**(a)****(b)****(c)**

Figure 5.5: The result of the compressed sensing example using ANOA. (a) The original image. (b) The image sampled from 10% information from the original image. The other 90% of the pixel-channel values are set to zeros. (c) The retrieved image just from the sampled image using FISTA algorithm and the isotropic TV regularisation. The image is taken from Pixabay and it can be used for free.

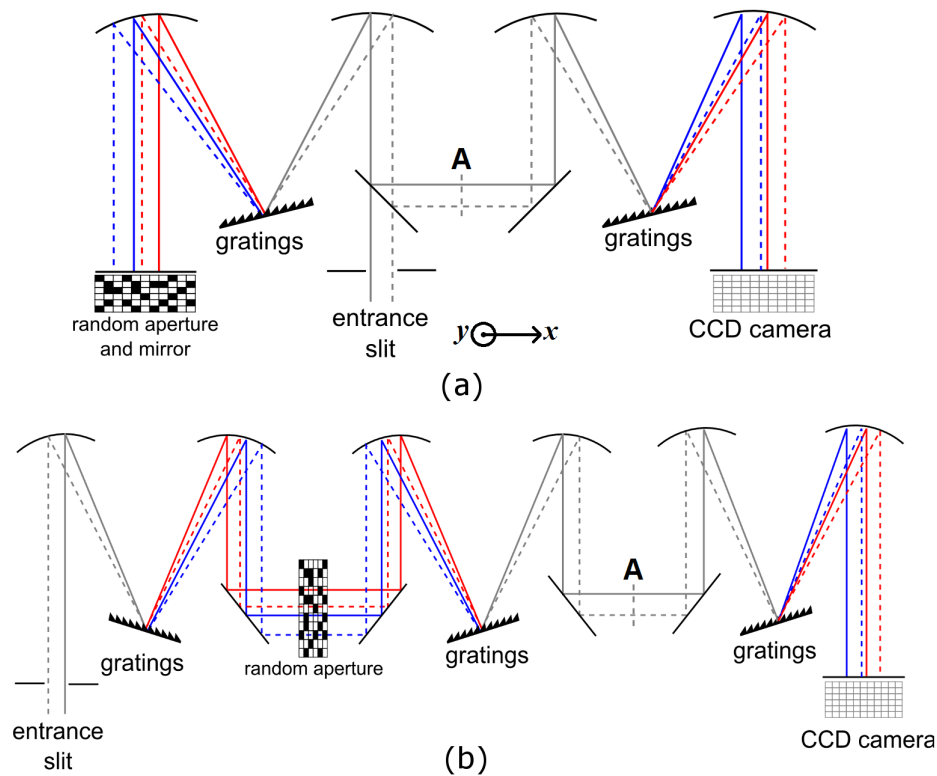


Figure 5.6: Schematic designs of the 3D spectrometer. Solid lines show the path of the ray entering from one side of the slit while the dashed lines for the ray on the other side. Blue and red lines are respectively representing high and low frequency components of the light. The x -axis is horizontal and the y -axis is coming out of the plane. The top picture (a) is the folded schematics of the design and the bottom picture (b) is the unfolded schematics.

Micromirror Device (DMD). And on the other side, there is a detector array to record the intensity of the light on that plane. One can use a Charge-Coupled Device (CCD) camera as the detector.

To retrieve the 3D information (2D spatial and 1D spectral profile) of the light entering the slit, it is essential to know how the 3D profile of the light is transformed in the spectrometer, or the datacube transformation. The datacube transformation of the light in this spectrometer design is shown in Figure 5.7.

From the entrance slit, the datacube is sheared by the gratings on the left side of the picture. After the light is reflected back by the random aperture and mirror, some part of the datacube is blocked and sheared back by the same gratings when it reaches plane A in Figure 5.6. Then by using another grating, the datacube is sheared to the opposite direction

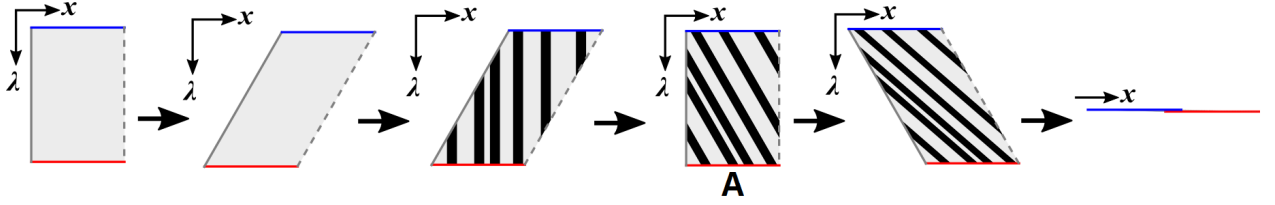


Figure 5.7: The datacube transformation of the light from entering the slit until reaching the CCD camera. The x -axis is corresponding to the axis in Figure 5.6(a) and the y -axis is coming out of the plane. The λ axis on the datacube represents the spectral axis of the light's profile. The fourth datacube is the spectral datacube on plane A in Figure 5.6. The last transformation is summing the datacube along the λ axis.

before it reaches the detector array. Because the detector array can only record the total intensity as function of positions, the datacube is then summed over the λ -axis. From this recorded intensity profile on the detector, one can employ a compressed sensing retrieval algorithm to obtain the original datacube, such as TwIST [84], FISTA [85], OWL-QN [86], and IHT [87].

In this spectrometer design, the detector measures the intensity in the spectral and spatial domain, while the randomisation also takes place in the spatial and spectral domain. This gives more randomisation in the measurement, and thus it has a high probability of having lower coherence and better retrieval ability compared to the previous methods [94, 95].

An equivalent design of this schematic is shown in Figure 5.6(b) where the design is unfolded. The unfolded design gives an advantage of higher light flux reaching the CCD camera. It also reduces the risk of getting double reflections from the beam splitter. In the folded design (Figure 5.6(a)), at least 25% of the light is loss in the beam splitter. This does not happen in the unfolded design because it does not involve a beam splitter. However, more gratings and mirrors are required for this unfolded design.

There are a few similar approaches to the 3D spectrometer that employ a similar transformation: Single Disperser design for Coded Aperture Snapshot Spectral Imaging (SD-CASSI) [94] and Double Disperser CASSI (DD-CASSI) [95]. SD-CASSI puts the random aperture at the beginning and then the datacube is sheared once before it comes to the screen. DD-CASSI shears the datacube first, then it comes to the random aperture, and

then re-shears back before it comes to the screen. DD-CASSI is similar to our design without the right part of Czerny-Turner in Figure 5.6. Both designs can retrieve simple shape spectral profiles, but are unable to retrieve accurately the spectral profile when it needs a further post-processing. This is an advantage of my design where the spectral profile needs a further post-processing as in spectral interferometry.

5.4 Numerical tests

In order to test the design, I performed some numerical tests. The tests were performed in Python and the signals were retrieved using ANOA library in Python. In this case, the program determines the gradients of the loss from equation (5.7), \mathcal{L} , with respect to the spectral cube's elements, \mathbf{c} . The program also determines \mathbf{c} that minimises the loss \mathcal{L} .

A code was written to simulate the datacube transformation using concave toroidal mirrors with focal lengths of $f = 15$ cm and gratings with 300 lines/mm. The random aperture in the simulation had 100×100 square pixels with side length of $20 \mu\text{m}$ each. The detector array had $20 \mu\text{m}$ -size square pixels with 100×100 pixels.

The first case used a spatially chirped light source, where the light had different central frequency for different x -position in this case. For every x -position, the light had a Gaussian spectral profile with the full width half maximum (FWHM) was 4.05 nm and the central wavelength was around 633 nm. The chirp was 31.25 nm/mm. Gaussian noise with σ 10% of the maximum spectral intensity was also added to test the robustness. The width of the slit was $200 \mu\text{m}$ or about 10 times of the detector's pixel size.

For the retrieval algorithm, OWL-QN algorithm was employed [86] with the datacube size of $(N_y, N_x, N_\lambda) = (100 \times 11 \times 90)$ voxels. OWL-QN is a second order optimisation algorithm for L_1 regulariser. Three dimensional Discrete Cosine Transformation (3D-DCT) domain was chosen as the signal's sparse domain.

From the measured intensity profile shown in Figure 5.8(a), the complete 3D datacube

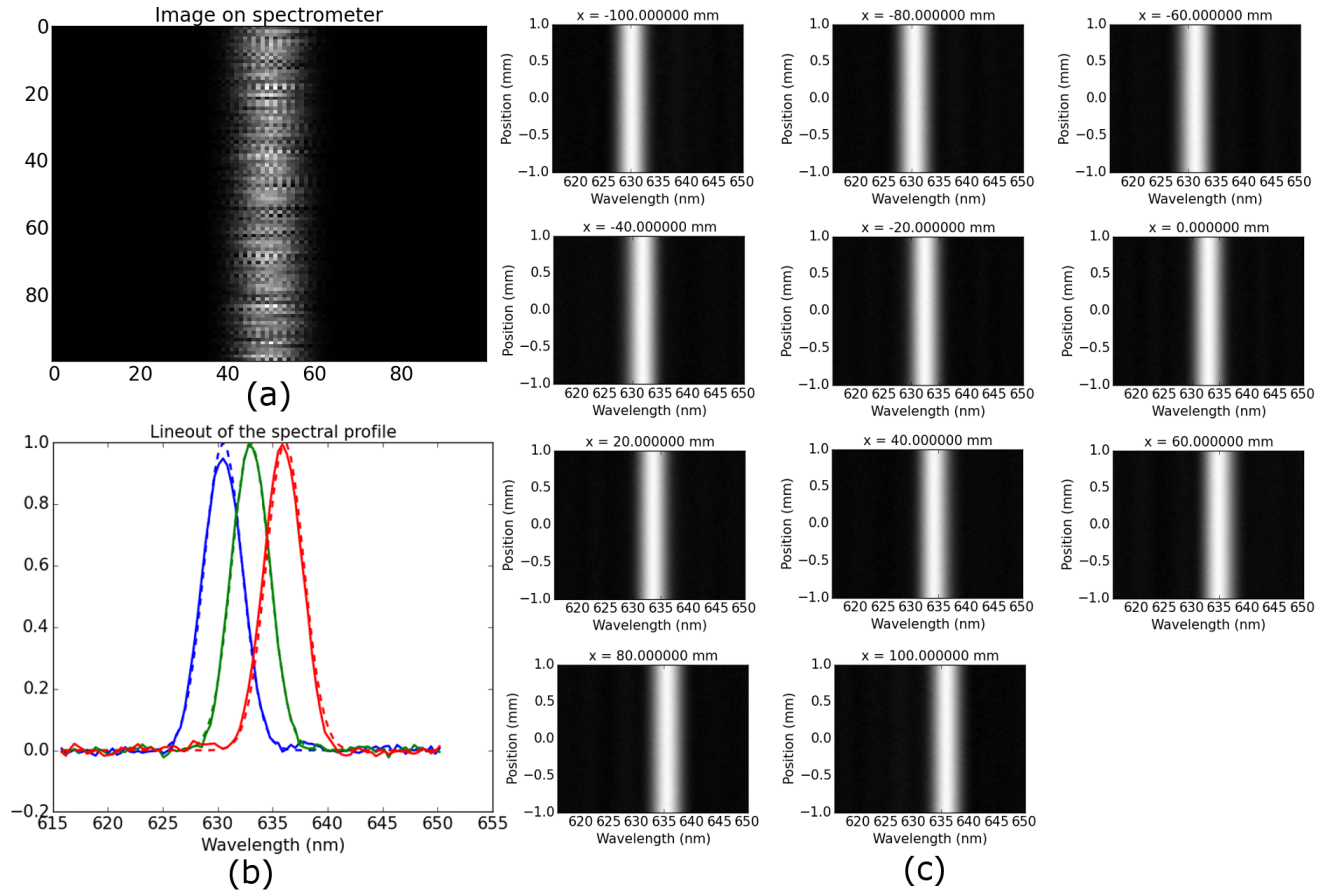


Figure 5.8: (a) The image captured on the CCD camera in the 3D spectrometer. (b) Lineout of the spectrum from the leftmost slice (blue), middle slice (green), and the rightmost slice (red). (c) The spectral intensity profiles for every slices.

profile of the light entering the slit is retrieved, as shown in Figure 5.8(c). The figures clearly show the central wavelength of the light is different for slices in different x -position. As shown in Figure 5.8(b), the central lineouts of the retrieved signal agree very well with the original signal. The spectral profiles in Figure 5.8(c) are also in excellent agreement with the original profiles.

Another test was done based on spectral interferometry case [16, 27, 71]. In the case of spectral interferometry, there were two short laser pulses. One pulse, called the probe pulse, had some phase modulation while another pulse, the reference pulse, did not have phase modulation. The pulses were temporally chirped. They co-propagated with each other and were temporally separated. The spectral profile of these two pulses contains information of the phase modulation of the probe pulse. By applying a high pass filter and

obtaining the phase of the spectral profile, one can extract the phase modulation of the probe pulse. Implementing this in the 3D spectrometer poses a greater challenge since the spectral interferometry needs a post-processing (i.e. phase extraction) of the retrieved signal to obtain useful information.

In this simulated case, the phase of the probe pulse was modulated spatially and spectrally. The modulation was then encoded into the spectral intensity according to the equation below,

$$I(x, y, \omega) = I_0(x, y, \omega) \left[1 + \cos \left(\omega\tau + \Delta\tilde{\phi}(x, y, \omega) \right) \right], \quad (5.10)$$

where I_0 is the spectral intensity profile for one pulse and $\tilde{\phi}(x, y, \omega)$ is the phase modulation of the probe pulse. In the 3D spectrometer, the spectral intensity datacube, $I(x, y, \omega)$, undergoes the transformation as in Figure 5.7 with $\lambda = 2\pi c/\omega$. The spectral intensity datacube was retrieved from the image captured by the 3D spectrometer and further the post-processing analysis was performed to obtain the phase modulation datacube, $\tilde{\phi}(x, y, \omega)$.

In the tested case, there were $N_x = 10$ slice of the spectral intensity profiles to be retrieved each with size $N_\lambda \times N_y = (800 \times 100)$ pixels. The datacube example is shown in Figure 5.9(a). The pattern on the 3D spectrometer's screen is shown in Figure 5.9(b). The size of the image on the screen is (809×100) pixels. Retrieving all the spectral intensity profiles was done using TwIST algorithm. One of the retrieved spectral intensity profiles and its comparison with the original spectral intensity profile is shown in Figure 5.10(a).

One advantage of this 3D spectrometer design is that it is also robust even with this post-processing. Figure 5.10 (b) show the extracted phase maps from the retrieved and original spectral profiles. The comparison for the central lineouts of the phase maps from all the spectral intensity profiles is shown in Figure 5.10 (c). This shows an excellent agreement between the retrieved phase maps and the original phase maps.

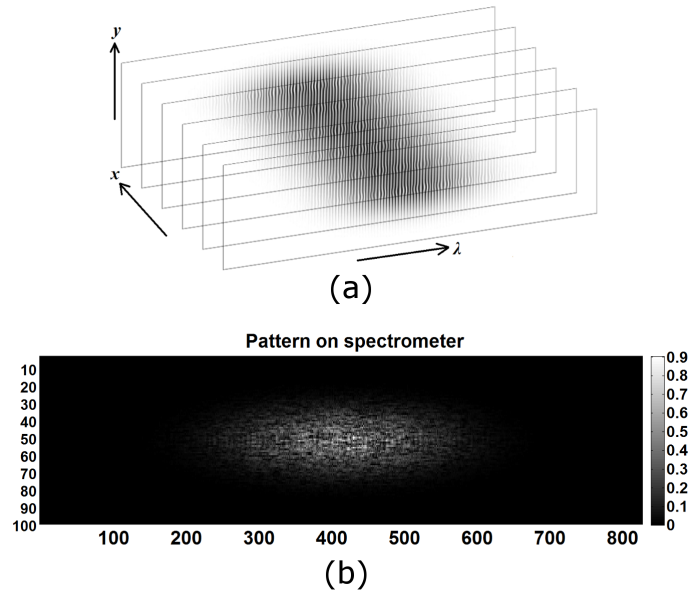


Figure 5.9: (a) Illustration of the datacube to be inverted. There are only $N_x = 6$ slices shown here, but in the test case, $N_x = 10$ is used. (b) The simulated pattern captured on the screen in the 3D spectrometer.

5.5 Conclusion

In this chapter, I have presented a novel design of 3D spectrometer that can retrieve multiple 2D spectral profiles in a single shot. The design consists of two or three Czerny-Turner type of imaging spectrometers with a random aperture. Numerical tests have shown that the spectrometer can retrieve up to ten sets of 2D spectral profiles just from a single image. The design is also robust when the retrieved signal requires further post-processing. This can be useful in diagnosing plasma wakefield experiments. A patent for the 3D spectrometer design has been filed at the time this was written.

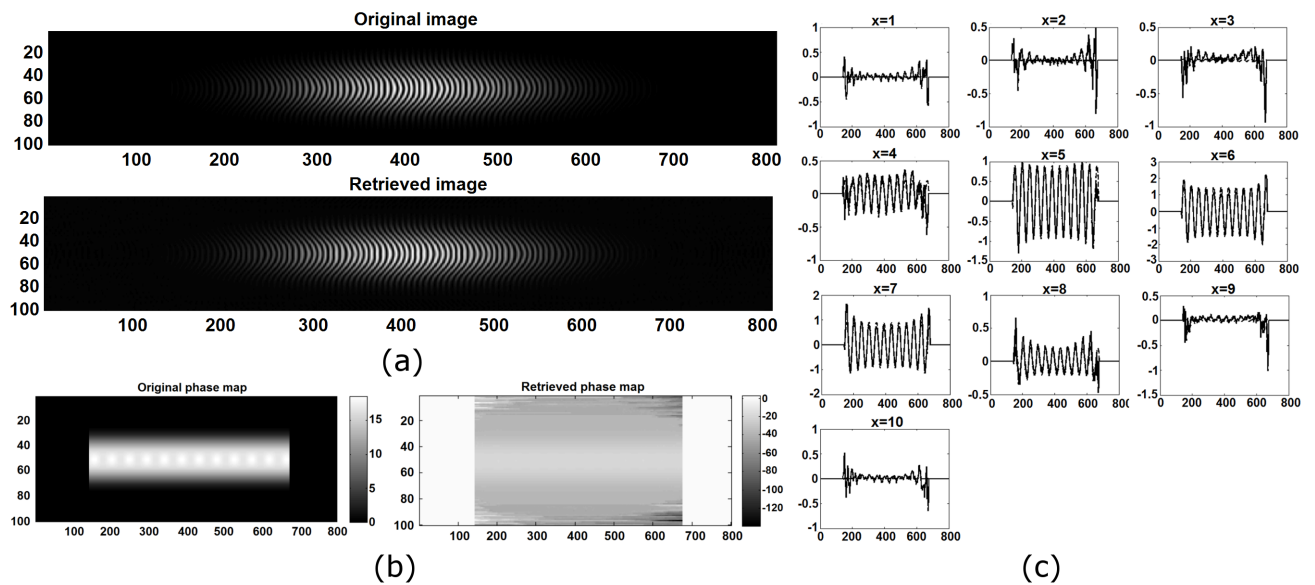


Figure 5.10: (a) Comparison of a slice of original spectral intensity profile and the profile retrieved from 5.9(b). There are 10 slices of the retrieved profiles in total. (b) The comparison between the extracted phase map from the original and the retrieved spectral profiles. (c) The central lineout of the phase map from every slice. Dashed lines are the original phase map profile and the solid lines are the retrieved phase map. Both profiles are in a great agreement.

Chapter 6

Quantitative shadowgraphy and proton radiography

Another way to diagnose plasma wakefield is via shadowgraphy. When light propagate inside the plasma, difference in the refractive index bent the light and thus forming intensity modulation on the screen. The same principle is also used in proton radiography where the beam is proton instead of light. Proton radiography is used extensively to diagnose magnetic field in laser-plasma systems. However, retrieving quantitative magnetic field or refractive index gradient from shadowgrams or radiograms (inverse problem) is a challenging task due to its non-linear relation. In this chapter, I describe my initial attempts and successful approaches to solve the shadowgraphy and proton radiography inversion problem. My successful approach uses tools from computational graphics. It was the first algorithm introduced in the plasma physics community to solve the shadowgraphy and proton radiography inverse problems in the non-linear regime and was published earlier this year in Physical Review E [25].

The previous chapters discussed diagnosing wakefields using spectral interferometry. In this chapter, I present an analysis to obtain the plasma wakefield modulation from different methods: those of optical shadowgraphy and proton radiography.

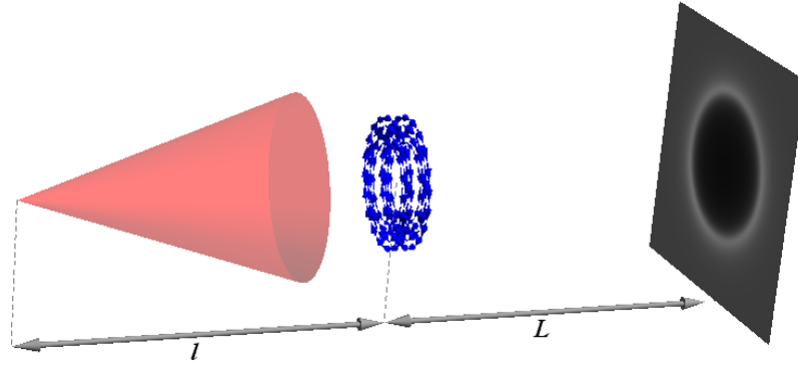


Figure 6.1: An example set up for proton radiography.

Shadowgraphy is a technique to visualise variation of refractive indices in a transparent medium [96, 97]. The deflection of light beam path results from the variation of refractive indices and this eventually causes a variation of intensity profile of the beam captured on a screen. This phenomenon is common in our daily life, for example the shadow of steam of boiling water or a candle. Some researchers use shadowgraphy to visualise flow [96], measure temperatures in combustion processes [98], and characterisation of optical systems [99].

Recently, a collaboration from Jena, Germany and Imperial College London used shadowgraphy to capture the density modulation of the plasma wakefield [23]. The variation of the density profile in the plasma wakefield induces a variation of the refractive indices of the plasma. When an optical probe beam pulse propagates perpendicularly to the wakefield, the beam pulse gets deflected and forms an intensity modulation profile on a screen after the interaction. This intensity modulation is related to the density modulation in the wakefield.

Another technique that is frequently used in plasma physics and shares the same underlying principles with the shadowgraphy is proton radiography. In proton radiography, a proton beam is fired to a small object with magnetic field and/or electric field in it. The magnetic/electric field in the object deflect the path of some protons, and thus creates an density modulation profile on a screen to capture the protons. An example set up is shown in Figure 6.1.

Despite of its wide use, shadowgraphy and proton radiography are only known as qualitative diagnostics. It is very hard to extract quantitative information out from the shadow-

grams and proton radiograms. One can easily obtain information about length and size from the shadowgrams, but it is hard to obtain other information like how strong the magnetic field at several points in the object, how much the refractive indices varies in the diagnosed system, etc. Some progress has been made, principally in the field of, quantitative shadowgraphy, but they only work for small deflections [24, 96, 100, 101]. The difficulties for large deflections comes from the non-linear nature of the shadowgraphy and proton radiography.

My work during my third year of my DPhil involved solving the quantitative shadowgraphy and proton radiography problems for large deflections. As shadowgraphy and proton radiography techniques are applied to a variety of research topics in physics and engineering, I believe it will be applied advantageously to other areas besides plasma physics.

For the rest of this chapter, I use the terms ‘shadowgraphy’ and ‘proton radiography’ interchangeably. Unless stated, the principles and algorithms below apply for both methods.

6.1 The theory of shadowgraphy and proton radiography

There are three cases considered in this chapter. The first case is charged beams propagating into objects with electric fields and the second case is with magnetic fields. The last case is a beam of light travelling through a transparent medium with varying refractive indices. In these cases, it is assumed that the longitudinal velocities of the beams are much larger than the transversal velocities of during the interaction with the objects, i.e. $\mathbf{v}_{\parallel} \gg \mathbf{v}_{\perp}$ and the transversal velocity is initially zero, $\mathbf{v}_{\perp} = 0$. It is also assumed that the longitudinal velocities of the beams do not change much, so it can be assumed to be constant.

6.1.1 Deflections with electric and/or magnetic fields

For the first and second cases, the transversal acceleration experienced by the beam are respectively,

$$\mathbf{a}_{\perp}^{(1)} = \frac{q}{m} \mathbf{E}_{\perp}; \quad \mathbf{a}_{\perp}^{(2)} = \frac{q}{m} \mathbf{v}_{\parallel} \times \mathbf{B}; \quad (6.1)$$

where q is the beam's charge, m is the beam's mass, \mathbf{v} is the beam's velocity, and \mathbf{E} and \mathbf{B} are the electric and magnetic fields, respectively. The longitudinal and transversal vectors are defined as $\mathbf{h}_{\parallel} = (\mathbf{h} \cdot \hat{\mathbf{z}})\hat{\mathbf{z}}$ and $\mathbf{h}_{\perp} = \mathbf{h} - \mathbf{h}_{\parallel}$ if the beams are moving in the z -axis. If an object contains both electric and magnetic fields, both terms are added together.

After the interaction with the objects, the beams' transversal velocity is given by $\mathbf{v}_{\perp} = \int \mathbf{a}_{\perp} dt$. Because it is assumed that the longitudinal velocities do not change much, the time integration can be changed to be $dt = dz/v_{\parallel}$. The deflection angles after the beams exit the object can be approximated as $\boldsymbol{\theta} = \mathbf{v}_{\perp}/v_{\parallel}$. Thus, the deflection angles of the beams are given by

$$\boldsymbol{\theta}^{(1)} = \frac{q}{2W} \int \mathbf{E}_{\perp} dz \quad (6.2)$$

$$\boldsymbol{\theta}^{(2)} = -\frac{q}{mv_{\parallel}} \int \mathbf{B} \times d\mathbf{z}, \quad (6.3)$$

where W is the kinetic energy of the beam (assuming non-relativistic beams), $W = mv_{\parallel}^2/2$.

In terms of potentials, the transverse electric field can be expressed in terms of the electric potential, ϕ , as $\mathbf{E}_{\perp} = \nabla_{\perp} \phi$ and the magnetic field as $\mathbf{B} = \nabla \times \mathbf{A}$ with \mathbf{A} as the magnetic vector potential. Expressing the equations above in terms of potentials yield the equations

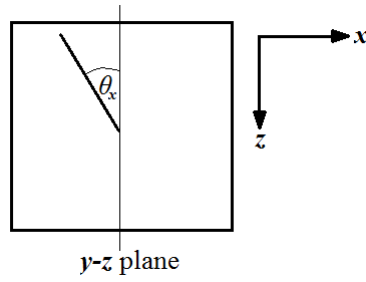


Figure 6.2: A case for light refraction.

below,

$$\boldsymbol{\theta}^{(1)} = \nabla \left[\frac{q}{2W} \int \phi \, dz \right] \quad (6.4)$$

$$\boldsymbol{\theta}^{(2)} = \nabla \left[\frac{q}{mv_{\parallel}} \int \mathbf{A} \cdot d\mathbf{z} \right], \quad (6.5)$$

where the identity $d\mathbf{z} \times (\nabla \times \mathbf{A}) = \nabla(\mathbf{A} \cdot d\mathbf{z})$ is used in the second equation. The integrations inside the square brackets produce 2D profiles on an $x - y$ plane, therefore $\nabla_{\perp} = \nabla$ in these cases.

6.1.2 Deflection with varied refractive indices

Consider a light propagating on an $x - z$ plane with an angle θ_x to the z axis. The light is refracted by an $y - z$ plane as depicted in Figure 6.2. According to Snell's law of refraction, the relations between index of refractive of the medium and the light's angle is

$$n(\mathbf{r}) \cos \theta_x = \text{constant}. \quad (6.6)$$

For simplicity, assume that the light's path is perpendicular to the y axis and is not refracted by the $x - z$ plane. The refraction on the y axis (by an $x - z$ plane) will be considered later. By taking the derivative of equation (6.6) with respect to the time, the

equation below is obtained,

$$\frac{dn}{dt} \cos \theta_x - n \sin \theta_x \frac{d\theta}{dt} = 0. \quad (6.7)$$

As the light is moving with speed c/n with an angle θ_x with the z axis, the light velocity in the medium is,

$$\frac{dx}{dt} = \frac{c}{n} \sin \theta_x \quad (6.8)$$

$$\frac{dz}{dt} = \frac{c}{n} \cos \theta_x. \quad (6.9)$$

Thus, from the equation above, we can substitute the derivatives dn/dt and $d\theta_x/dt$ as

$$\begin{aligned} \frac{dn}{dt} &= \frac{\partial n}{\partial x} \frac{dx}{dt} \\ &= \frac{\partial n}{\partial x} \frac{c}{n} \sin \theta \end{aligned} \quad (6.10)$$

$$\frac{d\theta_x}{dt} = \frac{c}{n} \cos \theta_x \frac{d\theta_x}{dz}. \quad (6.11)$$

The term for $\partial n/\partial z$ is omitted in the equation above because we only consider refraction by the $y - z$ plane. Refraction on $x - y$ plane does not have any effect on the beam's angle as long as the light comes in and out from two parallel $x - y$ planes of the object.

Substituting the equations above with equation (6.7), we obtain

$$\theta_x = \frac{\partial}{\partial x} \int \ln n \, dz. \quad (6.12)$$

The equation is also used in [101]. The angle on y axis also takes the same form as the equation above. Thus, the deviation of the beam after coming out from the object, $\theta =$

v_{\perp}/v_{\parallel} , is

$$\boldsymbol{\theta} = \nabla \int \ln n \, dz. \quad (6.13)$$

6.1.3 Intensity on screen

The three cases (charged beam with magnetic and/or electric fields and light beam with different refractive indices) share the similar property. The deflection angle after the beams coming out from the diagnosed objects can be represented as $\boldsymbol{\theta}(x_0, y_0) = \nabla \Phi(x_0, y_0)$, where Φ is the integrated property (magnetic/electric field or refractive index) of the object as shown in equations (6.2), (6.3), and (6.13). In this section and for the rest of this chapter, I denote the coordinate inside the object with subscript 0 and the coordinate for the screen without the subscript 0. The deflection potential for the shadowgraphy with light and proton radiography cases, respectively, are

$$\Phi(x_0, y_0) = \int \ln \eta(x_0, y_0, z_0) \, dz_0 \quad (6.14a)$$

$$\Phi(x_0, y_0) = \frac{q}{2W} \int \phi(x_0, y_0, z_0) \, dz_0 \quad (6.14b)$$

$$\Phi(x_0, y_0) = \frac{q}{mv} \int \mathbf{A}(x_0, y_0, z_0) \cdot dz_0. \quad (6.14c)$$

If in a case there is magnetic and electric field, the second and third equations above should be added together to produce the deflection potential.

If the distance between the object and the screen is L , the beams at position (x_0, y_0) on the object will be projected to (x, y) on the screen with the relations between the two

positions as follows,

$$x = x_0 + \boldsymbol{\theta} \cdot \hat{\mathbf{x}}L \quad (6.15)$$

$$y = y_0 + \boldsymbol{\theta} \cdot \hat{\mathbf{y}}L.$$

The equations valid for parallel probe beams. For diverging beams using the paraxial approximation, the equations above can be modified simply by replacing $x_0 \rightarrow x_0(1 + L/l)$ and $y_0 \rightarrow y_0(1 + L/l)$ where l is the distance between the beam source and the object. In all cases in this chapter, it is assumed that the object is much smaller than L and l .

Assume the intensity of the beam on the object is I_0 . Now consider a small square with the bottom left and upper right corners are (x_0, y_0) and $(x_0 + dx_0, y_0 + dy_0)$. The total flux of the beam going through the square is $I_0 dx_0 dy_0$. On the screen, the square's corners positions are transformed according to the equations (6.15), i.e.

$$(x_0, y_0) \rightarrow (x, y)$$

$$(x_0 + dx_0, y_0) \rightarrow \left(x + \frac{\partial x}{\partial x_0} dx_0, y + \frac{\partial y}{\partial x_0} dx_0\right) \quad (6.16)$$

$$(x_0, y_0 + dy_0) \rightarrow \left(x + \frac{\partial x}{\partial y_0} dy_0, y + \frac{\partial y}{\partial y_0} dy_0\right)$$

$$(x_0 + dx_0, y_0 + dy_0) \rightarrow \left(x + \frac{\partial x}{\partial x_0} dx_0 + \frac{\partial x}{\partial y_0} dy_0, y + \frac{\partial y}{\partial x_0} dx_0 + \frac{\partial y}{\partial y_0} dy_0\right).$$

The square on the object is a parallelogram on the screen. From the transformed square's corners positions above, the area of the transformed square is given by

$$\begin{aligned} A &= dx_0 dy_0 \left(\frac{\partial x}{\partial x_0} \frac{\partial y}{\partial y_0} - \frac{\partial y}{\partial x_0} \frac{\partial x}{\partial y_0} \right) \\ &= dx_0 dy_0 \left| \frac{\partial(x, y)}{\partial(x_0, y_0)} \right|, \end{aligned} \quad (6.17)$$

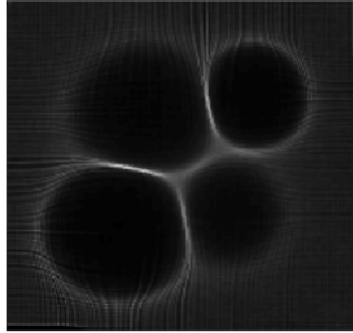


Figure 6.3: An example of radiogram with caustics.

where the term $\partial(x, y)/\partial(x_0, y_0)$ is the determinant of the Jacobian matrix of (x, y) with respect to (x_0, y_0) . As the flux that going through a square on the object plane is the same with the flux on the parallelogram on the screen, the intensity distribution on the screen can be written as

$$I(x, y) = \frac{I_0(x, y)}{\left| \frac{\partial(x, y)}{\partial(x_0, y_0)} \right|}. \quad (6.18)$$

For general case that includes parallel and diverging beam, I_0 is the intensity on the screen without deflections from the object. In the equation above, the deflection potential is encoded in the denominator, thus the intensity is related non-linearly with the deflection potential.

Moreover, the denominator in the equation above can be zero. Therefore, theoretically the intensity of the shadowgraphy or proton radiography can reach infinity. That happens when there are more than one beam line converge into the exact same position on the screen. The sharp increase of the intensity on the screen are known as *caustics*. An example of caustics is shown in Figure 6.3.

Retrieving the deflection potential from a radiogram or shadowgram is a daunting task because of its non-linear behaviour. At the time I commenced this project, there has been no introduced algorithm to invert the radiogram or shadowgram to obtain the deflection potential in non-linear regime. Among of the previous efforts are published by Kugland, *et al.* [24] and Graziani, *et al.* [102]. However, the methods only work for the linear regime or small deflections on the beam. This is the first algorithm introduced into the plasma physics

community to solve the proton radiography or shadowgraphy inversion problem.

6.2 Computational graphics tools

As the algorithm in this chapter uses a lot of computational graphics tools, it would be helpful to present a brief explanation about the tools I used in the algorithm. These are Voronoi [103] and weighted Voronoi diagram (power diagram), and Lloyd's algorithm [104].

6.2.1 Voronoi and power diagram

Consider a rectangular plane with several sites occupying some positions on the plane. For every position on the plane, there is a site that is closest to the position. A collection of positions that are closest to a particular site is called as *Voronoi cell*. Mathematically, the Voronoi cell corresponding to the i -th site at $\mathbf{r}_{0i} = (x_{0i}, y_{0i})$ covers all positions $\mathbf{r}_0 = (x_0, y_0)$ where \mathbf{r}_0 satisfies the condition below for all j ,

$$\|\mathbf{r}_0 - \mathbf{r}_{0i}\|^2 \leq \|\mathbf{r}_0 - \mathbf{r}_{0j}\|^2. \quad (6.19)$$

A Voronoi diagram is a diagram where all the cells are drawn on the plane.

In some cases, not all sites are weighted the same. Some sites with larger weights tend to increase the sizes of their cells. There are a few methods to assign weights on the sites, but the one that I used in this chapter is the additional weights method. This is called as the *power diagram*. If a power diagram has weights represented in a vector, \mathbf{w} , with the i -th site has weight of w_i , the condition for a cell in the equation (6.19) becomes

$$\|\mathbf{r}_0 - \mathbf{r}_{0i}\|^2 - w_i \leq \|\mathbf{r}_0 - \mathbf{r}_{0j}\|^2 - w_j. \quad (6.20)$$

When setting $\mathbf{w} = 0$, the power diagram equals to the Voronoi diagram. Figure 6.4(a)-(c)

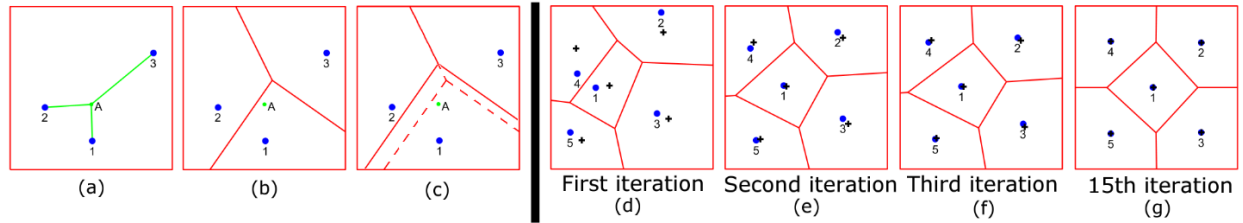


Figure 6.4: (a) An example of a case with 3 sites and a point A which is closest to the site 1. (b) A Voronoi diagram constructed with the 3 sites. Here the point A belongs to the cell of the site 1 because it is closest to the site 1. (c) A power diagram with the same sites as in the previous picture, but with weight equals to 1 assigned to the site 1. The dashed lines show the lines for the Voronoi diagram, where all weights are zeros. (d)-(g) An example of Lloyd's algorithm with sites denoted by blue circles and centroids with black plus signs (+).

shows examples of Voronoi diagram and power diagram.

6.2.2 Lloyd's algorithm

Lloyd's algorithm [104] is an algorithm to divide a bounded region (approximately) evenly into several smaller regions. It utilises Voronoi diagrams in the algorithm.

The Lloyd's algorithm starts by deploying N number of sites randomly onto the bounded plane and construct the Voronoi diagram based on the sites positions. Once each Voronoi cell is constructed, then calculate the position of the centroid of each cell. Move every site on the plane to the centroid of its corresponding cell. After that, the Voronoi diagram based on the new positions of the cells is constructed. This process is repeated several times until some number of iterations is reached or the maximum displacement of the cells is below a certain value. An example of the Lloyd's algorithm is depicted on Figure 6.4(d)-(f).

If the plane's density is not uniform, then the centroid of each cell is computed by taking into account the density distribution on the plane. This is similar to computing the centroid of 2D objects with non-uniform density. Other than that, an improvement to the Lloyd's algorithm can be made by deploying the sites initially using the simple rejection method [105]. The sites are initially deployed randomly, then for each site, generate a uniform random number. If a site is at a position with lower density, it has a low probability

to be accepted and high probability to be rejected. All rejected sites are then redeployed and random numbers for each newly redeployed sites are generated. The process is repeated until all sites are accepted.

This algorithm can be used to produce stippling pictures [106]. A stippling picture is a picture that consists of dots with same size placed randomly on the image so that it resembles the original picture. An example of the stippling picture I produced as a by-product of this project is shown in Figure 1 at the front of this thesis.

6.3 Quantitative shadowgraphy algorithms

In order to implement quantitative shadowgraphy algorithms, I need to simulate how the beam deflected from the object forms the intensity profile on the screen, or the “forward” transform. This section contains my algorithm on doing the forward transform, the first unsuccessful attempt to do the inverse transform via a simple gradient descent, and the successful attempt to obtain the inverse transform.

6.3.1 Forward transform

It was relatively straight-forward to simulate the proton trajectory going through an object with some deflection potential. Some assumptions made in this case were (1) the proton beam travelled in a straight line while interacting with the object, the direction of the velocity only changed after the interaction with the object, (2) the beam’s speed was constant from the source to the screen, (3) the deflection potential of the object did not change much during the interaction, and (4) interactions among the beam were neglected.

The first step was to calculate the deflection potential of the object. Thus the deflection angle was determined by $\theta = \nabla\Phi$. From the deflection angle, the projected position of a proton from (x_0, y_0) on the object plane to (x, y) on the screen was obtained by applying equation (6.15).

While some proton radiography codes [107] rely on simulating huge numbers of protons and binning the protons that lie on every pixel (Monte Carlo method), my method made use of computational graphics technique to reduce the memory usage of the simulation and made the intensity on the screen derivable with respect to the deflection potential. The method is simple. Assume that on the object plane between $[x_0, x_0 + \Delta x_0)$ and $[y_0, y_0 + \Delta y_0)$, the intensity is I_0 and the total flux inside this rectangular plane is $I_0 dx_0 dy_0$. Four corners of the small plane are then mapped on the screen based on equations (6.15) using the calculated θ . The rectangular plane on the object plane becomes a polygon on the screen. The intensity contributed by this polygon to a pixel on the screen plane, (x_p, y_p) with size $(\Delta x_p, \Delta y_p)$ is given by

$$\Delta I_p = \frac{A_{\text{clip}}}{A_{\text{par}}} \frac{\Delta x_p \Delta y_p}{\Delta x_0 \Delta y_0} I_0, \quad (6.21)$$

where A_{clip} is the clipping area between the parallelogram and the pixel and A_{par} is the area of the parallelogram in pixel unit. This is illustrated on Figure 6.5. Every contributions from every rectangular pixel on the object plane to every pixel on the screen is added to obtain the proton intensity profile on the screen.

Using this method, it was sufficient to simulate the trajectory of $(N_x + 1)(N_y + 1)$ protons only, which were located at the corners of every pixel on the object plane. The result of implementing this technique in comparison with binning technique with various numbers of protons are shown in Figure 6.6.

Figure 6.6 shows that by using the pixel-filling method described in this subsection, one produces a very smooth image within a reasonable time. It is smoother than the Monte Carlo (MC) method, even when the MC method used 100 particles per pixel (the pixel-filling algorithm only uses ~ 1 particle per pixel). Besides the precision of the result, this method also provided the gradient of each pixel intensity with respect to the deflection potential. This will be explained in the next subsection.

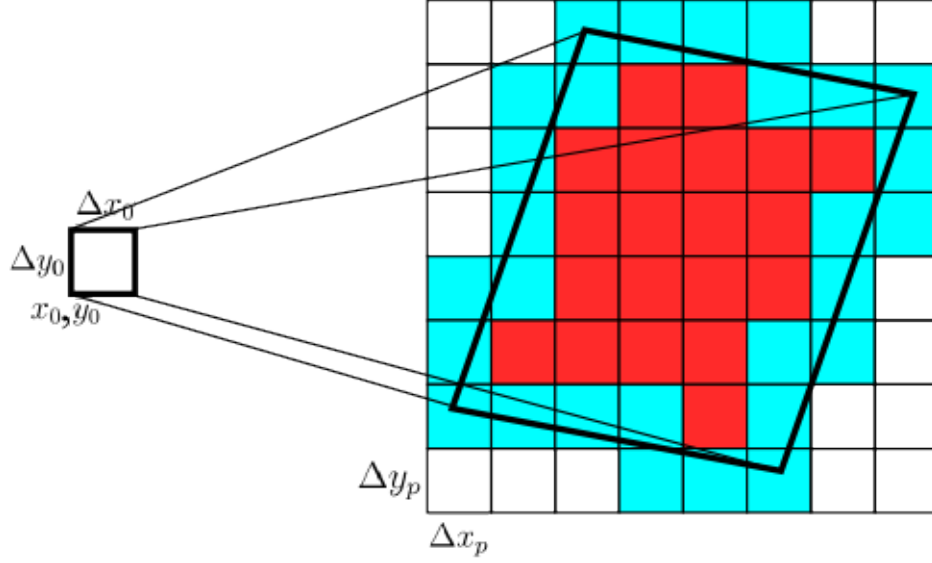


Figure 6.5: Projections of 4 protons from object plane (left) to the screen (right). The square on the object plane have flux of $I_0 \Delta x_0 \Delta y_0$ that is brought to the screen. The flux is divided uniformly between the pixels crossed and enclosed by the polygon according to equation (6.21). Cyan pixels are pixels that are crossed by the circumference of the polygon and red pixels are pixels that are fully enclosed by the polygon.

6.3.2 Inverse transform with simple gradient descent

The first idea that came to me to solve the quantitative proton radiography problem was the use of a simple gradient descent algorithm. Suppose that we have a shadowgraph image, $\vec{\mathbf{I}}_p^{(t)}$, discretised on $N_{xp} \times N_{yp}$ pixels. The vector hat is a flattening operator to change a 2D matrix into a 1D vector with the same number of elements. The forward transformation, as explained in the previous subsection, from the deflection potential of the object, Φ , to the image on the screen, $\vec{\mathbf{I}}_p$ is expressed as $\vec{\mathbf{I}}_p(\Phi)$. The objective of the gradient descent method in this section is to find Φ_s to minimise a loss function,

$$\Phi_s = \arg \min_{\Phi} \mathcal{L}(\Phi) = \frac{1}{2} \|\vec{\mathbf{I}}_p^{(t)} - \vec{\mathbf{I}}_p(\Phi)\|^2. \quad (6.22)$$

This was done via gradient descent method, where the gradient of the loss function with

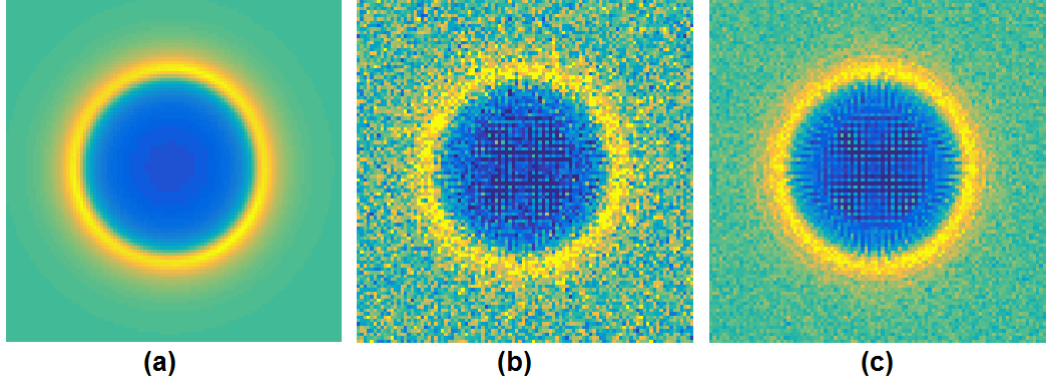


Figure 6.6: Results of various forward transform algorithms of proton radiography with the same deflection potential profiles on 100×100 pixels using (a) the algorithm explained in the subsection 6.3.1 with 10,201 particles, (b) Monte Carlo simulation with 100,000 particles, and (c) Monte Carlo simulation with 1,000,000 particles. The running times to produce each picture are (a) 6.0 seconds, (b) 3.2 seconds, and (c) 37.9 seconds, respectively.

respect to every elements of the deflection potential is

$$\nabla_{\Phi} \mathcal{L}(\Phi) = - \sum_{i,j} \left(I_{p(i,j)}^{(t)} - I_{p(i,j)}(\Phi) \right) \nabla_{\Phi} I_{p(i,j)}, \quad (6.23)$$

where the subscript (i, j) denotes the element of the matrix at the i -th row and j -th column. The gradient of the loss consists of taking the derivative of each pixel's intensity on the screen with respect to each element of the deflection potential and summing for all elements in the intensity matrix.

Since all ∇ are accompanied with a Φ -subscript, for simplicity of notation, the subscripts from the nabla symbols are removed in the rest of this sub-section. All nablas are with Φ , unless indicated.

The algorithm works as follows. It starts from an initial profile of Φ . For simplicity, Φ is initialised with an all zeros matrix. The algorithm then computes the gradient of the loss function as in equation (6.23) and update the deflection potential as

$$\Phi \rightarrow \Phi - \alpha \nabla \mathcal{L}(\Phi), \quad (6.24)$$

where α is the learning rate, a parameter that user needs to choose.

Implementing a gradient descent algorithm is fairly straightforward. However, the real challenge in this method is to calculate $\nabla I_{p(i,j)}$, the derivative of each pixel's intensity on the screen with respect to every element in the deflection potential, Φ from every pixel (i_0, j_0) on the object plane. This can be split into easier terms by the chain rule,

$$\frac{\partial I_p}{\partial \Phi} = \sum_{i_0, j_0} \left[\frac{\partial I_p}{\partial x} \frac{\partial x}{\partial \theta_x} \sum \frac{\partial \theta_x}{\partial \Phi} + \frac{\partial I_p}{\partial y} \frac{\partial y}{\partial \theta_y} \sum \frac{\partial \theta_y}{\partial \Phi} \right], \quad (6.25)$$

where all the superscript (i_0, j_0) are omitted for clarity. The main summation sums every term from every pixel on the object plane, (i_0, j_0) . The summation inside the bracket sums the change in angle from pixel (i_0, j_0) by the change of deflection potential around the pixel.

The change in position with respect to the angle can be obtained easily from equation (6.15) which is simply $\partial x / \partial \theta_x = \partial y / \partial \theta_y = L$. As the angle is $\theta = \nabla \Phi$, the change in angle with respect to the deflection potential can be written as

$$-\frac{\partial \theta_x^{(i_0, j_0)}}{\partial \Phi^{(i_0, j_0)}} = \frac{\partial \theta_x^{(i_0, j_0)}}{\partial \Phi^{(i_0, j_0+1)}} = 1 \quad (6.26)$$

$$-\frac{\partial \theta_y^{(i_0, j_0)}}{\partial \Phi^{(i_0, j_0)}} = \frac{\partial \theta_y^{(i_0, j_0)}}{\partial \Phi^{(i_0+1, j_0)}} = 1. \quad (6.27)$$

The last terms, $\partial I_p / \partial x$ and $\partial I_p / \partial y$, are a bit tricky. They can be computed by the study of equation (6.21). The derivative equation is divided into three cases, where (1) the polygon from pixel (i_0, j_0) does not enclose or cross the pixel (i, j) , (2) the polygon fully encloses the pixel (red pixels in Figure 6.5), and (3) the polygon partially encloses the pixel (cyan pixels

in Figure 6.5). The equations are given below respectively,

$$\left[\frac{\partial I_p}{\partial x} \right]_{(i_0, j_0)} = 0. \quad (6.28)$$

$$\left[\frac{\partial I_p}{\partial x} \right]_{(i_0, j_0)} = -\frac{\Delta x_p \Delta y_p}{\Delta x_0 \Delta y_0} I_0 \frac{A_{\text{clip}}}{A_{\text{par}}^2} \frac{\partial A_{\text{par}}}{\partial x} \quad (6.29)$$

$$\left[\frac{\partial I_p}{\partial x} \right]_{(i_0, j_0)} = \frac{\Delta x_p \Delta y_p}{\Delta x_0 \Delta y_0} I_0 \partial \left(\frac{A_{\text{clip}}}{A_{\text{par}}} \right) / \partial x. \quad (6.30)$$

The same equations apply for partial derivative in y .

The area for a polygon with corners in (x_i, y_i) is given by

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i-1} - x_{i+1} y_i), \quad (6.31)$$

where the corners numbering is done in clockwise direction. The index is circular, e.g. $x_{-1} = x_{N-1}$ and $x_N = x_0$.

The clipping between the polygon and the pixel was performed using the Sutherland-Hodgman algorithm [108]. Putting all these equations together, one is now able to compute the gradient of an intensity on the screen with respect to the change in the deflection potential, and thus compute the gradient of the loss, $\nabla \mathcal{L}(\Phi)$ from equation (6.23). Clearly, this is a non-trivial problem.

Results from gradient descent

The algorithm was implemented in 1D cases with a Gaussian deflection potential. I tested the cases with varying values of μ , a parameter from [24] to indicate the characteristic of the proton radiogram. Caustics appear with value of μ around the order of unity, depending on

the shape of the deflection potential. In the Gaussian case, μ is defined as,

$$\mu = \sqrt{\pi} \frac{e\phi_0 l}{W a}, \quad (6.32)$$

where e and W is the electron's charge and kinetic energy, respectively, ϕ_0 is the amplitude of the deflection potential, l is the distance from the beam source to the object, and a is the Gaussian $1/e$ -distance. In this case, the caustics appears around $\mu \approx 2.4$. If μ is much smaller than 1, the deflection potential can be obtained by Poisson's equation solver [101]. However, the solution for μ comparable to 0.1 or larger was not available at that time, let alone $\mu \sim 1$.

The gradient descent algorithm turned out to work pretty well in the 1D case, even after caustics appear. Figure 6.7 shows the results of the proton radiography inversion from 1D radiograms to the deflection potential for various values of $\mu > 1$ with some additional white noise.

One weakness in the gradient descent technique I employed is that it tended to overfit the proton radiogram's intensity. Therefore, the deflection potential it retrieved was different because of noise. An example is given by Figure 6.8 with $\mu = 2.0$ and 10% white noise. As seen in the figure, the projected proton radiogram from the retrieved deflection potential matches really well with the given proton radiogram intensity. However, the retrieved deflection potential did not agree well with the ground truth. This is a sign of overfitting.

In 2D case, the gradient descent algorithm did not perform as well as in the 1D case. The inversion results of 2D proton radiograms are shown in Figure 6.9. From the figure, one can see that the retrieved deflection potential always underestimates the true deflection potential.

A possible explanation for this failure is the algorithm converges to a local minimum, because gradient descent cannot find the global minimum if the loss function has several local minima. However, looking at the equation (6.23), $\nabla \mathcal{L}$ is not zero unless the intensity difference between the given and retrieved radiograms are zero (i.e. the term ∇I_p cannot be zero unless $I_p = 0$). Therefore, the failure is most probably because the gradient descent

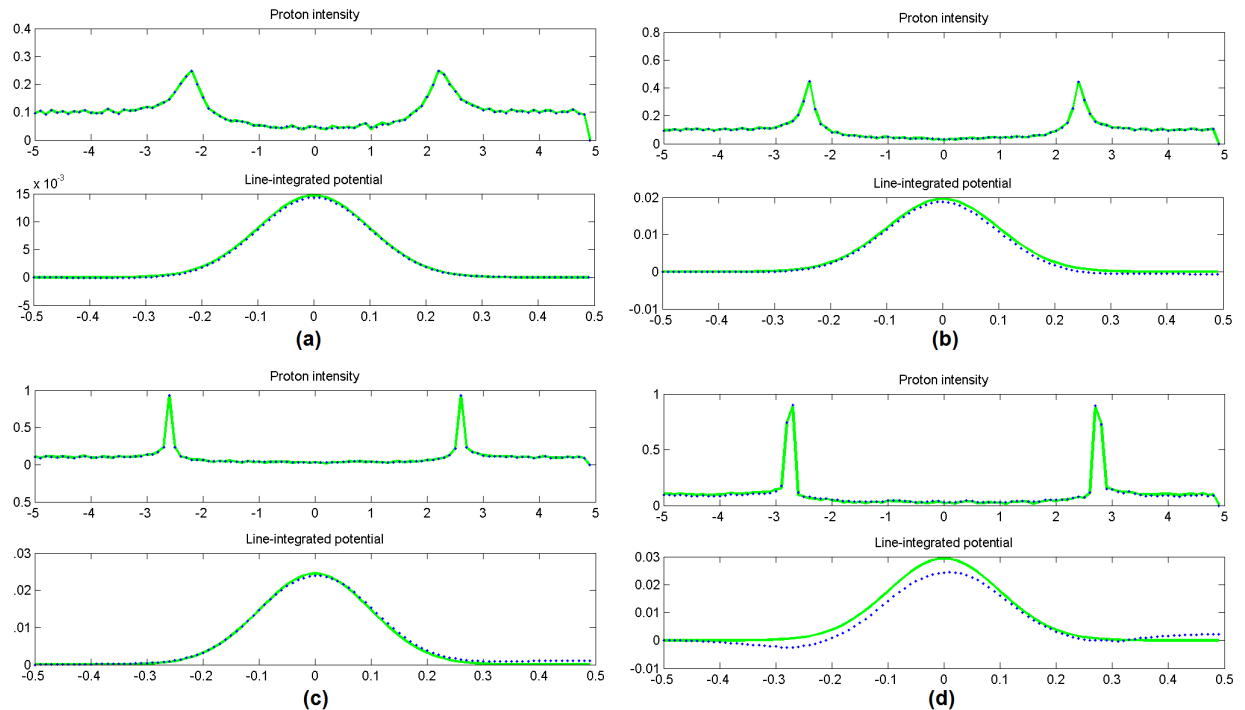


Figure 6.7: The results of 1D proton radiography inversion technique using gradient descent. The green line is the plot calculated given the deflection potential (ground truth) while the blue dots are the plots calculated given the radiogram intensity (inverse proton radiogram). The values of μ are (a) 1.5, (b) 2.0, (c) 2.5, and (d) 3.0, respectively. The caustics appear at $\mu \approx 2.4$. A Gaussian noise was added to the every radiograms with deviation of (a-b) 5% and (c-d) 7% of the mean intensity on the radiograms.

algorithm gets into a shallow valley in the loss function and makes the algorithm converge very slowly. One famous example of function with a shallow valley is the Rosenbrock function [109]. A solution to this problem, in future studies, is to employ a second order gradient descent algorithm, such as gradient descent with momentum [110] or L-BFGS [111, 112].

6.3.3 Inverse transform with power diagram

As the previous method was eventually unsuccessful in being able to retrieve quantitative information from the shadowgraph, I began to employ different methods. While reading about caustics, I found a link of a group designing surface of a glass to form an image in its shadow [113]. It is similar to shadowgraphy and proton radiography problems, so I traced back the algorithm that they did, as explained in this subsection.

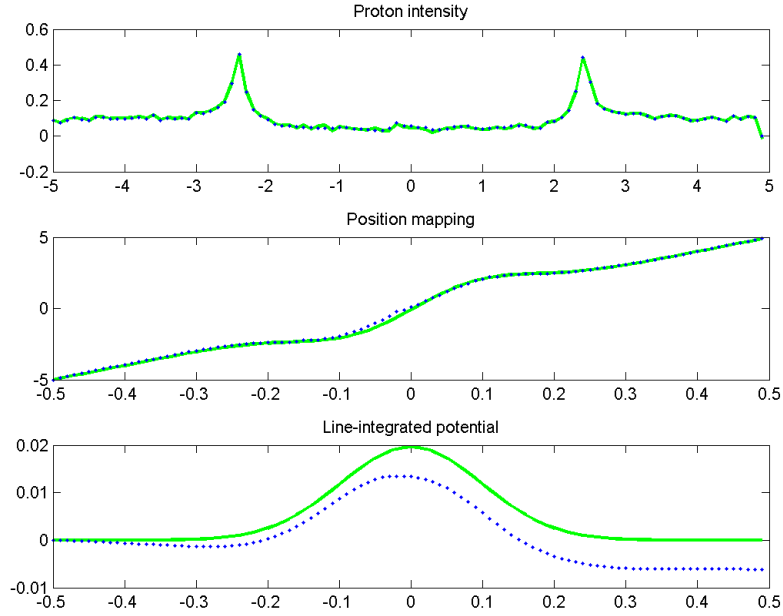


Figure 6.8: An example of overfitted results of gradient descent algorithm of 1D proton radiography inversion with $\mu = 2.0$ and 10% white noise.

The algorithm to perform quantitative shadowgraphy for non-linear cases is not new. It was invented in 1998 [114]. However, the application of the algorithm to the quantitative shadowgraphy and proton radiography problems is novel. My contribution is introducing the algorithm from computer graphics community to the plasma physics community.

Figure 6.10 shows examples of intensity profiles of the beam without and with the deflections from the object. The intensity profile without the deflections is called as “source” profile and the profile with the deflections is called as “target” profile. Assuming that the total flux between two profiles are the same, the question can be posed as “How to move the beam’s particles from the source profile to make the target profile so that the total distance by the particles is minimum?”. This is known as the “transport problem” by Monge [115].

Denote the source profile as $I_0(x_0, y_0)$ and the target profile as $I(x, y)$. Initially, a number of sites are deployed randomly on the source plane with a simple rejection method [105]. If there are more sites deployed, the resolution of the resulted picture increases. However, it also slows down the algorithm. Typical number of sites for an image with total N_{pix} pixels

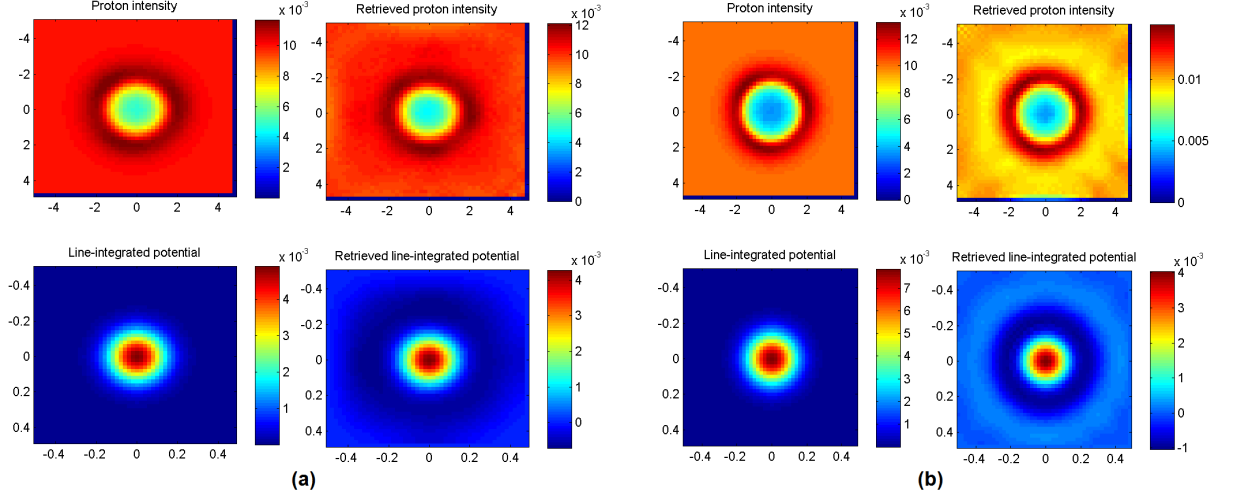


Figure 6.9: Inversion results for 2D Gaussian deflection potential with (a) $\mu = 0.5$ and (b) $\mu = 0.8$ without noise for both cases. The left pictures are the ground truths and the right pictures are the retrieved parameters.

is about $\sim N_{pix}/2$. Once the sites are deployed, Lloyd's algorithm is performed to make the sites regularly structured. From my experience, usually 50 iterations is sufficient to reach a good convergence. This produces a Voronoi diagram where each cell has approximately the same flux.

The regularly structured sites are then redeployed to the target plane by keeping their positions unchanged. Denote the i -th Voronoi cell on the source plane as V_i and the i -th power cell with weights \mathbf{w} on the target plane as $P_i^{\mathbf{w}}$. Also denote $S(V_i)$ as the total flux from the cell V_i on the source plane and $T(P_i^{\mathbf{w}})$ as the total flux from the cell $P_i^{\mathbf{w}}$ on the target plane. On the target plane, the weights for the cells are going to be optimised so that the total flux from the Voronoi cell on the source plane equals to the flux from the power cell on the target plane. Aurenhammer [114] found that to make the fluxes equal, the cost function below needs to be minimised with respect to the weights, \mathbf{w} ,

$$f(\mathbf{w}) = - \sum_i \left[w_i S(V_i) + \int_{P_i^{\mathbf{w}}} (\|\mathbf{r} - \mathbf{r}_{0i}\|^2 - w_i) I(\mathbf{r}) d\mathbf{r} \right], \quad (6.33)$$

where \mathbf{r}_{0i} and w_i are the position and the assigned weight of the i -th site, respectively. Note

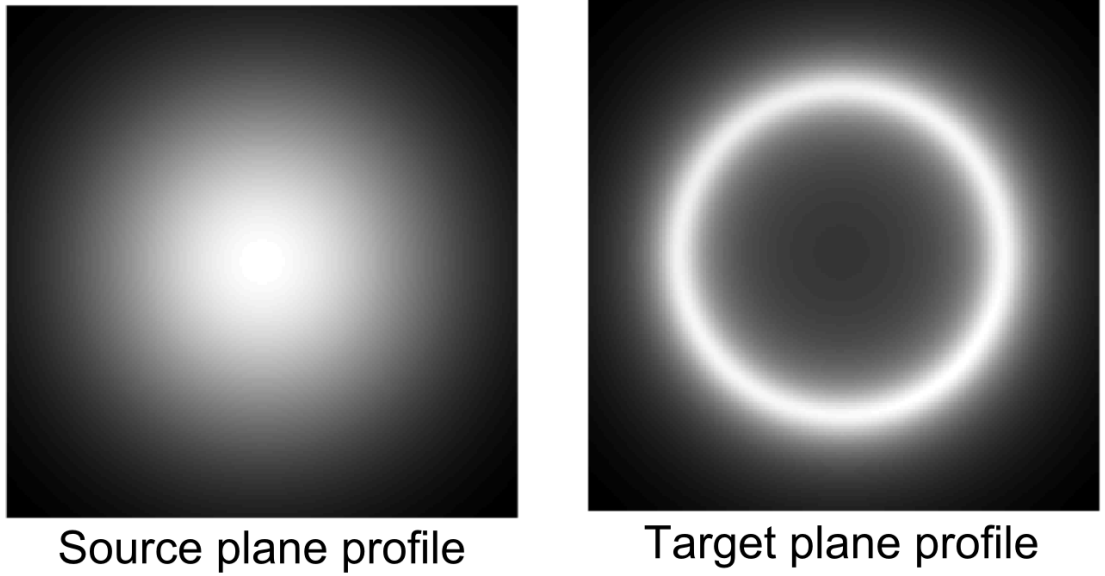


Figure 6.10: An example of a profile of the beam on the screen without the deflections from the object (source profile, left) and with the deflections (target profile, right).

that $\int_{P_i^{\mathbf{w}}} I(\mathbf{r}) d\mathbf{r} = T(P_i^{\mathbf{w}})$. To minimise the cost function above, it is easier and faster to use gradient descent, where the gradient of the cost function with respect to the i -th cell's weight, w_i , is given by

$$\frac{\partial f(\mathbf{w})}{\partial w_i} = T(P_i^{\mathbf{w}}) - S(V_i). \quad (6.34)$$

It is also worth noting that during the optimisation process, the sites' positions do not change. Only the weights are changed. During the optimisation, sometimes there are sites that do not have cells. If this is the case, then the integration on the right hand side of equation (6.33) is zero and $T(P_i^{\mathbf{w}}) = 0$. The cost function and its gradient have been employed to design surfaces of transparent objects to produce caustic designs [113].

Once the weights have been optimised, the centroids of all power cells are computed. The displacement from the centroid of the initial Voronoi cell to the centroid of the power cell becomes the displacement of the beam. However, the displacement is only known at the positions where the sites are. To determine the displacement at other positions, an interpolation is required.

In this case, the natural neighbour interpolation is utilised. However, natural neighbour

interpolation can only determine the values of the positions inside the convex hull of the sites. To solve this problem, four sites that are closest to each corner are moved to the corner of the plane. This makes the convex hull of the sites covers all the plane, thus, the natural neighbour interpolation can be used. Moving the sites to the corners makes the result picture distorted around the corner. It can be resolved by increasing the resolution by increasing the number of sites.

As the displacement is proportional to the derivative of the deflection potential, i.e. $\mathbf{r} - \mathbf{r}_0 = L\nabla\Phi$ from equations (6.15), the deflection potential, Φ can be obtained by integrating the deflection in one direction, either x or y . The result from the optimisation of the cost function in equation (6.33) is curl-free, therefore the integration result along the x or y direction should be approximately the same. In my implementation, I integrate the displacement in both directions and obtain the deflection potential as the average value of the both integration results.

The pseudo-code of the algorithm is shown in Algorithm 1 as well as the illustration on the algorithm on Figure 6.11.

Algorithm 1 Inverse shadowgraphy and proton radiography

- 1: **Input:** a shadowgram or a proton radiogram image
- 2: **Output:** Φ , the 2D deflection potential of the object
- 3:
- 4: *% Initialisation*
- 5: Deploy sites randomly on the source plane, \mathbf{x} and \mathbf{y}
- 6: **repeat**
- 7: Construct the Voronoi diagram with sites at \mathbf{x} and \mathbf{y} on the source plane
- 8: Calculate the centroid of each region, \mathbf{x}_c and \mathbf{y}_c
- 9: $\mathbf{x} \leftarrow \mathbf{x}_c$; $\mathbf{y} \leftarrow \mathbf{y}_c$
- 10: **until** any stopping conditions reached
- 11: Construct the Voronoi diagram with sites at \mathbf{x} and \mathbf{y} on the source plane
- 12: Calculate $\mathbf{S}(\mathbf{V})$
- 13:
- 14: *% Gradient-based optimization*
- 15: $\mathbf{w} \leftarrow \mathbf{0}$
- 16: **repeat**
- 17: Construct the power diagram with \mathbf{x} , \mathbf{y} , and \mathbf{w} on the target plane
- 18: Calculate $\mathbf{T}(\mathbf{P}^{\mathbf{w}})$ for each site
- 19: Calculate $f(\mathbf{w})$ and $\Delta\mathbf{w} = \nabla_{\mathbf{w}}f(\mathbf{w})$
- 20: Update $\mathbf{w} \leftarrow \mathbf{w} - \alpha\Delta\mathbf{w}$
- 21: **until** any stopping conditions reached
- 22:
- 23: *% Finalisation*
- 24: Construct the power diagram with \mathbf{x} , \mathbf{y} , and \mathbf{w} on the target plane
- 25: Obtain the centroid positions, \mathbf{x}_P and \mathbf{y}_P
- 26: Assign the displacement, $\mathbf{x}_P - \mathbf{x}$ and $\mathbf{y}_P - \mathbf{y}$, to each site
- 27: Move 4 sites closest to the corners to the corners
- 28: Get the displacement of each pixel using natural neighbour interpolation
- 29: Integrate the displacement in x or y axis to obtain Φ

6.3.4 Implementation

There are a lot of computer science techniques employed in the algorithm. The first one is the simple rejection technique. In the implementation of the simple rejection, one needs acceptance probability values for all positions on the plane. In the pixelated source plane, the

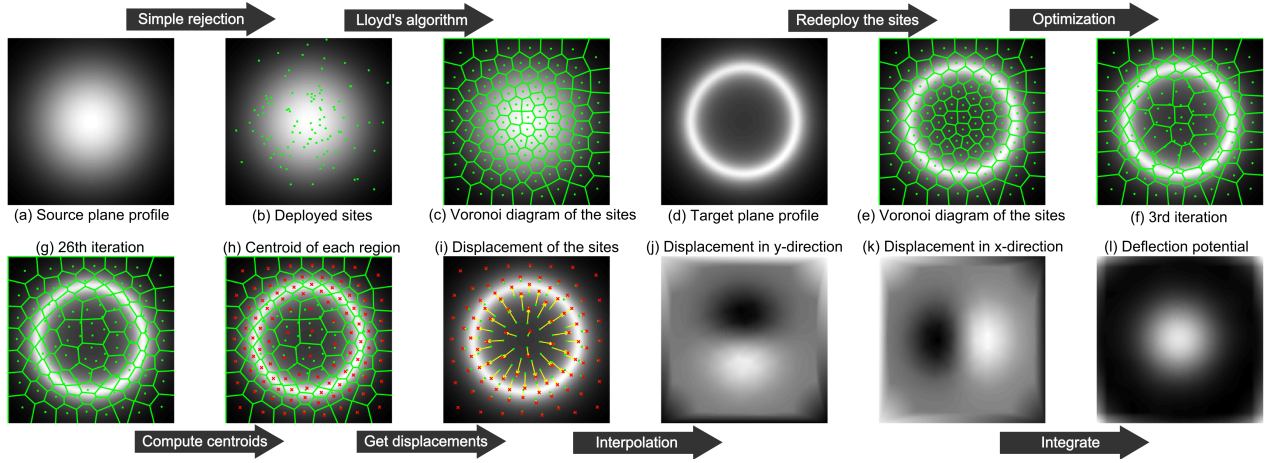


Figure 6.11: Illustration of the algorithm with 100 sites. Initially, the sites deployed randomly on the source plane with a simple rejection method is shown in figure (b). In figure (c), the sites are spaced relatively more regular as a result of Lloyd's algorithm. The deployed sites are redeployed at the same position on the target plane profile (d)-(e). Optimisation process is performed to make each cell has the same flux on the target plane with the flux on the source plane (f)-(g). In figure (h), the centroid of each cell is shown by red crosses. The displacement for each cell is then computed and shown in figure (i)-(k). Integration of the displacement yields the deflection potential of the object (l). The distortion at the corners in figure (l) is caused from moving the 4 sites to the corner as in line 27 in Algorithm 1

intensity at the position $x \in [i\Delta, i\Delta + \Delta)$ and $y \in [j\Delta, j\Delta + \Delta)$ (Δ is the pixel size) has the same value of the intensity at pixel (i, j) . The acceptance probability value at position (x, y) is given by the intensity at that position divided by the maximum intensity on the plane, i.e. $r_{max}(x, y) = I(x, y) / \max_{x,y} I(x, y)$. For every site, a random number, r , is generated from uniform distribution from 0 to 1. If $r \leq r_{max}$, the site is accepted. Otherwise, it is rejected and redeployed. This simple rejection method makes the sites gather in the higher intensity region.

To obtain the power diagram, algorithm 1 is employed using convex hull and transformation to dual space. Voronoi diagram can be obtained by setting all weights as zero in the power diagram. There is an implementation of power diagram in MATLAB, but it runs slowly for large numbers of sites. I wrote my own implementation in MATLAB to make it scale well with the number of sites. It works in a reasonable time even there are one million sites. The codes can be found on [116].

To calculate function (6.33), the values of $S(V_i)$ and $T(P_i^w)$, the flux on the source and

target plane of the cell, respectively, and $\int_{P_i^w} \|\mathbf{r} - \mathbf{r}_{0i}\|^2 I(\mathbf{r}) d\mathbf{r}$, the moment of inertia of the cell. The centroid of the cell also need to be calculated. In the pixelated source and target plane profile, the intensity inside a pixel is assumed to be constant. Therefore, it is easier to calculate the flux, centroid, and the moment of inertia of the cell by doing weighted sum of the flux, centroid, and the second moment of individual pixels that intersect with the cell. Weight of an individual pixel in obtaining those parameters is the pixel intensity value. Obtaining the intersection between the cells and the pixels is done by Sutherland-Hodgman algorithm [108] since the pixels and the cells are convex. The intersected region is a convex polygon with uniform density. The area, centroid position, and moment of inertia of a convex polygon with N vertices are

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_{i+1}y_i - x_iy_{i+1}) \quad (6.35a)$$

$$x_c = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_{i+1}y_i - x_iy_{i+1}) \quad (6.35b)$$

$$y_c = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_{i+1}y_i - x_iy_{i+1}) \quad (6.35c)$$

$$I_z = \frac{1}{12} \sum_{i=0}^{N-1} [(x_i^2 + x_ix_{i+1} + x_{i+1}^2) + (y_i^2 + y_iy_{i+1} + y_{i+1}^2)](x_{i+1}y_i - x_iy_{i+1}), \quad (6.35d)$$

where (x_i, y_i) is the coordinate of a vertex of the polygon, ordered in the clockwise direction. Note that $(x_N, y_N) = (x_0, y_0)$.

As the gradient of the cost function in equation (6.33) is available, it is better to employ a second order gradient descent for a faster convergence. However, second order gradient descent requires $O(n^2)$ memory where n is the number of parameters to be optimised. If there are one hundred thousand sites, the second order gradient descent will require memory to store $\sim 10^{10}$ values. This is hardly available in an ordinary computer. An alternative to do the second order gradient descent is using the Limited-memory Broyden-Fletcher-Goldfarb-

Shanno (L-BFGS) algorithm [111]. The complete implementation code can be found at <https://github.com/mfkasim91/invert-shadowgraphy>.

6.4 Benchmark with simulations

6.4.1 Gaussian magnetic field proton radiography

The first test for the algorithm was to retrieve a Gaussian deflection potential in a case of proton radiography with a toroidal magnetic field. Consider a proton beam with energy $W = 14.7$ MeV diverging from a source, propagating through an object along the z axis, and get onto a screen. The line-integrated magnetic field inside the object is given by

$$\int \mathbf{B} \times d\mathbf{z} = D_m \exp\left(-\frac{\|\mathbf{r}\|^2}{2\sigma^2} + \frac{1}{2}\right) \frac{1}{\sigma} \mathbf{r} \quad (6.36)$$

where D_m is the maximum value of the line-integrated magnetic field and σ corresponds to the size of the toroidal magnetic field.

In the first benchmark, the magnetic field size was $\sigma = 30 \mu\text{m}$, distance from the source to the object was $l = 1.3$ mm, the object to screen distance was $L = 2$ cm, and value of D_m was varied from 10 MG. μm to 340 MG. μm . This system gives magnification of $M = L/l = 15.4$ from the object to the screen. Caustics appeared at $D_m \geq 190$ MG. μm , so the benchmark included cases where caustics appeared.

The proton radiograms were generated based on equation (6.36) and the parameters above. Gaussian noise was added to the radiograms with standard deviation 10% of the average intensity on the radiograms. Examples of the radiograms of this system with various D_m values are shown in Figure 6.12.

From the proton radiograms, the line-integrated magnetic field was retrieved using the power diagram method and Poisson's equation solver [101]. The results is shown on Figure

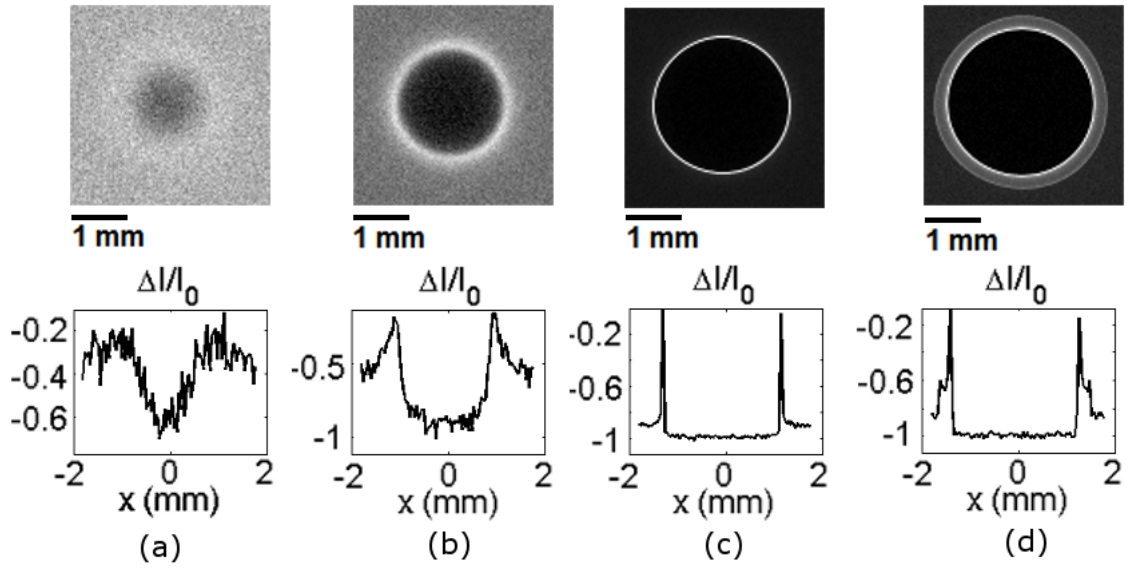


Figure 6.12: Various examples of the simulated radiograms with value of D_m equals to $\{24, 97, 194, 290\}$ $\text{MG}\cdot\mu\text{m}$ respectively from (a) to (d). These values correspond to (a) linear, (b) non-linear, (c) caustics, and (d) branching caustics case. The bottom pictures show the central lineout of the intensity profiles of the radiograms.

6.13.

The figures show that the power diagram method can retrieve the magnitude of the line-integrated magnetic field with error less than 10% even the value is as large as $D_m \approx 300 \text{ MG}\cdot\mu\text{m}$. This is a huge improvement over the Poisson's equation solver where it can retrieve accurately (with less than 10% error) when the value is less than $D_m \lesssim 30 \text{ MG}\cdot\mu\text{m}$. It is also seen on the figures that the power diagram method can still accurately retrieve the line-integrated magnetic field even when caustics appear.

6.4.2 Arbitrary shape

The next benchmark for the power diagram method was to test whether it retrieved the line-integrated magnetic field in cases where the magnetic field had an arbitrary shape. The tested cases generally includes toroidal magnetic fields with different strengths, sizes, and profiles. In 4 out of 5 tested cases, the strength of the line-integrated magnetic field was set so that caustics appear on the screen. In the last case, the case of a coaxial cable was

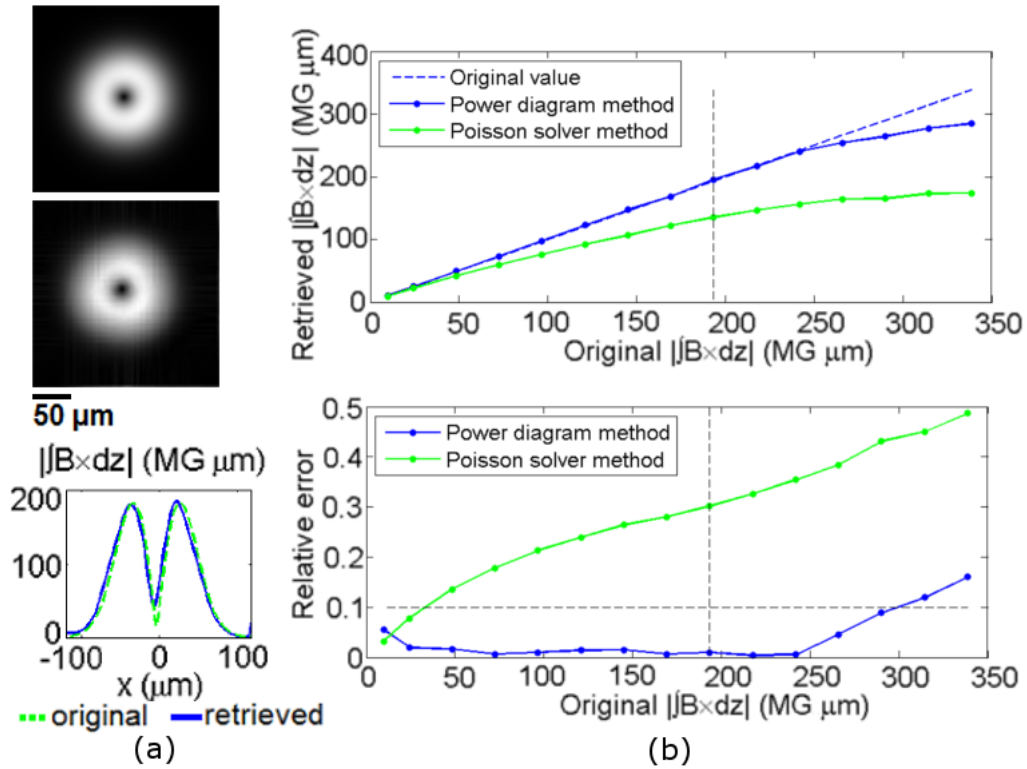


Figure 6.13: (a) (top) Original and (middle) retrieved line-integrated magnetic field's magnitude and (bottom) the central lineout of the profiles with $D_m = 190 \text{ MG} \cdot \mu$, when caustics just appear. (b) (top) Comparison of the maximum value of the line-integrated magnetic field between the original values, retrieved by power diagram method and the Poisson's equation solver. (bottom) The relative errors of the retrieved value of the line-integrated magnetic field by the power diagram method and the Poisson's equation solver with horizontal dash line shows 10% of the relative error. The vertical dash line shows the value where caustics appear.

considered where some part of the proton beam was blocked. The proton intensity on the screen without deflection was uniform.

The benchmark results on arbitrary shape magnetic fields can be seen in Figure 6.14. The figures show that the power diagram method can reliably retrieve the line-integrated magnetic field from the radiograms with caustics even with arbitrary structures and blocks in some part of the beam.

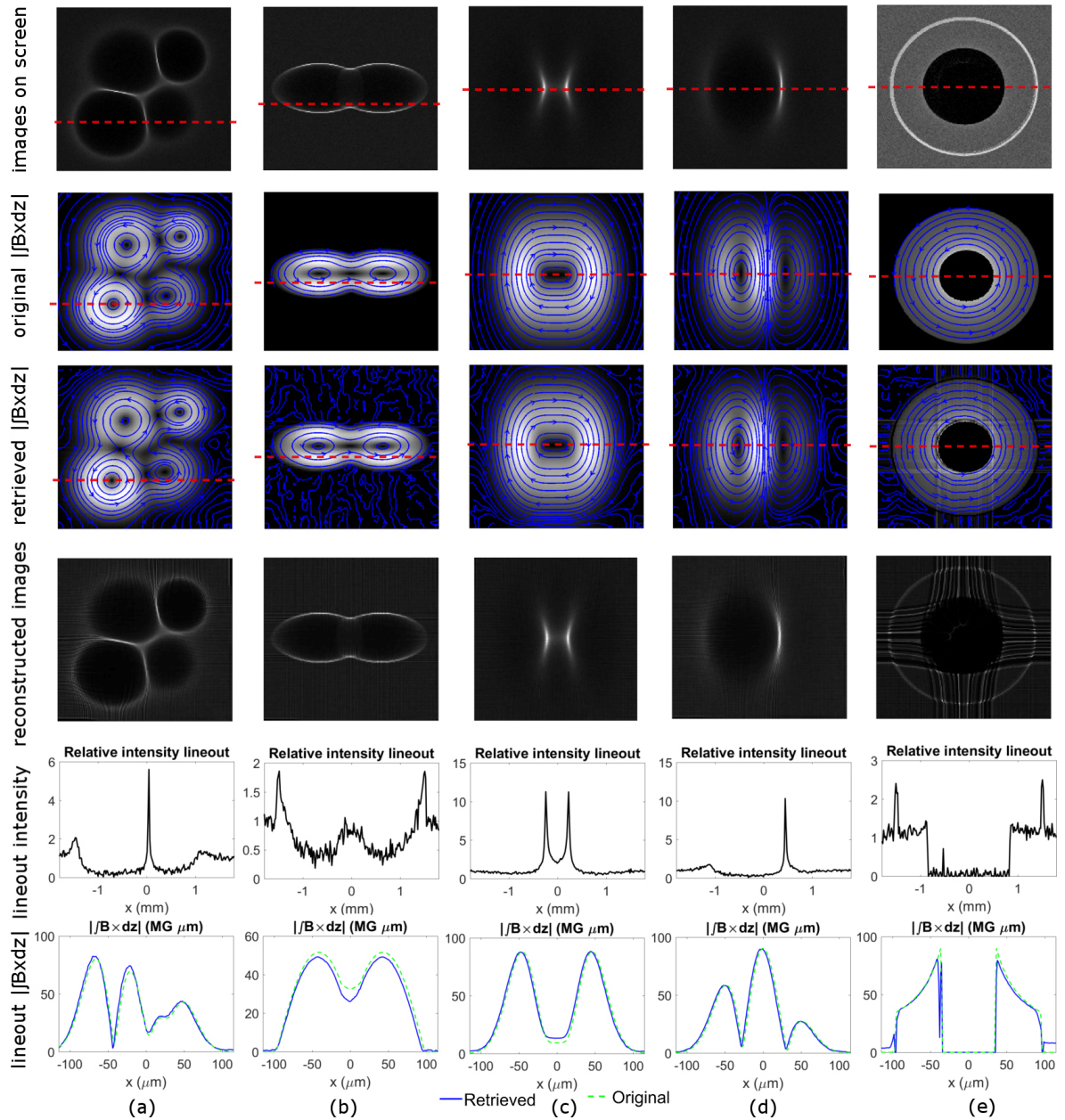


Figure 6.14: Benchmark results on arbitrary structures magnetic field. (a) A case with 4 Gaussian toroidal magnetic field with same direction, but different sizes and strengths. (b) Two \cos^2 ellipsoidal toroidal magnetic field with same direction and strength. (c) Two identical Gaussian ellipsoidal toroidal magnetic field that are close to each other. (d) Two Gaussian ellipsoidal toroidal magnetic field with different strengths and directions. (e) Magnetic field inside two coaxial cables where the outer and inner cables block the proton beam. The last two rows show the relative intensity lineout and line-integrated magnetic field at the positions shown by the red dash lines on the first three rows. Gaussian noise with 10% of the average intensity is added to the simulated radiogram images on screen (the top row).

6.5 Test with experimental results

The objective of this chapter was to develop a method to analyse proton radiography or shadowgraphy images from real experiments to get useful quantitative results. Therefore, the algorithm was tested on real experimental results from a plasma wakefield shadowgraphy experiment by Sävert, *et al.* [23].

There are several challenges in analysing experimental results that do not exist in simulations. The first one is that the beam's intensity profile without deflections (source profile) is not always constant from shot-to-shot, or maybe even unavailable. The second problem is that experimental results usually contain low frequency noise (i.e. beam's intensity modulation at larger length scale) that is amplified by the power diagram method.

There are some preprocessing techniques that can be applied to solve these problems. The simplest one is to assume the source profile is uniform and apply a high pass filter to the shadowgrams or proton radiograms (target profile). Another solution is to apply total-variation denoise (TV denoise) [117] to the target profile and denote the result as the source profile. TV denoise acts as a low pass filter that removes the high frequency modulation from the target profile. Both preprocessing techniques give approximately the same results. However, the first one is much faster than the second one. Thus, in this section, I applied the first preprocessing technique to the experimental results.

If a wakefield is formed in the plasma, the plasma density is modulated, hence the refractive index. The refractive index, η , of a plasma depends on the plasma density, n , as

$$\eta(\mathbf{r}_0) = \sqrt{1 - \frac{n(\mathbf{r}_0)e^2}{m\varepsilon_0\omega^2}}, \quad (6.37)$$

where m and e respectively are mass and charge of the electron, ε_0 is the permittivity constant, and ω is the probe's frequency. Substituting the refractive index, η , from the equation above to the deflection potential equation for shadowgraphy (equation (6.14a))

yields

$$\Phi(x_0, y_0) = \frac{e^2}{m\varepsilon_0\omega^2} \int n(\mathbf{r}_0) dz. \quad (6.38)$$

Here the approximation $\omega_p^2 = n_0 e^2 / m\varepsilon_0 \ll \omega^2$ is used. From the equation above, it is possible to retrieve the line-integrated plasma density, $\int n dz$, from the shadowgraphy method.

Analysis of experimental results from A. Sävert, *et al.* [23] is presented here. In their experiment, plasma wakefields were driven by short laser pulses with energy 750 mJ, 35 fs duration, and central wavelength 810 nm. The driver pulse was focused to produce the peak intensity of $6 \times 10^{18} \text{ Wcm}^{-2}$ and normalised intensity $a_0 \approx 1.7$. A helium gas jet was used to generate plasma with density of $1.65 \times 10^{19} \text{ cm}^{-3}$. The wakefield generated in the plasma by the driver pulse was captured with shadowgraphy method using a probe pulse with duration (5.9 ± 0.4) fs with the central wavelength 810 nm.

The results of inverting shadowgraphs from the experiment using power diagram method is shown in Figure 6.15. In this case, a high pass filter was applied to the shadowgraphs as the preprocessing and to the inverted results as the postprocessing. Figure 6.15 (b) shows the inverted results of the shadowgraphs, i.e. $\int n dz$, from Figure 6.15 (a). Although there are some distortions at the edges and corners of the figures caused by the postprocessing high pass filter, Figure 6.15 (b) shows the features of the wakefield. The central lineout of the line-integrated plasma density is shown in Figure 6.15 (c). This shows that the power diagram method is robust to process real experimental results when used with preprocessing and postprocessing high pass filters.

6.6 Conclusions

In this chapter, I have described the algorithms I developed and implemented in solving the proton radiography and shadowgraphy inverse problem. In simulating the proton radiography and shadowgraphy (forward problems), I used a filling intensity method instead of

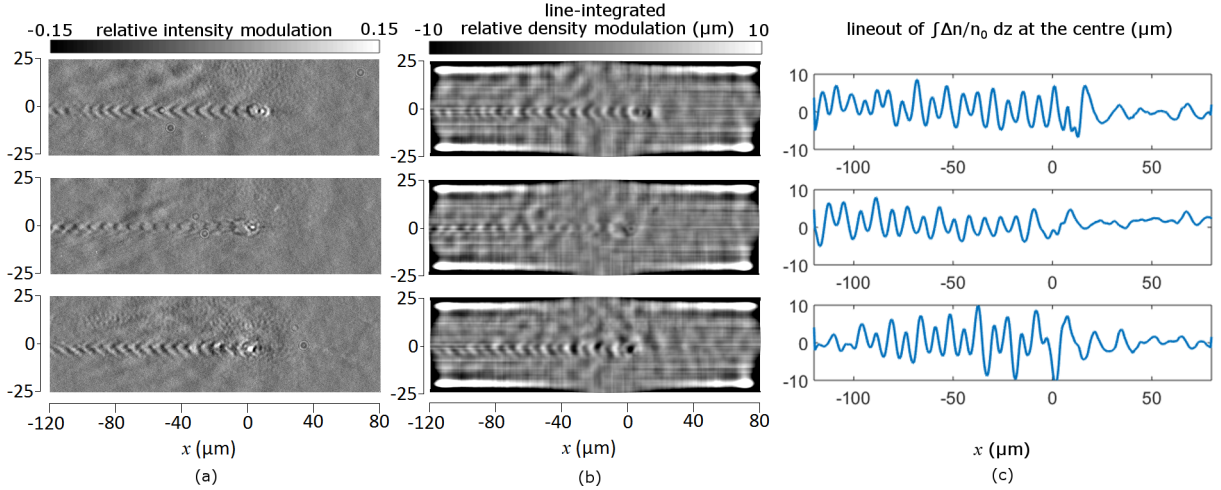


Figure 6.15: (a) Shadowgraph obtained by Sävert, *et al.* [23] in a plasma wakefield experiment. (b) The results of inverting shadowgraph figures using the power diagram method with high pass filter as the preprocessing and postprocessing. (c) Central lineout from the figures (b).

Monte Carlo simulations. It gave much better pictures in a relatively short amount of time, compared to the Monte Carlo simulations.

My first attempt in solving the inverse problem used the gradient descent algorithm. Gradient descent algorithm works very well for 1D cases, even when caustics appear. However, it failed in the 2D case probably because it got into a shallow valley in the function that reduces the gradient's magnitude. If this hypothesis is correct, then it might be solvable by employing a second order gradient descent algorithm in future studies.

The main algorithm presented in this paper is proton radiography and shadowgraphy inversion with Voronoi and power diagram. This method successfully retrieved the deflection potential on the object given the 2D radiograms, even when the caustics appear. It also provides excellent results when tested using real experimental data. This is the first algorithm introduced in the plasma physics community to solve the proton radiography inverse problem in the non-linear regime.

Chapter 7

Optimisations in plasma physics

A lot of problems in plasma physics can be posed as optimisation problems. Among the examples are selecting laser and plasma parameters to maximise electron beam energy and fitting parameters to experimental results. Realising that, I looked into machine learning and optimisation communities to see the available algorithms that can be used in plasma physics. While doing that, I developed a novel optimisation algorithm to do infinite-dimensional optimisation or 1D shape optimisation. The developed algorithm was tested against other algorithms developed recently and worked better than the tested algorithms. On the other hand, I also developed software tools to bridge physics simulations and optimisation algorithms. The software suite was built to help physicists doing optimisation of plasma physics systems via simulations. In this chapter, I show an example of optimisation results of laser plasma systems to maximise the beam energy gain. As the result, the software can find optimum parameters that agrees with W. Lu *et al.* results [50] without explicitly being taught.

During my DPhil, I had many discussions on optimisation and machine learning techniques with my supervisors, in particular, how one can apply them to plasma physics and accelerator research. There are a lot of problems in plasma physics that can be posed as optimisation problems. An example in plasma wakefield is “*what are the parameters of the*

laser pulse and the plasma to get the highest electron energy?” In that example, the objective is to maximise the electron energy and the input parameters are parameters of the laser pulse and plasma (e.g. plasma length, density, pulse duration, energy, etc.). Another example of problem that can be posed as an optimisation problem is fitting: *“what are the parameters of the system so that the simulated result is the most similar to the experimental result?”* In this case, the objective is to minimise the error between the simulation result and the experimental result.

Without losing generality, the optimisation problem can be posed as below,

$$\mathbf{x}_{\min} = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x}), \quad (7.1)$$

where \mathbf{x} is the parameters to be optimised arranged in a vector and $f(\mathbf{x})$ is the evaluation function to be minimised. Maximisation of a function, $g(\mathbf{x})$, can be performed by taking the negative its value, i.e. $f(\mathbf{x}) = -g(\mathbf{x})$.

There are two classes of optimisation algorithm based on the utility of the gradient of the evaluation function: gradient-based and gradient-less. An example of gradient-based optimisation algorithm is gradient-descent algorithm [110, 111] which is widely used in the machine learning community to train the model. An example of the gradient-less optimisation algorithm is genetic algorithm [118] and Bayesian optimisation [119–123]. The latter is gaining interest in machine learning community as means to optimise the hyper-parameters of their models.

Ideally, optimisation should be done during real experiments. However, if the optimisation during an experiment is infeasible, the optimisation should be done in simulations to get an idea of the optimal parameters that should be explored in the experiment. In most optimisation problems in plasma physics I can think of, the gradient of the evaluation function is hard to obtain because of its non-linear nature (e.g. how much the electron energy changes if the pulse is stretched a bit, etc.). Therefore, the focus in this chapter is on gradient-less optimisation algorithms.

Moreover, as the cost of doing a simulation and/or an experiment in plasma physics is relatively high (e.g. one simulation can take up to 1000 core-days), the number of evaluations of the evaluation function must be kept as low as possible. This is the most challenging part in the optimisation algorithm.

There are two contributions to the scientific communities written in this chapter. The first one is a novel algorithm to do infinite dimensional optimisation (shape optimisation) efficiently. The paper about the algorithm [30] was accepted to be presented at Bayesian Optimisation workshop at NIPS 2016, one of the largest machine learning conferences. Another one is a framework for physicists to optimise parameters using simulations.

7.1 Infinite dimensional optimisation

The main objective of the infinite dimensional optimisation or shape optimisation is to determine the best shape to minimise some objective. An example of the problem in this class is “*what is the trajectory of an object under the gravity influence so that it can travel from one point to another point as fast as possible?*” (i.e. *brachistochrone* problem). An example in plasma accelerator case might be “*what is the longitudinal shape of the plasma density profile, given other parameters, so that one can get the highest electron energy?*”

The input argument of this optimisation problem is a shape. To fully describe an arbitrary shape, one needs infinitely many points, even if the shape is one-dimensional shape, hence the name infinite dimensional optimisation.

To solve the problem, I used a Simultaneous Optimistic Optimisation (SOO) [124,125] approach and modify it so that it can solve the shape optimisation problem. I call the approach as Multi Level Simultaneous Optimistic Optimisation (ML-SOO) [30]. SOO works for optimisation with D -dimensional input arguments with bounded values, where D is finite. Extending it to ML-SOO makes it works as well for infinite dimensional input arguments. Before explaining about my approach (ML-SOO), the SOO algorithm will be explained first. In this section, optimisation is sought for maximisation instead of minimisation.

7.1.1 Simultaneous Optimistic Optimisation

Consider a function, $f : \mathcal{X} \rightarrow \mathbb{R}$ with $\mathcal{X} \subset \mathbb{R}^D$, a bounded set in \mathbb{R}^D . In most cases, \mathcal{X} is a hyper-rectangle with D dimensions, i.e. each parameter has fixed lower and upper bound values. The function has at least a global optimiser, $x^* \in \mathcal{X}$ of f , i.e.

$$f(x^*) \geq f(x) \quad \forall x \in \mathcal{X}. \quad (7.2)$$

Also, for all $x \in \mathcal{X}$, it is assumed that $f(x^*) - f(x) \leq l(x^*, x)$. The function $l : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a semi-metric function between two points in \mathcal{X} , for example $l(x_1, x_2) = \|x_1 - x_2\|^2$. In simpler words, the assumption states that the values of $f(x)$ around the global optimiser x^* are close to the global optimiser value, $f(x^*)$. An example of function that does not satisfy the assumption is a delta function, where the global optimiser value is not close to the values around it.

With the properties and assumptions above, R. Munos [124] advised optimistic optimisation algorithms to find the global optimum value of f . One of the algorithms, Simultaneous Optimistic Optimisation (SOO), does not even need to know the knowledge of the semi-metric function, l . This is the approach that is explained in this subsection.

SOO is a tree-based search algorithm which partitions the search space, \mathcal{X} into several cells. Initially, there is only one cell, which contains all the search space. The central point of the cell is evaluated. This is the root of the tree. Then the cell is divided into K smaller cells and the centre points of the cells are evaluated. The smaller cells become the children of the divided cell. If a cell has a centre value larger than any cells with larger or the same size, then the cell is divided. The tree's depth is limited by a function of the number of evaluations, $h_{max}(n)$. The pseudo-code of the algorithm is shown in 2, where the i -th cell of depth h is denoted as (h, i) and the central value of the cell is $x_{h,i}$.

Algorithm 2 Simultaneous Optimistic Optimisation (SOO)

```

1:  $\mathcal{T} \leftarrow (0, 0)$ ;  $n \leftarrow 0$ ; and  $\mathcal{L}_{\mathcal{T}} :=$  any nodes in  $\mathcal{T}$  without children
2: repeat
3:    $v_{max} \leftarrow 0$ 
4:   for  $h \leftarrow 0$  to  $\min(\text{depth}(\mathcal{T}), h_{max}(n))$  do
5:      $(h, i) \leftarrow \arg \max_{(h,j) \in \mathcal{L}_{\mathcal{T}}} f(x_{h,j})$ 
6:     if  $v_{max} \leq f(x_{h,i})$  then
7:       Expand  $(h, i)$  to  $K$  children
8:       Evaluate the values of the children
9:       Add them to  $\mathcal{T}$ 
10:       $v_{max} \leftarrow f(x_{h,i})$ ;  $n \leftarrow n + K$ 
11: until any stopping conditions
12: return  $x_n^+ \leftarrow \arg \max_{(h,i) \in \mathcal{T}} f(x_{h,i})$ 

```

One way to divide the cell into K smaller cells is to divide along the cell's longest axis. Choosing an odd value of K in this case can be advantageous because the central child has the same value as its parent, so only $K - 1$ evaluations needed while the cell size is reduced by K .

The regret of the algorithm after n function evaluations is defined as $r_n = f(x^*) - f(x_n^+)$, where $f(x_n^+)$ is the maximum value found so far, i.e. how far the found maximum value is from the true global optimum value of the function. For some functions, the regret is even exponential, i.e. $r_n = \mathcal{O}(e^{-C\sqrt{n}/D})$ for some constant, C , as shown in [124]. SOO is also combined with Gaussian Processes to give better empirical performance and theoretical guarantee, as seen in BaMSOO [125] and IMGPO [126].

7.1.2 Multi-level SOO

SOO is a powerful algorithm to optimise function with moderate dimensions (e.g. 10 dimensions). However, to optimise functionals that have infinite dimensions, some modifications are required.

Suppose that we have a 1D function, $f : \mathcal{X} \rightarrow \mathcal{F}$ where $\mathcal{X}, \mathcal{F} \subset \mathbb{R}$, which is an input of

a functional, $J : \mathcal{F}^{\mathcal{X}} \rightarrow \mathbb{R}$. An example of the functional J is the time taken for an object to travel from one point to the another point in space with the trajectory function, f . The problem considered here is to find a function, f_* , to maximise the black-box functional, J , i.e. $f_* = \arg \max_{f \in \mathcal{F}^{\mathcal{X}}} J[f]$.

It is assumed that the end points of f are fixed, i.e. at $f(x_a)$ and $f(x_b)$, where $x_a < x_b$. The functional J is also assumed to be Fréchet differentiable and f_* is twice differentiable. In order to solve it numerically, the function $f(x)$ is discretised at regular spacing points and then the number of discrete points in $f(x)$ is gradually increased. The values of $f(x)$ for $x \in \mathcal{X}$ is obtained by linearly interpolating the values from the discretised points.

Let one denote $f^{(l)}(x)$ as the discretised function of $f(x)$ at $n = 2^l - 1$ points, excluding the end points, with spacing $h = (x_b - x_a)/2^l$, and then interpolate linearly. The number of discretised points will always be counted without the end points, unless indicated. The set of all possible discretised functions is denoted as $\mathcal{F}^{\mathcal{X}^{(l)}}$, where $\mathcal{F}^{\mathcal{X}^{(1)}} \subset \mathcal{F}^{\mathcal{X}^{(2)}} \subset \dots \subset \mathcal{F}^{\mathcal{X}}$. The discretised optimiser function, f_* , is denoted by $f_*^{(l)}(x)$. The global optimiser in $\mathcal{F}^{\mathcal{X}^{(l)}}$ is defined as $f_+^{(l)}(x) = \arg \max_{f \in \mathcal{F}^{\mathcal{X}^{(l)}}} J[f]$. The global optimiser in $\mathcal{F}^{\mathcal{X}^{(l)}}$ is not necessarily the same as $f_*^{(l)}$, but $f_+^{(l)} \rightarrow f_*^{(l)}$ as $l \rightarrow \infty$. Finding $f_+^{(l)}$ is a finite dimension optimisation problem with $D = 2^l - 1$ dimensions. Figure 7.1 illustrates the notations in this paragraph.

Assumption 1 For a Fréchet differentiable functional, $J[f]$, there exists $L > 0$ and $\eta \in (0, 1)$ where $\eta L \|f_* - f\|_1^2 \leq J[f_*] - J[f] \leq L \|f_* - f\|_1^2$ for all $f \in \mathcal{F}^{\mathcal{X}}$.

Lemma 2 The L_1 -distance between $f_+^{(l)}$ and f_* is bounded by $\|f_* - f_+^{(l)}\|_1 \leq \alpha p^{-l+1}$ with $p = 4$ for some $\alpha \in \mathbb{R}^+$.

The variable α depends on the intrinsic parameters of J and f_* . From the lemma above, the search space can be decreased by 4^{-l} as l increased, provided α is chosen to be large enough. The variable α can be regarded as the width of the initial search space for $l = 1$.

Based on lemma 2, the initial idea was to start from $l = 1$, find $f_+^{(l)}$ using SOO, increase

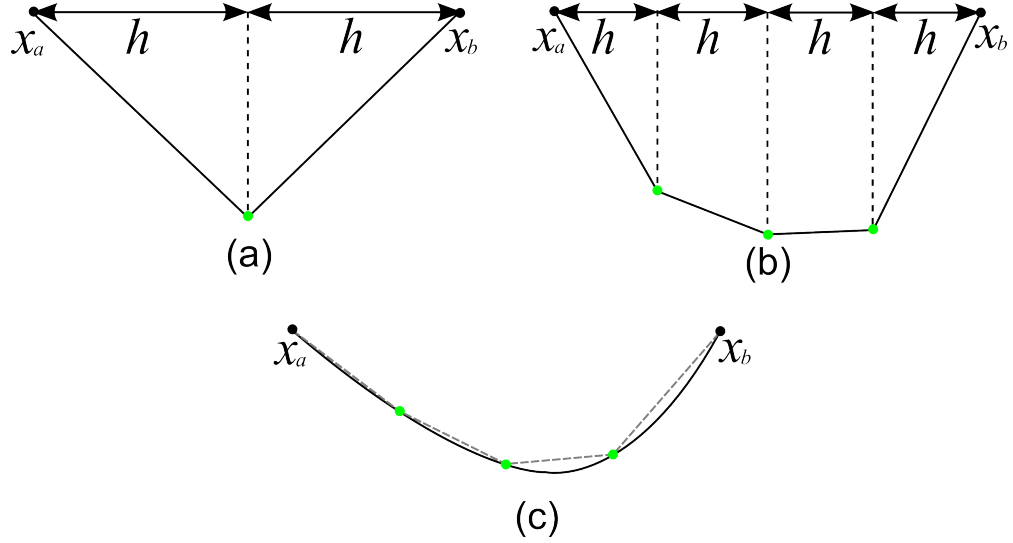


Figure 7.1: (a, b) Examples of discretised functions, $f^{(l)}(x)$ with (a) $l = 1$ and (b) $l = 2$ with linear interpolation. (c) An example of $f_*(x)$ for an arbitrary case, represented in the black solid line, as well as its discretised function, $f_*^{(l)}$ with $l = 2$, shown in the gray dashed line.

l by 1, repeat the same by reducing the search space by factor $p = 4$, and then repeat until it reaches any terminating conditions. However, this approach requires a lot of functional evaluations since to find $f_+^{(l)}$ until a very good approximation requires non-trivial numbers of functional evaluations.

Another approach to solve the problem on the previous paragraph is to reduce the search space on the additional dimensions from the beginning, so there is no need to search for the global optimum solution for each $l, f_+^{(l)}$. For example, the search space for $\mathcal{F}^{\mathcal{X}(2)}$ is the space covered by l_1 -balls with radius α/p and centres all along $\mathcal{F}^{\mathcal{X}(1)}$. An illustration for this case is shown in Figure 7.2.

Based on the reduced search space approach, it is possible to perform optimisations using SOO as in Algorithm 2 with the use of several rules in dividing cells as below, letting $p = 4$.

(1) The initial cell is one dimensional with a width of w and $l = 1$. The position of the central point of the cell represents the value of a point in $x = (x_a + x_b)/2$.

(2) If all dimensions in the cell have lengths less than or equal to p^{-l} , then new 2^l dimensions are added to the cell. The new dimensions represent the position of the new

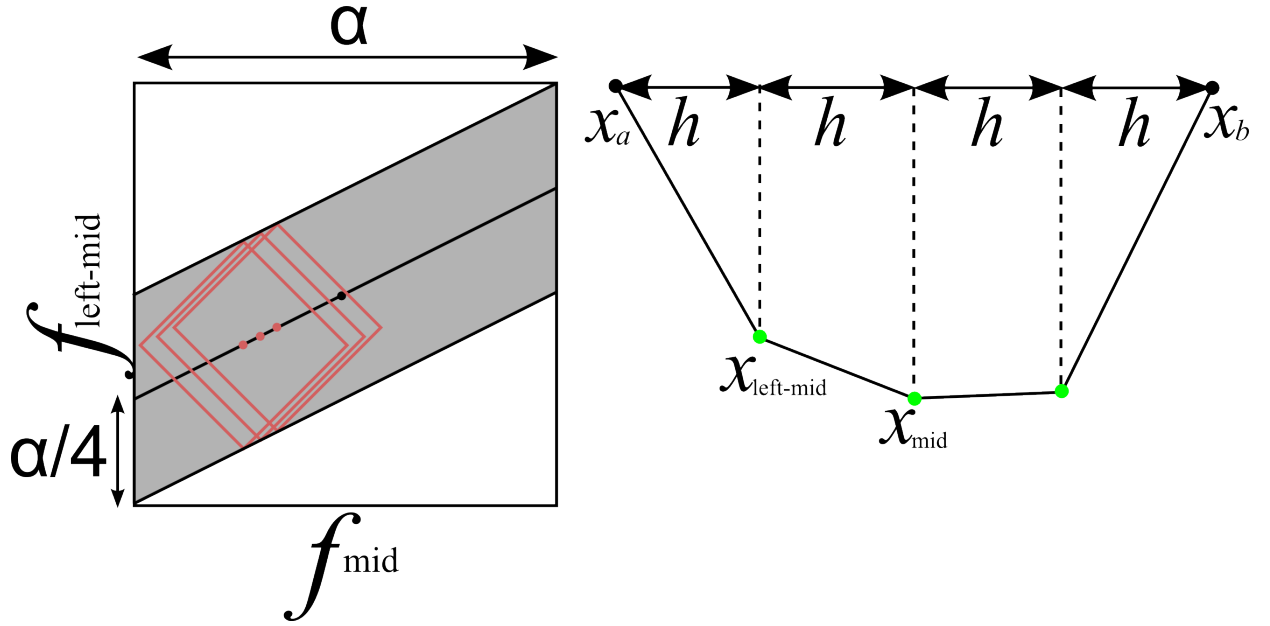


Figure 7.2: (Left) The reduced search space for $l = 2$, but only shown for two dimensions: the value of left point with the value of the middle point of the input function. The red squares are the l_1 -balls with radius $\alpha/4$. The shaded area is the search space for f_{mid} and f_{left} . The gradient of f_{mid} with respect to f_{left} is due to the linear interpolation, i.e. the value of f_{left} should change when f_{mid} is changed. (Right) Illustration of where the left point and the middle point takes place in a function with $l = 2$.

points in the discretised $f^{(l+1)}(x)$ which is placed in the middle positions of discretised points in $f^{(l)}(x)$. The new dimensions have widths of p^{-l} each. The value of l is also increased by 1.

(3) A cell is divided along the longest dimension into $K = 3$ smaller cells as in [?]. If there are more than one dimension that have the same length, then the ‘oldest’ dimension is chosen.

(4) If a cell is divided along an ‘old’ dimension, the change in position in the dimension will also change the position in the ‘newer’ dimensions around it.

As an example let $p = K = 3$. The first cell has $l = 1$ and only specifies $D = 2^l - 1 = 1$ points on f , which is $x_1^{(1)}$, the middle point between x_a and x_b . On the first division, the cell is divided into 3 smaller cells along the first dimension. After the division, the children’s dimension width is now 3^{-1} and it is equal to p^{-l} with $p = 3$ and $l = 1$. Therefore, $2^l = 2$ new dimensions are added to the children cell which correspond to the middle points between x_a

and $x_1^{(1)}$, and between $x_1^{(1)}$ and x_b .

With $l = 2$ and $D = 2^l - 1 = 3$ dimensions, let one denote the discretised points as $x_i^{(2)}$ with $i \in \{1, 2, 3\}$. The second dimension is the oldest dimension because it was the same dimension as the dimension with $l = 1$. The first and third dimensions are the newer dimensions. As all the dimensions now has width of 3^{-1} , the older dimension is divided first, i.e. $i = 2$. Because an older dimension is divided, some of its children have different positions in the second dimension and thus change the positions in the first and third dimensions. If one of the children is shifted in the second dimension by $\Delta x_2^{(2)}$ with respect to its parent's position, then the first and third dimensions also shift by $\Delta x_2^{(2)}/2$, because linear interpolation is used. Figure 7.3 illustrates the steps of the algorithm.

For the next division, as the longest dimensions are the first and the third dimension, it can choose any dimension to be divided. As the divided dimension is the newest dimension, it does not shift the positions in other dimensions. The division process of cells is repeated until the algorithm stops.

One can use different methods of interpolation other than linear interpolation. If a different interpolation method is employed, it changes the upper limit of p , but it also needs different assumptions on the optimal function, f_* . For example, quadratic interpolation yields the upper limit of p to be 16, but the fourth derivative of f_* needs to be defined and bounded. Increasing the order of polynomial in the interpolation does not always help, because the upper bound of m -th derivative of f_* can increase as in Runge's phenomenon [127].

7.1.3 Technical implementation details

Working with small number when representing a cell's dimension width can be prone to floating point errors. Therefore, for each D dimensional cell we define an array, $d_i^{(l)}$, with length of D that represents the log of the width of the cell divided by the initial cell's width, $d_i^{(l)} = -\log_K(w_i^{(l)}/w)$. Therefore, comparing the dimension's width can be done by

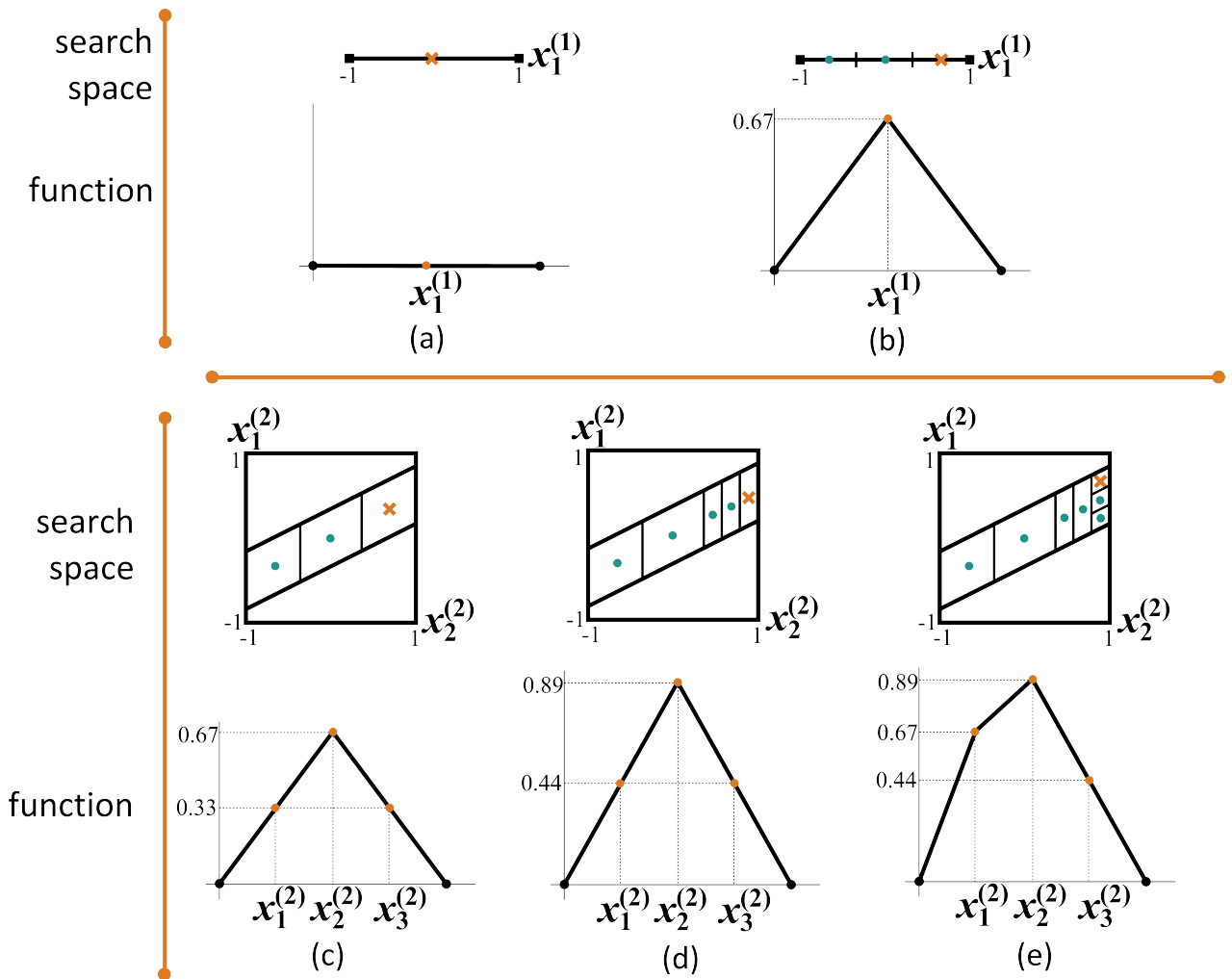


Figure 7.3: Illustration of the cell division rule in the ML-SOO. The functions illustrated at the bottom part is the representation of the cross in the search space.

comparing $d_i^{(l)}$.

For some l , comparing which dimension is older can be done by counting how many consecutive zeros at the least-significant bits of its dimension in its binary representation. For example, for $l = 3$ and thus it has $D = 7$ dimensions and numbered from 1-7 ($x_1^{(3)}$ is the closest to the left end and $x_7^{(3)}$ is the closest to the right end), the 4th dimension is the oldest because it has the 2 consecutive zeros on its least significant bits, i.e. 100, then followed by 2nd (010) and 6th (110) dimensions.

If the dimension has q consecutive zeros on its least significant bits, then changing the dimension affect $2^q - 1$ other dimensions on each side. For example, with $l = 3$, changing the 4th dimension (100, $q = 2$) will also change the 1st, 2nd, and 3rd dimensions as well as the 5th, 6th, and 7th dimensions. On the other hand, changing the 6th (110, $q = 1$) dimension also changes the 5th and 7th dimensions. This applies for any values of l .

The amount of change of the j -th dimension if i -th dimension changes can be calculated using linear interpolation. If i , in its binary representation, has q consecutive zeros on its least significant bits, the change of the j -th dimension with respect to the i -th dimension change can be written as

$$\frac{\partial x_j^{(l)}}{\partial x_i^{(l)}} = \begin{cases} 1 - (i - j)2^{-q} & : \text{if } (i - 2^q) < j \leq i \\ 1 - (j - i)2^{-q} & : \text{if } i < j < (i + 2^q) \\ 0 & : \text{otherwise.} \end{cases} \quad (7.3)$$

7.1.4 Analysis

A regret after n functional evaluations is defined by

$$r_n = J[f_*] - J[f_{n*}] \quad (7.4)$$

where f_{n^*} is the maximum input of J found among n functional evaluations. The algorithm employed here is a special case of SOO [124]. Therefore, the derivation of the regret follows the steps in [124] first and then it is specified to the case in this paper.

Assumption 3 For a semi-metric, $d(f_1, f_2) = L\|f_1 - f_2\|^2$ for any $L > 0$, there exists a decreasing sequence $\delta(h) > 0$ for any cell (h, i) at depth $h \geq 0$, where $\sup_{f \in F_{(h,i)}} d(f_{h,i}, f) \leq \delta(h)$ with $F_{(h,i)}$ is the Hilbert space covered by the cell (h, i) .

In other words, the assumption states that the cell size, which is defined by the maximum l_2 distance in the Hilbert space, decreases as h increases. This assumption holds in our algorithm as the cells are divided as h increases.

If the cell containing the global optima, f_* is at depth h_* , the regret is bounded by $r(h_*) = J[f_*] - \max_{(h,i)} J[f_{h,i}] \leq \delta(h_*)$. The inequalities can be obtained from assumptions 1 and 3. Thus, the regret bound at depth h can be obtained by calculating cell size at depth h . For the sake of simplicity, it is assumed that $p = K = 3$ and the width of the root cell is $w = 1$. Although we used $p = 4$ instead of 3, the upper bound of the resulted regret will still be applicable.

Lemma 4 Consider a cell with some value of l with $D = 2^l - 1$ dimensions with width for each dimensions are contained in an array $\mathbf{w}^{(l)}$. The maximum L_1 distance of the elements in the cell is

$$S(\mathbf{w}^{(l)}) = \frac{2}{D+1} \sum_{i=1}^D \sum_{j=1}^D J_{j,i}^{(l)} w_i^{(l)} + \frac{K^{-l+1}}{2(K-1)},$$

where $J_{j,i}^{(l)} = \frac{\partial x_j^{(l)}}{\partial x_i^{(l)}}$ is an element of a Jacobian matrix as in equation (7.3).

For a consistent division algorithm, the widths for each dimensions of a cell are always the same with other cells at the same depth. Based on the equation in lemma 4, the cell size is minimised if the dimension with the largest $\sum_{j=1}^D J_{j,i}^{(l)} w_i^{(l)}$ is divided. However, in practice

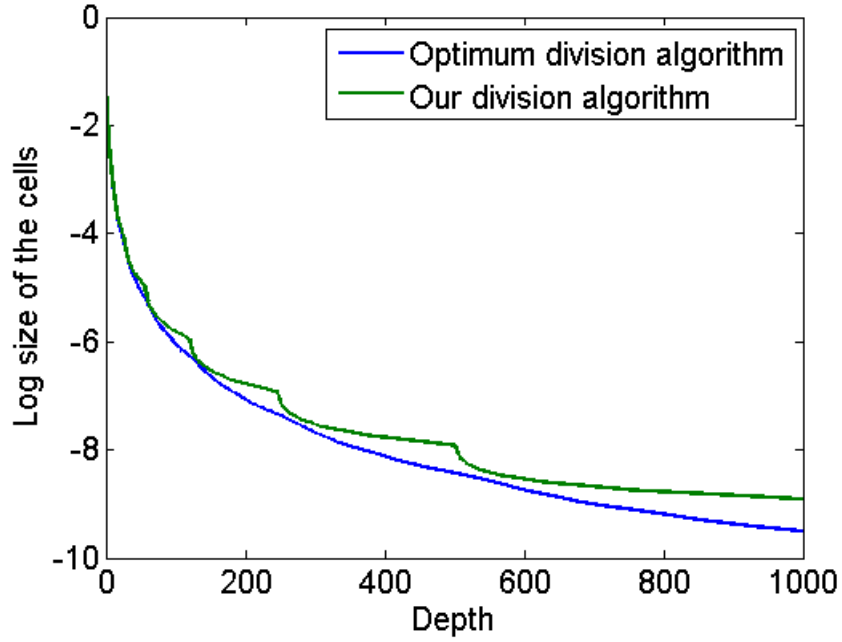


Figure 7.4: The log size of the cells, relative to the initial cell size, as function of the depth in the tree. With our division algorithm, the cell sizes are not really much different with the cell sizes divided by the optimum division algorithm. In practice, our division algorithm gives better performance.

this does not work as good as the proposed division algorithm. Figure 7.4 shows the log of cell size versus the tree depth using the proposed division algorithm and the optimum division algorithm by dividing the largest $\sum_{j=1}^D J_{j,i}^{(l)} w_i^{(l)}$. It is also seen on the figure that the discrepancy between the cell size divided using the proposed algorithm with the optimum division algorithm is not really significant. For $h = 1000$, the cell size divided with optimum division algorithm is almost half of the cell size divided by the proposed algorithm.

From Figure 7.4, it is seen that there are sudden decrease on cell size at some points. This happens when new dimensions are added to the cell and the oldest dimension is divided. When new dimensions are added, all dimensions have the same width. Thus, when the oldest dimension is divided, it gives large decrement as it has the largest $\sum_{j=1}^D J_{j,i}^{(l)} w_i^{(l)}$ among all dimensions. The upper bound on the cell size as function of depth can be derived from this point of depth and then generalised to other depths.

Lemma 5 *If the deepest cell containing the global optima, f_* is at depth h_* , the regret is*

bounded by

$$r(h_*) \leq L \left(\frac{h_*}{2} \right)^{-2 \log_2 K} \left[\log_2 \left(\frac{h_*}{2} \right) + \frac{K}{2(K-1)} \right]^2.$$

In order to determine the global optima depth, h , with the functional evaluations, N , we define a set I_h as number of cells which the central value, $J[f_{h,i}]$, satisfies $J[f_*] - J[f_{h,i}] \leq \delta(h)$. As in SOO algorithm, the deepest cell that contains f_* does not get divided if it is dominated by other cells, which are members of I_{h_*} . Thus, the lower bound of h_* as function of number of node expansions, n , is given below.

Lemma 6 [Lemma 2 in R. Munos (2011)] For any depth $0 \leq h \leq h_{\max}(n)$, wherever $n \geq \sum_{j=0}^h |I_j| h_{\max}(n)$, we have $h_* \geq h$.

With odd numbers of K and number of node expansions, n , the number of functional evaluations is given by $N = n(K-1)$.

Definition 7 There exist constants, β and k , such that $|I_h|$ is bounded by $|I_h| \leq \beta k^h$ where $k \in [1, K]$.

For $k = 1$, the number of elements in I_h is constant for all depth in the tree. If all elements in the tree is in $|I_h|$, then $k = K$. With definition 7 and lemma 6, we can substitute the deepest depth of the cell containing f_* with the number of functional evaluations to lemma 5.

Theorem 8 For N functional evaluations with K is an odd number, the regret is bounded by $r(h)$ in lemma 5 with

$$h_* \geq \log_k \left[\sqrt{\frac{N}{K-1} \frac{k-1}{\beta} + 1} \right] - 1$$

for $k \in (1, K]$ and

$$h_* \geq \frac{1}{\beta} \sqrt{\frac{N}{K-1}} - 1$$

for $k = 1$.

Thus, $r_N = \mathcal{O}[(\log_k N)^{-2 \log_2 K} (\log_2 \log_k N)^2]$ and for $k = 1$, $r_N = \mathcal{O}[N^{-\log_2 K} (\log_2 N)^2]$.

7.1.5 Numerical experiments

To test the performance of the algorithm, some numerical experiments on physical cases have been performed. The algorithm is benchmarked against SOO [124] and IMGPO [126]. Those algorithms are also gradient-free optimisations. As the other algorithms are designed for fixed dimensions search space, it is tested with $D = 7$ dimensions.

The algorithms are tested for solving brachistochrone and catenary problems. The first case is brachistochrone problem, where the algorithms need to determine the path between 2 points in 2D space so that an object starting from the first point can travel in influence of gravity to the other point as fast as possible. The gravity is assumed to be 1. The positions of the end points are $(0, 0)$ and $(1, 0)$ and the beads have initial velocity $v_0 = \sqrt{2/(2 + \pi)}$. The minimum time required in this case is $\pi/\sqrt{2 + \pi}$. The second case is catenary problem, which is the minimum area when the path between two end points is rotated along the x -axis. The end points are located at $(0, 0)$ and $(1, 0)$. The third case is brachistochrone problem with linear drag, $F_d = -\beta v$ with $\beta = 0.02$. The end points are located at $(0, 0)$ and $(1, 0.417)$. The object has initial velocity of $v_0 = 1.19$. The brachistochrone with drag is a complicated problem where the solution and Fréchet derivative cannot be obtained in a straightforward manner. The actual solutions for the test cases are cycloid, hyperbolic cosine, and the equation (35) from [128], respectively. Output of the functional is the negative value of time required to travel such shape. It returns a very large number if it is impossible to reach the other end.

In SOO and IMGPO, a cell is divided into three smaller children along the longest

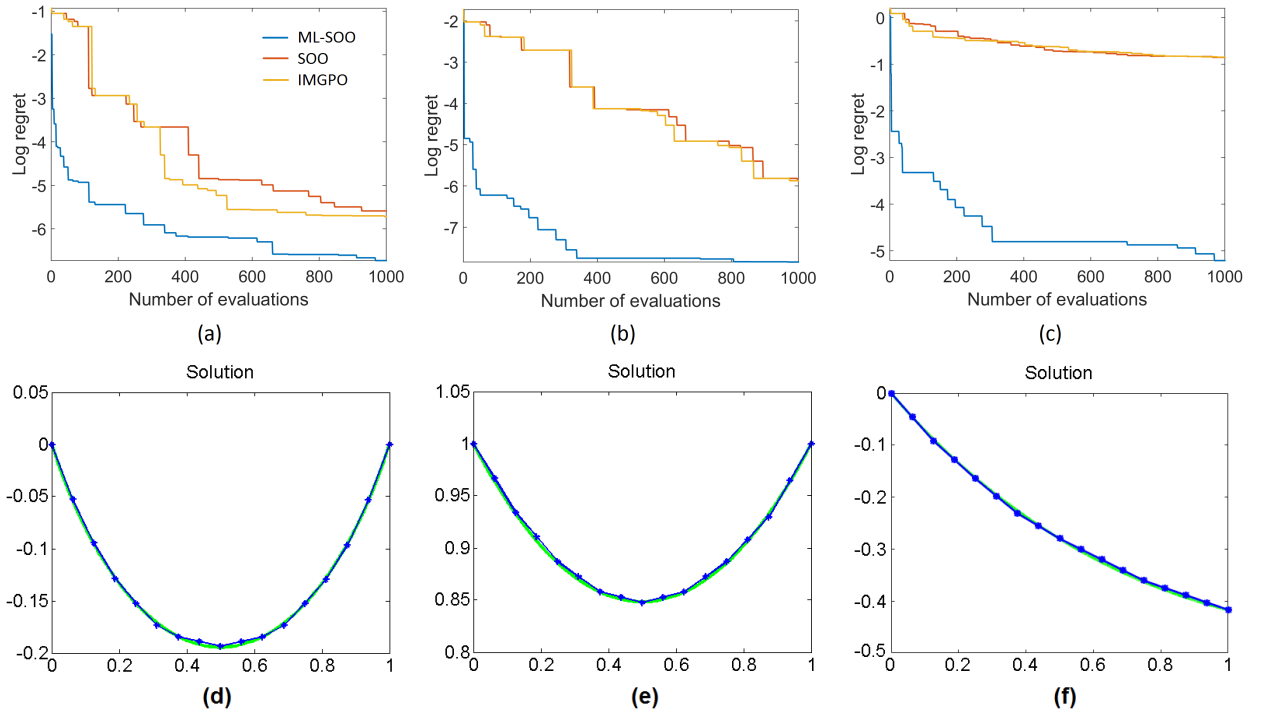


Figure 7.5: Top: Log regret versus number of functional evaluations using multi-level SOO, SOO, and IMGPO for (a) the brachistochrone problem, (b) the catenary problem, and (c) the brachistochrone problem with drag. Bottom: Comparison of the solutions obtained by the algorithm with the actual solutions.

dimension. The parameters that were used in IMGPO are the parameters used in [126]. The search space for the algorithms is confined to $[0, 1]$ for each dimension. For the new algorithm, the initial bound is set to be $[-1, 1]$, except in the second case where the bound is $[-2, 2]$. The bound is reduced by factor of 4 every time new dimensions are added.

The performance of the algorithms in all test cases is shown in Figure 7.5(a)-(c). The solutions obtained by the new algorithm are also plotted with the optimum solution in Figure 7.5(d)-(f). As demonstrated in the figures, the new algorithm performs better than the other algorithms in all test cases. The solutions obtained by the new algorithm are also quite close to the optimum solutions. In all cases, the solution found by the new algorithm consists of 15 points which is already more than the dimensions tested by other algorithms. The number of discretised points in the solutions can be increased by running the algorithm for longer.

7.2 Optimisation via simulations

As stated previously, there are many problems in physics that can be posed as optimisation problems. There are many of optimisation algorithms already available and it is a very active field of research, especially during the blossoming of machine learning in the last few years. Most of the optimisation algorithm works as follows: (1) it suggests a single or multiple values of parameters to be evaluated or tested or simulated that is to be thought to be optimum, (2) receives the evaluation results, and (3) repeat the steps (1)-(2) until maximum number of evaluations or other termination conditions reached. Unfortunately, in many cases in physics, the cost of evaluating or simulating a set of parameters is expensive. In some cases, supercomputers are required.

Fortunately, there has been some algorithms that can get reasonable results within reasonable numbers of evaluations. Among the examples are CMA-ES [129] and Bayesian Optimisations [119]. For high dimensions (high number of parameters), CMA-ES is the current state-of-the-art of the optimisation algorithms. It is based on evolutionary strategy and works as follow in D -dimensional space (i.e. there are D parameters to be optimised):

1. it is initialised by setting an initial central point, μ , and the initial covariance matrix, \mathbf{K} ,
2. the algorithm generates N random points with Gaussian distribution with central position μ and the covariance \mathbf{K} in the D -dimensional space,
3. it evaluates the values of the generated points and sort the points based on their values,
4. it updates the central position and the covariance matrix based on the m best points and their previous values,
5. repeat from step 2 until any termination criteria is reached.

Some of the advantages of using CMA-ES are that the evaluation process can be parallelised and also its hyper-parameters already have the default values.

Despite its ability in optimising systems with many parameters, the use of CMA-ES

or optimisation algorithms seems not employed much in physics. In my case in plasma accelerator physics, the problem is that it is hard to bridge the optimisation algorithms and the simulations. Because of that, I decided to make a framework to bridge the optimisation algorithms with physics simulations.

7.2.1 Software architecture

The software requires the simulations executed using a workload management platform, e.g. LSF, as the simulations are submitted as jobs. This is common in modern supercomputers. The software itself does not need to be executed in the same machine, as long as it has SSH access to the computing machine.

The architecture of the software is shown in Figure 7.6. It consists of a database and 3 main blocks: optimiser, submitter, and job-checker. The operations of those 3 blocks are executed sequentially as follows:

1. [**optimiser**] reads the config.json file;
2. reads the database to resume the previous state;
3. puts the suggested parameters to the database as new entries;
4. [**submitter**] obtains unsubmitted jobs from the database;
5. submits the jobs to the computing server;
6. checks if the jobs are submitted successfully;
7. if yes, marks the jobs as submitted, otherwise, resubmits;
8. [**job-checker**] gets the unfinished jobs from the database;
9. checks if any job has finish;
10. downloads the output of the finished jobs and does the post-processing to convert the result into a number to be optimised (e.g. electron energy);
11. marks the jobs as finished;

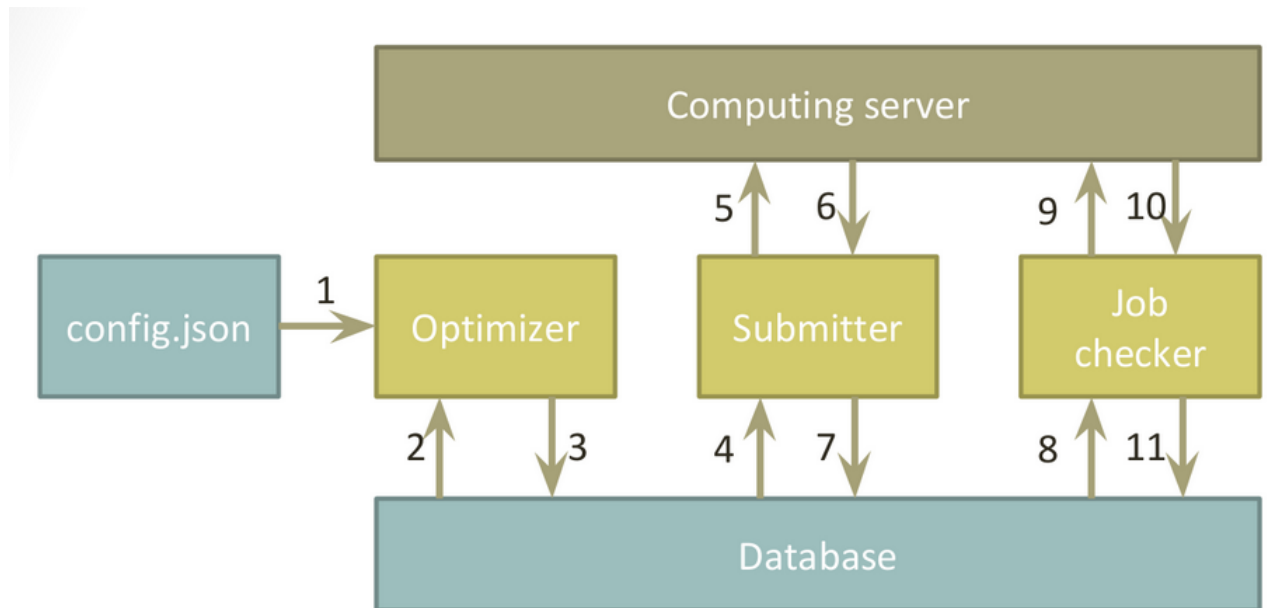


Figure 7.6: Software architecture of the optimisation framework.

12. repeats the process from step 2

As the optimisation of a system could take very long time (days, weeks, or even months), the software needs to be robust and able to resume the optimisation progress if something interrupts, so that the users only need to set up the optimisation process, run it, and leave it until it finishes.

Although the software is written in Python, the simulations to be executed can be written in any language, as long as it can be executed in the computing machine. It is also easier if the simulation program receives a text file as its input. There is also no need to change the simulation program as the optimisation software only changes the input of the simulation program and read the simulations output.

I call the software **TABE**, originated from a Buginese polite word. The codes are uploaded and maintained on GitHub (<https://github.com/mfkasim91/tabe>).

7.2.2 Inputs from users

To run the software appropriately, the users need to write some text files and scripts. Those are:

- `config.json` - the file to specify the other file names, details of the connection to the computing machine if applicable, and simulation parameters;
- `input_template` - template for the input of the simulations that will be filled with the parameters suggested by TABE;
- `submission_file` - shell/batch/cmd script that will be executed to start a simulation;
- `preprocess` (optional) - a python script to transform the parameters suggested by the optimiser into parameters specified in the input template;
- `postprocess` - a python script to process the output of the simulation into a number that the users want to optimise.

In the next sub-subsection, each file is explained with an example of minimising a simple quadratic function, $f(a, b) = a^2 + b^2$.

File `config.json`

An example `config.json` file is shown in Figure 7.7. This is formatted as a JavaScript Object Notation (JSON) file, so the syntax needs to follow JSON. The reason why I chose JSON is because of its simplicity and its ability to express complex data structures.

Here are the explanations of each variable in the file.

- `experiment_name`: Specifies the name of an optimisation experiment and will also be the name of the optimisation process database. When running this software, it looks for the database with the same name. If found, then it reads the database and resume the optimisation process. Otherwise, it creates new database and start the optimisation process from the beginning.

```
1 {
2   "experiment_name": "test",
3   "ssh": {
4     "hostname": "host.url.com",
5     "username": "username",
6     "password": "p4**w.rd",
7     "working_directory": "/your/complete/working/directory/at/the/host/"
8   },
9   "scheduler": "LSF",
10  "populations": 16,
11  "submission": {
12    "input_template": "os-stdin.template",
13    "submission_file": "submission",
14    "preprocess": "preprocess.py",
15    "postprocess": "postprocess.py"
16  },
17  "parameters": {
18    "a": {
19      "type": "float",
20      "size": 1,
21      "min": 0,
22      "max": 1,
23      "initial": 0.2
24    },
25    "b": {
26      "type": "float",
27      "size": 1,
28      "min": 0,
29      "max": 1,
30      "initial": 0.2
31    }
32  }
33 }
```

Figure 7.7: An example of config.json file.

- **ssh** (optional): The parameter to connect to the remote computing server via SSH. This variable is not required if the optimisation software runs on the same machine as the computing machine.
- **scheduler**: The type of job scheduler on the computing machine. Example of the schedulers are LSF and PBS. If run on machine without the scheduler, it can be specified as **bash** for Linux machine.
- **populations**: The number of multiple simulations running in a time. This depends on the availability of the computing machine. Increasing populations can lead to a better solution, with more computational cost.
- **submissions**: The relative path to the required files, including the name of the files. For the `input_template` name, it needs to have an extension of `.template` as this extension will be removed when writing the template for the input file.
- **parameters**: The parameters the optimiser need to optimise.
 - **type** (optional, default: `float`): The type of the parameter. Currently only support `float` type of parameter.
 - **size** (optional, default: `1`): The number of elements in the parameter. All the parameters are formed in a vector, even if the `size` is `1`.
 - **min** and **max**: The lower and upper bound of the variable. These parameters can be written in a number or an array with the same size as `size`. If it is a number, then it is applied for all elements in the parameter. If it is a vector, then the bound is applied element-wise.
 - **initial** (optional, default: `min+max/2`): The initial central value of the optimiser. If not specified, it starts from the central point.

Files `input_template` and `preprocess`

The simulation software that runs on the computing machine needs to read a file as its input. The input template file name needs to resembles `<file input name>.template` where the

```
1 {a_square} {b_square}
```

Figure 7.8: An example of input template file (`os-stdin.template`).

```
2 def main(x):
3     y = {
4         'a_square': x['a'][0]**2,
5         'b_square': x['b'][0]**2
6     }
7     return y
```

Figure 7.9: Preprocess file example (`preprocess.py`).

`.template` extension will be removed to be the input file for the simulation software.

In this case, let us say the simulation software named `add_two_numbers` takes an input of a file `os-stdin` that contains two numbers and write the summation result to the output file, `output.txt`. If one wants to calculate the quadratic sum of two numbers, $f(a, b) = a^2 + b^2$, then the input file needs to contains the values of a^2 and b^2 . The input template file in this case should be as shown in Figure 7.8. The variables in the curly brackets will be replaced by some values given by the optimiser. If an actual curly bracket is needed, it should be written as double curly brackets, e.g. `{{` is rendered `}`.

In the input template file, the parameters are `a_square` and `b_square`. However, in the `config.json` file, the parameters are `a` and `b`. To bridge between the `config.json` and `input_template`, a preprocess file is needed. This preprocess file is a Python script that only contains a `main` function. The main function receives variable `x` that is a dictionary of the variables specified from the `config.json` and should return a dictionary that contains keys listed in `input_template` file. Figure 7.9 is an example of a preprocess file used in this case. The preprocess file needs to be written in Python.

File submission_file

After the optimiser software have written the input file, it will submit the job by executing the submission file. This should be a batch script (for windows) or a bash/shell script. An example of the submission file is shown in Figure 7.10. Line 1-5 is the line to set parameters for the LSF scheduler. Line 7 is to execute the simple simulation software (`add_two_numbers`).

```
1 #BSUB -q scarf
2 #BSUB -n 1
3 #BSUB -W 1:00
4 #BSUB -o %J.log
5 #BSUB -e %J.err
6
7 add_two_numbers
```

Figure 7.10: Submission file example (submission).

```
3 def main():
4     f = open('output.txt', 'r')
5     s = eval(f.read())
6     f.close()
7     print(s)
8
9 if __name__ == '__main__':
10    main()
```

Figure 7.11: A simple example of the postprocess code (postprocess.py).

The submission file, input file, and the output files of the simulation software are contained in a folder named based on the job ID. The submission file is also executed from that directory. Therefore, the simulation program should be accessible anywhere.

File postprocess

The output of a simulation software can be in any form. However, the output desired by this optimisation software is a single number for each simulation. To bridge this, a postprocess file is required.

The postprocess file's job is to read the simulation output, calculate the desired output to be optimised (e.g. electron's maximum energy), and print out the number. This file needs to be written in Python. An extension for MATLAB postprocess script is currently undergoing.

In this example case, the summation of the two numbers are written in an output file. Therefore, the postprocess file in this script just needs to read the content of the output file and print out the result. In some cases, the postprocess script does more complicated tasks. Figure 7.11 shows the example of the postprocess file.

Running the program

After all the files are set, to run the program is simple. Of course, the user needs to download the code and its requirement first. The code and examples can be obtained from <https://github.com/mfkasim91/tabe> [130]. The same page contains the requirements and a tutorial on getting started.

The program can be executed from a command window or terminal. From the terminal, the user needs to navigate (`cd`) to the directory where the `config.json` file is contained. We call this as the experiment directory. Then, user can just simply execute the TABE main file from the experiment directory. It can be done by typing `python /<directory-to-tabe>/main.py` (and changing the `/directory-to-tabe/` with the actual directory).

7.3 Optimising laser plasma wakefield

7.3.1 Longitudinally uniform density

The first case to use TABE is optimising laser plasma wakefield. In this case, I tried optimising the plasma density, plasma length, density waveguide factor, laser pulse duration, waist size, and its focal spot position. There are 6 variables to be tuned to maximised the integral longitudinal electric field over the propagation distance, $\int E_z ds$, with a moving window with the speed of light, c . This is approximately the energy gain for externally injected electron beam.

The simulations are kept as cheap as possible but maintaining reasonable accuracy. Once the optimum parameters to maximise the electron energy gain are obtained, one can run a more accurate and more expensive simulation to get better results.

The bounds for plasma density are $1.2 \times 10^{16} \text{ cm}^{-3}$ to $3.5 \times 10^{19} \text{ cm}^{-3}$. To search for parameters with several orders of magnitude range, it is better to take the log of the

parameter as the input parameter. In this example, the log density bounds are 37 to 45. The plasma length is searched from 1 mm to 20 mm. Quadratic radial density profile is also applied in this case where the density is given by $n(r)/n_0 = 1 + f_r \omega_p^2 r^2 / c^2 \pi^2$, where f_r is the density radial factor to be optimised. The bound for the f_r is from 0 to 1. In the simulation, the ions are fixed.

The laser pulse focal position is varied from 0 to 1, where 0 means it is focused at the beginning of the plasma, 1 is at the end, and other values are interpolated linearly (e.g. 0.5 means the pulse is focused at the middle). The laser pulse duration and waist size are (3 to 13) ω_p and (1 to 20) ω_p/c , respectively. Variable ω_p is the plasma frequency. The laser total energy is kept 15 J and it is linearly polarised. The parameters search space is set according to parameters for GEMINI laser in the Target Area 3 in the Central Laser Facility, STFC Rutherford Appleton Laboratory (RAL).

A particle-in-cell (PIC) code called OSIRIS was used as the simulation program. The simulations were executed on SCARF, a supercomputer machine from the RAL, STFC. One simulation is set to run for maximum 24 hours in 64 cores and 16 simulations are submitted simultaneously every batch.

Optimisation results

The optimisation runs for 20.5 days with 304 simulations performed. The maximum electron energy gain, $\int E_z ds$, found is 2.47 GeV. Figure 7.12 shows the negative of electron energy gain against the simulation ID and time. The total cores utilisation is 3280 core-days.

The optimum parameters found by TABE to optimise the electron energy gain are:

- plasma density: $n_0 = 5.24 \times 10^{17} \text{ cm}^{-3}$ (plasma wavelength: $\lambda_p = 46.2 \text{ } \mu\text{m}$);
- plasma length: 19.2 mm;
- plasma density radial factor ($\Delta n/r^2$): $9.53 \times 10^{14} \text{ cm}^{-3}/\mu\text{m}^2$;
- laser pulse duration: 109.3 fs (full-width half-max: 54.6 fs);

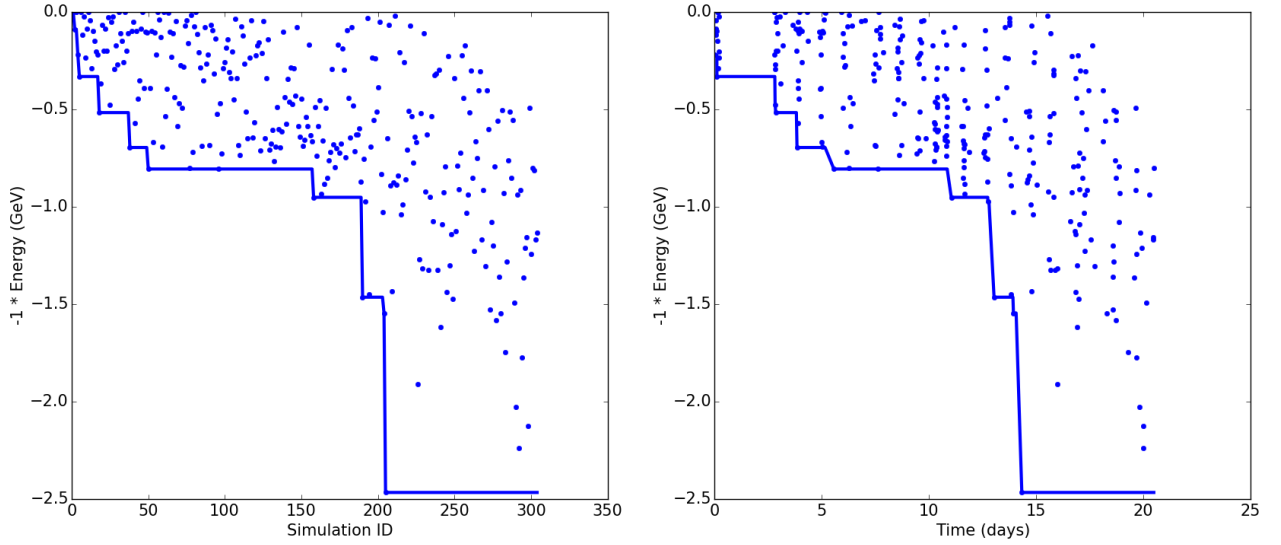


Figure 7.12: The electron energy gain plotted against the simulation ID and time for uniform longitudinal density. The dots are individual results for every simulations and the lines are the minimum results found so far.

- laser pulse waist size: $w_0 = 17.0 \mu\text{m}$ (Rayleigh length: 1.13 mm, $a_0 = 3.76$);
- laser pulse focal position: 0.21 mm from the beginning.

In order to look into the simulation more detail, I plot the average electric field profile felt by every position in the moving window with speed c and compare it with the electric field profile at the beginning and at the end. It is shown in Figure 7.13. The left picture shows the wakefield is shifted backward relative to the moving window. The picture on the right also shows that the electric field at the optimum position (i.e. the position on the moving window where the energy gain is maximum) is decreasing as it propagates while the maximum electric field is increasing. This is due to dephasing effect. However, at the end of simulation, the electric field at the optimum position is still around 100 GeV/m. If the simulation is run longer, then the energy potential gain can be more than what was obtained in this simulation.

During the simulation, plasma bubble was formed. The peak electric field with a bubble is given by [50] $E_z \approx (mc\omega_p)/e\sqrt{a_0} \approx 135 \text{ GeV/m}$ using the optimum parameters found by the software. The picture on the right of Figure 7.13 shows the electric field is about 100-150 GeV/m, which agrees with the prediction.

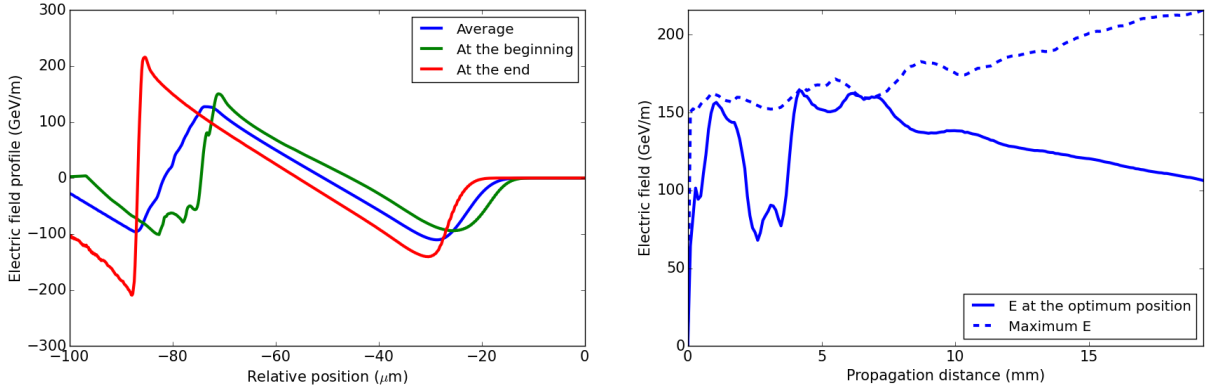


Figure 7.13: (Left) The electric field profile in the moving window with the speed of light, c . (Right) The electric field at the position where the average electric field is maximum plotted against the propagation distance. The dashed line shows the maximum electric field at a given time and propagation distance.

Based on W. Lu *et al.* [50], the optimum condition for the spot size of linearly polarised laser pulse, such that it minimises the oscillation of the spot size, is given by $k_p w_0 \approx 1.12\sqrt{a_0}$. With $k_p = 2\pi/\lambda_p = 1.36 \times 10^5 \text{m}^{-1}$, $w_0 = 17 \mu\text{m}$, and $a_0 = 3.76$, it is found that $k_p w_0 / (1.12\sqrt{a_0}) = 1.06 \approx 1$ which means the w_0 selected by the algorithm is optimum. Another optimality condition in [50] is that the pulse length is approximately the same length as the radius of the bubble, $c\tau_{FWHM} \lesssim R \approx w_0$. The FWHM duration of the pulse is 54.6 fs, thus $c\tau_{FWHM} = 16 \mu\text{m} \approx w_0$. This shows that the optimisation software can learn about the plasma wakefield system and choose the optimum parameters without explicitly being taught.

7.3.2 Varied plasma density

After optimising plasma wakefield with 6 parameters, I tried adding more parameters to the optimisation problem. The new parameters are the relative plasma density at 8 longitudinal positions with respect to the base plasma density. The 8 positions are spaced regularly with $x = 1/9, 2/9, \dots, 8/9$. The density at $x = 0$ and $x = 1$ are kept 0 and all other density is interpolated linearly. The relative plasma density for the 8 positions are bounded between 0 and 2. If the relative plasma density is 2, then the density at that longitudinal position is

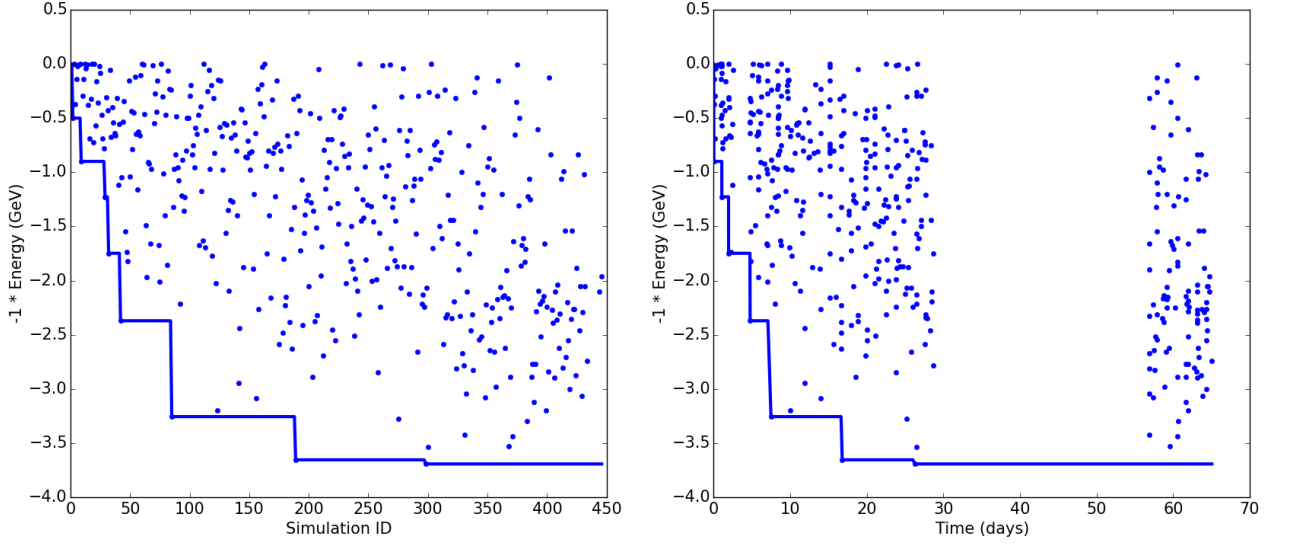


Figure 7.14: The electron energy gain plotted against the simulation ID and time with varied longitudinal density. The dots are individual results for every simulations and the lines are the minimum results found so far. The large gap on the right plot is due to inactivity of TABE.

twice of the base density, which is also a parameter. The other parameters and their bounds from the previous subsection are kept the same. The goal also remains the same as the previous subsection, i.e. $\int E_z ds$, the electron energy gain. In this case, the optimisation problem now has 14 parameters to be optimised.

Optimisation results

It took 65 days to finish this optimisation process, with 28 days non-activity. There are 435 simulations performed. The optimisation progress is shown in Figure 7.14.

In this process, TABE found a set of parameters that can get $\int E_z ds = 3.7$ GeV electron energy gain. The set of parameters are listed below:

- plasma base density: $n_0 = 1.57 \times 10^{18} \text{ cm}^{-3}$ (plasma wavelength: $\lambda_p = 26.7 \text{ }\mu\text{m}$);
- plasma length: 14.7 mm;
- plasma density radial factor ($\Delta n/r^2$): $4.93 \times 10^{15} \text{ cm}^{-3}/\mu\text{m}^2$;
- laser pulse duration: 67.9 fs (full-width half-max: 33.9 fs);

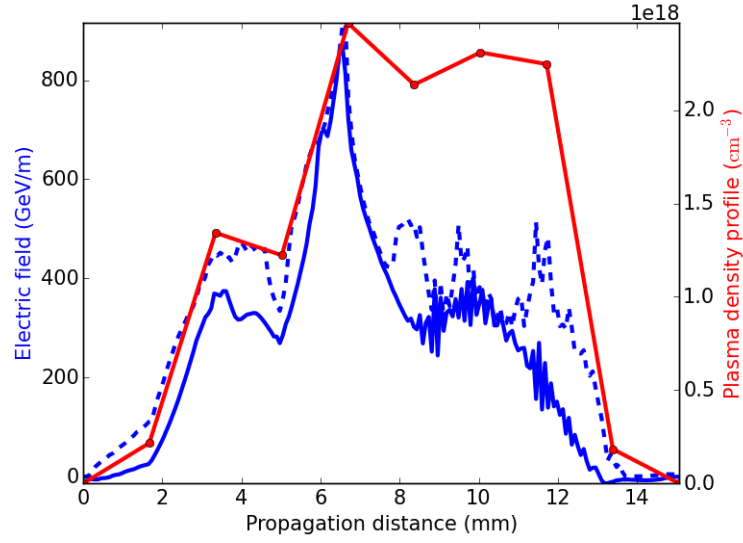


Figure 7.15: (Blue, left axis) The electric field of the optimum position that gains the most electric potential as a function of the propagation distance. The maximum electric field at a given time and propagation distance is shown in the dashed blue lines. Note that this maximum electric field's relative position is changing therefore it is hard for a beam to get the maximum electric field all the time. (Red, right axis) The plasma density profile as a function of the propagation distance in the plasma.

- laser pulse waist size: $w_0 = 17.1 \mu\text{m}$ (Rayleigh length: 1.14 mm, $a_0 = 4.74$);
- laser pulse focal position: 1.3 mm from the beginning;
- plasma density relative profile: [0.138, 0.855, 0.778, 1.57, 1.36, 1.47, 1.43, 0.117].

The Figure 7.15 shows the electric field at the optimum place with the optimum parameters given above as a function of the propagation distance as well as the plasma density profile. As shown in the Figure, the electric field is peaked around 7 mm where the plasma density is also increased to more than $2.0 \times 10^{18} \text{ cm}^{-3}$ and then decreases right away.

One interesting thing to note here is the electric field can exceed 800 GeV/m when the density is about $2.5 \times 10^{18} \text{ cm}^{-3}$. From W. Lu *et al.* work [50], the peak electric field is $E_z = mc\omega_p/e\sqrt{a_0}$ or around $E_z \approx 331 \text{ GeV/m}$ in this case with $n = 2.5 \times 10^{18} \text{ cm}^{-3}$ and $a_0 = 4.74$. The electric field in the simulation is $E_{sim} = 870 \text{ GeV/m}$, which is around 2.6 times of the calculated peak electric field. Comparing this scaling with the previous case (i.e. the uniform plasma density case where $E_z = 135 \text{ GeV/m}$ and $E_{sim} = 150 \text{ GeV/m}$),

the electric field in the simulation is much higher than the calculated electric field. This is probably due to the pulse focusing and shortening of the laser pulse when it is around 3-5 mm from the beginning and thus increasing the pulse intensity, a_0 . The focusing is due to self-focusing effect and the plasma channel that acts as the external guiding. The shortening is because when it drives the wakefield, the front part of the pulse travels slower than the back due to difference in the plasma wakefield density with wakefield.

7.4 Conclusions

In this chapter, I have presented a novel algorithm to do shape optimisation based on Simultaneous Optimistic Optimisation (SOO) [124]. The shape optimisation algorithm has been tested with several cases in physics, such as brachistochrone problem. The algorithm can find the optimal solutions with reasonable number of simulations. It also surpasses the benchmark algorithms based on SOO.

Besides the algorithm, the software I developed to bridge physics simulations and optimisation algorithms has also been presented. The software is robust so that the user can just set the optimisation process once and leave it until the process finish or it finds reasonable results. If the process is interrupted, it is also possible to resume the previous state of the optimisation process. The software suite has been named TABE.

TABE has already been tested to optimise laser-driven plasma wakefield parameters using simulations with different conditions. When the longitudinal plasma density was set uniform, TABE can find the optimal laser pulse duration and laser pulse waist size in agreement with analytic models. When TABE was used to investigate the effect of modifying the longitudinal plasma density profile, it found a profile that increased the electron bunch energy compared to the uniform case by around 50%. This prediction will be verified in future experiments on the GEMINI facility.

Summary and future works

This thesis contains some works on inferring quantitative information on plasma wakefield diagnostics using optical probes with spectral interferometry and shadowgraphy. A method to obtain quantitative information of magnetic and/or electric field in a plasma with proton radiography is also presented in this thesis. Aside from the diagnostic parts, software framework that incorporates simulation programs to optimise plasma systems is described in the last chapter.

In Chapter 2, the idea of using frequency modulation of a co-propagating probe pulse to measure the wakefield density profile was tested using numerous one dimensional Particle-in-Cell (PIC) simulations. In each simulation, an intense short driver pulse with period \sim fs was fired into a uniform plasma to drive a wakefield. Then, a long probe pulse with several hundreds of femtoseconds delay and much smaller intensity was fired and covered several wakefield periods. After some propagation distance, the frequency modulation of the probe pulse was then extracted and matched with the obtained integrated wakefield density profile. The simulations were repeated for different peak intensities of the driver pulse to drive wakefields with different amplitudes. The results showed that the integrated wakefield density profiles calculated from the probe's frequency modulations agreed well with the integrated wakefield density profiles obtained directly from the simulations. Some limitations that could spoil the measurement results were also discussed.

If the measurement using a co-propagating probe was discussed in Chapter 2, the next chapter (Chapter 3) presented a study of a wakefield measurement using a probe crossing the wakefield in an oblique angle. The wakefield density modulation was obtained from the phase

modulation of the probe pulse measured by a spectral interferometry. In the chapter, analytic equations relating the phase modulation of the probe pulse to the wakefield density were derived. The equations were generalisation of Abel transforms to cover non-perpendicular probe cases. Measurement simulations using PIC were also performed. The results from the simulations showed that the measured wakefield density modulations were accurate with error less than 15%. Limitations on the crossing angle to obtain a small measurement error were presented in the chapter.

The next chapter, Chapter 4, presented details of an experiment on plasma wakefield diagnostic. The main objective of the experiment was to test the wakefield diagnostic method with the crossing probe pulse described in the previous chapter. The experiment was performed for 12 weeks in the Gemini laser's facility's Target Area 2 in the Central Laser Facility, STFC Rutherford Appleton Laboratory. As the wakefield driver pulse, an 800 nm, 45 fs, and 0.5 J Ti:S laser pulse focused with $f/17$ off-axis parabolic mirror was used. The probe pulse to diagnose the wakefield was a 400 nm, 600 fs, 0.5 mJ laser pulse crossing the wakefield with 8° angle and delayed by 0-200 fs behind the driver pulse. As the target, helium, nitrogen, and methane gasses were used. The scanned plasma electron density was $(5-10) \times 10^{18} \text{ cm}^{-3}$. With the helium and nitrogen gas, there was no modulation seen. With methane gas, there were some modulations seen with similar wavelengths with the expected wakefield wavelengths. However, the modulation wavelengths were generally longer than the expected wakefield wavelengths. The discrepancy was explained using some theories developed for cluster plasma and possibility of elongated wakefield due to the beam loading effect. The exact cause of the discrepancy is a subject of a future study.

In Chapter 5, a novel idea of three dimensional spectrometer was presented. The design of the 3D spectrometer is similar to the 2D Czerny-Turner spectrometer, except that it has a random aperture and the slit is supposed to be open wider in the new design. In the 3D spectrometer, the signal is captured by a screen / CCD and the 3D signal is retrieved using a compressed sensing method. Numerical simulations that emulated the signal's transformations in the 3D spectrometer were done to test the performance of the design. As the results, the 3D spectrometer can retrieve up to 10 sets of two dimensional spectra images

just from a single image captured in the spectrometer. Moreover, the retrieved signals were also accurate enough such that a further post-processing can be applied to them. A patent for the 3D spectrometer design has been filed. Future works for the 3D spectrometer is to build a working prototype and test it in a real laser-plasma experiment.

In contrast to the previous chapters, in Chapter 6, other methods to diagnose plasma were discussed. They were proton radiography and shadowgraphy. By the time I started the project, there were no well-known algorithm to retrieve quantitative information from proton radiography and shadowgram images in the non-linear regime. The first attempt to solve this problem was using a gradient descent algorithm. However, the employed gradient descent algorithm was not good enough to reach the convergence. The second attempt was using power diagrams and computational graphics techniques to get the quantitative information. This method worked very well and also robust to noise. It can retrieve quantitative information from proton radiography and shadowgram images in the non-linear regime, except in the regime where it is mathematically impossible to solve.

The last chapter in this thesis deals with optimisation algorithms and the applications on the plasma wakefield. New software framework, which is called TABE, was described in this chapter and how it incorporates any simulation softwares to optimise systems (including plasma systems). TABE was tested by running hundreds of simulations in a remote cluster computer to optimise the potential energy gain from laser-driven plasma accelerators. After running it for 20 days, it found the optimum parameters that agreed well with a well-accepted reference in the field. It also found a new way to enhance the electron energy by changing the longitudinal plasma density.

In the software side, TABE has a lot of potentials for improvements, such as incorporating more optimisation algorithms. Adding Markov-Chain Monte-Carlo (MCMC) and multi-objective algorithms is a part of future works, so that it can find several parameters to optimise several values. From the physics side, trying TABE to optimise laser driven plasma accelerators with two different plasma densities is worth doing. Exploring optimum conditions for other plasma systems, such as hole boring, target normal sheath acceleration, and fitting spectra, could also give new results in the field.

Appendix A

Derivations

A.1 Oblique angle Abel transformation and inverse

A.1.1 Forward transform

In this part of appendix, I will show the derivation of the modified Abel transformation when the probe is crossing in an oblique angle, not in perpendicular direction in relative to the object. The case is shown in Figure A.1 where the probe crosses the object with an angle θ .

Consider the object's parameter to be diagnosed is $f(r, z)$ and thus the probe's parameter changes as it diagnosed through the object as

$$F(y, z_0) = \int_{-\infty}^{\infty} f(r, z) ds, \quad (\text{A.1})$$

where $\int \cdot ds$ is the integral of the probe along its propagation time. Without losing generality, we assume that at position $s = 0$, the probe crosses the axis of the object. Therefore, at position s , the part of the probe at z_0 at $z = z_0 + s \cos \theta$. Now we can apply Taylor expansion

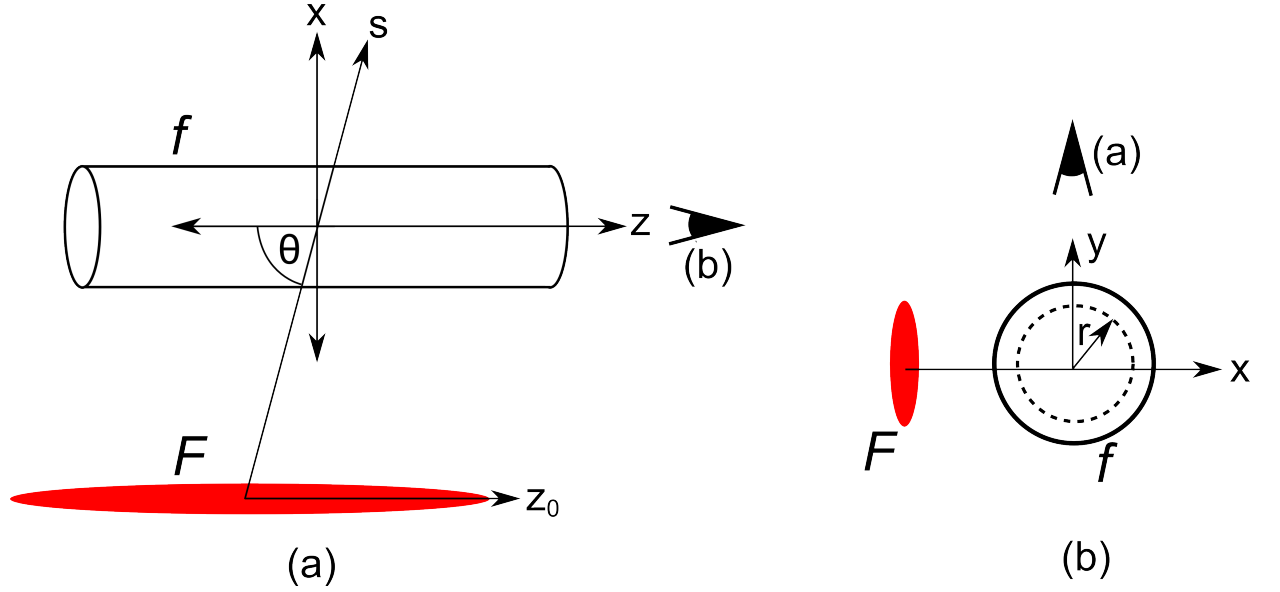


Figure A.1: The cases considered in this section by (a) looking along the y axis and (b) looking along the z axis. The probe is denoted by F and the object by f .

to $f(r, z)$ in the equation above,

$$F(y, z_0) = \int_{-\infty}^{\infty} \sum_{n=0}^{\infty} \frac{\partial^n f}{\partial z^n}(r, z_0) \frac{1}{n!} (s \cos \theta)^n ds. \quad (\text{A.2})$$

As $x = s \sin \theta$, we can substitute the integral above to the integral with dx ,

$$F(y, z_0) = \sum_{n=0}^{\infty} \frac{\cot^n \theta}{n! \sin \theta} \int_{-\infty}^{\infty} \frac{\partial^n f}{\partial z^n}(r, z_0) x^n dx. \quad (\text{A.3})$$

From the Figure A.1(b), $x = \pm (r^2 - y^2)^{1/2}$, and $dx = \pm r dr / \sqrt{r^2 - y^2}$. Thus, to change the integration above into dr , we need to split the argument from $x = (-\infty, 0]$ and $x = [0, \infty)$. Fortunately, for odd n , the function inside the integral becomes an odd function (as $\partial^n f / \partial z^n$ is an even function) and for even n , the function becomes an even function. Therefore, we can eliminate the odd terms of n and we can get the integral values of even n by integrating

$x = [0, \infty)$ and multiply it by 2. From here, we get

$$F(y, z_0) = 2 \sum_{j=0}^{\infty} \frac{\cot^{2j} \theta}{(2j)! \sin \theta} \int_y^{\infty} \frac{\partial^{2j} f}{\partial z^{2j}}(r, z_0) (r^2 - y^2)^{j-1/2} r \, dr. \quad (\text{A.4})$$

This is the trickiest part. We can apply Fourier Transform in the z_0 direction to get,

$$\begin{aligned} \tilde{F}(y, k_0) &= 2 \sum_{j=0}^{\infty} \frac{\cot^{2j} \theta}{(2j)! \sin \theta} \int_y^{\infty} (ik)^{2j} \tilde{f}(r, k_0) (r^2 - y^2)^{j-1/2} r \, dr \\ &= \frac{2}{\sin \theta} \int_y^{\infty} \left[\sum_{j=0}^{\infty} \frac{(-1)^j}{(2j)!} \left(k \cot \theta \sqrt{r^2 - y^2} \right)^{2j} \right] \tilde{f}(r, k_0) \frac{r \, dr}{\sqrt{r^2 - y^2}}, \end{aligned} \quad (\text{A.5})$$

where the variables with tilde hat are the Fourier-transformed variables and k_0 is the frequency in z_0 . We know that $\cos x = \sum_{j=0}^{\infty} (-1)^j x^{2j} / (2j)!$, so we can replace the term in the square brackets with cosine term,

$$\tilde{F}(y, k_0) = \frac{2}{\sin \theta} \int_y^{\infty} \cos \left(k \cot \theta \sqrt{r^2 - y^2} \right) \tilde{f}(r, k_0) \frac{r \, dr}{\sqrt{r^2 - y^2}}. \quad (\text{A.6})$$

Now it is relatively straight forward to simplify. By making a substitution for variable $\sqrt{r^2 - y^2}$ and $\cos \left(k \cot \theta \sqrt{r^2 - y^2} \right)$, the equation above can be simplified to

$$\tilde{F}(y, k_0) = -\frac{2}{\sin \theta} \int_y^{\infty} \frac{\sin \left(k \cot \theta \sqrt{r^2 - y^2} \right)}{k \cot \theta} \frac{\partial \tilde{f}}{\partial r}(r, k_0) \, dr. \quad (\text{A.7})$$

This is the modified Abel transformation if the probe is not coming in a perpendicular direction. Setting $\theta = \pi/2$ gives us the original Abel transformation.

A.1.2 Inverse transform

The most interesting part in a transformation is the inverse transform. In the real application, people can easily measure the probe's parameters, F , and interested in f . The transformation above only gives us F if f is known. Therefore, the inverse transformation of equation (A.7) is needed. If the forward transform is already tricky, the inverse transform is much trickier.

First let us start by applying the Fourier transformation of equation (A.7) in the y direction and denoting k_0 as k_z ,

$$\int_{y=-\infty}^{\infty} \tilde{F}(y, k_z) e^{-ik_y y} dy = b \int_{y=-\infty}^{\infty} \int_{r=y}^{\infty} \cos(k_z a \sqrt{r^2 - y^2}) \tilde{f}(r, k_z) \frac{r}{\sqrt{r^2 - y^2}} e^{-ik_y y} dr dy, \quad (\text{A.8})$$

where $a \equiv \cot \theta$ and $b \equiv 2/\sin \theta$ for simplification. Reverse the integral order,

$$\hat{F}(k_y, k_z) = 2b \int_{r=0}^{\infty} \tilde{f}(r, k_z) r \int_{y=0}^r \cos(k_y y) \cos(k_z a \sqrt{r^2 - y^2}) \frac{dy}{\sqrt{r^2 - y^2}} dr. \quad (\text{A.9})$$

We substituted $\int_{-\infty}^{\infty} \tilde{F} e^{-ik_y y} dy$ into $2 \int_0^{\infty} \tilde{F} \cos(k_y y) dy$ because \tilde{F} is an even function on y .

Substituting $y = r \sin \phi$, we obtain

$$\hat{F}(k_y, k_z) = 2b \int_{r=0}^{\infty} \tilde{f}(r, k_z) r \int_{\phi=0}^{\pi/2} \cos(k_y r \sin \theta) \cos(k_z a r \cos \phi) d\phi dr. \quad (\text{A.10})$$

We can prove that

$$\int_{\phi=0}^{\pi/2} \cos(a \sin \phi) \cos(b \cos \phi) d\phi = \frac{\pi}{2} J_0 \left(\sqrt{a^2 + b^2} \right), \quad (\text{A.11})$$

where $J_0(x)$ is the Bessel function of the first kind. Therefore,

$$\begin{aligned}\hat{F}(k_y, k_z) &= 2b \int_{r=0}^{\infty} \tilde{f}(r, k_z) r \frac{\pi}{2} J_0\left(\sqrt{k_y^2 + a^2 k_z^2} r\right) dr \\ &= \frac{1}{\sin \theta} 2\pi \int_{r=0}^{\infty} \tilde{f}(r, k_z) J_0(k_r r) r dr,\end{aligned}\quad (\text{A.12})$$

where $k_r^2 = k_y^2 + a^2 k_z^2$. The equation above shows that applying the Fourier transformation to $\tilde{F}(y, k_z)$ in y direction equals to applying the Hankel transformation to $\tilde{f}(r, k_z)$ in r direction.

Inverting the Hankel transformation is similar to its transformation, i.e.

$$\begin{aligned}\tilde{f}(r, k_z) &= \sin \theta \frac{1}{2\pi} \int_{k_r=0}^{\infty} \hat{F}(k_y, k_z) J_0(k_r r) k_r dk_r \\ &= \sin \theta \frac{1}{2\pi} \int_{k_r=0}^{\infty} \int_{y=-\infty}^{\infty} \tilde{F}(y, k_z) e^{-ik_y y} dy J_0(k_r r) k_r dk_r.\end{aligned}\quad (\text{A.13})$$

$\tilde{F}(y)$ is symmetric in $y = 0$, so it can be written as

$$\tilde{f}(r, k_z) = \sin \theta \frac{1}{\pi} \int_{k_r=0}^{\infty} \int_{y=0}^{\infty} \tilde{F}(y, k_z) e^{-ik_y y} dy J_0(k_r r) k_r dk_r.\quad (\text{A.14})$$

Reversing the order of the integral and writing $\tilde{F}(y, k_z) = -\int_y^{\infty} \tilde{F}'(y', k_z) dy'$, where $\tilde{F}'(y', k_z) = \partial \tilde{F} / \partial y(y', k_z)$, lead us to

$$\tilde{f}(r, k_z) = -\frac{\sin \theta}{\pi} \int_{y=0}^{\infty} \int_{y'=y}^{\infty} \int_{k_r=0}^{\infty} \tilde{F}'(y', k_z) \cos(k_y y) J_0(k_r r) k_r dy' dk_r dy.\quad (\text{A.15})$$

Changing the order of integration to $\int_{y'} \int_{k_r} \int_y$ gives us

$$\begin{aligned} \tilde{f}(r, k_z) &= -\frac{\sin \theta}{\pi} \int_{y'=0}^{\infty} \tilde{F}'(y', k_z) \int_{k_r=0}^{\infty} J_0(k_r r) k_r \int_{y=0}^{y'} \cos(k_y y) dy dk_r dy' \\ &= -\frac{\sin \theta}{\pi} \int_{y'=0}^{\infty} \tilde{F}'(y', k_z) \int_{k_r=0}^{\infty} J_0(k_r r) k_r \frac{\sin\left(\sqrt{k_r^2 - a^2 k_z^2} y\right)}{\sqrt{k_r^2 - a^2 k_z^2}} dk_r dy'. \end{aligned} \quad (\text{A.16})$$

On the second line, we substitute $k_y = \sqrt{k_r^2 - a^2 k_z^2}$.

From Gradshteyn-Ryzhik's book (edition 7, sec. 6.677 equation 1), we know that

$$I = \int_a^{\infty} J_0\left(b\sqrt{x^2 - a^2}\right) \sin(cx) dx = \begin{cases} 0 & : 0 < c < b \\ \frac{\cosh(a\sqrt{c^2 - b^2})}{\sqrt{c^2 - b^2}} & : 0 < b < c \end{cases} \quad (\text{A.17})$$

$$I = \int_0^{\infty} J_0(bz) z \frac{\sin\left(\sqrt{z^2 + a^2} c\right)}{\sqrt{z^2 + a^2}} dz.$$

We can substitute $a \rightarrow iak_z$, $b \rightarrow k_r$, $c \rightarrow y$, and $z \rightarrow k_r$ from the equation above to solve the integral in equation (A.16). Therefore, we obtain

$$\tilde{f}(r, k_z) = -\frac{\sin \theta}{\pi} \int_r^{\infty} \frac{\partial \tilde{F}}{\partial y}(y, k_z) \frac{\cosh\left(k_z \cot \theta \sqrt{y^2 - r^2}\right)}{\sqrt{y^2 - r^2}} dy. \quad (\text{A.18})$$

A.2 Oblique angle tomography

In the previous section, the transformations for oblique crossing angle probe for cylindrically symmetric object are presented. In this section, we will consider if the object is not cylindrically symmetric. If this is the case, then more than one probes are required to capture the full profile as in tomography [73]. The derivations here follow closely the derivation in

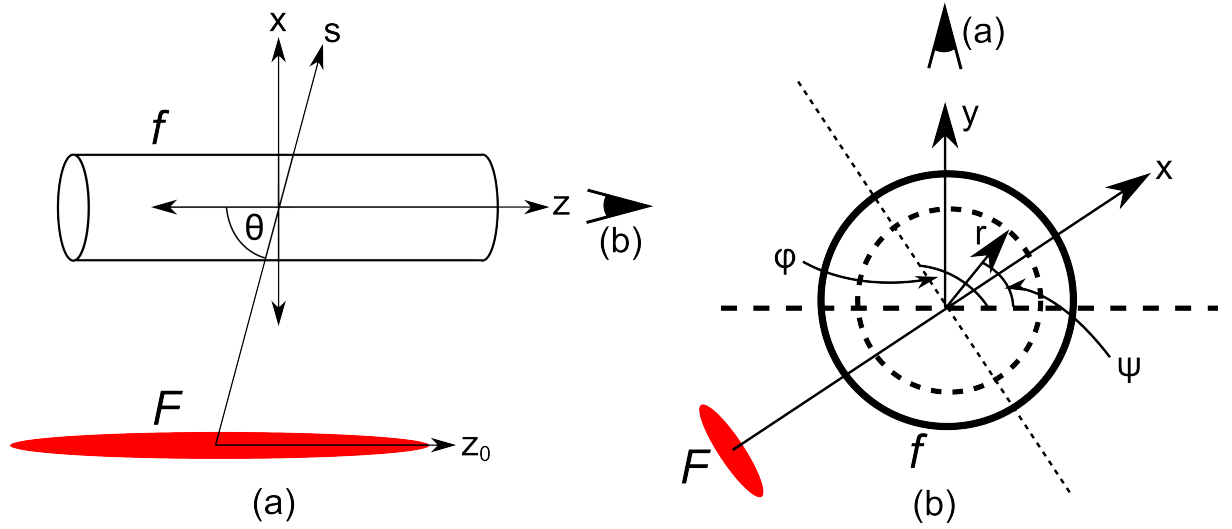


Figure A.2: The cases considered in this section by (a) looking along the y axis and (b) looking along the z axis. The probe is denoted by F and the object by f .

Cormack [73] with some modifications.

A.2.1 Forward transformation

Let us denote the parameter of the object is $f(r, z, \psi)$ and the probe's imprinted parameter is $F(y, z, \phi)$. These are similar to the previous section, except with additional angular terms, ψ and ϕ to capture the full 3D profiles. Except these two terms, all other terms are re-used from the previous section. Figure A.2 show the illustration of this case. From equation (A.3), we obtain

$$F(y, z_0, \phi) = \sum_{n=0}^{\infty} \frac{a^n}{n! \sin \theta} \int_{-\infty}^{\infty} \frac{\partial^n f}{\partial z^n}(r, z_0, \theta) x^n dx, \quad (\text{A.19})$$

with $a \equiv \cot \theta$. We can relate ψ and ϕ as

$$\psi = \begin{cases} \phi - \arccos(y/r) & : x > 0 \\ \phi + \arccos(y/r) & : x \leq 0. \end{cases} \quad (\text{A.20})$$

Using multipole series, we can substitute

$$F(y, z_0, \phi) = \sum_{m=-\infty}^{\infty} F_m(y, z_0) \exp(im\phi) \quad (\text{A.21})$$

$$f(r, z, \psi) = \sum_{m=-\infty}^{\infty} f_m(r, z) \exp(im\psi) \quad (\text{A.22})$$

into equation (A.19) to produce

$$\sum_{m=-\infty}^{\infty} F_m(y, z_0) \exp(im\phi) = \sum_{n=0}^{\infty} \frac{a^n}{n! \sin \theta} \int_{-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \frac{\partial^n f_m}{\partial z^n}(r, z_0) \exp(im\psi) x^n dx. \quad (\text{A.23})$$

Now substitute $x = \pm\sqrt{r^2 - y^2}$ and (A.20) to the equation above, and then split the integral for $x > 0$ and $x \leq 0$,

$$\begin{aligned} \sum_{m=-\infty}^{\infty} F_m(y, z_0) \exp(im\phi) &= \sum_{m=-\infty}^{\infty} \exp(im\phi) \sum_{n=0}^{\infty} \frac{a^n}{n! \sin \theta} \int_{|y|}^{\infty} \frac{\partial^n f_m}{\partial z^n}(r, z_0) (r^2 - y^2)^{(n-1)/2} \\ &\quad [(-1)^n e^{im \arccos(y/r)} + e^{-im \arccos(y/r)}] r dr. \end{aligned} \quad (\text{A.24})$$

From the equation above, we can get the equation for F_m ,

$$F_m(y, z_0) = \sum_{n=0}^{\infty} \frac{a^n}{n! \sin \theta} \int_{|y|}^{\infty} \frac{\partial^n f_m}{\partial z^n}(r, z_0) (r^2 - y^2)^{(n-1)/2} [(-1)^n e^{im \arccos(y/r)} + e^{-im \arccos(y/r)}] r dr. \quad (\text{A.25})$$

Splitting between the even and odd terms of n produces,

$$\begin{aligned}
 F_m(y, z_0) &= 2 \sum_{j=0}^{\infty} \frac{a^{2j}}{(2j)! \sin \theta} \int_{|y|}^{\infty} \frac{\partial^{2j} f_m}{\partial z^{2j}}(r, z_0) \cos [m \arccos(y/r)] (r^2 - y^2)^{(2j-1)/2} r \, dr \\
 &\quad - 2i \sum_{j=0}^{\infty} \frac{a^{2j+1}}{(2j+1)! \sin \theta} \int_{|y|}^{\infty} \frac{\partial^{2j+1} f_m}{\partial z^{2j+1}}(r, z_0) \sin [m \arccos(y/r)] (r^2 - y^2)^j r \, dr.
 \end{aligned} \tag{A.26}$$

As before, we need to apply the Fourier transform on z_0 direction,

$$\begin{aligned}
 \tilde{F}_m(y, k_z) &= 2 \sum_{j=0}^{\infty} \frac{a^{2j}}{(2j)! \sin \theta} \int_{|y|}^{\infty} \left(i k_z \sqrt{r^2 - y^2} \right)^{2j} \tilde{f}_m(r, k_z) \cos [m \arccos(y/r)] \frac{r \, dr}{\sqrt{r^2 - y^2}} \\
 &\quad - 2i \sum_{j=0}^{\infty} \frac{a^{2j+1}}{(2j+1)! \sin \theta} \int_{|y|}^{\infty} \left(i k_z \sqrt{r^2 - y^2} \right)^{2j+1} \tilde{f}_m(r, k_z) \sin [m \arccos(y/r)] \frac{r \, dr}{\sqrt{r^2 - y^2}}.
 \end{aligned} \tag{A.27}$$

We know that $\cos x = \sum_{j=0}^{\infty} (ix)^{2j}/(2j)!$ and $\sin x = -i \sum_{j=0}^{\infty} (ix)^{2j+1}/(2j+1)!$, so we can simplify the equation above into

$$\begin{aligned}
 \tilde{F}_m(y, k_z) &= \frac{2}{\sin \theta} \int_{|y|}^{\infty} \cos \left(k_z a \sqrt{r^2 - y^2} \right) \tilde{f}_m(r, k_z) \cos [m \arccos(y/r)] \frac{r \, dr}{\sqrt{r^2 - y^2}} \\
 &\quad + \frac{2}{\sin \theta} \int_{|y|}^{\infty} \sin \left(k_z a \sqrt{r^2 - y^2} \right) \tilde{f}_m(r, k_z) \sin [m \arccos(y/r)] \frac{r \, dr}{\sqrt{r^2 - y^2}}.
 \end{aligned} \tag{A.28}$$

Using trigonometry identity $\cos(a - b) = \cos a \cos b - \sin a \sin b$, it can be further simplify into,

$$\tilde{F}_m(y, k_z) = \frac{2}{\sin \theta} \int_{|y|}^{\infty} \cos \left[k_z a \sqrt{r^2 - y^2} - m \arccos(y/r) \right] \tilde{f}_m(r, k_z) \frac{r \, dr}{\sqrt{r^2 - y^2}}. \tag{A.29}$$

A.2.2 Inverse transformation

The inverse transformation of equation (A.29) is

$$\tilde{f}_m(r, k_z) = -\frac{\sin \theta}{\pi} \int_{y=r}^{\infty} \frac{\partial \tilde{F}_m}{\partial y}(y, k_z) \frac{\cosh \left[k_z a \sqrt{y^2 - r^2} - m \operatorname{arccosh}(y/r) \right]}{\sqrt{y^2 - r^2}} dy. \quad (\text{A.30})$$

To proof this, substitute \tilde{f}_m from equation above to the equation (A.29) to obtain,

$$\begin{aligned} \tilde{F}_m(y, k_z) &= -\frac{2}{\pi} \int_{s=|y|}^{\infty} \frac{\partial \tilde{F}_m}{\partial s}(s, k_z) \cdot \\ &\int_{r=|y|}^s \frac{\cos \left\{ k_z a \sqrt{r^2 - y^2} - m \arccos(y/r) \operatorname{arccosh} \left[k_z a \sqrt{s^2 - r^2} - m \operatorname{arccosh}(s/r) \right] \right\}}{\sqrt{r^2 - y^2} \sqrt{s^2 - r^2}} r dr ds. \end{aligned} \quad (\text{A.31})$$

Using the software Mathematica, I found that the r -integral for even and odd m and $y > 0$ is $\pi/2$. Therefore,

$$\tilde{F}_m(y, k_z) = -\frac{2}{\pi} \int_{s=y}^{\infty} \frac{\partial \tilde{F}_m}{\partial s}(s, k_z) \frac{\pi}{2} ds = \tilde{F}_m(y, k_z), \quad (\text{A.32})$$

which completes the proof.

Bibliography

- [1] T. Tajima and J. M. Dawson, Phys. Rev. Lett. **43**, 267 (1979).
- [2] S. P. D. Mangles, C. D. Murphy, Z. Najmudin, A. G. R. Thomas, J. L. Collier, A. E. Dangor, E. J. Divall, P. S. Foster, J. G. Gallacher, C. J. Hooker, *et al.*, Nature **431**, 535 (2004).
- [3] C. G. R. Geddes, Cs. Toth, J. van Tilborg, E. Esarey, C. B. Schroeder, D. Bruhwiler, C. Nieter, J. Cary, and W. P. Leemans, Nature **431**, 538 (2004).
- [4] J. Faure, Y. Glinec, A. Pukhov, S. Kiselev, S. Gordienko, E. Lefebvre, J.-P. Rousseau, F. Burgy, and V. Malka, Nature **431**, 541 (2004).
- [5] W. P. Leemans, B. Nagler, A. J. Gonsalves, Cs. Tóth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, and S. M. Hooker, Nature Phys. **2**, 696 (2006).
- [6] V. Malka, J. Faure, Y. A. Gauduel, E. Lefebvre, A. Rousse, and K. T. Phuoc, Nat. Phys. **4**, 447 - 453 (2008).
- [7] I. Blumenfeld, C. E. Clayton, F.-J. Decker, M. J. Hogan, C. Huang, R. Ischebeck, R. Iverson, C. Joshi, T. Katsouleas, N. Kirby, *et al.*, Nature **445**, 741-744 (2007).
- [8] M. Litos, E. Adli, W. An, C. I. Clarke, C. E. Clayton, S. Corde, J. P. Delahaye, R. J. England, A. S. Fisher, J. Frederico, *et al.*, Nature **515**, 92-95 (2014).
- [9] A. Caldwell, K. Lotov, A. Pukhov, and F. Simon, Nature Phys. **5**, 363 (2009).

- [10] R. Assmann, *et al.*, AWAKE Collaboration, CERN-SPSC-2013-013, <http://cds.cern.ch/record/1537318>, and arXiv:1401.4823 (2013).
- [11] K. V. Lotov, A. P. Sosedkin, A. V. Petrenko, L. D. Amorim, J. Vieira, R. A. Fonseca, L. O. Silva, E. Gschwendtner, and P. Muggli, *Physics of Plasmas* **21**, 123116 (2014).
- [12] P. Muggli, E. Adli, R. Apsimon, F. Asmus, R. Baartman, A.-M. Bachmann, M. Barros Marin, F. Batsch, J. Bauche, V. K. Berglyd Olsen, *et al.*, arXiv:1708.01087 (2017).
- [13] A. Modena, Z. Najmudin, A. E. Dangor, C. E. Clayton, K. A. Marsh, C. Joshi, V. Malka, C. B. Darrow, C. Danson, D. Neely, *et al.*, *Nature* **377**, 606 (1995).
- [14] C. Joshi, W. B. Mori, T. Katsouleas, J. M. Dawson, J. M. Kindel, and D. W. Forslund, *Nature* **311**, 525 (1984).
- [15] E. Esarey, C. B. Schroeder, and W. P. Leemans, *Rev. Mod. Phys.* **81**, 1229 (2009).
- [16] N. H. Matlis, S. Reed, S. S. Bulanov, V. Chvykov, G. Kalintchenko, T. Matsuoka, P. Rousseau, V. Yanovsky, A. Maximchuk, S. Kalmykov, *et al.*, *Nature Phys.* **2**, 749 (2006).
- [17] R. A. Fonseca, L. O. Silva, F. S. Tsung, V. K. Decyk, W. Lu, C. Ren, W. B. Mori, S. Deng, S. Lee, T. Katsouleas, *et al.*, *Lecture Notes in Computer Science Vol. 2329, III-342* (Springer, Heidelberg, 2002).
- [18] R. G. Hemker, arXiv:1503.00276 (2015).
- [19] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (CRC Press, 2004).
- [20] Z. Li, H.-E. Tsai, X. Zhang, C.-H. Pai, Y.-Y. Chang, R. Zgadzaj, X. Wang, V. Khudik, G. Shvets, M. C. Downer, *Phys. Rev. Lett.* **113**, 085001 (2014).
- [21] E. J. Candes, J. Romberg, and T. Tao, *IEEE Trans. Inform. Theory*, **52**, pp. 489-509 (2006).
- [22] D. L. Donoho, *IEEE Trans. Inform. Theory*, **52**, pp. 1289-1306 (2006).

- [23] A. Sävert, S. P. D. Mangles, M. Schnell, J. M. Cole, M. Nicolai, M. Reuter, M. B. Schwab, M. Möller, K. Poder, O. Jäckel, *et al.*, arXiv:1402.3052 (2014).
- [24] N. L. Kugland, D. D. Ryutov, C. Plechaty, J. S. Ross, and H.-S. Park, *et al.*, *Rev. Sci. Instrum.* **83**, 101301 (2012).
- [25] M. F. Kasim, L. Ceurvorst, N. Ratan, J. Sadler, N. Chen, A. Sävert, R. Trines, R. Bingham, P. N. Burrows, M. C. Kaluza, and P. Norreys, *Phys. Rev. E* **95**, 023306 (2016).
- [26] M. F. Kasim, N. Ratan, L. Ceurvorst, J. Sadler, P. N. Burrows, R. Trines, J. Holloway, M. Wing, R. Bingham, and P. Norreys, *Phys. Rev. ST Accel. Beams* **18** 032801 (2015).
- [27] M. F. Kasim, J. Holloway, L. Ceurvorst, M. C. Levy, N. Ratan, J. Sadler, R. Bingham, P. N. Burrows, R. Trines, M. Wing, and P. A. Norreys, *Phys. Rev. ST Accel. Beams* **18**, 081302 (2015).
- [28] M. F. Kasim, K. Glize, L. Ceurvorst, F. Keeble, J. Sadler, N. Ratan, D. Symes, N. Bourgeois, A. Savin, A. Ross, *et al.*, in preparation for *Phys. Rev. Lett.*
- [29] M. F. Kasim, P. A. Norreys, P. N. Burrows, UK patent application no. GB1712357.1 (2017).
- [30] M. F. Kasim and P. A. Norreys, BayesOpt Workshop at NIPS and arXiv:1611.05845 (2016).
- [31] N. F. Y. Chen, M. F. Kasim, L. Ceurvorst, N. Ratan, J. Sadler, M. C. Levy, R. Trines, R. Bingham, and P. Norreys, *Phys. Rev. E* **95**, 043305 (2017).
- [32] C. E. Clayton, K. A. Marsh, A. Dyson, M. Everett, A. Lal, W. P. Leemans, R. Williams, and C. Joshi, *Phys. Rev. Lett.* **70**, 37 (1993).
- [33] P. Gibbon and A. R. Bell, *Phys. Rev. Lett.* **61**, 1599 (1988).
- [34] C. Joshi, T. Tajima, J. M. Dawson, H. A. Baldis, and N. A. Ebrahim, *Phys. Rev. Lett.* **47**, 1285 (1981).

- [35] J. Krall, A. Ting, E. Esarey, P. Sprangle, and G. Joyce, *Phys. Rev. E* **48**, 2157 (1993).
- [36] K. Nakajima, D. Fisher, T. Kawakubo, H. Nakanishi, A. Ogata, Y. Kato, Y. Kitagawa, R. Kodama, K. Mima, H. Shiraga, *et al.*, *Phys. Rev. Lett.* **74**, 4428 (1995).
- [37] D. Strickland and G. Mourou, *Opt. Commun.* **56**, 219 (1985).
- [38] V. Malka, S. Fritzler, E. Lefebvre, M.-M. Aleonard, F. Burgy, J.-P. Chambaret, J.-F. Chemin, K. Krushelnick, G. Malka, S. P. D. Mangles, *et al.*, *Science* **298**, 1596 - 1600 (2002).
- [39] W. P. Leemans, A. J. Gonsalves, H.-S. Mao, K. Nakamura, C. Benedetti, C. B. Schroeder, Cs. Tóth, J. Daniels, D.E. Mittelberger, S.S. Bulanov, *et al.*, *Phys. Rev. Lett.* **113**, 245002 (2014).
- [40] X. Wang, R. Zgadza, N. Fazel, Z. Li, S. A. Yi, X. Zhang, W. Henderson, Y.-Y. Chang, R. Korzekwa, H.-E. Tsai, *et al.*, *Nat. Commun.* **4**, 1988 (2013).
- [41] J. Faure, C. Rechatin, A. Norlin, A. Lifschitz, Y. Glinec, and V. Malka, *Nature* **444**, 737-739 (2006).
- [42] H. T. Kim, K. H. Pae, H. J. Cha, I J. Kim, T. J. Yu, J. H. Sung, S. K. Lee, T. M. Jeong, and J. Lee, *Phys. Rev. Lett.* **111**, 165002 (2013).
- [43] S. Steinke, J. van Tilborg, C. Benedetti, C. G. R. Geddes, C. B. Schroeder, J. Daniels, K. K. Swanson, A. J. Gonsalves, K. Nakamura, N. H. Matlis, *et al.*, *Nature* **530**, 190193 (2016).
- [44] A. Rousse, K. T. Phuoc, R. Shah, A. Pukhov, E. Lefebvre, V. Malka, S. Kiselev, F. Burgy, J.-P. Rousseau, D. Umstadter, and D. Hulin, *Phys. Rev. Lett.* **93**, 135005 (2004).
- [45] S. Kneip, C. McGuffey, J. L. Martins, S. F. Martins, C. Bellei, V. Chvykov, F. Dollar, R. Fonseca, C. Huntington, G. Kalintchenko, *et al.*, *Nat. Phys.* **6**, 980 - 983 (2010).
- [46] J. M. Cole, J. C. Wood, N. C. Lopes, K. Poder, R. L. Abel, S. Alatabi, J. S. J. Bryant, A. Jin, S. Kneip, K. Mecseki, *et al.*, *Plasma Phys. Control. Fusion* **58**, 014008 (2015).

- [47] J. M. Cole, J. C. Wood, N. C. Lopes, K. Poder, R. L. Abel, S. Alatabi, J. S. J. Bryant, A. Jin, S. Kneip, K. Mecseki, *et al.*, *Sci. Rep.* **5**, 13244 (2015).
- [48] S. Gessner, E. Adli, J. M. Allen, W. An, C. I. Clarke, C. E. Clayton, S. Corde, J. P. Delahaye, J. Frederico, S. Z. Green, *et al.*, *Nat. Commun.* **7**, 11785 (2016).
- [49] J. A. Holloway, P. A. Norreys, A. G. R. Thomas, R. Bartolini, R. Bingham, J. Nydell, R. M. G. M. Trines, R. Walker, and M. Wing, *Sci. Rep.* **7**, 3985 (2017).
- [50] W. Lu, M. Tzoufras, C. Joshi, F. S. Tsung, W. B. Mori, J. Vieira, R. A. Fonseca, and L. O. Silva, *Phys. Rev. ST Accel. Beams* **10**, 061301 (2007).
- [51] C. D. Decker, W. B. Mori, K.-C. Tzeng, and T. Katsouleas, *Phys. Plasmas* **3**, 2047 (1996).
- [52] J. M. Dawson, *Rev. Mod. Phys.* **55**, 403 (1983).
- [53] J. Derouillat, A. Beck, F. Pérez, T. Vinci, M. Chiaramello, A. Grassi, M. Flé, G. Bouchard, I. Plotnikov, N. Aunai, *et al.*, arXiv:1702.05128 (2017).
- [54] P. Sprangle, E. Esarey, J. Krall, and G. Joyce, *Phys. Rev. Lett.* **69**, 2200 (1992).
- [55] W. B. Mori, *IEEE Journal of Quantum Electronics* **33**, 1942 (1997).
- [56] T. J. Antonsen Jr. and P. Mora, *Phys. Rev. Lett.* **69**, 2204 (1992).
- [57] Y. R. Shen, *The Principles of Nonlinear Optics* (Wiley, New York, 1984).
- [58] T. Tajima, Y. Kishimoto, and M. C. Downer, *Cluster Plasma and its Dispersion Relation*, <http://w3fusion.ph.utexas.edu/ifs/ifsreports/813-Cluster.pdf>.
- [59] S. C. Wilks, J. M. Dawson, W. B. Mori, T. Katsouleas, and M. E. Jones, *Phys. Rev. Lett.* **62**, 2600 (1989).
- [60] C. D. Murphy, R. Trines, J. Vieira, A. J. W. Reitsma, R. Bingham, J. L. Collier, E. J. Divall, P. S. Foster, C. J. Hooker, A. J. Langley, *et al.*, *Phys. Plasmas* **13**, 033108 (2006).

- [61] R. M. G. M. Trines, C. D. Murphy, K. L. Lancaster, O. Chekhlov, P. A. Norreys, R. Bingham, J. T. Mendonça, L. O. Silva, S. P. D. Mangles, C. Kamperidis, *et al.*, *Plasma Physics and Controlled Fusion* **51**, 024008 (2009).
- [62] J. Schreiber, C. Bellei, S. P. D. Mangles, C. Kamperidis, S. Kneip, S. R. Nagel, C. A. J. Palmer, P. P. Rajeev, M. J. V. Streeter, and Z. Najmudin, *Phys. Rev. Lett.* **105** 235003 (2010).
- [63] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion* vol 1 (Springer, New York, 2006).
- [64] J. M. Dias, L. Oliveira e Silva, and J. T. Mendonça, *Phys. Rev. ST Accel. Beams* **1**, 031301 (1998).
- [65] J. T. Mendonça, *Theory of Photon Acceleration* (CRC Press, Bristol, 2001).
- [66] T. D. Arber, K. Bennett, C. S. Brady, A. Lawrence-Douglas, M. G. Ramsay, N. J. Sircombe, P. Gillies, R. G. Evans, H. Schmitz, A. R. Bell and C. P. Ridgers, *Plasma Phys. Control. Fusion* **57**, 113001 (2015).
- [67] E. L. Lindmann, *J. Comp. Phys.* **18**, 66 - 78 (1975).
- [68] E. Wigner, *Phys. Rev.* **40**, 749 (1932).
- [69] K. Nishikawa, *Advances in Plasma Physics* vol 6 ed A. Simon and W. B. Thomson (Wiley, New York, 1976).
- [70] R. Bingham and C. N. Lashmore-Davies, *Plasma Phys. Control. Fusion* **21**, 433 (1979).
- [71] K. Y. Kim, I. Alexeev, and H. M. Milchberg, *Appl. Phys. Lett.* **81**, 4124 (2002).
- [72] A. Buck, M. Nicolai, K. Schmid, C. M. S. Sears, A. Sävert, J. M. Mikhailova, F. Krausz, M. C. Kaluza, and L. Veisz, *Nature Phys.* **7**, 543548 (2011).
- [73] A. M. Cormack, *Journal of Applied Physics* **34**, 2722 (1963).

- [74] N. H. Abel, Rolution de quelque problèmes à l'aide d'intégrales définies, 1823. In *Oeuvres Complètes* **1**, pp. 11-27 (Christiania, Oslo, 1881).
- [75] C. J. Dasch, *Applied Optics* **31** (8) pp. 1146-1152 (1992).
- [76] M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat
- [77] M. Tzoufras, W. Lu, F. S. Tsung, C. Huang, W. B. Mori, T. Katsouleas, J. Vieira, R. A. Fonseca, and L. O. Silva, *Phys. Plasmas* **16**, 056705 (2009).
- [78] L. Gao, J. Liang, C. Li, and L. V. Wang, *Nature* **516**, pp. 74-77 (2014).
- [79] M. Narasimha and A. Peterson, *IEEE Trans. Commun.* **26**, pp. 934-936 (1978).
- [80] A. Haar, *Mathematische Annalen* **69**, pp. 331-371 (1910).
- [81] E. J. Candes, *Comptes Rendus Mathématique*, **346**, pp. 589-592 (2008).
- [82] E. J. Candes and M. B. Wakin, *IEEE Signal Process. Mag.*, **25**, pp. 21-30 (2008).
- [83] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, *IEEE Signal Process. Mag.*, **25**, pp. 83-91 (2008).
- [84] J. M. Bioucas-Dias and M. A. T. Figueiredo, *IEEE Trans. Image Process.* **16**, pp. 2992-3004 (2007).
- [85] A. Beck and M. Teboulle, *SIAM J. Imaging Sciences* **2**, pp. 183-202 (2009).
- [86] G. Andrew and J. Gao, *Proc. Intl. Conf. Machine Learning*, pp. 33-40 (2007).
- [87] T. Blumensath and M. E. Davies, *arXiv:0805.0510* (2008).
- [88] <https://github.com/mfkasim91/anoa>
- [89] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Nature* **323**, pp. 533-536 (1986).
- [90] Theano Development Team, *arXiv:1605.02688* (2016).

- [91] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, <http://tensorflow.org> (2015).
- [92] R. Collobert, K. Kavukcuoglu, and C. Farabet, BigLearn, NIPS Workshop. No. EPFL-CONF-192376 (2011).
- [93] M. Czerny and A. F. Turner, *Zeitschrift für Physik* **61**, pp. 792-797 (1930).
- [94] A. Wagadarikar, R. John, R. Willett, and D. Brady, *Appl. Optics* **47**, pp. B44-B51 (2008).
- [95] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz, *Optics Express* **15**, pp. 14013-14027 (2007).
- [96] G. S. Settles, *Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media* (Heidelberg: Springer, 2001).
- [97] P. K. Panigrahi and K. Muralidhar, *Schlieren and Shadowgraph Methods in Heat and Mass Transfer* (Springer, New York, 2012).
- [98] R. W. Lewis, R. E. Teets, J. A. Sell, and T. A. Seder, *Applied Optics* **26** (17) pp. 3695-3704 (1987).
- [99] H. Canabal, J. Alonso, and E. Bernabeu, *Opt. Eng.* **40** (11) pp. 2517-2523 (2011).
- [100] A. Pogany, D. Gao, and S. W. Wilkins, *Rev. Sci. Instrum.* **68** (7) (1997).
- [101] G. Izarra and C. Izarra, *Eur. J. Phys.* **33**, pp. 18211842 (2012).
- [102] C. Graziani, P. Tzeferacos, D. Q. Lamb, and C. Li, arXiv:1603.08617 (2016).
- [103] F. Aurenhammer, *ACM Computing Surveys* **23**, No. 3 (1991).
- [104] M. McCool and E. Fiume, *Proc. of the Graphics Interface*, pp. 94-105 (1992).
- [105] G. Casella, C. P. Robert, and M. T. Wells, *Lecture Notes - Monograph Series* **45**, pp. 342-347 (2004).

- [106] A. Secord, Proc. Intl. Symp. Non-photorealistic Animation and Rendering (2002).
- [107] M. C. Levy, D. D. Ryutov, S. C. Wilks, J. S. Ross, C. M. Huntington, F. Fiuza, D. A. Martinez, N. L. Kugland, M. G. Baring, and H. -S. Park, Rev. Sci. Instrum **86**, 033302 (2015).
- [108] I. E. Sutherland and G. W. Hodgman, Communications of the ACM **17**, pp. 32-42 (1974).
- [109] H. H. Rosenbrock, The Computer Journal, **3**: 175-184 (1960).
- [110] N. Qian, Neural Networks **12**: 145-151 (1999).
- [111] D. C. Liu and J. Nocedal, Mathematical Programming, Vol. 45, Issue 1, pp. 503-528 (1989).
- [112] M. Schmidt. minFunc: unconstrained differentiable multivariate optimization in Matlab (2005). URL: <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>
- [113] Y. Schwartzburg, R. Testuz, A. Tagliasacchi, and M. Pauly, ACM Trans. Graph. **33**, **4**, Article 74 (2014).
- [114] F. Aurenhammer, F. Hoffmann, and B. Aronov, Algorithmica, Vol. 20, Issue 1, pp. 61-76 (1998).
- [115] G. Monge, *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, pp. 666-704 (1781).
- [116] <https://uk.mathworks.com/matlabcentral/fileexchange/56633-bounded-power-diagram>
- [117] A. Chambolle, J. Math. Imag. Vis., vol. 20, pp. 8997 (2004).
- [118] L. Davis, *Handbook of genetic algorithms* (Van Nostrand Reinhold, New York, 1991).
- [119] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Massachusetts, 2006).

- [120] H. J. Kushner, *J. Basic Eng.* **86**, pp. 97-106 (1964).
- [121] D. R. Jones, M. Schonlau, and W. J. Welch, *Journal of Global Optimization* **13** 455 (1998).
- [122] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. *International Conference on Machine Learning* (2010).
- [123] N. d. Freitas, A. J. Smola, and M. Zoghi, *International Conference on Machine Learning* (2012).
- [124] R. Munos, *Annual Conference on Neural Information Processing Systems* (2011).
- [125] Z. Wang, B. Shakibi, L. Jin, and N. d. Freitas, *International Conference on Artificial Intelligence and Statistics* (2014).
- [126] K. Kawaguchi, L. P. Kaelbling, and T. Lozano-Pérez, *Annual Conference on Neural Information Processing Systems* (2015).
- [127] C. Runge, *Zeitschrift für Mathematik und Physik*, **46** pp. 224-243 (1901).
- [128] A. S. Parnovsky, *Acta Phys. Pol. A*, **93** Supplement, S-55 (1998).
- [129] N. Hansen, arXiv:1604.00772 (2016).
- [130] <https://github.com/mfkasim91/tabe>