Author:
Dias Paulino Da Costa, Frederico

Title:
Simulation of observable in-flight shockwave shadows

Simulation of Observable In-Flight Shockwave Shadows

By

FREDERICO DIAS PAULINO DA COSTA

Department of Aerospace Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements for award of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

DECEMBER 2021

Word count: 76812

ABSTRACT

Sunlight impinging on an aircraft interacts with the shock waves that form due to the high-speed flow and produce an optical phenomenon comprising of a shadow and caustic pattern extending along the wing span. This direct natural shadowgraph feature is visible under particular observer, vehicle operation and illumination conditions. Despite similarities to simulations of traditional fluid flow visualisation images, the phenomenon has not been replicated in its entirety before this research. The capability to simulate the phenomenon in a realistic manner is a prerequisite for future research. This work will allow the relationship between the observed visual artefacts, and the trans/super sonic compressible flow over an aircraft, to be established.

In this research, a computational framework to simulate the behaviour of light in compressible flow is developed. Underpinned by the discipline of geometrical optics and based on underlying computational fluid dynamics solutions, the framework is composed of physically accurate models of all of the elements of the shock wave shadow formation and visualisation. These are used to predict the flow field, replicate light scattering and digitally generate realistic shock wave shadow images.

In a fully tested working implementation, both natural and artificial (wind tunnel) shadowgraphy experiments are accurately reproduced. To obtain representative synthetic shockshadows, it is determined that light must reach the shock waves in a grazing incidence condition. Moreover, it is found that the computation of the final illumination distribution depends on the spatial sampling and on the radiant energy gathering characteristics used in an adaptive statistical density estimation method.

DEDICATION

To Rosa, my beloved aunt and godmother, who was an example of courage and resilience.

ACKNOWLEDGEMENTS

I would like to first thank Dr Raf Theunissen, as without him none of this would have been possible. Raf, my friend, thank you for the opportunity, for our friendship, for the hours spent together brainstorming and for the major inspiration you have provided me since we first met. I will be forever grateful for all that.

Dr Dorian Jones and Dr Ann Gaitonde, thank you for the help while I developed this work. A particular thanks must also be given to Dr Simon Lock, who always promptly helped me with my requests. To all the other University staff who I met either through teaching or through the aerospace representation, thank you for this incredible experience.

I would also like to thank Mike Finch and Paul White from Airbus. Our meetings were important to consolidate my progress in this research.

I am very thankful for all the friends I have made and all the amazing people I had the chance to meet over these past four years. Those who share my passion for climbing and who came from my country, thank you for making my life far from home easier. A special thanks to Bella, without whom I would not have been able to get to the end and overcome these challenging times.

Finally, I am so thankful for the love, encouragement, mentoring and, most of all, understanding from my family. All that I have achieved is due to your endless support. Thank you.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

TABLE OF CONTENTS

LIST OF TABLES

TABLE **Page**

LIST OF FIGURES

FIGURE **Page**

CHAPTER 1

INTRODUCTION

Technological advances in the Aerospace industry pushed the flight speed of aircraft to magnitudes in the vicinity of or even above the speed of sound, which were once believed unachievable. This new realm of flight aerodynamics exposed some new and unfamiliar phenomena, which affected the overall performance of high-speed aircraft. An example of these phenomena is a *shock wave*, which occurs when matter, irrespective of the state, undergoes a rapid compression. Having higher amplitudes than, for example, acoustic or sound waves, shock waves delimit a three-dimensional boundary across which an abrupt variation of the thermodynamic properties of the medium is observed, including density. As a mechanical high-rate wave, this discontinuity propagates in space with supersonic speeds conditioned by the magnitude of the pressure change that originated it. The development of experimental studies into the underlying physics of shock waves has always been primarily based on the occurrence of the phenomenon in gases, given the translucence and high compressibility of substances in this state and the common use of optical investigation methods [1]. However, shock waves are actually studied for many different applications, ranging from ballistics and explosions to seismology, astrophysics and gas dynamics [2].

1.1 Classical Optical Studies

Being a natural physical phenomenon, shock waves have long been explored by pioneers, whose achievements provided not only the most fundamental insights into supersonic flow but also the parallel development of high-speed diagnostic techniques, together forming the current state of knowledge regarding shock waves. These pioneers were physicists, mathematicians, researchers, experimentalists and simply curious individuals who attempted to explain the unique character of shock waves. Over the course of approximately three decades in the second half of the nineteenth century, the motto of Ernst Mach that ’Seeing is understanding’ [1, 2] was explicitly followed and the most significant methods of experimental fluid flow visualisation were devised, and some valuable measurements and recordings were made.

In the context of the present research, some of the names mentioned in the all-encompassing chronology presented by Krehl [2] stand out. Perhaps the most significant contributions are those from Ernst Mach, whose broad experience influenced everyone who followed in his footsteps. Mach, with the help of other researchers, was responsible for some groundbreaking achievements in the fields of gas dynamics and high-speed photography. These achievements include the very first laboratory confirmation of the supersonic speed of shock wave propagation and its dependence on the shock strength, the first photograph of a shock wave and even the first record of density variation at the shock wave using an early version of the interferometry technique [2–5].

This earlier construction of the optical interferometer is attributed to Jules Jamin, but was later further developed by Ludwig Mach and Ludwig Zehnder, independently, originating the so-called Mach-Zehnder interferometer device [2, 6–8]. The interferometer was also used by August Toepler in fluid dynamics, who became mostly known for his own invention, the schlieren technique (see Figure 1.1). The technique enabled the direct visualisation of discontinuities in air, which he then described, for the first time, using the term *shock wave* [2, 9, 10]. Vincenz Dvorák built on the work of Toepler and greatly simplified the schlieren technique to create a new method, then named shadowgraphy, that provides similar images without requiring such complex experimental apparatus [2, 11]. Lastly, Ludwig Prandtl effectively stated that ’the thickness of shock layers range within the wavelengths of visible light.’ [2, 12]. For further details of all the significant milestones in the understanding of shock waves, see [2].

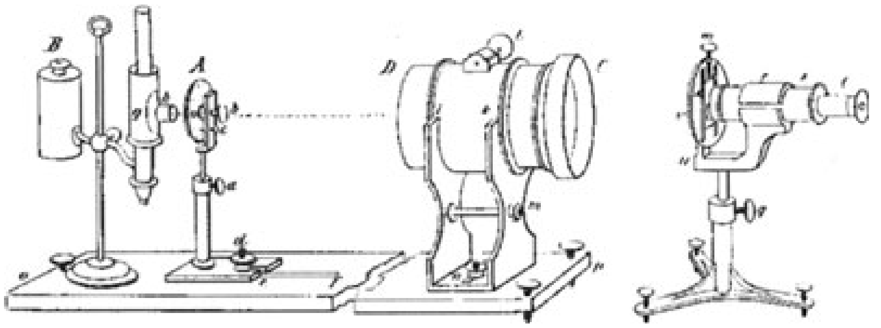


Figure 1.1: Optical apparatus used by Toepler to develop the schlieren technique. Image reproduced from [1].

Prandtl's statement is of great importance, particularly in the context of the classical studies of shock waves, and in understanding how the respective optical systems (and modern developments) process light to form images that allow one to visualise a fluid flow. The previously mentioned characteristic of shock waves reveals that the behaviour of light involves more than the scattering, i.e. reflection and refraction, that can be approximately modelled by treating light in these systems as simple beams or rays parallel to the propagation trajectory. This is a fundamental abstraction in the idealised and simpler domain of Geometrical Optics. In fact, the behaviour attributed to the wave nature of light, e.g. interference and diffraction, can also be observed in these studies. Therefore, in principle, the wavelength of light needs be taken into account and the visualisation problem has to be also or alternatively addressed from the perspective of Physical Optics [13].

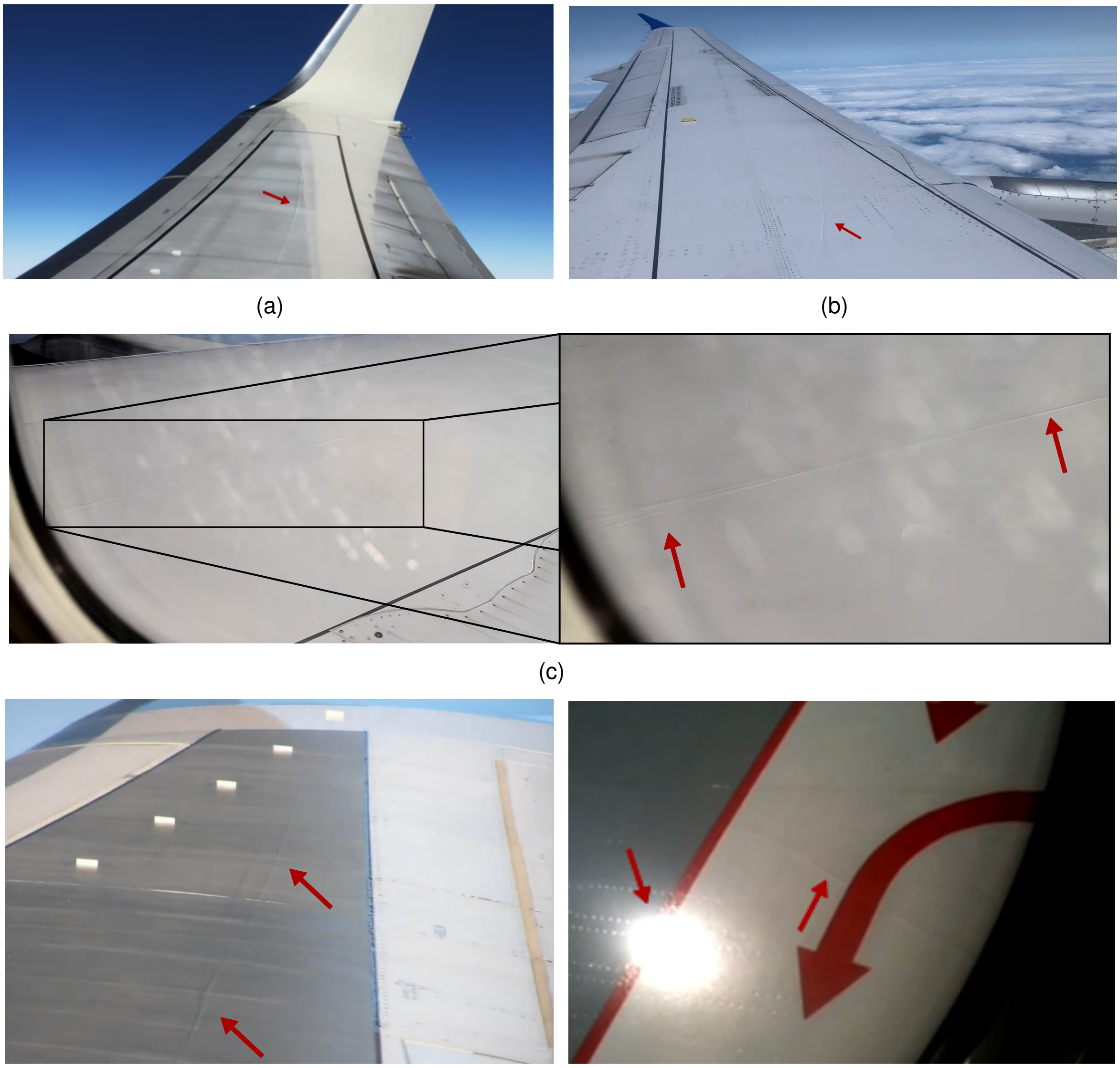
1.2 A Natural Phenomenon

Settles [1] noted that a typical shock wave encountered in compressible flow, travelling just over the sonic speed, induces a very small variation of around 10^{-5} in the refractive index of air due to its characteristic density gradient. This inhomogeneity in the optical medium in turn causes the refraction of light and a deflection of only approximately 10^{-3} degrees or less [1]. Although such a weak discontinuity and small light bending are usually invisible to the naked eye, the light displacement, deviation and interference resulting from the interaction with compressible flow features can be very pronounced. This is due to the complex wave-particle duality of light and the intricate characteristics of shock waves themselves (e.g. shape, internal structure and general behaviour within the flow). The outcome is that image features associate with shock waves can occur naturally, when the conditions are right and 'one knows where and how to look' [1], or even be artificially produced. Through the establishment of these visualisation requirements, from illumination to formation and viewing, and development of suitable mechanisms that can generate or enhance these conditions, as with the use of special lights and/or optical systems, similar visual effects can be attained in a controlled environment. The experimental flow visualisation techniques briefly mentioned above, i.e. shadowgraphy, interferometry and schlieren imaging, are examples of that.

The diverging high intensity light that is emitted from the sun, which after travelling a very long distance arrives at the earth as a parallel beam, can serve as an appropriate light source to generate direct natural shadowgraphs. These shadowgraphs of a refractive index field can be cast on any approximately flat and diffuse surface [1], via direct illumination or even from indirect illumination by sunlight reflected from the moon. In the 1940s, for example, in response to reports from pilots regarding aeroplane instabilities during flight, Major Frederic Austin Borsodi conducted tests on a P-51 Mustang, diving from altitudes around 35,000 feet and reaching Mach numbers up to 0.83. During these tests, Borsodi became possibly the first man to witness and historically record a shock wave 'as a shimmering line of light and shadow extending spanwise from the root on the upper surface of the wing' [14]. Thereafter, the same visual effect was again described during other flight tests related to the development of the Lockheed P-80 Shooting Star, the first American jet-propelled fighter aeroplane, where the light and dark bands were observed on the wing tip fuel tank intersection [15].

These days airline passengers can often see this shadow and caustic (envelope of refracted rays [13]) pattern [16] over the wing when the Sun is approximately overhead [1]. A quick search on the YouTube website (*http://youtube.com*) reveals several examples of video recordings where the oscillatory optical phenomenon appeared in the out-of-the-window view. Figure 1.2 reproduces some frames captured from those videos, where one can see the specific pattern sequence of a shadow followed by a bright fringe in the leading to trailing edge direction (Figures 1.2(a) and 1.2(b)); the different shapes including bifurcations in those patterns (Figures 1.2(c) and 1.2(d)); and even the glare or sun reflection highlight indicating its positioning above the wing (Figure 1.2(e)). These frames indicate that what is visible from the airborne perspective is completely dependent on the particular vehicle operation, illumination, observer and recording conditions.

Direct natural sunlight shadowgraphy is therefore an interesting and potential valuable subject for study. This research is in response to this fundamental question: what can be inferred about the shock wave and the compressible flow field in general from the observed shadow and caustic patterns? Despite being a very straightforward question, answering this question presents considerable challenges. This may be the reason why there are not many investigations related to this specific phenomenon in the literature. Although these visual effects have not been thoroughly explored from the experimental perspective, the field of fluid flow visualisation, especially in the aerospace industry, has long been harnessing the natural behaviour of light in compressible flow to study, analyse and evaluate naturally invisible flow features. Therefore, learning how these traditional experimental techniques were devised and how they work can be helpful in terms of complementing the understanding of the sunlight shadowgraphy and its computational simulation. The following section thus reviews these flow visualisation techniques. Next, the historical reports of observation of the actual phenomenon are also reviewed, since the insights already available may prove to be useful in further understanding shock wave shadows and how they arise.



(a)

(b)

(c)

(d)

(e)

Figure 1.2: Direct natural sunlight shadowgraphy visible during transonic flight. The images are from [17–21], respectively, and are in the public domain. Note that (c) shows the frame image on the left-hand side and a zoomed in section around the shockshadow of the same on the right-hand side.

1.3 Fluid Flow Visualisation

Fluid flow visualisation as a discipline encompasses a variety of diagnostic tools that are employed in general thermofluids studies. For shock wave research, flow visualisation is used to convert the invisible, e.g. a fully transparent medium and physical phenomena that occur within it, into something that can be seen. Classical optical experimental techniques are particularly valuable being a nonintrusive means of obtaining information about the flow, without the addition of foreign particles or materials. Mostly, this is qualitative information in the form of patterns recorded in images. However, sometimes quantitative information can also be obtained when these methods are combined with additional data sources and processing. This information includes the overall aspects and properties of the flow and is mostly free of possible measurement disturbances that would otherwise influence the flow itself and the results. Through these methods the now observable features of the flow can be detected, such as shock waves, enabling further investigation and understanding of its phenomenological aspects and behaviour mechanisms [22, 23].

Optical experimental techniques essentially see the compressible flow as a simple refractive index field [22]. This is possible from the standpoint of Physical Optics, which is the field of study predominantly interested in the wave nature of light and its electromagnetic characteristics [13]. The work of Lorentz and Lorenz provided a useful constant of proportionality between the oscillating electric field which is the light wave and the corresponding charge configuration established in the fluid molecules under the influence of this field [24]. A further development of their formula resulted in a direct relationship between the index-of-refraction and the density of a fluid for light of any wavelength [22, 24]. The relationship is somewhat complex, as it involves the properties of an electron, molecular properties of the fluid and oscillatory features of the electromagnetic wave. One of its forms valid for gases and liquids is attributed to Clausius and Mosotti [22], however this formulation is outside the scope of this thesis and hence is not presented here.

A simpler relationship can be found for gases, which have a refractive index approximately equal to one. In this case the refractive index of the fluid (n) is directly proportional to its density (ρ). The relationship is simplified to the Gladstone-Dale relation [22] and is written as

$$n - 1 = K\rho \tag{1.1}$$

where K is the so-called Gladstone-Dale constant. This constant depends on the gas composition and on the wavelength of the light interacting with the fluid [22, 25]. Table 1.1 highlights the weakly dispersive characteristic of the constant for air at $T = 288K$, varying only slightly across a range of wavelengths, including the visible spectrum [22].

In air, a transparent medium, the index-of-refraction distribution established by changes in the thermodynamic properties of the fluid are naturally invisible to the naked eye. However, they can be made visible by setting up specific optical apparatus based on how these variations affect light propagation. Light, as an electromagnetic wave, will have its main characteristics

Table 1.1: Gladstone-Dale constant for air at T=$288K$. Table reproduced from [22].

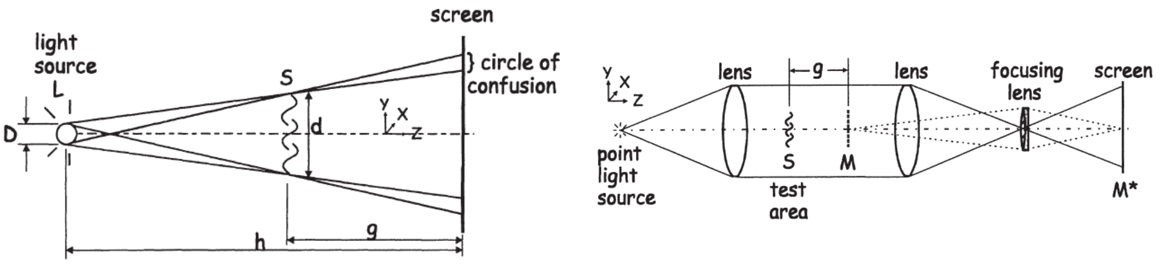
K, cm^3/g	Wavelength, μm
0.2239	0.9125
0.2250	0.7034
0.2259	0.6074
0.2274	0.5097
0.2304	0.4079
0.2330	0.3562

determined by the interaction with the fluid [25]. In an isotropic inhomogeneous medium, where the refractive index is a function of location in space $n = n(x, y, z)$ [26] and the Gladstone-Dale relation holds true [24], there will be simultaneously a change in the direction of propagation and a phase shift of a light beam traversing the discontinuities present in the medium [22, 25]. These disturbances of light rays can be measured as a geometrical displacement, an angular deflection, a retardation or a combination of these quantities [25] with respect to a scenario in which light rays remain undisturbed [22]. In practice, an arrangement containing an image plane is used to capture the observable patterns on a picture [22], thus photographing the light disturbances instead of the flow itself [23]. Each class of optical flow visualisation methods described below focus on one of those quantities and, as line-of-sight or integral along the optical path methods, they empirically record the modification in location, impingement angle and time with which light reaches the sensing device [22, 23].

1.3.1 Shadowgraphy

Shadowgraphy is the simplest of the classical flow visualisation methods [1, 22, 23, 25]. Its simplicity lies in the fact that even without any equipment, and for a field of view containing an inhomogeneous medium, the visual patterns of a shadowgraph can be quickly generated on any approximately flat diffusive surface, which reflects light equally in all directions, by shining a light on the domain [1, 22].

Shadowgraphy visualisation can be set up either in a direct or a focused configuration, represented in Figure 1.3. If a diverging light source is used in the direct configuration, the magnification of the surface of inhomogeneity on the shadowgraph and the sensitivity or contrast reproduced on the final image are controlled only by the positions of the light and the screen relative to the refractive field itself. Theoretically, sensitivity is maximised in this case of direct shadowgraphy if the source is located far from everything and the discontinuity occurs exactly in between the positions of the source and of the screen. The further the diverging light is positioned the more it effectively behaves as a collimated beam or parallel light source, and the sensitivity available in this condition is actually doubled in comparison to when the light is located closer, for a fixed distance between the refractive field and the screen. Parallel light is therefore the arrangement of choice experimentally, realisable through the use of optical apparatus, such as lenses and mirrors, or even naturally exploiting sunlight, as per the shock wave shadow phenomenon.



(a)

(b)

Figure 1.3: (a) Direct and (b) focused configurations of the shadowgraphy experimental technique with a diverging light source. Images reproduced from [27].

Despite the high sensitivity achievable with direct shadowgraphy, it still suffers from image blur due to the fact that the light source is not in reality a singular point, but rather a three-dimensional shape of some sort, and also because of the wave aspect of light, which means that diffraction effects can also occur. Because of these blurs and distortions, contact shadowgraphy is often times practised, in which case the screen, or a photographic film, is placed as close as it is practically feasible to the inhomogeneous medium. This approach reveals several details of the flow field (e.g. shock waves and boundary layers), at the cost of image contrast, though being still well adequate for visualisation purposes. Shock wave shadows cast over aircraft wings are a clear example of the feasibility of contact shadowgraphy, considering that the inhomogeneity is literally attached to the projection surface, i.e. the wing itself.

In the focused shadowgraphy configuration, a parallel light source is also set, and two additional lenses are placed behind the inhomogeneous medium, focusing and projecting the shadowgraph onto the screen. These optical instruments, also present in the schlieren system, greatly facilitate the adjustments of magnification, sensitivity and resolution of the shadowgraph, instead of relying on changes of the physical location of the recording film to do so [1].

Figure 1.4 depicts an example of direct shadowgraphy [1], where the lit candle is illuminated by a simple torch and the "shadow" of the flame or, better, the indication of the presence of convective plume [1] is then cast on a white wall. Parallel light rays traversing the gradient refractive index field around the flame refract, deviate from their original path and arrive at the background wall at different locations. It can be seen in Figure 1.4 that this new distribution of energy on the wall translates to the two main features of a shadowgraph: namely the concentration of light in one location and deficit of light in the vicinity of this illumination highlight or caustic feature. In other words, there is a region where light is now scarce resulting in the shadow or dark band that gives the technique its name, and another region where those disturbed light rays have interacted in different ways with the uniform light forming a caustic pattern or a bright fringe (a coherent caustic) [1, 22, 25].

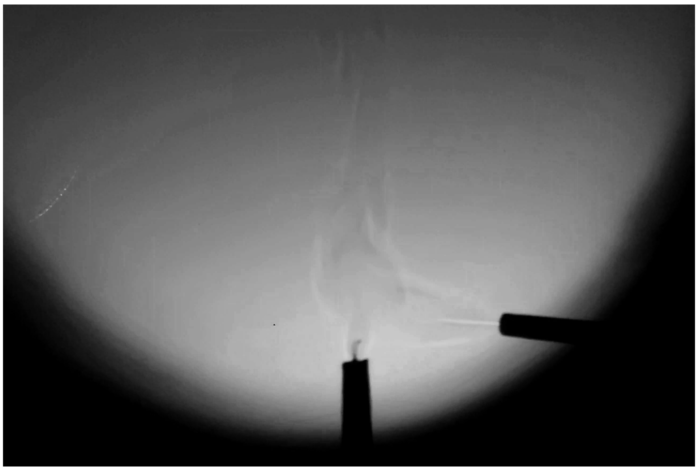


Figure 1.4: Classical shadowgraph of a candle plume.

For the case of shock wave visualisation, the specific sequence of dark and bright bands then fading back to the background level in the flow direction is the most common pattern, assuming a standard experimental setup of light source and recording plane on opposite sides of the flow discontinuity. Then the discontinuity front coincides with the outer edge of the shadow, making it a useful method for locating a shock wave surface and measuring its distances to the solid body immersed in compressible flow. Although the dark band upstream boundary marks where light was initially disturbed and the respective contour of the refractive object in two dimensions, other features such as the shadow band width and strength or bright band position, width and strength usually can not be directly correlated with a physical aspect of the flow, as they correspond to distorted representations of reality. The caustic is the clear example of this distortion, since the envelope of refracted rays can spread out or concentrate across the screen depending on the characteristics of the shock wave that originated it, but the assessment of the exact trait responsible for this light behaviour can be very contradictory. In fact, a caustic pattern can lead to an incorrect visual interpretation of the flow feature [1, 22, 23]. A very curved shock wave, for example, could induce a high degree of light deflection, which would result in a sharp caustic. Similarly, a strong shock wave would cause intensive light refraction, from which a concentrated caustic would also arise. Given that shock waves are three-dimensional flow features, two shocks with different strength, shape and internal structure can therefore, in principle, produce a very similar if not equal visual pattern.

The shadowgraph technique visualises the geometrical *displacement* of the light beam or array of light rays and the relative changes in light intensity at a plane of observation with respect to undisturbed case are recorded by a photographic film. This form of light intensity normalised contrast is sensitive to variations in the second derivatives of the gas density. This characteristic inhibits the extraction of quantitative density information from shadowgraphs, which would require double integration. Figure 1.5 exemplifies the shadowgraph dependency on a variable Laplacian of the density field [1, 22].

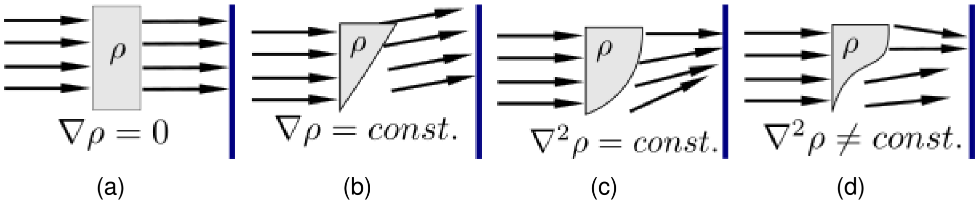


Figure 1.5: Schematic of light ray deflection behaviour in optical media equivalent to (a) a constant density field; (b) a constant density gradient field; (c) a constant Laplacian of the density field and; (d) a variable Laplacian of the density field.

As shown in Figure 1.5(a), the homogeneous refractive index media only retards (change in phase) the wave fronts. The linear variation of density, represented in Figure 1.5(b), deflects all light rays by the same angle and results in an uniform illumination. The quadratic distribution of density (Figure 1.5(c)) also produces a uniform illumination, but with lower exposure given the convergent lens effect of the media [22]. It is only the cubic variation in density (Figure 1.5(d)) that actually results in a nonuniform illumination on the image plane [25]. Therefore, discontinuities in the gradient of density like the ones associated with the formation of shock waves and turbulent structures in compressible gas flows are well detected and depicted by shadowgraphs, producing sharp variations in the illumination of the scene [1, 22, 23], as in the shadowgraph example shown on Figure 1.6.

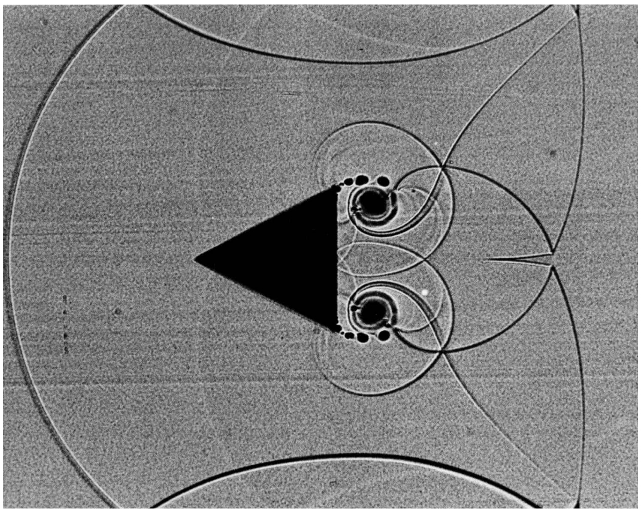


Figure 1.6: Shadowgraph of the interaction between shock waves and vortices around a finite wedge. Image reproduced from [28].

1.3.2 Schlieren Imaging

A schlieren imaging system, represented in Figure 1.7, is a somewhat more complex experiment than shadowgraphy. It requires an optical imaging unit containing not only a lens system similar to the one employed in the focusing shadowgraphy method, but also a cutoff device as the principal component. This device, commonly referred to as the knife edge (also the original name given to the method), blocks a portion of the light emitted by the source and, in doing so, it effectively reduces the exposure of the recording film and separates deflected from undisturbed background light. As a result, on the displaced image of the light source that forms at the plane of the cutoff device, the points corresponding to the previously intercepted disturbed beams appear darker. The points corresponding to the beams that were still free to travel in turn appear brighter. This method projects an inverted image of the inhomogeneity on the screen, which contains the visual feature that gives the more recent name of the technique. The schlieren feature is nothing more than a small gradient in the refractive index of the medium that distinguishes it from the surrounding homogeneous background. This type of feature can occur in different media, with different structures and levels of complexity, caused by variations in temperature, density or simply in the geometry of the medium. This illumination distribution, including the schlieren feature, then retrieves the shape and strength of the inhomogeneity. In general, the features in the image correlate explicitly with the refractive field that the method visualises. Because of the very high sensitivity that can be achieved in the technique, illumination details are even usually highlighted in comparison to the other techniques [1, 22, 23].

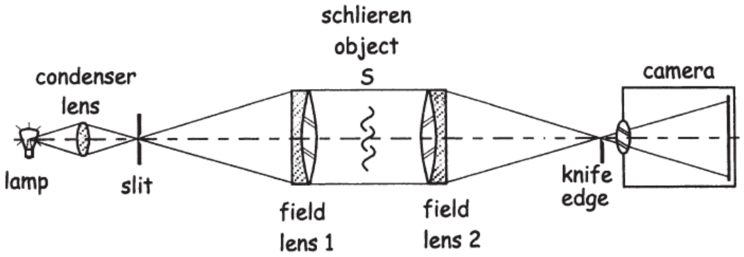


Figure 1.7: Typical setup of the schlieren experimental technique. Image reproduced from [27].

The schlieren imaging technique detects and measures the change in light direction or angular *deflection* that occurs after the traversal through the inhomogeneous refractive index field. The deflection is therefore visible in the resulting photograph and, moreover, it is proportional to the density gradient in the direction perpendicular to the edge of the cutoff device only. Systems composed of multiple or particularly oriented and shaped cutoff devices are then often used to capture deflections, and the respective density gradients, in more than one direction. This multidimensional possibility and the level of resolution (higher than in shadowgraphs) that can be achieved with an appropriate configuration makes schlieren particularly useful for the visualisation of aerodynamic features characterised by intrincate and weaker gradients patterns, as the ones shown in Figure 1.8 [1, 22, 23].

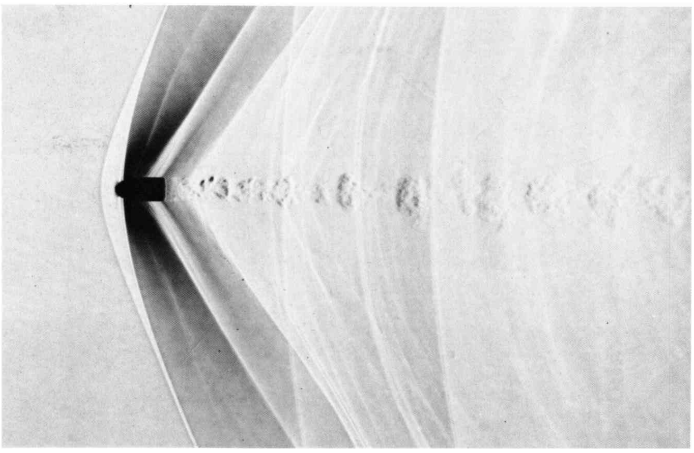


Figure 1.8: Schlieren photograph of a rifle bullet at supersonic speed. Image reproduced from [28].

Although the relative intensity of the schlieren features on the final image can be adjusted through changes of the aperture of the knife edge, there is a limit on the resolution that can be used. The resolution is related to the illumination contrast displayed in the image and to the sensitivity range of angular deflection that can be measured and represented in the image. The limitation on those factors is imposed both by a minimum illumination level that is required as the background reference, and by the appearance of diffraction effects. The latter, different from the schlieren feature itself, are usually undesirable. There is a fundamental trade-off between contrast and measuring range, characteristics that determine the sensitivity of the schlieren system [1, 22, 23].

If care is taken in setting the system in order to avoid these effects and reduce or even eliminate other possible optical aberrations, a schlieren image can be used to evaluate the refractive index and hence the density distribution within the flow field via single integration. The gradient dependence also means that some slowly varying or more gradual phenomena, such as expansion and compression zones, that are not visible in shadowgraphs now can be seen through schlieren images [1, 22, 23].

Some more modern systems, while still derived from the original, were completely modified to integrate other instruments such as the additional cutoff device. Another modification is the replacement of the knife edge by colour filters, which introduces changes in hue to the traditional continuous gray-scale monochromatic images, according to the amount of deflection that light has experienced. Although colour schlieren is supposedly more appropriate for visualisation of features by the human visual system, diffraction again affects the technique and the overall image quality obtained with this modification is much lower, especially if attempting to maintain the level of sensitivity achievable with the conventional system [1, 23].

1.3.3 Interferometry

Interferometry, as the name suggests, is based on the interference phenomena that occur between light waves that propagate in different optical domains. Although there is no significant difference between interference and diffraction [13], they are usually studied separately and only the latter is mentioned as having, in some cases, a significant contribution to the formation of a shadowgraph [1, 22]. Moreover, this study is specifically focused on the domain of Geometrical Optics, in which the light behaviour associated to its wave nature, e.g. interference, is not modelled, but only propagation and scattering. Therefore, interferometry techniques do not relate directly to this work, but are nevertheless described only for completeness.

Through the technique, an object light beam crosses a discontinuous refractive index medium and is slowed down or sped up and therefore phase shifted with respect to another, the reference beam. The phase shift is effectively associated with a difference in the optical path lengths and in the related times of travel of these waves. The stable interference image pattern that forms from this interaction between beams with different phases is composed of light and dark fringes. The fringes that form can be proportional to the gas density gradient and thus can represent lines where this variation rate is constant. Alternatively, they can correspond to lines of constant refractive index and, thus, the density increments between two fringes can be evaluated. This depends on the configuration of the interferometer [23, 25].

Interferograms are able to depict details of flow features such as shock waves (see Figure 1.9), boundary layers and shock wave-induced boundary layer separation in images. However, the qualitative interpretation of the images is still considerably more difficult compared to shadowgraphs and schlieren images and some compressible flow features can easily be missed during an initial inspection [23, 25].

1.3.4 Background-Oriented Schlieren

When a discontinuity is appropriately illuminated and observed against a background that contains some well-defined contrasting light and dark boundaries, it becomes a visible schlieren object [1]. Using the Sun as the light source and a cloudy sky as the background, for example, serve perfectly to create these natural background-distortion schlieren visualisations of shock waves above the wings of aircraft in transonic flight [1]. The possibility of schlieren imaging flow fields of large-scale, outside of an experimental facility and therefore without requiring most of the equipment previously used aroused considerable interest among researchers. With lower

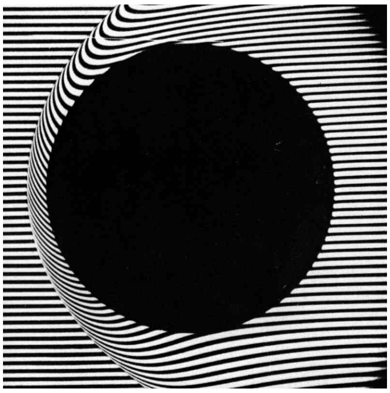


Figure 1.9: Finite fringe interferogram of a bow shock wave detached from a supersonic sphere. Image reproduced from [28].

costs of purchase, configuration and operation, the development of an actual experimental technique that seizes the background distortion as an additional source of information for the generation of schlieren-like images has been encouraged.

More recently renamed background-oriented schlieren, or simply BOS, the re-engineered versions of the technique are actually synthetically produced, in the sense that digital image processing and sometimes artificial backgrounds are also employed in the process. These backgrounds, whether artificial or natural, are selected amongst those that exhibit a randomly distributed high contrast structure on a small scale and an approximately uniform pattern on a large scale. This is essential if the computerised processes, which are analogous to typical Particle Image Velocimetry analyses, are to perform optimally. From two identically set and gathered pictures of the background, with only one containing the refractive index discontinuity of interest in the field-of-view, the usual optical unit of a schlieren system is then replaced by cross-correlation computational algorithms that compare those images and evaluate the resulting distortion in terms of apparent pixel shift. Natural BOS is even achieved reference-free, using instead two cameras acquiring images of the phase object at each instant, but separated in space and/or time. The distortion can then be directly related to the light angular deflection, given the geometry of the photographic setup and, thus, the classical visual effect of a schlieren image can also be generated through this synthetic BOS technique [29, 30].

At the NASA Armstrong Flight Research Center, researchers utilised the BOS technique to visualise not only schlieren images of shock waves around a single full-scale aircraft in supersonic flight [31], but also of the interaction between multiple shock waves emanating from two similar vehicles (see Figure 1.10) [32]. The experiments, associated with studies of noise in supersonic flight, were performed with special attention to the choice of natural background, imaging system, processing algorithms and positioning of both object and camera aircraft.

As suggested in [29] for effective BOS performance in an outdoor environment, the flight tests were carried out above a desert, which resembles an ideal background. State-of-the-art video equipment that is able to capture images with a very high definition and frame rate in a wide field of view was also required. Similarly, the subsequent storage of the large amount of data produced was necessary [31, 32]. Indeed, the natural BOS imaging proved to be very useful for this type of application where the traditional optical flow visualisation techniques are constrained [29].

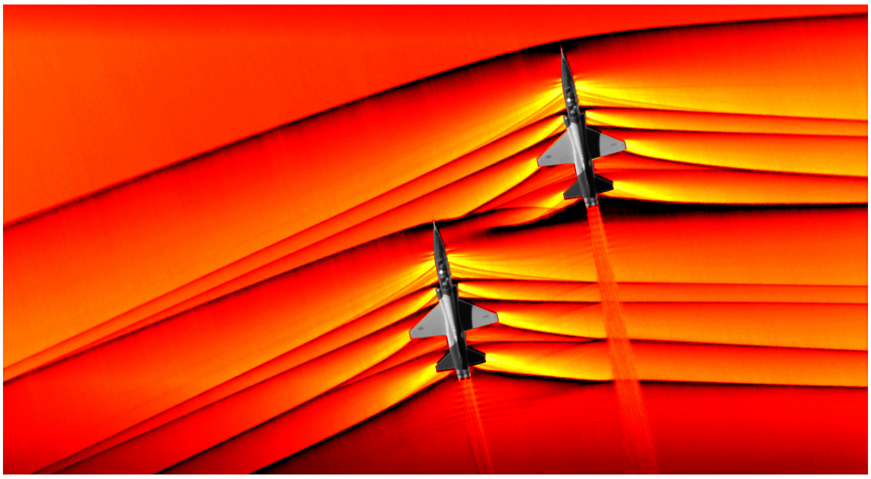


Figure 1.10: NASA Air-to-Air Background Oriented Schlieren image of shock wave interaction in supersonic flight. Image reproduced from [33].

1.4 Shock Wave Shadow Observations

Only a few other accounts of shock wave shadows or "shockshadows" exist in the literature. Rathert and Cooper [34] provided an initial investigation into the phenomenological aspects of this type of shadowgram or shadowgraph, both equivalent names for the same type of image. By recreating similar conditions to the first recorded observations of shockshadows, they were able to replicate sunlight shadowgraphs during high-speed dive tests with multiple single-engine fighter-type aeroplanes. This decoupled the appearance of the phenomenon from the specific type of aerial vehicle being flown. Highly skilled pilots flew the aeroplanes to render the shadowgram easily observable [34]. Normal shock waves were made visible in a planform configuration cast over the *unswept* wing surface and also in a profile perspective projected on the canopy, as shown in Figure 1.11. The pilots were able to see the shadowgrams because of their particular positions inside the cockpit, aligned with the shock front along the wing span, and these effects also appeared more clearly in photographs captured in line with the shock [34].

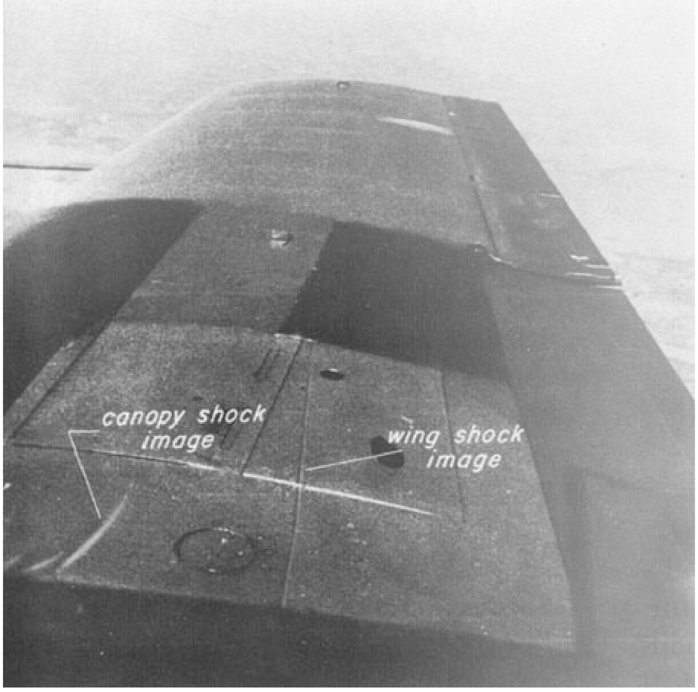


Figure 1.11: Record of shock wave's shadows cast over the wing surface and projected on the canopy of an aeroplane during high-speed dive tests. Image reproduced from [34].

The aeroplane attitude was controlled in order to achieve flight conditions that reached critical Mach number, i.e. sonic flow and shock wave formation at the wing, and a particular orientation with respect to the sun. The sun had to be 'near its zenith' or 'directly overhead' [34] with an elevation angle between $45°$ and $135°$ above the wing, and approximately within an azimuth angle of $\pm 20°$. Moreover, during summer months these shadowgrams were more easily reproduced on the wing and they could be visualised both when looking towards and away from the sun. The spectral dependence of light was also considered when the wing was painted with different colours, including dark blue and red, or kept as a pure aluminium surface, either polished or not. Interestingly, the red paint provided better images, indicating that the reflective properties of the surface at the higher end of the visible spectrum produced less blurred results or a sharper shadowgraph. By contrast, the rough unpainted wing resulted in caustics separated into segments, instead of the single smooth curve usually observed. These results demonstrated that the surface conditions also affect how light is reflected and refracted back towards the viewer [34].

Experimental pressure distribution data, representative of the type of aircraft used in the flight tests, suggested that the common expectation that the outer edge of the dark band of the shadowgraph nearly coincides with the shock wave location could thereby be confirmed. Visual inspection of the shadow during flight at different speeds, altitudes and attitudes also verified that the behaviour of the visual effect was indeed associated with the characteristics of its cause, moving forward or rearward as the strength of the shock wave was deliberately decreased or increased, respectively. Furthermore, the shadow seemed to also oscillate in accordance with the appearance of buffeting, suggesting a link between the shock wave itself, a possibly shock wave induced flow separation, and the vibrations caused by these respective aerodynamic excitations [34].

Larmore and Hall [35] described some common and other rather rare optical effects stemming from sunlight reflection and refraction that can be visualised by the airborne observer. However, they only briefly mentioned the in-flight shockshadow that appears on the inboard section of an aircraft wing, reproduced in Figure 1.12, and the unsteadiness characteristic of this visual phenomenon.

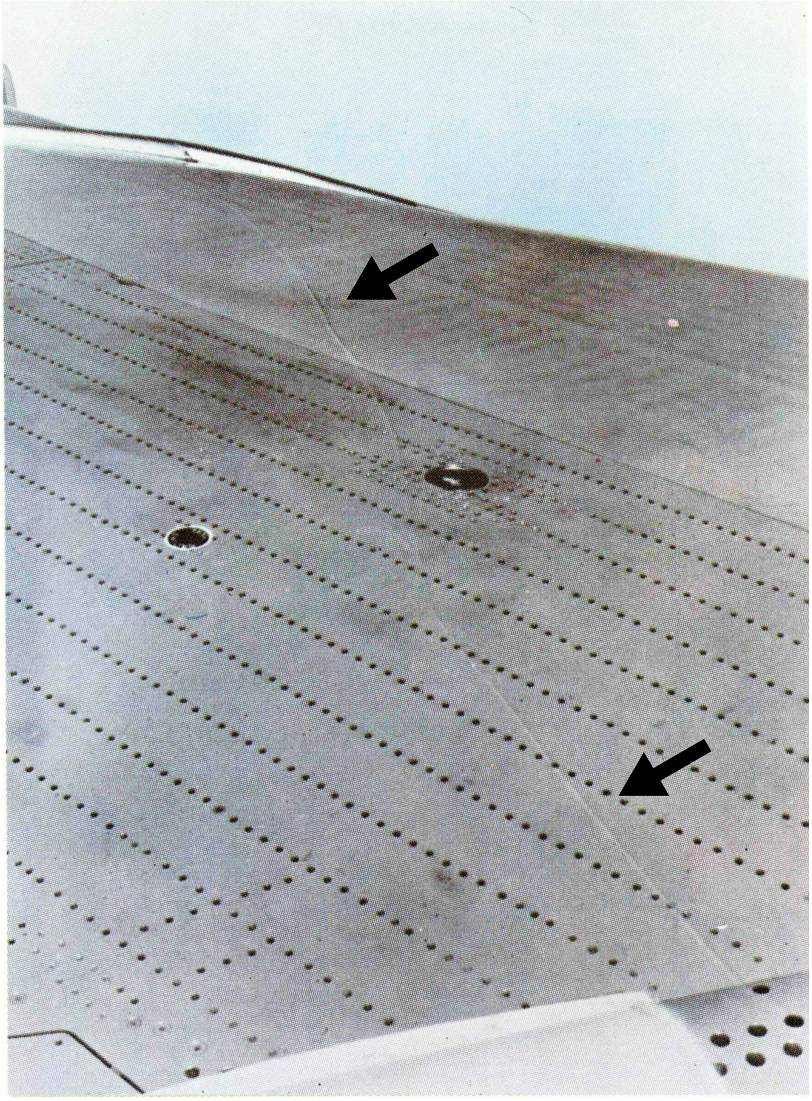


Figure 1.12: In-flight shockshadow at the inboard of an aircraft wing. Image reproduced from [35].

Crowder [36] described sunlight shadowgraphy as a simple experiment that does not require the same expensive facilities and complex optical instruments of the traditional flow visualisation techniques, apart from perhaps a camera for recording purposes. He suggested that the shadowgrams, which appeared when sunlight was tangent to the shock wave surface during demonstration tests, would then transform complete shock wave systems typical of full-scale commercial aeroplanes into a detectable feature that characterises the overall compressible flow. Nevertheless, by the time his work was published a study into the phenomenon as a means to validate analytical, experimental and/or numerical compressible data was not yet available [36].

Fisher et al. [16] then reported an analysis of the more precise conditions necessary for the natural shadowgraph manifestation in flight, given the air density change across shock waves present over aircraft wings and the consequential sunlight refraction, as represented in the schematic of Figure 1.13. This work, built on the research of Rathert and Cooper [34], evaluated the shockshadow formation from a geometrical perspective, relating the aircraft attitude and relative sun angles with respect to its wings to the visualisation of the phenomenon, thereby confirming the earlier findings from [34].

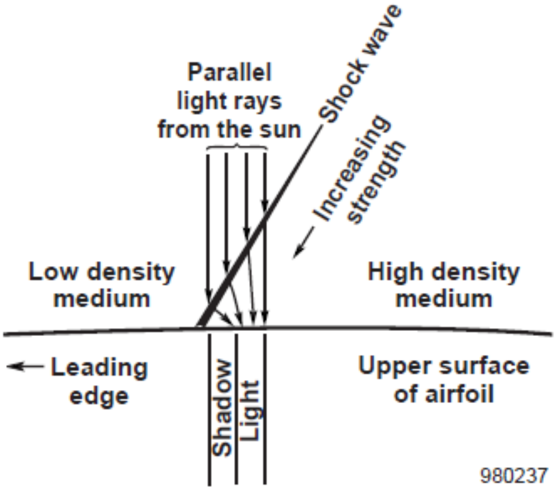


Figure 1.13: Underlying principle of shockshadow formation in aircraft wings. Image reproduced from [16].

They performed a couple of flights during which a transonic and high-altitude commercial aircraft, having swept as opposed to straight wings as in [34], executed banked turning manoeuvres. Simultaneously, an instrumentation system recorded flight data (e.g. time, Mach number, altitude, aircraft attitude and geopositioning) and photographs and videos were recorded during a limited number of shockshadow observations. This data was then used to determine the orientation of the sun with respect to the aircraft wing, i.e. sun elevation angle above the wing and azimuth angle relative to the wing quarter chordline, when the shockshadows were visible (see Figure 1.14).

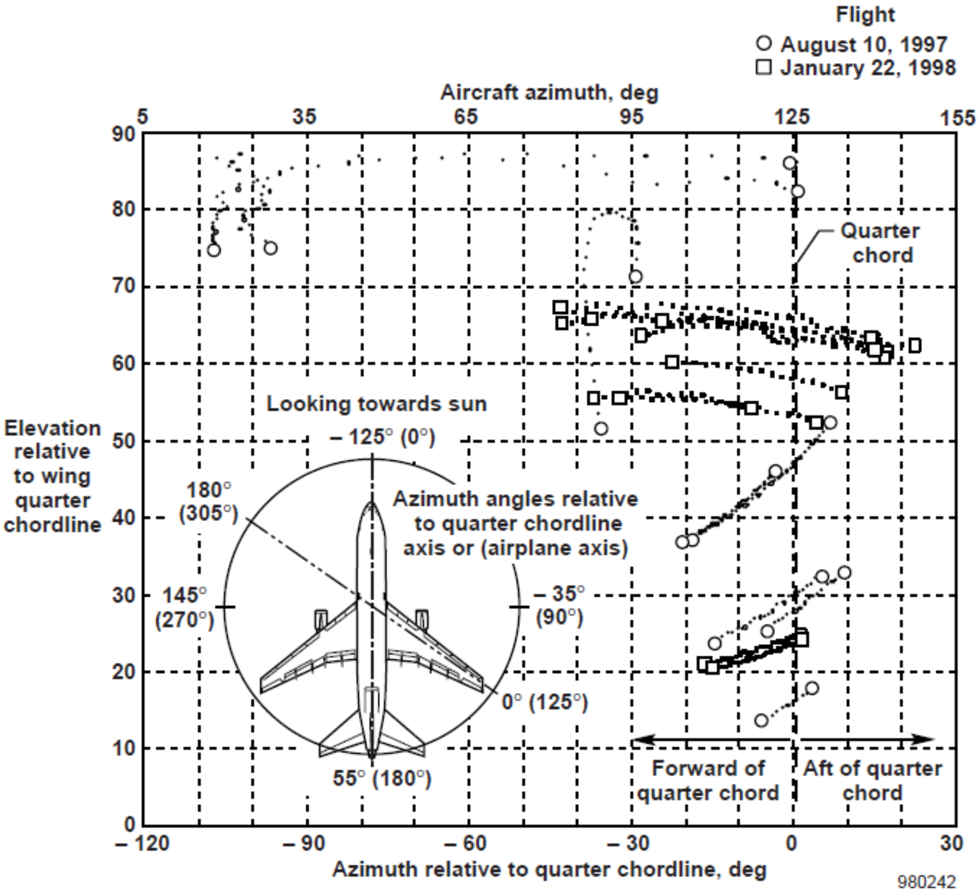
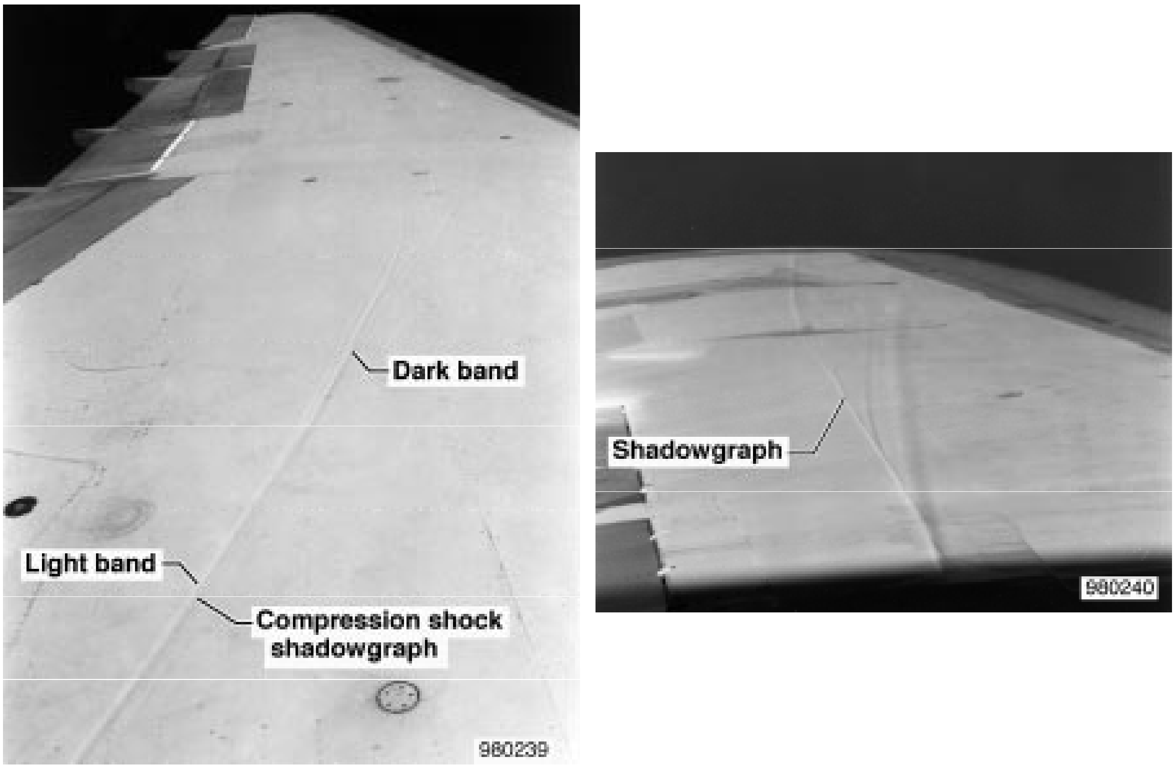


Figure 1.14: Sun azimuth and elevation angles, relative to the quarter chordline of an aircraft wing, required for shockshadow visualisation when looking towards the sun. Image reproduced from [16].

Using the swept wing quarter chordline as a reference, they determined that at low, very oblique, sun elevation angles above the wing the shadowgram was only briefly visible throughout the turns when the sun was nearly aligned with the reference axis, whether in front of or behind the observer. In these cases, they did not consider the effect to be an accurate representation of the shock wave location. Rather it was a projection of the shock portion near the wing tip as a shadowgraph at the inboard region of the wing, later also visualised as a background oriented schlieren object, or an image distortion against the background of scattered clouds. At elevation angles higher than $60°$, looking both towards and away from the sun (as seen in Figure 1.15), the shadowgram appearance was more consistent and the higher the angle the less important was the azimuthal orientation. In essence, they have concluded that bright sunlight and high sun angles were optimal in terms of shockshadow visualisation [16].



(a) (b)

Figure 1.15: Natural shadowgraph recorded in-flight when looking towards (a) and away from (b) the sun. Image reproduced from [16].

Lokos et al. [37] reported a clear link between an abrupt and substantial change in the hinge moments of both aileron and trailing-edge flap of an aeroelastic research aircraft and the shock wave translation over the respective hinge lines. The moments were measured through appropriate load and deflection detection instruments. The movement of the shock wave, or rather three different shock waves, was simultaneously observed as moving shadow patterns, recorded in real-time by onboard video cameras. Due to the transonic flight and ideal natural light conditions, sunlight shadowgraphs were occasionally captured, as shown in the single frame Figure 1.16. Similar to the earlier suggestion in [34], the detection of these shockshadows and measurement of these hinge moment variations again could be used as an indicative of the relationship between the periodic motion of shock waves and the onset of or contribution to aeroelastic instabilities and related unsteady loads on the vehicle.

Tauer et al. [38] have investigated further this relationship. In flight tests with again fighter aircraft, they painted the outboard region of the wing with a white glossy paint (see Figure 1.17), determined as the region of interest through computational aeroelastic and aerodynamic analysis. Video cameras were directed towards this region and tufts were also installed, the camera in the fuselage near the canopy responsible for recording airborne shadowgraphs. The

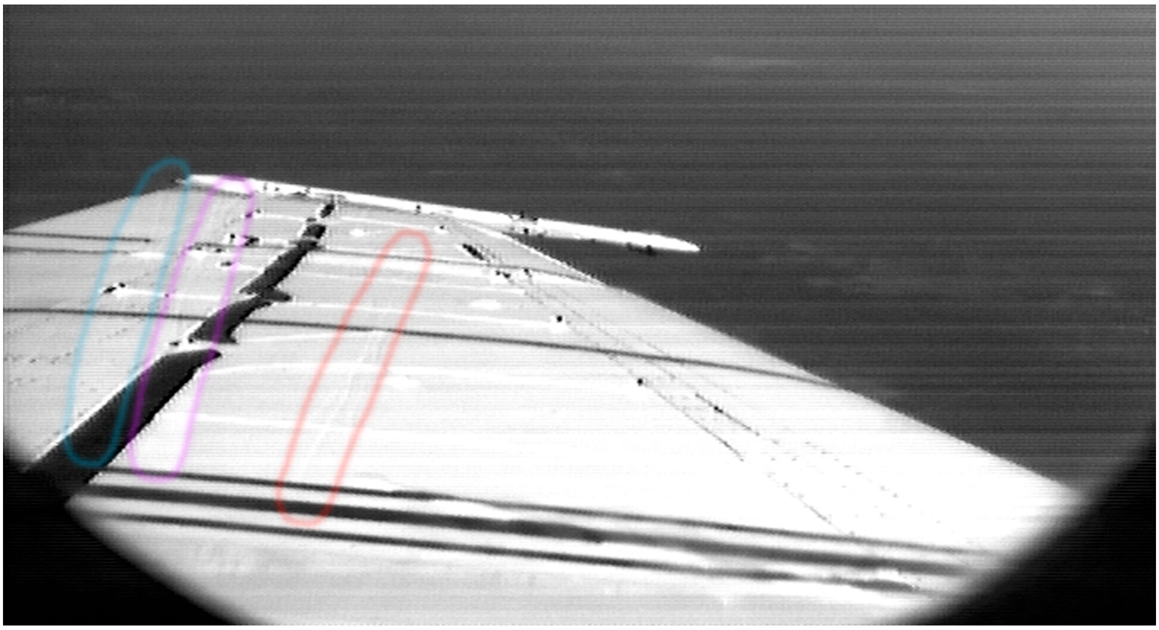


Figure 1.16: Sunlight shadowgraphs at three distinct locations and respective flow regions recorded during aeroelastic flight tests. Image reproduced from [37].

aircraft performed manoeuvres in order to establish the optimal orientation with respect to the sun, according to the analysis of Fisher et al. [16]. Their major interest was the determination, via in-flight flow visualisation, of the role of shock waves and shock wave-induced trailing-edge flow separation in the development of limit-cycle oscillations (LCOs), a type of unsteady flow phenomena. However, the tufts remained directed towards the trailing edge, indicating no flow separation. Although the obtained shadowgrams exhibited a near normal shock wave on the wing upper surface, and this was observed during LCO events, it did not move during the aeroelastic response of the wing. Shadowgraphs projected on the surface of the wing, such as the one presented in Figure 1.17, revealed the shock wave and these observations were corroborated by BOS images using a solar (panel) farm on the ground as the reference. Therefore, the presence of a shock wave could not be linked to flow separation and its observed unsteadiness in some scenarios was rather correlated to the mere alteration of the flight conditions [38].

1.5 Computational Flow Imaging

As Havener [39] once described, the ’science of generating digital images of theoretical fluid dynamic phenomena in formats that mimic optical observations of real flow fields’ [39], or simply Computational Flow Imaging (CFI), has been evolving since the traditional flow visualisation experimental methods became well established within the research community. Inspired by these experimental techniques, CFI relies on several computational models, from the flow field to the visualisation and measurement processes and the final goal of digital image synthesis.

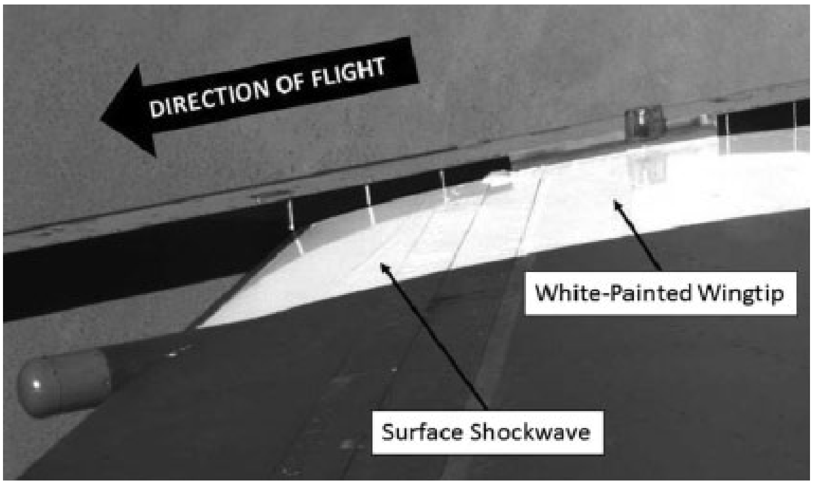


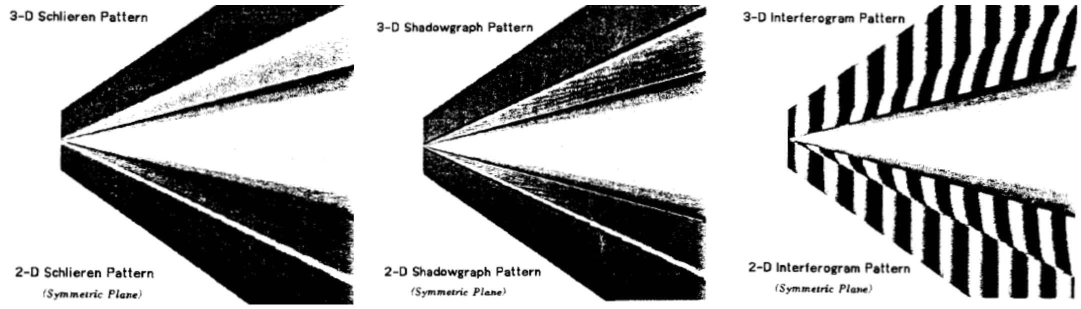
Figure 1.17: Airborne shadowgraphs on the wing surface recorded during aeroelastic flight tests. Image reproduced from [38].

The flow prediction is obtained either analytically from pure fluid mechanics or aerodynamics theory or numerically from Computational Fluid Dynamics (CFD) simulations. The reproduction of the visualisation process is based on the optics field of study. It either uses approximated Geometrical Optics where the 'wavelength of light is small in comparison to' [13] the dimensions of the media that light traverses or the more comprehensive and precise Physical Optics where the 'wave nature' [13] of light is considered. Through these fields, the light scattering effects that may occur within the predicted flow field and upon interaction with other elements present in the scene are covered. The generation and processing of digital images then complements the former models by simulating light behaviour inside the recording apparatus.

CFI offers many advantages that are not (as) easily attainable by any other means of flow visualisation or related engineering. Beyond the opportunity to observe invisible phenomena within a simulated flow, the behaviour and characteristics of flow features can be analysed, and thus experiments can be planned and the observed results and measurements can be assessed. CFI-Schlieren, CFI-Shadowgraphy and CFI-Interferometry all benefit from the latter. The whole experiment is designed and built virtually, thus creating a modifiable environment where: the techniques are optimised; their expected images investigated; and the flow field itself viewed from different perspectives. Another strength of CFI is the versatility of digital image processing, including options of computer zooming, perspective viewing, domain isolation and picture treatment. Moreover, in line with ongoing advancements in high-performance computing, CFI is currently still being developed to provide an alternative to CFD post processing and to be able to validate CFD codes via comparison with analogous experimental data. This is especially useful in the case of modern solvers that treat complex fluid flow problems and still lack a comprehensive amount of empirical information for comparison [39].

Havener [39] also explained that one of the conditions that the computational simulation has to satisfy in order to generate an image that can be considered a 'true representation of nature' [39] is that all of the aspects involved in the corresponding real image formation have to be adequately addressed in the models [39]. Tamura and Fujii [40] reinforced this point when describing the inadequate comparison often made between contour plots of computed scalar results, such as surface plots of air density stemming from CFD, and schlieren images or shadowgraphs. Shadowgraphs of two-dimensional flows, for example, do not depict 2D slices of the density field itself, but rather of the second derivative of density. If the flow is three-dimensional, the relationship between the 2D image on the recording plane or screen and the density distribution, or the result of the interaction between light and fluid flow, is even more complicated. Adding to this complexity, in real or simulated shockshadows the images are unconstrained, essentially visualising the whole 3D flow field and recording further 3D light interactions, including reflection from a surface (e.g. wing) and additional refractions during multiple traverses through the flow. Tamura and Fujii [40] have therefore stated that it is necessary to simulate the visualisation processes in place in order to compare computation with experiment [40]. Although some similarities exist between the experimental images and the standard computationally produced plots, the principles of image formation are not the same and the features depicted often represent different flow characteristics. A true accurate comparison between computational and experimental data then still requires the virtual modelling of the real experiment and respective visualisation and optical processes. Only under such conditions can the validation of the simulations, either CFD or CFI, be reliable and CFI be considered an accurate and useful postprocessing tool of CFD, enabling multiple viewings of the solution, possible interactivity and digital data extraction [40].

Tamura and Fujii [40] took advantage of the newly available graphics workstations to construct simulations of schlieren, shadowgraphy and interferogram images using libraries containing some of the classical computer graphics (CG) algorithms, in parallel to the early advancements seen in CFD due to the development of supercomputers. Based on two and three-dimensional supersonic CFD solutions concerning internal flow in an intake and external flow over an axis-aligned cone (see Figures 1.18), the analogous images were generated by shooting and tracing light rays from the pixels that compose the rendering window towards the compressible flow domain. The light intensity of the respective pictures were calculated by numerically integrating along the path of each light ray. Either through the density field itself, available in the CFD solution, or the numerically evaluated first derivatives of the same, according to the fundamental principles of the optical techniques. The original structured CFD grid was converted into a fully tetrahedral mesh and the integration was simplified by finding the faces of each tetrahedron intersected by a linear light ray and computing the resulting intensity from the respective face values, assuming a linear variation of gradients inside the cell, i.e. the trapezoidal rule. Aiming for computational efficiency, only some of the pixels were selected to trace light rays from and the intensity was simply interpolated amongst the remaining pixels [40].



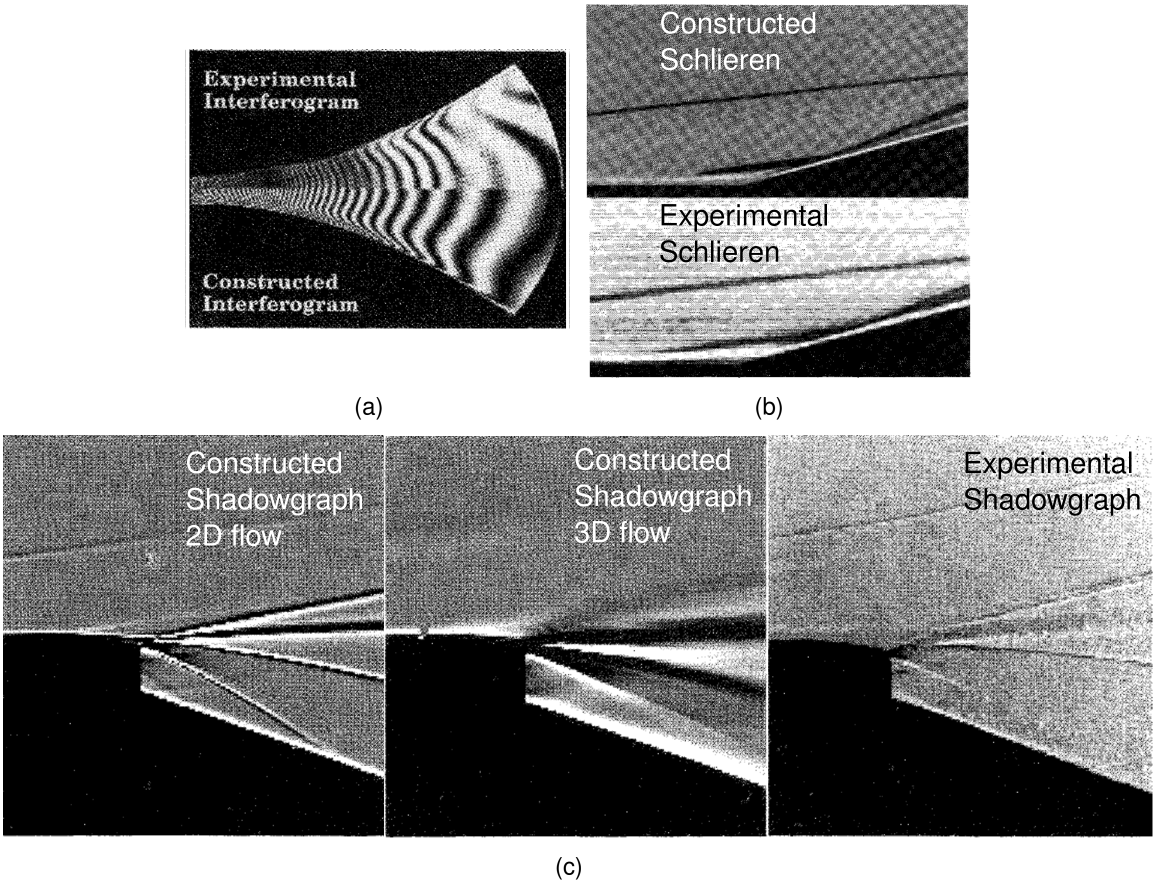
(a) (b) (c)

Figure 1.18: Simulated (a) schlieren, (b) shadowgraph and (c) interferometry images of a cone in supersonic flow. Images reproduced from [40].

The schlieren image of the CFD solution of the 2D intake demonstrated the differences between contour density surface plots and CFI. The schlieren images of the cone test case (Figure 1.18(a)) in turn revealed that 3D effects can play a considerable role in image formation, since the dimensional reduction resulted in some loss of details. More specifically, the region of density gradient between the oblique shock wave and the solid body was present within 3D CFD but not captured when 2D CFD was used. In contrast, on the shadowgraphs (Figure 1.18(b)), calculated as the divergence of the schlieren quantities, these differences were not as apparent due to the shadowgraphy sensitivity to variation of gradients of density or second derivatives. In the interferometry (Figure 1.18(c)), the 3D simulation even presented a stronger numerical error, originated and propagated during integration and interpolation operations, and some information may have been lost due to these numerical averaging processes. A shadowgraph and an interferogram of the 2D intake field were also presented. All these computationally simulated images enabled not only direct comparison with their experimental versions, but they also illuminated some flow features (e.g. shock and expansion waves) that are usually not well resolved in other forms of CFI, like plots of contours, velocity vectors or streamlines. In general, they have contributed to the 'personal graphics' [40] approach to CFD postprocessing, where the physics of the computed flow field and its interpretation are more relevant than photorealism and simple illustration [40].

Yates [41] followed a similar approach to the one adopted in [40] by integrating functions along the lines of sight to also construct computational schlieren, shadowgraph and interferogram images from simulated flow fields, as shown in Figure 1.19. At the time of the study, the still existing limitations in terms of computational resources prevented the appropriate tracing of nonlinear light rays by solving the appropriate governing equation. Instead minimal deflections through the inhomogeneous index-of-refraction field and a straight traversal perpendicular to the image plane were assumed. These assumptions were said to have no effect in discontinuity free regions of the flow and, in the presence of shock waves they introduced errors of lower order than the errors already present in the underlying CFD solutions and the experimental results to be used for comparison. The error introduced by this approximation was analysed analytically. Functions of the refractive index of air, instead of density, were then evaluated employing the Gladstone-Dale relation and integrated again using the trapezoidal rule within the grid cells and interpolation on the cell faces, where the values were averaged from the vertex data. Light intensity patterns formed on the image plane were then related to these integrals. The CFD and respective CFI mesh, however, were not restricted to tetrahedra. The integration was performed for all of the pixels in the image plane, avoiding accumulation of error due to numerical interpolation, and light was projected on the image plane for all types of simulations [41].

A simulated interferogram of a two-dimensional internal real gas flow in a hypersonic nozzle (Figure 1.19(a)) has proved to be easily compared against the experimental holographic technique. When both images were visualised side by side, the differences in the positions, shapes and orientations of the fringes became readily evident [41]. These differences were attributed to the discrepancies in the boundary layer thicknesses and density distributions of the simulation and experiment. Regarding a similar image of an ideal gas, but for a three-dimensional external flow over a ballistic model, the comparison with the experimental equivalent was not possible due to differences between the modelling and testing conditions. However, both images presented similar fringes. On the schlieren of a 2D compression corner (Figure 1.19(b)) the presence of shock wave, boundary layer and shear layer related to flow separation were observable and compatible in both computational and experimental images. The dark and light visual features were said to be prone to sharpness improvement upon the use of grid adaption during the CFD simulation. Similarly, on the shadowgraph of a 2D supersonic expansion ramp nozzle (Figure 1.19(c)) all of the features present in the experimental image, including shock waves, flow separation and expansion fan were also observed in the simulation. Although not exactly matching the experimental setup, the image plane position was adjusted to optimise its exposure and the consequent final visual aspect of the existing flow features. Concerning the same test case, 3D CFD solutions simulated in coarser levels of mesh refinement were also used in the image generation to analyse the effects of grid resolution and solution convergence (Figure 1.19(c)). As expected, the poorly resolved flow features (shock waves, shear layers and expansion fans) in the CFD resulted in blurred, diffused or even missing image features in the CFI [41], since 'the quality of the flowfield solution does affect the quality of the constructed images' [41].



(a) (b)

(c)

Figure 1.19: Computationally constructed and experimental (a) interferometry images of a hypersonic nozzle flow; (b) schlieren images of a compression corner, and; (c) shadowgraph images of a supersonic expansion ramp nozzle flow. Images reproduced from [42].

Samtaney et al. [43] exploited the common effects that an inhomogeneous flow field has over the light behaviour from the perspective of shock-capturing techniques typically used in CFD to capture and with grid refinement and/or mesh adaptation appropriately resolve the discontinuities in the flow solution. Again emulating the principles of traditional experimental flow visualisation methods but not exactly modelling the experiments per se, they have only treated mathematically the computed density field pertaining to a time step solution of an unsteady two-dimensional shock wave and contact interaction within a rectangular shock tube. Using the so-called filtering functions, which are well known numerical edge detection techniques available in the literature, the density jumps, gradient and Laplacian were estimated and used to compose the interferograms, schlieren images and shadowgraphs, respectively. These are effectively alternative ways to visualise the density field and, because all of the numerical derivation involved, the final images were highly susceptible to noise or numerical error. These errors were partially suppressed via image smoothing operators, such as a neighbour weighted averaging process and a Gaussian filter [43].

Brown et al. [44] proposed a completely different approach to the simulation of flow visualisation techniques, based entirely on physical optics rather than on the geometrical optics principles of ray tracing and integration of refractive index along the line of sight. By removing the constraining approximation on the wavelength of light, which invariably causes a loss of information regarding the shape of objects in the medium, effects that solid boundaries exert over light were also included in the modelling. Therefore, the diffractive as well as the refractive nature of light was captured in the wave propagation algorithm developed according to the scalar wave theory. The algorithm effectively projected, via space marching the paraxial wave equation, an optical light wave represented as a two-dimensional pixel grid through density fields obtained from CFD simulations. At each step, 2D Fast Fourier Transforms and the respective inverses were computed to retrieve the modulated light amplitude and phase, thereby plotted as a gray-scale map to create the numerical analogues of the experimental methods. In an initial test case for supersonic Prandtl-Meyer flow over a 2D expansion corner, the authors have shown on intensity plots of a shadowgraph (Figure 1.20(a)) and an interferogram (Figure 1.20(b)) the effects that the step size have on the final images. Specifically they have shown the appearance of Moire interference patterns when using a small computational resolution, and the dominance of diffraction effects over refraction in some regions of the domain. Moreover, on synthetic schlieren images (Figure 1.20(c)) they have tested different filters in the Fourier plane. This demonstrated not only that an appropriate choice of filter enables the reproduction of a normal schlieren-like image, but also the ability to separate image effects and render a final image where certain desired flow features, structures or interactions are isolated and enhanced. Due to the linearity property of a Fourier transform, the background can be removed from the schlieren images representing the expansion regions.

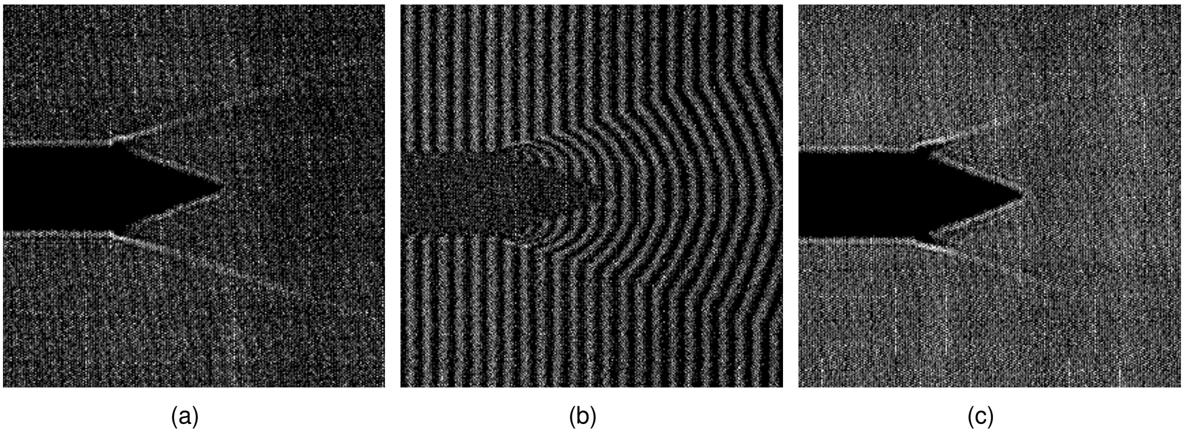

(a) (b) (c)

Figure 1.20: Synthetic (a) shadowgraph, (b) finite fringe interferogram and (c) schlieren (without filter) images of a supersonic Prandtl-Meyer flow. Images reproduced from [44].

Sun [45] innovatively avoided assumptions, for example of linearity, when employing the geometrical optics ray tracing technique to simulate light within a wind tunnel experiment model. In their model, light emitted from a spherical source was considered to traverse through: the test section containing the discontinuous flow field; the homogeneous medium; and all optical components of a typical focused shadowgraphy configuration outside of the section. Light refraction in the inhomogeneous medium was captured via Snell's Law of Refraction applied across the boundaries of the cells of the underlying structured grid. The method considered a face-based refractive index solution converted from density. An inviscid two-dimensional axially-symmetric CFD case of point explosion containing a shock wave and a contact surface was simulated by a proprietary solver. This used a gradient-based mesh refinement to resolve the flow features and speed up the ray tracing process, and was subsequently used to construct a pseudo-three-dimensional solution. The light intensity on an image screen was then mapped according to the effects of light refraction on the area illuminated by each light beam or set of rays, as shown in Figure 1.21 [45].

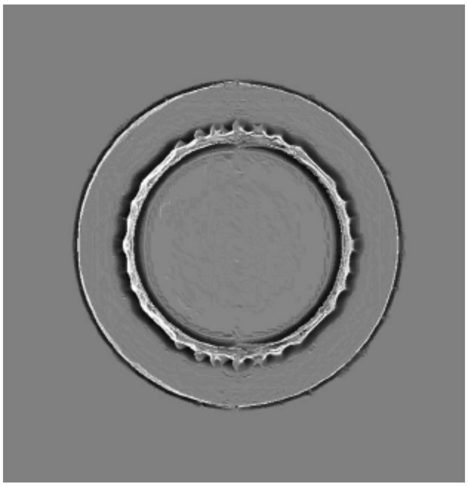
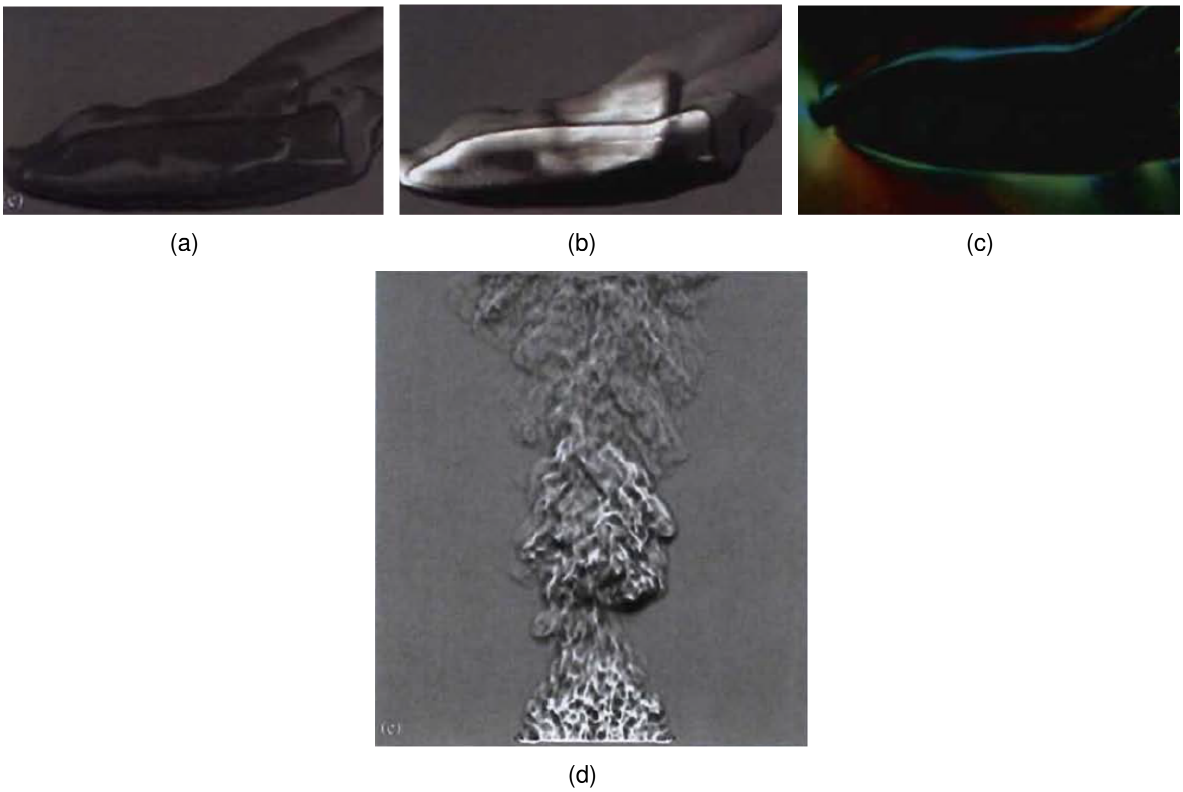


Figure 1.21: Computed shadowgraph of axially-symmetric point explosion. Image reproduced from [45].

The effects of the emitted light source direction were analysed and revealed a preference in terms of angle of incidence in the test section in order to achieve sharp image features related to the shock wave, i.e. the outer shadow. The light source size was also evaluated, indicating the appearance of geometric blur at the boundaries the more it deviated from an idealised point light source. Finally, the position of the recording system was also varied to check how it affects the shadowgraphy sensitivity, a well-known characteristic of the technique. The author therefore suggested that a complete replication of an optical system for flow visualisation experiments allows a researcher, experienced or not, to tune the optical setup without having to run any form of real experiment, which usually is a time intensive and financially expensive activity [45].

Brownlee et al. [46] removed the remaining assumption that refraction happens only through discrete surfaces, still embedded in the employment of the Snell's Law to capture refraction during ray tracing. This is not the case for some compressible flow features which actually form a continuous refractive index distribution over a certain region. This is especially true for discontinuities captured in CFD simulations, where for example a shock wave is usually spread over multiple grid cells that upon refinement tend to cover a smaller region. However, this shock surface is still nonetheless not a discrete surface. The authors therefore decided to utilise a piecewise linear approximation to the ray equation of geometrical optics, derived from Fermat's principle, to retrieve the curved paths traced by light rays through the inhomogeneous medium. Using this approximation together with an octree acceleration data structure, ray tracing became considerably more accurate and faster. This was mainly due to the step size of the integration automatically adapting to the gradients in the refractive index medium, evaluated a priori on the Central Processing Unit (CPU), while the whole ray casting algorithm was run on the Graphics Processing Unit (GPU). In order to synthesise numerical analogues of the flow visualisation experimental techniques, light rays were emitted in parallel along a grid (the light source) and nonlinearly traced throughout the domain. Once these rays reached the film plane they were accumulated as photons to yield the intensity values at the images. These intensities were subsequently filtered using a cone filter, inspired by the photon mapping technique, to reduce noise in the final rendering.

Due to the high level of computational efficiency achieved with this approach, they were able to generate several physically-based shadowgraphs and schlieren images with impressive sample sizes and at interactive and reasonably fast frame and convergence rates, providing researchers of the field with a 'familiar and intuitive visualisation' [46] with 'additional degrees of control' [46]. Data sets concerning a combustion scenario, fire emanating from coal, a helium plume and an aircraft on atmospheric re-entry were used for the shadowgraphs and schlieren images. In the helium plume case (Figure 1.22(d)), the proposed method was compared against a traditional volume rendering technique and an approximation of schlieren imaging without considering refraction. Using the ray equation, the images provided better results in terms of more information about the flow structure and enhancement of respective features. In the aircraft case (Figure 1.22(a),(b),(c)), a colour filter was also used and has shown more information than a simpler gray-scale image, for example even the formation of vortices. Lastly, on the combustion case the beneficial effect of using image filters was demonstrated, where the level of noise was considerably reduced when small sample sizes were used to render the image [46].



(a) (b) (c)

(d)

Figure 1.22: Computational (a) shadowgraph, (b) gray-scale schlieren and (c) colour schlieren images of an aircraft on atmospheric re-entry; and (d) computational schlieren image of an helium plume. Images reproduced from [46].

1.6 Engineering Relevance of Shockshadows

Although researchers have now acquired more than a century of knowledge on aerodynamic flow, including theoretical and experimental expertise, compressibility still presents a major challenge to aeroplane designers. Shock waves invariably appear when a fluid flow approaches the supersonic regime, whether it is within an engine intake, over a wing or even surrounding a whole aircraft. For the shock wave, several effects that impact the overall flight performance are initiated by its interaction with other flow features, such as boundary layers, wakes, vortices and neighbouring shock waves.

The appearance of shock waves, development of boundary layers, flow transition from laminar to turbulent, behaviour of turbulent structures and flow separation are all examples of viscous phenomena associated with the drag of vehicles. The shock wave system itself, due to the production of entropy, is in fact a source of drag in high-speed flight. The high gradients in pressure, temperature and velocity across a shock wave can induce local structural loads, intense heat transfer, loss of aerodynamic and/or propulsive efficiency, noise or even aeroelastic instabilities. In modern transport aircraft the formation of shock waves over the wing causes a rise not only in wave drag, but mainly in friction drag due to the interaction between the shock wave and the boundary layer. The separation of the boundary layer is also sometimes induced by the shock wave, creating an aerodynamic excitation that can strike other parts of the aeroplane, e.g. the tail, causing vibrations or buffet that can alter the flight stability [47].

All of these flow features demand high quality and thorough investigations in Aerospace, due to their intrinsic complexity and relevance in high-speed aerodynamics. In the transonic regime, where the operational envelope of modern commercial aircraft stand, the goals are to maximise the speed, maintain a certain standard of passenger comfort and reduce the fuel consumption [47]. These objectives inevitably rely on the characteristics of shock waves resultant from the aerodynamic design of vehicles for high-speed flight. In fact, this class of aerodynamics 'remains as a frontier to many fundamental questions' [48] about compressible flow and its investigation is considerably more challenging in comparison with the subsonic and supersonic regimes [49]. Since the second half of the twentieth century, wind tunnels [50] and experimental techniques (see Section 1.3) have thus been built specifically towards the understanding of the complex flow physics related to this regime [48]. However, there are still several limitations to the state of knowledge of transonic flow and to the capabilities of investigating flows within this regime [48, 49].

Regarding transonic industrial wind tunnel testing specifically, one of the major challenges to optical flow visualisation is the inherently ventilated test sections of all facilities. In these installations, the walls of the test section are not equipped with windows, but are rather perforated with specifically shaped and distributed holes or slots to prevent choking at this location [48, 50, 51]. Choking is the rise of local supersonic flows due to the obstruction effect caused by the model itself, which renders the wind tunnel inoperable and/or compromises the tests reliability, since the sonic regions can't be altered by changes in the freestream conditions and the model therefore becomes subject to velocities different than initially intended [49]. Cameras and lights could in principle be mounted directed towards these slots, but this solution is far from ideal, as the characteristics of testing equipment and models can greatly vary and thus the required viewport positions change [51]. The original designs of these facilities are therefore not adequate for optical access to the model region [48, 52], a practical aspect difficult to circumvent.

As exemplified in [48], the sometimes feasible modification of the test section to accommodate optical testing equipment would then involve not only the installation of special windows suitable to the harsh testing environment, involving high temperatures and pressure gradients, but also: the re-design of the wall ventilation; the analysis of shock/expansion wave(s) generated by the new wind tunnel section mounted to facilitate the experiment; the evaluation of sources of acoustic noise, drag and interference; and the possible adaptation of commercial instruments systems to the re-designed wind tunnel. Besides, these modifications require physical access to the test section, a complicated task by itself since sensitive devices would have to be moved through a very constricted space [51]. These factors then invariably convert to costs, of not only acquiring the equipment, modifying the installation, setting up the experiment and performing the actual measurements, but especially related to the time spent in these activities, a factor imperative to the success of any commercial facility [51]. The result is that these optical diagnostics techniques are usually expensive from a 'price-per-data-point perspective' [51].

Apart from the intellectual curiosity raised by a seemingly impossible shadow of an invisible entity cast by the sun [1], the shockshadow phenomenon therefore presents itself as a possible alternative method to the study of transonic and compressible flow in general [15]. Knowledge of the conditions under which the shockshadow becomes a visible element may lead to the conception of a technique that exploits the pattern as an additional source of information about the flow feature(s) that originated it [15]. This new information can then be used to improve old or possibly provide completely new insights into the physical processes of compressible flow, the definition of the role of a fluid flow visualisation technique [22].

Understanding, for example, the aspects of flight operation, including aircraft attitude, high-speed aerodynamic flow and atmospheric conditions that result in the formation of a shockshadow would enable researchers to predict its observation. The consideration of the characteristics of light emitted by the sun or perhaps from another (more) suitable source of illumination, e.g. lasers, would also allow the design of experiments around this particular phenomenon and possibly the further evaluation of its properties. Moreover, the assessment of the best scattering properties of a general wing or any other surface of projection, and the required specifications of a camera system necessary to record clear patterns, could possibly determine the creation of entirely new strategies to visualise compressibility phenomena [1]. However, at the time of writing this research, a systematic application of shadows and caustics generated by shock waves for the previously mentioned purposes of compressibility study has not yet been reported [36].

Sunlight shadowgraphy can indeed portray a single or a whole system of shock waves on surfaces of aircraft in the transonic flight regime and provide indication of their locations [36, 38]. As the influence of shock wave systems on the aerodynamic performance of these surfaces is of major importance, the plain detection of these flow features is classified as a practical representation of that flow [36]. Apart from flow visualisation and feature detection, the method has several other possible applications as a testing technique [34]. These include: tracking, measurement and characterisation of the shock wave system; detection of shock wave-induced flow separation; monitoring of aeroelastic instabilities through the shock wave oscillatory behaviour [34]; validation of analytical solutions; verification and validation of computational fluid dynamics solvers; and wind tunnel/flight testing correlation or validation of experiments [36]. These applications of the scattering of light [13, 22] can even be further elaborated from different perspectives other than direct shadowgraphy. If the wing surface or possibly the sky on a cloudy day, for example, is set up as an appropriately defined background, a background-distortion type of shadowgraph can also be explored for shock wave location. Other surfaces, for example the side of a fuselage or even a plate mounted either at inboard or outboard wing stations or somewhere within a wind tunnel test section, may also provide a feasible extension of the method [34]. This extension could then be employed to study the interaction of shock waves with boundary layers [22].

An obvious advantage of the shockshadow approach lies on its simplicity [34, 36], notably in comparison to other compressible flow optical visualisation techniques, given that the phenomenon can be achieved naturally. Therefore, it does not require: any specialised and complex devices [36], as for example the lasers, lenses, mirrors and screens frequently used with the traditional techniques (see Section 1.3); the introduction of mechanical probes or measurement instruments that invariably interfere with the fluid flow [22]; nor the addition of foreign materials to indirectly observe and evaluate the flow features, such as the phosphorescent/colored oils applied to models to reveal surface flow patterns [53], fluorescent paints used to measure surface flow properties (e.g. pressure and temperature) [53], or particles (e.g. smoke) seeded in the fluid to represent the flow behaviour [22]. Because of that, sunlight shadowgraphy is a suitable technique for applications outside of the limited laboratory environment of wind tunnel facilities, and it can be easily integrated into existing flight test campaigns of full-scale vehicles [36].

As a new method of fluid flow visualisation, the shockshadow is thus an optical, direct and nonintrusive technique that records surface visual patterns related to off-surface compressible flow features [22]. The method relies on the interaction of a light wave with a fluid flow and constitutes a compromise between the two approaches commonly used for obtaining information about this interaction, as described by Merzkirch [22]. From the perspective of computational flow imaging, a virtual shockshadow pattern would still be retrieved via integration along the path of the light transmitted through the compressible flow and respective refractive domain [22], similar to the principles of image formation in shadowgraphy and schlieren imaging. This indicates that, as with the classical techniques, qualitative and semiquantitative (e.g. shock location) data could be directly obtained from the shockshadow images given the effects of flow features on light. These features are not limited to shock waves but include others such as boundary layers and vortical structures, and the resulting data is in the form of two-dimensional slices of their structures, possibly delimiting their instantaneous positions at the time the picture was taken.

However, different from the traditional methods, the comparison with the undisturbed state of the emitted light would not be necessary, since in principle the pattern can be observed in its raw form (without image manipulation). More importantly, the disturbed light is not received on a two-dimensional screen or recording plane. Instead, the shockshadow is rather a record of the light scattered from the three-dimensional wing surface itself, i.e. in a certain position in space and in a certain direction, after multiple passages through the fluid flow [22]. This aspect is then comparable to the principles of image formation of Particle Image Velocimetry (PIV) and Laser Doppler Anemometry (LDA). These are both mature flow diagnostics techniques, with multiple commercial systems available, which measure local information about the seeded flow. The real quantitative information provided by the developments of these methods includes, apart from 2D velocity fields, the instantaneous three orthogonal components of flow velocity in a point (LDA), within a plane (stereoscopic PIV) or within a bounded volume (holographic PIV) [52]. Although widely adopted, these laser-based techniques are nevertheless still 2D in essence, relying heavily on complex optical systems, high-speed cameras and data processing algorithms. Moreover, they are always subject to the fundamental requirement of employing seeding particles small enough to faithfully represent the fluid motion, and the consequent errors associated with the finite size of real particles, and with the strong variation in the thermodynamic state and unsteadiness related to compressible flow features [22, 48].

Images of shockshadows, whether real photographs or computational simulations of the phenomenon, are essentially unconstrained two-dimensional representations of the full three-dimensional physical flow, the main characteristic that differentiates its application as a fluid flow visualisation technique from other classic or even modern methods. Based only on pure Optics, the phenomenon therefore opens the opportunity to the observation of compressible flow in multiple environments, requiring nothing but appropriate illumination and visualisation conditions, from different perspectives around the flow features, and with a strong potential to establish a new useful flow diagnostics tool.

1.7 Background to this Research

Thus far in the aerospace engineering literature there still exists a gap in the investigation of the shockshadow formation over wings of aircraft in high-speed flight. Although the physics behind natural direct shadowgraphy is well-known and the phenomenon has been purposefully analysed in flight under some circumstances in a few studies [16, 34, 36–38], it has not yet been thoroughly explored from either an experimental (wind tunnel or even flight test) nor a computational (CFI) perspective. Associated to the replication of shockshadows both in real laboratory and virtual simulation environments, rigorous experiments are yet to be devised around such a specific phenomenon, and all the exact conditions (apart from light source orientation) under which the shockshadow becomes a visible element are still unknown. The question if the visual pattern can be practically exploited to retrieve useful information, qualitative and possibly also quantitative, about the compressible flow and respective feature(s) that originated it is also still to be answered. If its formation could be fully predicted or artificially induced, its characteristics could be potentially correlated to some of the aspects of shock waves and thereby used to validate data from other sources.

Previous studies in CFI [40, 42–45, 54] successfully generated numerical analogues of shadowgraphs and schlieren images, however the implemented algorithms were specifically tailored for the simulation of these traditional flow visualisation techniques. Further, the modelling performed in these studies involved too many assumptions, including: 2D fluid flow; minimal deflections of light and straight traversal through the domain; specific variation of the density solution within the CFD mesh; lack of fidelity with the experiment itself, but rather only a postprocess of the CFD solution; light refraction captured through Snell's Law across discrete boundaries of the mesh; an approximate relationship between light intensity and line-of-sight integrals; no light reflection and pixel interpolation. The work of Brownlee et al. [46] was relatively more complete, as some of these assumptions were removed and the mathematical equation that governs light propagation was solved. It also comprised a more modern approach, since they took full advantage of available computer graphics resources. However, they still used a piecewise linear approximation of light rays and the method was only suitable for schlieren and shadowgraphy image simulation. In summary, these assumptions moved the simulations away from the physical reality of the flow visualisation experiments.

In truth, three-dimensional effects invariably influence the behaviour of light within the flow field and, although light indeed deflects minimally, the deviation from its original direction is still significant and one of the main contributors to the visual pattern formation. Because the shock wave itself typically has a 3D shape, light refraction when traversing across its complex internal structure is hardly a discrete process. Particularly in CFD, this discontinuity is usually captured as an inhomogeneous density solution diffused over multiple mesh cells rather than a discrete boundary, rendering it even difficult to precisely predict the density distribution. For these reasons, it is important to actually solve the mathematical equation that governs light propagation, in order to appropriately model refraction. This is also valid for light reflection, which inevitably occurs, for example, from the test model, during the photograph recording itself (from screen to camera), or from other artificial sources accidentally affecting the final image. It is therefore essential that both fluid flow and light behaviours computed are faithful representations of the phenomena taking place in the laboratory environment, that the mathematical modelling is suitable for this task, and that the computational simulations are as accurate as possible. Otherwise, the synthetic images become only simple alternatives to visually assess the results from flow prediction and not true numerical analogues of shadowgraphs and schlieren images.

The work of Onnink [55] concerned an initial attempt of actually simulating shock wave shadows, instead of the more common flow visualisation techniques. He also attempted to establish a relationship between the visual effect and the compressible flow feature, as it was postulated that the location of the shadow pattern along the wing could reveal the position of the shock wave and its illumination intensity could in theory be associated with the strength of the shock wave. Similar to Brownlee et al. [46], he solved the light propagation governing equation, but rather via numerical integration. However, all of the remaining models involved in the image simulation were incomplete, unrealistic and too prone to inaccuracies. Therefore the methodology was found unfeasible to be applied as a practical tool from which quantitative information could be extracted. These model limitations were mainly: the oversimplified 2D compressible flow field, which in turn produced effectively 1D computational shockshadow images; the disregard of the aerofoil surface curvature; the inaccurate light energy quantification; the lack of a reflective surface model and the absence of a viewing system in the process. The specifications of a device where the image is generated, e.g. a camera or an eye, were thus not taken into account. The wing surface geometrical shape and its light scattering characteristics were also not included via a light reflection model that would redirect light towards the viewing system and cause a second refraction during this traversal. Moreover, the light source model did not consider the wavelength dependence of light, which could result in the appearance of chromatic aberrations. Further, the generation of geometrical blurs in the images due to the finite size of the source (instead of an ideal point light source) was not considered. Noise was also a considerable problem in the illumination distributions obtained. In general, the method in [55] represented an incomplete scenario. Although it suggested the existence of a relation between flow and light features, it was the lack of fidelity that inhibited the ability of the computational models to provide quantitative information.

The research presented in this thesis therefore started from the recommendations for further developments of the analysis [55] and it addresses most of the simplifying assumptions that limited the accurate representation and simulation of nature. In doing so, it tries to answer, at a higher level of detail, what are the *exact* physical and numerical conditions that make the shock wave shadow phenomenon visible? To answer this question and address the limitations of the earlier study, it is necessary to explore different models of flow prediction, of light scattering and of digital image synthesis, and to investigate how the physical conditions translate to the numerical simulation.

First a more realistic compressible, viscous, and three-dimensional fluid flow solution is obtained using open-source CFD software. The mathematical equation that governs light propagation is solved to retrieve the nonlinear light ray paths in 3D, avoiding any simplifying assumptions with respect to the refractive index media or the light behaviour itself. Inspired by classical optics studies in the field of gradient-index ray tracing, light refraction is captured through coupled numerical integration, gradient estimation and interpolation algorithms performed on the underlying general 3D flow field domain. Based entirely on geometrical optics, this light simulation framework is also influenced by techniques from other fields, e.g. gradient estimation and shock detection in CFD, whilst harnessing the advantages of CFI by incorporating current computer graphics libraries and algorithms typically employed for computational image synthesis.

Moreover, this work focuses on modelling all of the principal aspects involved in the image formation: the inhomogeneous medium, light source, reflective surface, viewing system and light behaviour within these models. Therefore, modelling of the light source is conducted taking into consideration its spectral traits and the wavelength dependence of the radiant energy, appropriately quantified, arriving at the projection surface of the shockshadow. This projection surface (e.g. wing) will be modelled as a physically-based reflective material and the light scattering model therefore includes reflection as well as refraction. A practical viewing system is also included in the scene, which can be a simple light sensor, recording film or an actual camera system, allowing for flexibility in terms of configuration. The digital image synthesis process in such a viewing device is also appropriately modelled, verified and validated against suitable images. Whilst at first customised to the reproduction of shockshadows on aeroplane wings, the simulation framework constructed is not specific to this phenomenon, but rather a general virtual environment that can be used to generate artificial images of shockshadows and shadowgraphs. Initially, it is intended for flow visualisation and as an alternative to detect and/or portray the shockshadows and related shock waves. Nevertheless, the simulation framework is built envisaging a future aerodynamic data analysis tool and respective aerospace application.

1.8 Thesis Outline

The first half of this thesis focuses on models embedded in computational flow imaging to build the framework to visualise shockshadows in this research, i.e. compressible flow prediction, visualisation process and digital image synthesis. The key elements are presented in the roadmap in Figure 1.23.

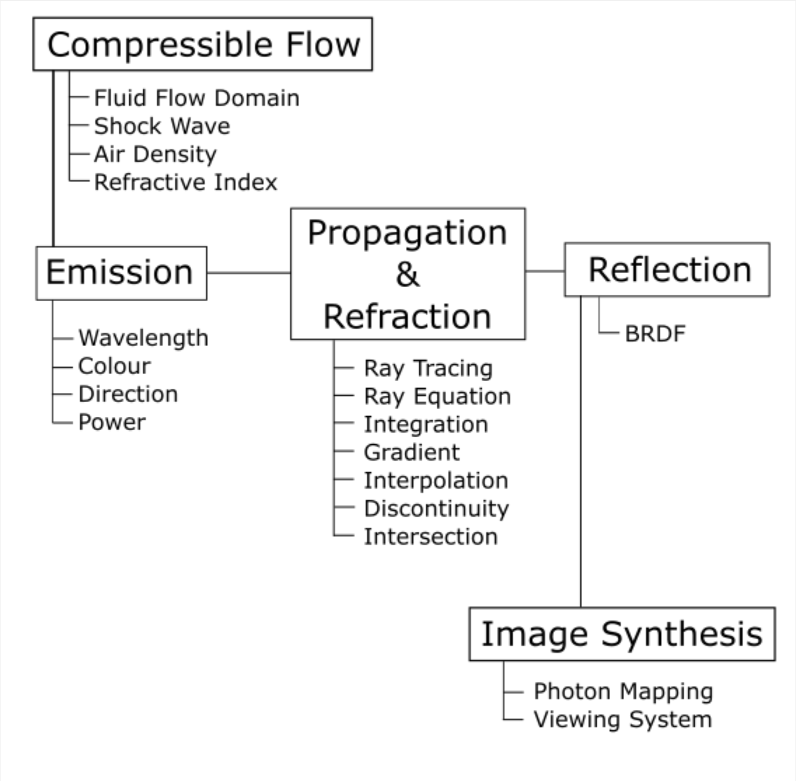


Figure 1.23: Framework roadmap.

In Chapter 2, an overview of the theory behind computational fluid dynamics is provided, which is in essence the foundation of the light simulation framework. Starting from the physical equations that govern fluid flows, the CFD methodologies are outlined, before more detail is given for the numerical techniques employed by the software used to simulate the compressible flow cases analysed in this research.

Chapter 3 details the light simulation framework systematically, in a sequence that corresponds to the stages undergone by light while traversing the predicted flow field. First, different ways to model the spectral radiant energy emitted from the sun, which is the light source, are explained. How this continuous distribution is converted into discrete values that can be displayed in pixels of an image or a screen is also described. Subsequently, the procedure to solve the Ray Equation, i.e. the gradient-index or nonlinear ray tracing algorithm, is presented. This explains how to capture the light refraction in the inhomogeneous optical medium accurately and efficiently, thereby tracing the paths followed by light rays before and after reflection at the wing surface. Emphasis is given to the dependence of the solution process on the numerical algorithms chosen to perform integration, gradient estimation, interpolation and intersection computation. Of particular importance is the trade-off between computational accuracy and efficiency. Thereafter, the estimation of the reflection is introduced, incorporated into the solution of the Rendering Equation for each picture element of a viewing system. This is achieved whilst evaluating the scattering properties of the reflective surface according to the suggested models. Finally, the perspective projection virtual camera is described alongside the fundamental technique used to generate computational images, i.e. photon mapping.

Chapter 4 reports on the verification and validation assessments performed by applying the implemented light simulation framework to three meaningful cases of light refraction. The cases are presented in order of complexity, whereby each module of the implementation is scrutinised with respect to the computer programming, accuracy of calculations, mathematical approach, representation of the physical reality and numerical errors in the simulated imagery.

Chapter 5 presents the computational shockshadow images simulated employing the new light simulation framework, developed as part of the main contribution of this work. This chapter therefore showcases the novel capability of this research, namely the first generation of physically accurate computational images exhibiting the formation of this specific visual phenomenon on aircraft wings. It concerns the targeted applications in two cases of compressible flow over wings. The first is artificially constructed to establish a virtual testing environment to study the applications to 3D aerodynamic simulations. The second is a realistic representation, whereby other aspects of computational image synthesis and shockshadow simulation are investigated.

Finally, Chapter 6 provides the reader with the principal conclusions drawn from the imagery and digital data generated in the previous chapter, and whilst developing the light simulation framework itself. Moreover, suggestions for future work are also given, mainly involving complementing implementations that would further improve this already versatile simulation tool.

CHAPTER 2

Computational Fluid Dynamics

The partial differential equations governing fluid mechanics were derived nearly two centuries ago and are called the Navier-Stokes equations. Numerical algorithms to solve these equations were initially limited by the available processing power and storage capability. However, these methods became viable with the evolution of electronic computers [56].

Computational Fluid Dynamics, or simply CFD, is therefore a modern aerodynamics discipline that complements experimental and theoretical approaches to the evaluation of fluid flow behaviour. Since there are few analytic solutions of the Navier-Stokes, CFD is an essential enabling technology employed extensively in aerospace engineering [56–59].

CFD encompasses a wide range of methods and approaches, but all discretise the fluid flow governing equations so that a numerical solution of the system can be obtained in time and space. The quality of the CFD solution and the accuracy with which it can provide both a qualitative and quantitative description of the flow is highly dependent on the discretisation itself [56–59]. The light modelling performed and analysed in this research relies on an abundant data set [46] and digital image processing techniques [44] to reproduce the shock wave shadow phenomenon. A complete set of information about the flow field is available from the CFD solution [40] for use in this study. Therefore this Chapter gives a general overview of CFD simulation, with a specific focus on the main methods employed in the present work.

2.1 Governing Equations

The Navier-Stokes equations that govern the behaviour of fluids are mathematical statements of conservation laws for mass, momentum and total energy. The equations of state close the system of coupled non-linear partial differential equations [58, 59]. In aerospace applications, the fluid is viscous, compressible and Newtonian, and body forces can be neglected. Hence in a three-dimensional Cartesian coordinate system (x,y,z), the differential formulation of conservation of mass or continuity (Eq. 2.1a), conservation of momentum in the three coordinate directions (Eqs. 2.1b, 2.1c, 2.1d) and conservation of total energy (Eq. 2.1e) can be written in conservative form as [58, 59]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \tag{2.1a}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y} + \frac{\partial \rho uw}{\partial z} = -\frac{\partial p}{\partial x} + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} \tag{2.1b}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} + \frac{\partial \rho vw}{\partial z} = -\frac{\partial p}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} \tag{2.1c}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho uw}{\partial x} + \frac{\partial \rho vw}{\partial y} + \frac{\partial \rho w^2}{\partial z} = -\frac{\partial p}{\partial z} + \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \tag{2.1d}$$

$$\begin{aligned} \frac{\partial \rho E}{\partial t} + \frac{\partial u(\rho E + p)}{\partial x} + \frac{\partial v(\rho E + p)}{\partial y} + \frac{\partial w(\rho E + p)}{\partial z} &= -\frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y} - \frac{\partial q_z}{\partial z} + \frac{\partial u\sigma_{xx}}{\partial x} + \\ \frac{\partial u\sigma_{yx}}{\partial y} + \frac{\partial u\sigma_{zx}}{\partial z} + \frac{\partial v\sigma_{xy}}{\partial x} + \frac{\partial v\sigma_{yy}}{\partial y} &+ \frac{\partial v\sigma_{zy}}{\partial z} + \frac{\partial w\sigma_{xz}}{\partial x} + \frac{\partial w\sigma_{yz}}{\partial y} + \frac{\partial w\sigma_{zz}}{\partial z} \end{aligned} \tag{2.1e}$$

where ρ, μ, $\overrightarrow{V} = [u,v,w]$, p, q_i, E and σ_{ij} are the density, dynamic viscosity, velocity, pressure, heat flux, total energy and viscous stress tensor of the fluid, respectively.

The viscous stress tensor for a Newtonian fluid is given in matrix notation as [58]

$$\sigma = \begin{bmatrix} 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\nabla\cdot\overrightarrow{V} & \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\nabla\cdot\overrightarrow{V} & \mu\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) \\ \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \mu\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) & 2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\nabla\cdot\overrightarrow{V} \end{bmatrix} \tag{2.2}$$

Assuming a perfect gas, the following equation can be used for the pressure [60]

$$p = \rho(\gamma - 1)\left(E - \frac{1}{2}(u^2 + v^2 + w^2)\right) \tag{2.3}$$

where $\gamma = c_p/c_v$ is the ratio of specific heats.

The heat flux is given by Fourier's law of heat conduction, written as [59]

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = -\frac{\mu c_p}{Pr} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} \tag{2.4}$$

where T, c_p and Pr are the temperature, specific heat at constant pressure and the Prandtl number, respectively.

Finally, for laminar flow, Sutherland's law is used to describe the dependence of dynamic viscosity on temperature [57]

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}} \left(\frac{T_\infty + 110}{T + 110}\right) \tag{2.5}$$

where μ_∞ and T_∞ are the free stream values, which are here used as the reference viscosity and temperature.

2.1.1 Reynolds Averaging

The Reynolds number of a flow can be regarded as a measure of the relative importance of viscous and inertial forces in the fluid dynamics [58]. When the Reynolds number is low, viscous forces dominate and turbulent instabilities are damped such that the flow is essentially laminar and characterised by a smooth and orderly flow. As the Reynolds number increases, the inertial forces become more significant compared to the viscous forces and turbulence develops. The flow exhibits randomness and unsteadiness, such that flow variables are no longer deterministic [61]. The transition from laminar to turbulent depends on: the initial disturbances, pressure gradients, wall roughness, heat transfer and compressibility effects, to name only some of the factors that affect the process [59].

Direct Numerical Simulation (DNS) of the complete Navier-Stokes equations can, in principle, be used to resolve turbulence. However, for high Reynolds number flows that arise in aerospace engineering, turbulent fluctuations in flow variables show considerable variation in dimension, strength and time scales (or frequency). Computational modelling of a flow with these characteristics is a difficult task for DNS and a dense mesh and short time step would be needed to resolve the small length and time scales [56, 58, 59]. This level of spatial and temporal refinement would entail prohibitive computational costs for such a simulation [58].

Commonly employed approaches to reduce costs involve the modification of the governing equations through statistical techniques [58, 61], e.g. filtering and averaging.

One of these approaches stems from the application of a high-pass spatial filter to the set of conservation equations, which will only directly simulate the mean flow and large scale turbulent structures, leaving small eddies to be resolved by sub-grid models [58, 59]. Examples of filters of this type are the box and Gaussian, which basically are multiplicative functions that become equal to zero at scales smaller than a threshold. By filtering out the small isotropic scales, a Large Eddy Simulation (LES) simulation concentrates on the eddies that are more complex to model due to their characteristic anisotropy and high energy content [58, 59]. However, significant and costly computational resources are still required in LES simulations at high Reynolds numbers [59].

For engineering applications it is more common to solve the Reynolds Averaged Navier-Stokes equations (RANS), which is a computationally cheaper approach [58, 59]. While RANS intends to resolve the impact on the mean flow of all scales of turbulence [58], like LES it results in additional terms and a modified set of equations. In this averaging approach the instantaneous flow variables are described as a mean value (indicated by a bar) plus fluctuating components (indicated by an apostrophe), expressed as [58, 61]

$$\phi(x,y,z,t) = \overline{\phi}(x,y,z,t) + \phi'(x,y,z,t) \tag{2.6}$$

where ϕ can represent any flow quantity. If we consider a time, rather than ensemble, averaging description then mean values can be defined by [58, 61]

$$\overline{\phi}(x,y,z,t) = \frac{1}{T}\int_{t}^{t+T} \phi(x,y,z,t)dt \tag{2.7}$$

where the time scale T is larger than turbulent time scales, but smaller than the time scales of unsteady flow phenomena of interest. For incompressible flows, the Navier-Stokes equations can then simply be rewritten in terms of these time averaged variables, with the resulting equations being identical to the laminar equivalent, except for the additional Reynolds stress tensor. The tensor is composed of velocity correlations, which are approximated by turbulence models thus closing the system of equations [61]. However, for compressible flows the rewritten Navier-Stokes equations contain density-velocity correlation terms as well as velocity correlations, which also need to be modelled [61].

This issue can be overcome using the Favre or density weighted averaging procedure for velocity, temperature and total energy. In this case, instantaneous flow variables are written as a sum of a mass-averaged component (indicated by a tilde) and a fluctuation (indicated by double apostrophes) [61]

$$\phi(x,y,z,t) = \tilde{\phi}(x,y,z,t) + \phi''(x,y,z,t) \tag{2.8}$$

where the mean is now density weighted as [61]

$$\tilde{\phi}(x,y,z,t) = \frac{\overline{\rho\phi}}{\overline{\rho}} \tag{2.9}$$

In this equation, the overbar denotes averaging performed using Eq. 2.7. The details of the additional steps and assumptions involved in the derivation of the Favre averaged equations are not presented here, but can be found in the book by Wilcox [61]. The final compressible equations in terms of time and mass averaged variables (overbars - now omitted for clarity) look very similar to the original compressible Navier-Stokes equations (Eq. 2.1), but with additional terms related to turbulence (represented by a T superscript)

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \tag{2.10a}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho u v}{\partial y} + \frac{\partial \rho u w}{\partial z} = -\frac{\partial p}{\partial x} + \frac{\partial(\sigma_{xx} + \sigma_{xx}^T)}{\partial x} + \frac{\partial(\sigma_{yx} + \sigma_{yx}^T)}{\partial y} + \frac{\partial(\sigma_{zx} + \sigma_{zx}^T)}{\partial z} \tag{2.10b}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho u v}{\partial x} + \frac{\partial \rho v^2}{\partial y} + \frac{\partial \rho v w}{\partial z} = -\frac{\partial p}{\partial y} + \frac{\partial(\sigma_{xy} + \sigma_{xy}^T)}{\partial x} + \frac{\partial(\sigma_{yy} + \sigma_{yy}^T)}{\partial y} + \frac{\partial(\sigma_{zy} + \sigma_{zy}^T)}{\partial z} \tag{2.10c}$$

$$\frac{\partial \rho w}{\partial t} + \frac{\partial \rho u w}{\partial x} + \frac{\partial \rho v w}{\partial y} + \frac{\partial \rho w^2}{\partial z} = -\frac{\partial p}{\partial z} + \frac{\partial(\sigma_{xz} + \sigma_{xz}^T)}{\partial x} + \frac{\partial(\sigma_{yz} + \sigma_{yz}^T)}{\partial y} + \frac{\partial(\sigma_{zz} + \sigma_{zz}^T)}{\partial z} \tag{2.10d}$$

$$\begin{aligned}\frac{\partial \rho E}{\partial t} + \frac{\partial u(\rho E + p)}{\partial x} + \frac{\partial v(\rho E + p)}{\partial y} + \frac{\partial w(\rho E + p)}{\partial z} = -\frac{\partial(q_x + q_x^T)}{\partial x} - \frac{\partial(q_y + q_y^T)}{\partial y} - \frac{\partial(q_z + q_z^T)}{\partial z} + \\ \frac{\partial u(\sigma_{xx} + \sigma_{xx}^T)}{\partial x} + \frac{\partial u(\sigma_{yx} + \sigma_{yx}^T)}{\partial y} + \frac{\partial u(\sigma_{zx} + \sigma_{zx}^T)}{\partial z} + \frac{\partial v(\sigma_{xy} + \sigma_{xy}^T)}{\partial x} + \frac{\partial v(\sigma_{yy} + \sigma_{yy}^T)}{\partial y} + \\ \frac{\partial v(\sigma_{zy} + \sigma_{zy}^T)}{\partial z} + \frac{\partial w(\sigma_{xz} + \sigma_{xz}^T)}{\partial x} + \frac{\partial w(\sigma_{yz} + \sigma_{yz}^T)}{\partial y} + \frac{\partial w(\sigma_{zz} + \sigma_{zz}^T)}{\partial z}\end{aligned} \tag{2.10e}$$

where the turbulent heat flux can be calculated as

$$\begin{bmatrix} q_x^T \\ q_y^T \\ q_z^T \end{bmatrix} = -\frac{\mu c_p}{Pr_t} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} \tag{2.11}$$

and the turbulent shear stress terms σ^T are obtained from a turbulence model, as previously mentioned. The classical models deal with these introduced unknown correlations based on the Boussinesq approximation, which assumes that the apparent shear stress tensor is proportional to the known mean velocity gradients [58]

$$\sigma^T = \mu_t \begin{bmatrix} 2\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) & \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & 2\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) & \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) \\ \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) & 2\frac{\partial w}{\partial z} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \end{bmatrix} \tag{2.12}$$

The turbulent eddy viscosity (μ_t) is the quantity calculated according to the turbulence model of choice. Note that most texts refer to these compressible Favre averaged equations as the Reynolds Averaged Navier-Stokes Equations or RANS equations.

2.1.2 Turbulence Models

Turbulence models can be zero, one or two equation models. Zero-equation models are based on Prandtl's mixing-length hypothesis and modern variants of his model, and compute the turbulent eddy viscosity and consequently the turbulent shear stresses via a simple algebraic formula [58, 59], e.g. the Baldwin-Lomax model. In order to perform the same computations, one-equation models solve the transport differential equation for the turbulent kinetic energy, κ. The solution procedures available for this equation involve simple approximations of the respective terms or more elaborated functions and coefficients for their evaluation. The widely employed Spalart-Allmaras model [61] solves a transport equation for the eddy viscosity, but can be considered in the same class of models as other single κ-equation models. Two-Equations models are also widely used and the $\kappa - \omega$ is well suited for separated flows containing weak adverse pressure gradients within the boundary layer near wall surfaces [58]. ω is in this case the specific dissipation, or rate of conversion of kinetic energy into thermal energy per unit volume, and the turbulent viscosity is calculated as [58]

$$\mu_t = \rho \frac{\kappa}{\omega} \tag{2.13}$$

2.1.3 Inviscid Flow

In many aerodynamic flow scenarios, viscous effects can be neglected reducing the complexity of the fluid flow problem and therefore the cost of solving the governing equations computationally. For example, in high-speed compressible flows inertia effects are most important in the regions where shock waves form, and viscous effects and turbulence are mainly confined to the boundary layer attached to solid surfaces [56]. In these cases, where the boundary layer is thin and remains attached, the Navier-Stokes equations can be approximated by the simpler and easier to solve Euler Equations, given as [62]

$$\frac{\partial \rho u}{\partial t}+\frac{\partial \rho u^2}{\partial x}+\frac{\partial \rho uv}{\partial y}+\frac{\partial \rho uw}{\partial z}=-\frac{\partial p}{\partial x} \tag{2.14a}$$

$$\frac{\partial \rho v}{\partial t}+\frac{\partial \rho uv}{\partial x}+\frac{\partial \rho v^2}{\partial y}+\frac{\partial \rho vw}{\partial z}=-\frac{\partial p}{\partial y} \tag{2.14b}$$

$$\frac{\partial \rho w}{\partial t}+\frac{\partial \rho uw}{\partial x}+\frac{\partial \rho vw}{\partial y}+\frac{\partial \rho w^2}{\partial z}=-\frac{\partial p}{\partial z} \tag{2.14c}$$

$$\frac{\partial \rho E}{\partial t}+\frac{\partial u(\rho E+p)}{\partial x}+\frac{\partial v(\rho E+p)}{\partial y}+\frac{\partial w(\rho E+p)}{\partial z}=0 \tag{2.14d}$$

2.1.4 Non-dimensional Vector Forms of the Equations

The RANS (2.10) and the Euler (2.14) equations are usually solved in a non-dimensional form [58] and also rewritten more compactly in vector form. The non-dimensionalisation variables presented in Table 2.1 are typically used, where the infinity subscript indicates quantities evaluated at the farfield.

Then the non-dimensional vector form of the RANS equations is expressed as

$$\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial(\mathbf{F}^I-\mathbf{F}^V)}{\partial x}+\frac{\partial(\mathbf{G}^I-\mathbf{G}^V)}{\partial y}+\frac{\partial(\mathbf{H}^I-\mathbf{H}^V)}{\partial z}=0, \tag{2.15}$$

where **Q** is the vector of unknowns, defined as

$$\mathbf{Q}= \quad \left[\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho E\right]^T \tag{2.16}$$

The inviscid flux vectors $\mathbf{F}^I$, $\mathbf{G}^I$ and $\mathbf{H}^I$, are defined as

Quantity	Scaling
density	ρ_∞
velocity	U_∞
length	L
temperature	T_∞
total energy	U_∞^2
dynamic viscosity	μ_∞
pressure	$\rho_\infty U_\infty^2$
time	$\frac{L}{U_\infty}$

Table 2.1: Nondimensionalisation of variables

$$\begin{aligned} \mathbf{F}^I = &\quad \left[\rho u,\ \rho u^2+p,\ \rho uv,\ \rho uw,\ (\rho E+p)u\right]^T \\ \mathbf{G}^I = &\quad \left[\rho v,\ \rho uv,\ \rho v^2+p,\ \rho vw,\ (\rho E+p)v\right]^T \\ \mathbf{H}^I = &\quad \left[\rho w,\ \rho uw,\ \rho vw,\ \rho w^2+p,\ (\rho E+p)w\right]^T \end{aligned} \tag{2.17}$$

and the viscous flux vectors $\mathbf{F}^V$, $\mathbf{G}^V$ and $\mathbf{H}^V$ are given by

$$\begin{aligned} \mathbf{F}^V = &\quad \left[0,\ \sigma^*_{xx},\ \sigma^*_{xy},\ \sigma^*_{xz},\ u\sigma^*_{xx}+v\sigma^*_{xy}+w\sigma^*_{xz}-q^*_x\right]^T \\ \mathbf{G}^V = &\quad \left[0,\ \sigma^*_{yx},\ \sigma^*_{yy},\ \sigma^*_{yz},\ u\sigma^*_{yx}+v\sigma^*_{yy}+w\sigma^*_{yz}-q^*_y\right]^T \\ \mathbf{H}^V = &\quad \left[0,\ \sigma^*_{zx},\ \sigma^*_{zy},\ \sigma^*_{zz},\ u\sigma^*_{zx}+v\sigma^*_{zy}+w\sigma^*_{zz}-q^*_z\right]^T \end{aligned} \tag{2.18}$$

The shear stress tensor σ^* now includes laminar and turbulent components, as in

$$\sigma^* = \left(\mu+\mu_t\right)\begin{bmatrix} 2\frac{\partial u}{\partial x}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right) & \left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right) & \left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) \\ \left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right) & 2\frac{\partial v}{\partial y}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right) & \left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right) \\ \left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) & \left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right) & 2\frac{\partial w}{\partial z}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right) \end{bmatrix} \tag{2.19}$$

and so does the heat flux q^*

$$\begin{bmatrix} q_x^* \\ q_y^* \\ q_z^* \end{bmatrix} = -\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_T}\right) c_p \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} \tag{2.20}$$

The vector form of the Euler equations is then given by equation (2.15) without the viscous fluxes, i.e.

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}^I}{\partial x} + \frac{\partial \mathbf{G}^I}{\partial y} + \frac{\partial \mathbf{H}^I}{\partial z} = 0, \tag{2.21}$$

2.1.5 Integral Form of the Vector Equations

The unsteady Navier-Stokes can also be written in integral form. Then integrating (Eq.2.15) over a non-moving control volume V the Navier-Stokes equations become

$$\iiint_V \left(\frac{\partial \mathbf{Q}}{\partial t}\right) dV + \iiint_V \left(\frac{\partial(\mathbf{F}^I - F^V)}{\partial x} + \frac{\partial(\mathbf{G}^I - G^V)}{\partial y} + \frac{\partial(\mathbf{H}^I - H^V)}{\partial z}\right) dV = 0 \tag{2.22}$$

The time derivative can be moved outside of the integral, as the control volume is fixed. Then using the Gauss divergence theorem to transform the volume integrals of the fluxes into surface integrals gives

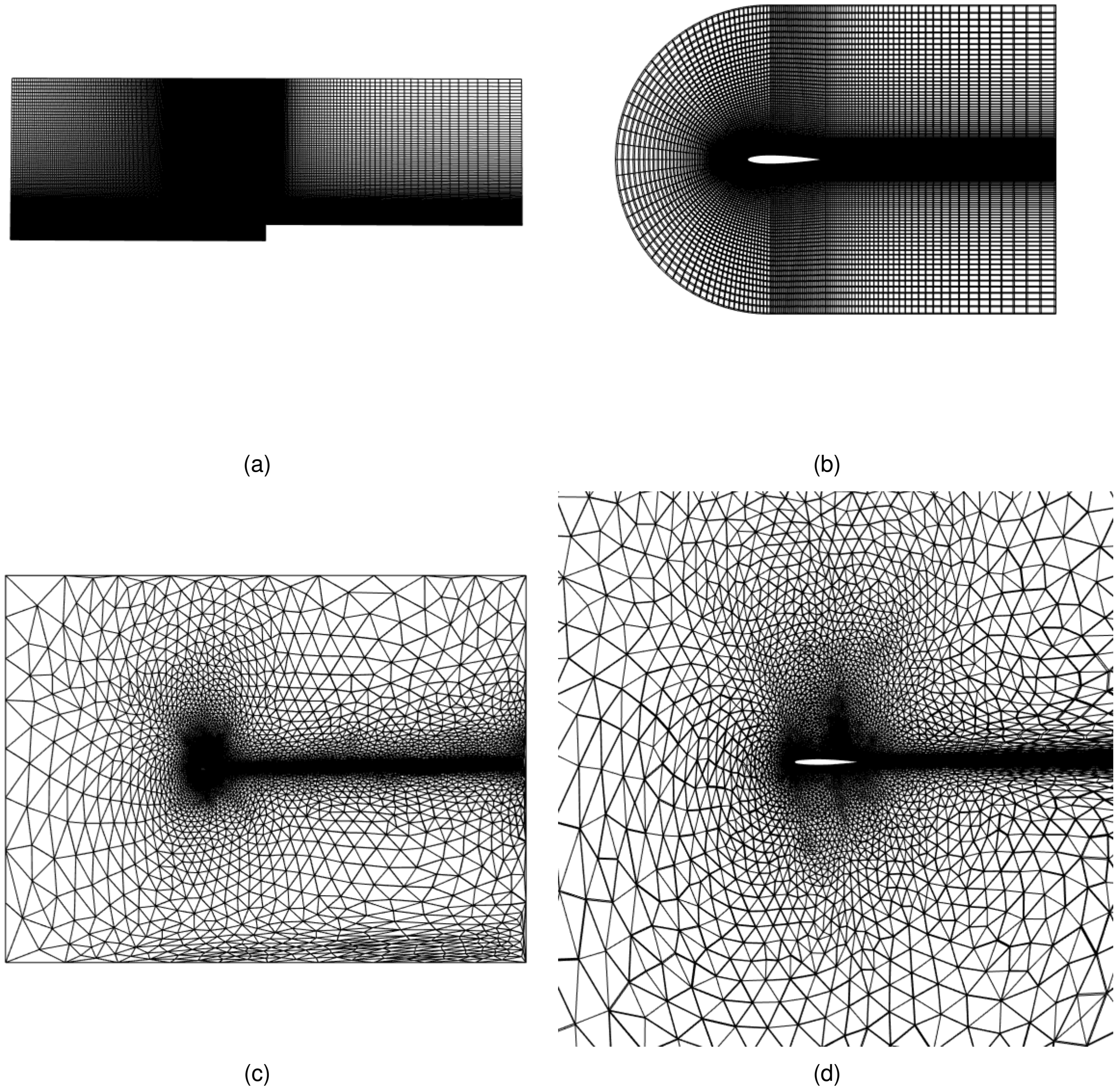
$$\frac{\partial}{\partial t} \iiint_V \mathbf{Q} dV + \iint_{\partial V} \left(\frac{\partial(\mathbf{F}^I - F^V)}{\partial x} + \frac{\partial(\mathbf{G}^I - G^V)}{\partial y} + \frac{\partial(\mathbf{H}^I - H^V)}{\partial z}\right) dA = 0 \tag{2.23}$$

where A is the surface area of the volume V. Similarly, for a fixed mesh the integral form of the Euler equations (Eq. 2.21) become

$$\frac{\partial}{\partial t} \iiint_V Q dV + \iint_{\partial V} \left(\frac{\partial \mathbf{F}^I}{\partial x} + \frac{\partial \mathbf{G}^I}{\partial y} + \frac{\partial \mathbf{H}^I}{\partial z}\right) dA = 0 \tag{2.24}$$

2.2 Domain Discretisation

An important aspect of most CFD simulations is the requirement to geometrically discretise the physical domain delineating the region around a certain solid body that is subject to a fluid flow. The governing equations are put into discrete form and solved on this computational grid, or mesh, and the efficiency of algorithms and the accuracy of respective solutions depend on the mesh properties [56]. The mesh therefore divides the domain into discrete elements, or cells, composed of a finite number of boundary faces, which are in turn defined by edges and vertices, or grid nodes [58]. Meshes are then typically classified as structured or unstructured (see Figure 2.1).



(a)

(b)

(c)

(d)

Figure 2.1: Examples of (a) regular Cartesian structured mesh; (b) C-type structured grid and; (c), (d) tetrahedral unstructured mesh.

A structured mesh encompasses lines and points that can be numbered consecutively and logically indexed, as for example in a simple Cartesian grid. This characteristic directly provides neighbour connectivity information and simplifies the programming, data access and storage during the solution of the equations. There exists some special types of this grid, e.g. boundary-fitted non-orthogonal O, C and H meshes. Also within this class are multiblock meshes, which allow for some localised refinements. However, for geometrically complex domains generating structured grids can by extremely difficult [56].

For geometrically complex domains, an unstructured mesh has the advantage that control volumes can take arbitrary convex polyhedral shapes to adapt to the domain, e.g. hexahedra, prisms, pyramids or tetrahedra in 3D. The trade-off is the requirement for explicit specification of nodes and connectivity and the resulting more complicated coding of the system of equations [56].

The accuracy of the CFD solution greatly depends on the mesh elements topology. The simplest algorithms that evaluate convective and diffuse fluxes and interpolate variables usually provide better approximations on quadrilaterals or hexahedra, compared to triangles or tetrahedra. This is due to the multiple symmetries and parallelism between cell faces. This preference for quadrilaterally-faced elements is especially true near solid boundaries. Particularly when resolving the boundary layer in viscous computation, or if there is a need to construct grid lines following some streamlines of the flow [56].

Another factor that affects the accuracy is the distribution of elements within the computational domain. A smoothly varying solution distribution would be ideal, where there is an optimal smooth transition between neighbouring cells with different sizes. This can be controlled by the user, who normally has prior knowledge about regions in the flow under analysis where the variables are expected to vary considerably and hence require a higher level of mesh refinement and higher number of elements. The smaller spatial resolution would reduce the discretisation error and capture any flow discontinuity. This can also be achieved through multiple iterations, where the solution obtained with an initial coarser mesh indicates to the user where to refine locally in the subsequent grid, forming a solution adaptive mesh generation technique. Similarly, systematic mesh refinements while maintaining the grid topology and relative distribution of grid points is a method to estimate and analyse discretisation errors [56].

It should be noted that grid generation is usually a fundamental stage of CFD simulation, consuming the largest amount of time spent by an experienced user in pre-processsing. The goal being to optimise the quality of the mesh as much as possible, as this generally results in a more accurate solution [56].

2.3 Flow Solver Approaches

Mesh based CFD solvers can be split into three types according to the approach used to discretise the governing partial differential equations into an appropriate form that can be solved numerically [56, 58]. These types are: finite difference, finite element and finite volume.

The finite difference method (FDM) employs the differential form of the conservation equations (Eq. 2.15) to obtain one system of algebraic equations per grid node. Approximations of the partial derivatives then lead to the solution in terms of variable values at these nodes. Usually applied to structured meshes and exploiting their readily available neighbour and connectivity information, these approximations of first and second order derivatives are obtained through Taylor series expansion or polynomial fitting based on local nodal values and the mesh topology. In general, FDM is an effective method to be applied to simple geometries and regular grids, provided that care is taken in enforcing conservation [56].

The finite element method (FEM) is closely related to the finite volume method as both solve the equations in integral form, but belong to different classes of the same weighted residual methods. FEM is developed from the Galerkin finite element formulation and the method assumes an analytical representation of the solution within each grid cell. This is based on the unknown variable values at the respective vertices and so-called shape functions, typically low-order piecewise polynomials. Similar functions are also used as weights when integrating the conservation equations (Eq. 2.23) to result in algebraic equations that interconnect the nodal values and minimise the residual, i.e. the error introduced by the approximations [56, 63].

The Finite Volume Method (FVM) is the numerical solution method most widely used in commercial and open-source CFD codes due to its general applicability to a wide range of fluid flow problems in polyhedral grids [64]. The FVM also applies the integral form of the equations to control volumes that cover the computational domain, which may be the mesh cells themselves or auxiliary or dual mesh cells [56, 64]. Since flux leaving one control volume will enter the adjacent control volume, the method is conservative at the mesh cell level. The method is also mathematically simple [64] and other numerical approximation techniques of interpolation, differentiation and integration can be easily embedded in its formulation. In this work the flow solvers used to generate the CFD data are finite volume methods.

2.4 Finite Volume Methods

2.4.1 Solution Storage

There are many possibilities in terms of distribution of solution storage locations within the computational domain for the system unknowns [56]. In the cell-centred (CC) approach, variables are stored at the centroids of cell elements and the control volumes coincide with the cells formed during the mesh discretisation. In the vertex-centred approach (VC), the variables are stored at the grid nodes and the control volumes are usually formed by connecting the centres of the neighbouring cells directly or also considering the centroids of the adjacent edges, forming the the so-called dual cell [58].

Diskin et al. [65] and Diskin and Thomas [66] studied the performance of both these approaches for grids with different levels of regularity and isotropy using multiple differencing schemes. In both the inviscid and viscous flows studied, the approaches had shown very similar computational complexity, discretisation errors (second-order of accuracy) and convergence rate [65, 66]. However, since the VC approach requires additional storage and special treatment of the boundaries and respective boundary conditions, the CC approach is more widely employed for CFD simulations in the aerospace community [58].

In some solvers hybrid storage solutions are used [67]. For example, in the pressure correction method applied to incompressible flow, the pressure is stored on mesh nodes and velocities are stored at face midpoints. For this scheme this offers a solution to the problem of chequerboard instabilities and simplifies the calculation of fluxes at cell faces [67].

2.4.2 Overview of Solvers for Steady Flow

A large number of different finite volume algorithms have been developed to solve the Navier-Stokes and Euler equations for compressible flow. For steady flow, some solvers directly solve the steady form of the equations, however a common and efficient approach is to solve the unsteady equations and use the same time-stepping or time-marching approach used for unsteady flows. The solution is advanced in time until it reaches a steady-state as the time derivatives tend to zero. The key differences between time-marching schemes then lies in the different approaches to spatial discretisation (the evaluation of the convective and dissipative fluxes) and time integration [56, 58]. More detail is given below for the two codes used in this research.

2.4.3 Boundary Conditions

The discretised system of algebraic equations obtained following spatial and temporal discretisation is composed of a set of conservation equations for each control volume. In order to solve these equations, boundary conditions are also required. Physical boundary conditions are required to enforce flow physics, for example no flow through a solid surface. These boundary conditions have to be enforced for all CFD schemes. Other boundary conditions are numerical boundary conditions that arise due to the stencil of the specific scheme used. Cells located at the boundary of the computational domain have a reduced number of neighbouring cells and the scheme stencil may require information from outside the flow domain which is unknown. Hence such cells must be treated differently to ensure the robustness of the simulation code [56, 58, 64].

2.5 Software Used in this Research

CFD is used to provide a rich data set in this research. However it is not a central focus of the research and hence the choice of solver is essentially arbitrary and any suitable code could be used. The approach taken was to make use of the open-source CFD software available. These codes normally provide a comprehensive tutorial collection and corresponding user guide to preprocess, run and postprocess the examples. Two different software packages widely adopted in CFD research were used to predict the flow field for the three main test cases analysed. The cases concern compressible flow over:

1. an axis-aligned cylinder;
2. the NACA 0012 aerofoil;
3. the Onera-M6 Wing.

Case 1 was selected due to the existence of shadowgraphy experimental data associated to this physical problem, which is then used to validate the light simulation framework (see Chapter 3 and Section 4.2). Cases 2 and 3 are, in turn, classical problems in aerodynamics and CFD, frequently employed in the validation of solvers, given also the availability of experimental studies of all forms and respective reports on these examples. Involving two and three-dimensional analyses of transonic wings, respectively, these physical problems represent a logical progress towards a realistic simulation of a shockshadow over an aircraft. In fact, an additional case for the flow over a full civil aircraft model research configuration (developed and shared by Airbus for multi-disciplinary studies) was simulated using the TAU flow solver [68] on one of the standard meshes provided. However, the light propagation process could not be completed, because difficulties were encountered in converting between the TAU and VTK structures. Some elements of the polyhedral mesh could not be processed despite many efforts. These flow solutions are therefore not discussed further here, but are discussed in the Future Work Section of Chapter 6.

2.5.1 OpenFOAM

The first software package used in this research is OpenFOAM (Open Source Field Operation and Manipulation), an open-source object-oriented CFD framework, that originated at Imperial College, London, and was later developed by OpenCFD Ltd. The code is written in C++ and composed of multiple libraries capable of simulating a diverse range of fluid flow problems

[69]. The underlying CFD mesh is generated using the OpenFOAM pre-processing utility, *blockMesh*, which decomposes the computational domain into one or more three-dimensional structured hexahedral blocks. Grid refinement is controlled by specifying the number of cells and the grading, or cell/edge expansion rate defined as the ratio between subsequent lengths, in each of the global coordinate directions [70].

OpenFoam employs the finite volume method to discretise the governing equations and a cell-centred approach where the mesh cells themselves are the control volumes [58, 71]. The specific solver applied in this work to simulate flow around the aerofoil and the cylinder, is *rhoPimpleFOAM*. This solves the transient compressible RANS flow equations using a combination of the PISO and the SIMPLE iterative algorithms [72, 73]. Both the axis-aligned cylinder case and the NACA 0012 case are run as RANS simulations.

For these simulations, a $\kappa - \omega$ model is used to close the system of equations and model turbulent fluctuations in the flow. The solution is time-stepped until a steady flow is reached using the implicit first-order Euler scheme. The solution is then output using the OpenFOAM post-processing utility, *paraFoam* [70], which is conveniently built upon the Visualization Toolkit (VTK) as its data and graphics processing tool. VTK is an open-source software for manipulating and displaying scientific data [74].

2.5.1.1 The PISO-SIMPLE (PIMPLE) Scheme

The PISO-SIMPLE or PIMPLE scheme is an iterative scheme used widely in open-source and commercial CFD software. The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) method [75] was originally developed for steady incompressible flows. This original method consists of an iterative predictor-corrector procedure for the pressure and velocity fields [56, 59]. Each outer iteration starts from the pressure field estimate of the previous iteration, and a velocity field is then computed from this estimate via the momentum equations [75]. Although the pressure and velocity fields then satisfy the momentum conservation equations, they do not necessarily satisfy the continuity equation. Thus the fields must be corrected to enforce mass conservation, which in turn leads to violation of the momentum conservation equation. This process is repeated iteratively until both the continuity and momentum equations are satisfied simultaneously [56, 58]. Note that under-relaxation of the corrections is needed in order to achieve stability and convergence [73]. Traditionally, this type of algorithm works well on staggered arrangements of the variables on the grid. Instead of evaluating both velocity and pressure at the same grid points and using the same control volumes, which simplifies programming but can generate nonphysical and/or oscillatory pressure solutions (e.g. the checkerboard issue) [56, 58, 59], the velocity components are rather computed on a face-centred staggered grid.

Variants of the method relate to differences in the calculation of the corrections and the addition of inner iterations within each outer iteration. Examples include the SIMPLER (SIMPLE Revised) method [56, 59, 64] and the SIMPLEC (SIMPLE Consistent) method [59]. The PISO (Pressure Implicit with Splitting of Operators) method [76] algorithm was specifically developed for transient flows and corrects pressure and velocities twice within an inner iteration [59]. Unlike the SIMPLE algorithm the pressure-velocity coupling equation is consistent and the time derivatives are included, hence under-relaxation is not needed. However there is a stability constraint on the time step. The PIMPLE method combines the PISO and SIMPLE methods and has the advantage of allowing larger time steps whilst maintaining stability. The extension of the above methods to compressible flows requires the solution of the energy equation and the equation of state to obtain density once pressure and temperature have been updated, as well as modifications to the correction equations.

2.5.2 SU2

The second software used is the SU2 toolset. This is also an open-source parallelised C++ framework, designed at Stanford University to solve partial differential equations (PDEs), such as CFD, perform multiphysics analysis and tackle constrained optimisation tasks [77, 78]. A mesh for the Onera-M6 wing is provided within the embedded tutorials for the software, however to allow the mesh to be adapted to adequately resolve the shock a mesh adaptation software is also coupled with SU2. This allows the inherent flexibility of the unstructured mesh structure to be harnessed to optimise the trade-off between solution accuracy and solver efficiency. AMG-Lib [79], written in C and Fortran, is integrated to the CFD suite through a Python wrapper, pyAMG [80], which controls the whole mesh anisotropic adaptation process. Using a metric-based approach [81], a unit mesh is generated in a Riemannian metric space. Subsequently, a metric tensor field evaluated in this space is used to express requirements in terms of size, shape and orientation and local operators, such as edge collapse and point insertion on edge, are applied to modify the mesh and improve it with respect to a pre-established goal [81]. One type of goal is the minimisation of the error in the evaluation of a scalar function, e.g. the pressure field. Finally, the flow solution is interpolated on the now adapted mesh.

SU2 employs the finite volume method to discretise the governing equations, but uses a vertex-centred approach with control volumes around the vertices constructed from a median-dual grid connecting the centroids and midpoints of all the faces, cells and edges of the primal mesh [78, 82]. The specific solver applied in this work to simulate the compressible flow around the wing implements the Jameson-Schmidt-Turkel (JST) scheme for the convective fluxes [82, 83] and the implicit Euler method [83] for time integration. The test case considered is well predicted by inviscid CFD so the Euler equations are solved and the solution is also output in the VTK [74] format [77].

2.5.2.1 The Jameson-Schmidt-Turkel (JST) Scheme

The JST Scheme [84] is a widely used scheme in aerodynamics. The method was developed to address the deficiencies of potential flow solvers when predicting transonic flows with shock waves that are typical of aircraft in cruise. Solutions of the steady Euler equations were restricted by the available computing power and an efficient method was needed. A finite volume approach was taken as the basis for the solver because it allows for arbitrary meshes and could perform calculations on the same meshes used for potential flow simulations [84] for a direct comparison. In the original method a cell-centred approach was adopted, with the control volumes being the mesh cells of a structured mesh. Before the development of the JST scheme, Euler methods were mostly time-stepping schemes, but none appeared to be able to consistently reach steady-state solutions [85].

The time-marching approach was also adopted for the JST scheme, but the convergence issues were overcome. The convective fluxes were evaluated using cell face values for variables, which were obtained as the average of the values on either side of the face. In order to prevent oscillations in the solution and drive convergence to steady state, artificial dissipation terms were developed. After experimentation the dissipation terms in the scheme were based on a combination of second order and fourth order differences. The second order terms are needed near shock waves to prevent oscillations, but are not needed in smooth parts of the flow. The fourth order difference terms are needed in smooth areas of the flow to ensure convergence, but second order differences are not needed. Hence the dissipation adapts according to a sensor term based on the local pressure gradient. In the original code time integration was performed using a multi-stage Runge-Kutta scheme. For steady-state problems convergence was possible through the use of local time-stepping where the solution was advanced to a different time level at each step. The scheme was robust and was thus adopted widely throughout industry and was developed further within the literature including to: RANS simulations [86, 87], unstructured meshes [88, 89], unsteady flow on fixed and moving meshes [90–93], cell-vertex schemes [94, 95] and incompressible flow including free surface and cavitation problems [96–98].

In the SU2 implementation used for the wing calculations, the JST approach is used to evaluate the convective fluxes. The formulation is cell-vertex and uses a median-dual grid which has the cell vertex inside the dual-mesh cells which are used as control volumes. Unlike the original Jameson implementation, time integration uses an implicit Euler scheme [82, 83] since these are the modules used in a tutorial case within SU2 for the Onera-M6 wing.

CHAPTER 3

LIGHT SIMULATION FRAMEWORK

This Chapter describes the background and theory behind the methods implemented to model the behaviour of light within a compressible flow and thus to generate computational shockshadow images. The construction of a light simulation framework based on underlying computational fluid dynamics simulations is the main focus of the work undertaken in this research. The key contribution is the conception of a novel approach that brings together existing methods in computer graphics to build the first physically realistic simulation of this phenomenon including the generation of shockshadow images.

Here, the light simulation framework is presented broken down into the consecutive stages undergone by light whilst travelling through the inhomogeneous medium to produce the shockshadow phenomenon. These include a model of the sun, as a light source that provides the energy required for direct natural shadowgraphy; the Ray Equation from geometrical optics to retrieve the nonlinear paths of light rays within the refractive index medium; the Reflection Equation and a practical viewing system as a means to computationally synthesise an image.

The development of a light simulation framework in this research was completed in two phases. In the initial implementation the code was written in the C++ programming language and based primarily on the Computational Geometry Algorithms Library (CGAL [99]). This approach was employed to develop prototypes of all of the chosen models for light emission, propagation, refraction and reflection. CGAL was adopted as it is widely used within the computer graphics field and hence this seemed the most appropriate tool to develop the light simulation. However, during the project it became clear that CGAL had limited capability to produce the quantities needed to implement the process and lacked the flexibility to handle the diversity of CFD meshes and data structures.

Therefore, for the final implementation, codes were translated to Python and operations performed using the open-source Visualization Toolkit (VTK [74]). This approach has yielded a general and versatile simulation framework that can integrate CFD data from multiple sources and solvers. Since these CFD solutions and corresponding computational grids can vary greatly in complexity, format and physical aspects, a highly flexible implementation framework like that provided by VTK is required. Throughout the descriptions in this chapter, both implementations are detailed. It should be noted that the first results, and the validation and verification assessments, were undertaken with the initial implementation. The later simulations and the final conclusions were drawn based on the final implementation.

Given that there are many concepts and processes that comprise the simulation framework described in this chapter, the roadmap previously presented in Chapter 1 is reproduced here in Figure 3.1 to guide the reader through the stages of light propagation. Previously, in Chapter 2, the methodology used to model compressible flow with shock waves was described. From these solutions, the air density field is extracted and converted into the refractive index medium through the Gladstone-Dale relation (see Eq. 1.1 in Section 1.3), which then dictates the behaviour of light. The path of light starts at emission, goes through the actual propagation and refraction, before and after reflection, and ends at the image synthesis process.

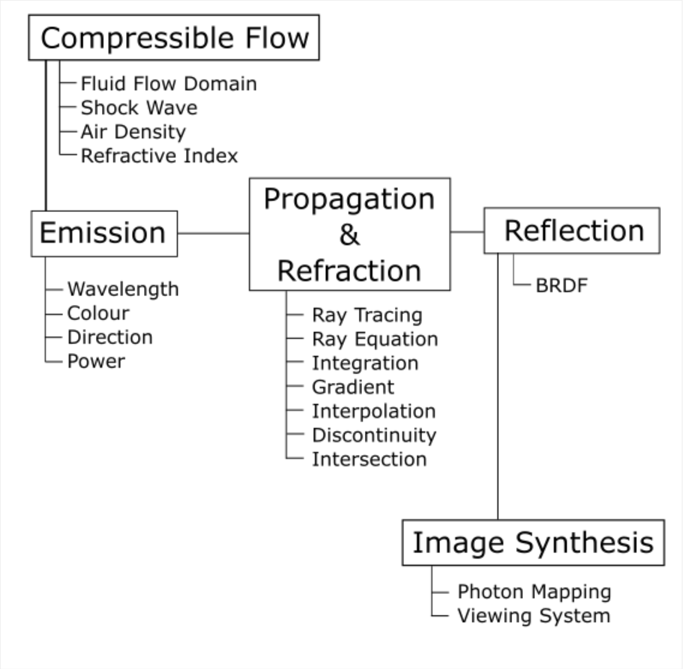


Figure 3.1: Roadmap for this Chapter.

3.1 Light Emission

In rendering engines where physics has a lower priority than the visual aspect of the images generated, light sources are usually represented as a single value of brightness or a RGB colour triplet. The latter is sometimes already compressed in the range $[0,1]$ or $[0,255]$ for 8-bit per channel images. However, in the real world, the range of illumination available varies drastically between the different types of light sources. This is especially true if those naturally occurring, like the sun, are compared against the ones artificially created, such as a tungsten-filament bulb or a fluorescent lamp [100]. When physical realism is required, the challenge then becomes to correctly model these real light sources when rendering. Although the assumption that they can be similarly represented as a white RGB $[1.0,1.0,1.0]$ or a randomly selected brightness value [101] can result in visually appealing images, this is physically incorrect. In fact, the amount of energy emanating from each light source and contributing to the image formation is distinct. Moreover, the spectral distribution of this radiant energy when interacting with the spectral reflective properties of objects under illumination will influence the final perception of their colours, which will differ depending on the source [100].

3.1.1 Spectral Radiant Energy

Radiometry is the study of the propagation of electromagnetic radiation in an environment [102]. Radiometric theory provides the physical foundation upon which rendering engines can be built. By describing radiative transfer phenomena based on geometric optics fundamentals, the behaviour of light is assumed to be linear, energy conserving, in equilibrium and not involving polarisation, fluorescence nor phosphorescence [103]. In the modelling of physically-based light sources in a rendering engine, two radiometric quantities are of importance. The first is the radiant flux or power, Φ, which is the total radiant energy passing through a region in space per unit time. The second is irradiance, E, the flux density (per unit surface area) *arriving* at a surface [103, 104]. Mathematically, the radiant flux and the irradiance are described as [104]

$$\Phi = \frac{dQ}{dt} \tag{3.1a}$$

$$E = \frac{d\Phi}{dA} \tag{3.1b}$$

where: $Q = hc/\lambda$ is the radiant energy; $h = 6.62620 \times 10^{-34}\ Js$ is the Planck's constant; $c = 2.998 \times 10^{8}\ m/s$ is the speed of light in a vacuum and λ is the wavelength of the light. Radiant flux and irradiance have units of watts (W) and watts per square meter (W/m^2), respectively. The solution to realistic light source modelling lies therefore on the spectral characteristics of these quantities, which can be represented by a spectral distribution, i.e. a description of the amount of light at each wavelength [103]. More specifically, in the present research the radiant flux distribution in the visible spectrum (i.e. wavelengths between approximately $390\ [nm]$ and $770\ [nm]$ [105]) is the power from the light source used when solving the Rendering Equation with the Photon Mapping technique [102]. These are the models of image synthesis described in Section 3.4. This radiant power can be obtained from respective irradiance distributions.

The book by Iqbal [105] presents data and methodology for determining quantitatively the solar radiation that reaches the surface of the earth. The energy emission from the sun concentrates in the thermal range of wavelengths, i.e. from $200\ [nm]$ to $10^6\ [nm]$, and depends on complex radiation processes. The sun has an equivalent surface temperature of $5777\ [K]$ and the spectral radiant exitance or radiant flux density *leaving* its surface is often modelled under the assumption that it behaves as a blackbody. By definition, a blackbody is an ideal surface that emits and absorbs the maximum content of electromagnetic radiation at each and all wavelengths, at a certain temperature (T_b), originating from all directions. Planck's Law can then be used to calculate the radiant flux density ($E_{b\lambda}$) stemming from a blackbody in a vacuum or in a medium where the index-of-refraction is close to one. The blackbody radiation equation is written as [105]

$$E_b(\lambda, T) = \frac{C_1}{\lambda^5 [exp(C_2/\lambda T_b) - 1]} = \frac{2\pi h c^2}{\lambda^5 [exp((hc/k)/\lambda T_b) - 1]} \tag{3.2}$$

where h is Planck's constant, $k = 1.3807x10^{-23}\ [J/K]$ is Boltzmann's constant, c is the speed of light in the medium, $C_1 = 3.7427x10^8\ [W\mu m^4 m^{-2}]$ and $C_2 = 1.4388x10^4\ [\mu mK]$ are the first and second radiation constants. Therefore, considering the example of a surface normal to the light rays placed at the mean distance between the sun and earth, $r_0 = 1.4960x10^8\ [km]$, and the mean radius of the solar disk as $r_{sun} = 6.9598x10^5\ [km]$, the extraterrestrial solar spectral irradiance arriving at this surface is determined by Eq. 3.3 and plotted over the spectrum in Figure 3.2.

$$E_{surf}(\lambda) = \frac{\phi_{sun}(\lambda)}{4\pi r_0^2} = \frac{E_b(\lambda, 5777K)\ 4\pi r_{sun}^2}{4\pi r_0^2} \tag{3.3}$$

Knowledge of the extraterrestrial solar spectrum is valuable in a variety of fields of study, from astrophysical research to photovoltaic power systems deployed in high-orbit satellites and vehicles. Moreover, terrestrial solar energy applications also have an interest in this distribution, as the radiation that reaches the earth's surface influences solar power generation and even

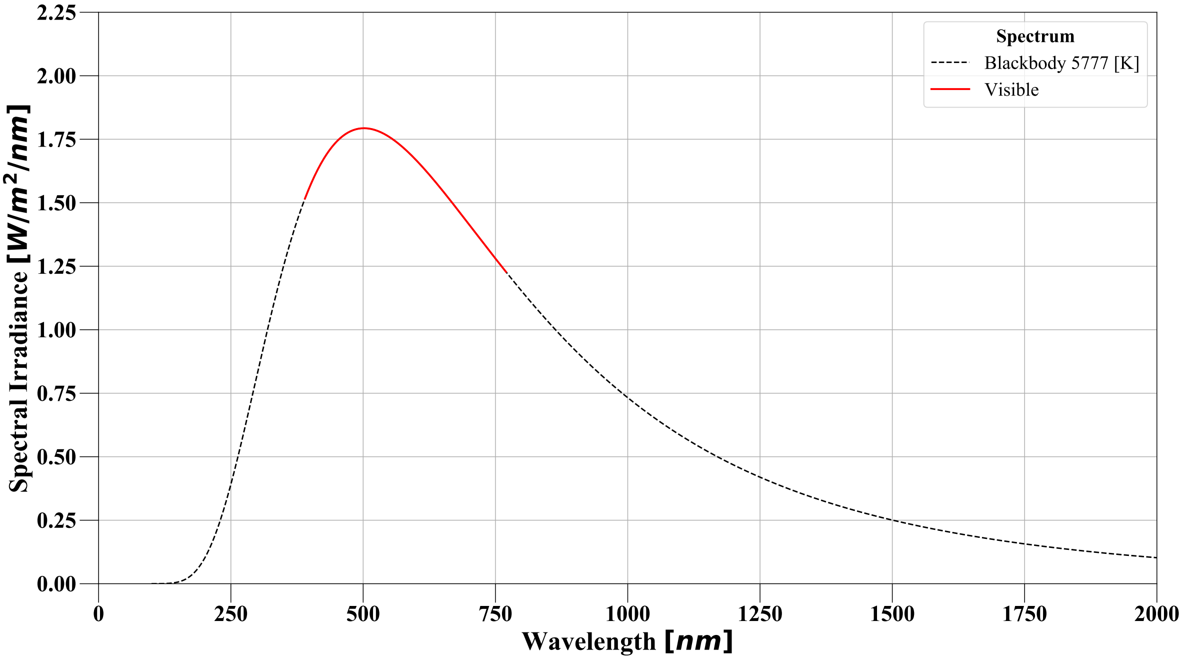


Figure 3.2: Extraterrestrial spectral irradiance from the sun as blackbody at $T = 5777\ K$ evaluated at the mean sun-earth distance.

global warming. These depend on the energy arriving on the outer edge of the atmosphere and the subsequent attenuation by the atmosphere's constituents [105, 106].

However, Planck's blackbody radiation equation for the energy emission from the sun is an idealisation which does not capture the real situation. The sun is a gaseous body with a complex physical structure with multiple layers whose actual measured energy output exhibits significant differences from the idealised blackbody curve. For example, there is a shift in the wavelength where maximum emission occurs towards the ultraviolet region [105].

Early experiments to measure the solar spectrum were entirely terrestrially based using technologically limited techniques. The extraterrestrial distribution was then determined a posteriori via extrapolation to a zero air mass condition. Inevitably, this approach resulted in inaccuracies. With the development of high-altitude measuring instruments, such as observatories, research aircraft, atmospheric balloons and spacecraft, a multitude of extraterrestrial solar spectra have been gathered. However these have been obtained under different conditions, covering different bands of the spectrum and having different wavelength resolutions. Consequently the current approach to determine the terrestrial distribution is to use radiative transfer models that take into account the atmospheric interaction with light. Given the existence of multiple sources of information, complete extraterrestrial (and terrestrial) spectra are built as a combination of the available partial data and modelling methodologies, however these represent only average conditions [106].

A complete composite extraterrestrial spectra was developed by the National Aeronautics and Space Administration (NASA) and Thekaekara [107]. It was later revised by the World Radiation Center (WRC) and Wehrli [108] to address some issues concerning the experiments that the former spectrum was based on. Both of these spectra have been long in circulation and have been widely adopted. The NASA/Thekaekara spectrum served as the basis for a standard in terms of extraterrestrial solar spectral irradiance intended for aerospace application, published by the American Society for Testing and Materials (ASTM) [109]. The standard (ASTM E490 - [110, 111]) was modified in 2000 and it now employs partial spectra from other sources. The WRC/Wehrli spectrum was specifically designed to better match the solar constant recommended by the the World Meteorological Organization (WMO), $1367\ [W/m^2]$, who then adopted the spectrum also as a standard. The solar constant is ’the rate of total solar energy at all wavelengths incident on a unit area exposed normally to the rays’ [105] evaluated at the mean sun-earth distance. This spectrum is also the extraterrestrial reference used in the ASTM G173 standard for terrestrial applications [112, 113]. Figure 3.3 depicts both the NASA/Thekaekara and WRC/Wehrli solar extraterrestrial spectral irradiance distributions.

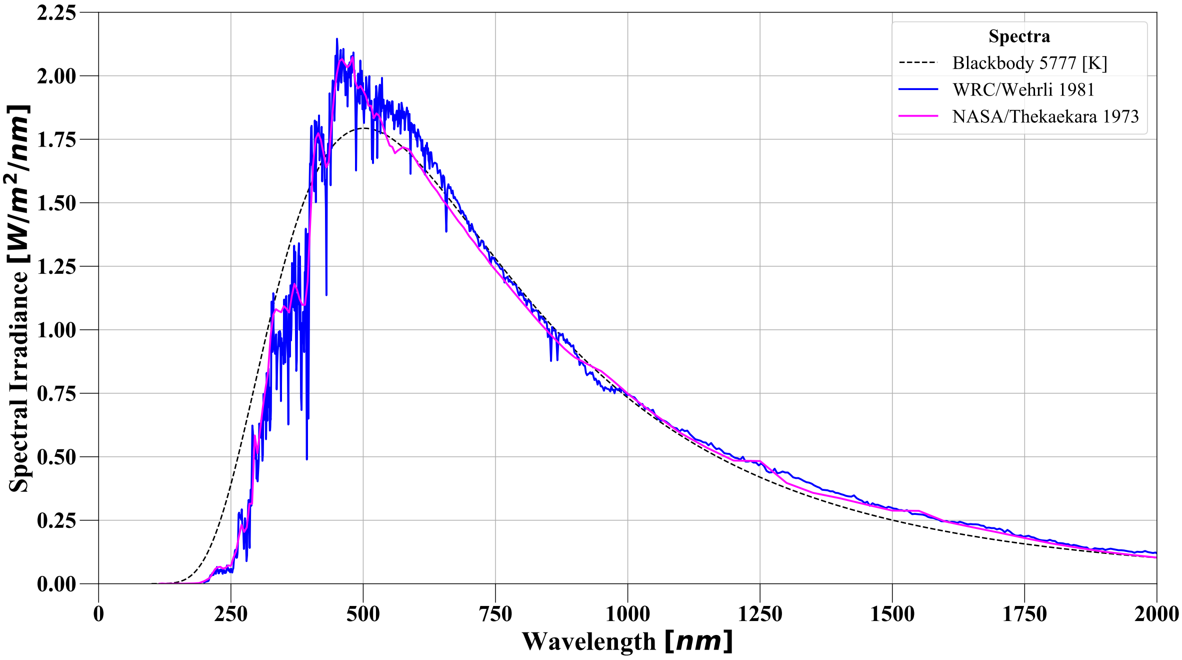


Figure 3.3: NASA/Thaekara and WRC/Wehrli composite solar irradiance spectra.

3.1.2 Colour Representation

Colour becomes a quantitative attribute only when the light scattering properties of the environment are analysed from the physiological view of the human visual system. Colour is a phenomenon with an intrinsic psychological nature. It can be measured in a way equivalent to the photodetection that occurs inside human eyes, i.e. considering the three cone cell responses to a stimulus that are sensed, and together compose human vision [100]. Cones are photoreceptors present in the retina, divided in three categories according to their sensitivity to long, medium or short wavelength ranges. The combination of the signals produced by these cells is then responsible for colour vision. Colorimetry is therefore based on Grassmann's Laws of additive colour matching. These are mathematical rules for combining three primary colours and completely matching any other colour stimulus, according to the trichromatic generalisation. This combination is nothing more than the simple addition of power distributions of different lights over the wavelength spectrum which has an inherent assumption of linearity [100].

Each of the three colour components is thus the result of a sensitivity imposed on the power distribution (Φ) being received as a stimulus at each wavelength λ, integrated over the spectrum, as demonstrated in Eq. 3.4. In practical cases, these sensitivities are colour-matching functions ($\overline{r}$,$\overline{g}$,$\overline{b}$) which, when integrated, specify the proportions of each of the three primaries required to generate a colour match, resulting in the so-called tristimulus values (R,G,B). The red (R), green (G) and blue (B) primaries are presented only as an example, as other colour spaces also exist. The trichromatic generalisation underpins methods for measuring, comparing (matching) and communicating colours [100].

$$R = \int_{\lambda} \Phi(\lambda) \cdot \overline{r}(\lambda) d\lambda \tag{3.4a}$$

$$G = \int_{\lambda} \Phi(\lambda) \cdot \overline{g}(\lambda) d\lambda \tag{3.4b}$$

$$B = \int_{\lambda} \Phi(\lambda) \cdot \overline{b}(\lambda) d\lambda \tag{3.4c}$$

Data from experiments on colour matching performed in the 1920s and 1930s was used by the Commission Internationale de l'Éclairage (CIE) to establish a standard in terms of colour matching functions and a respective set of primaries [100]. The well known and still used CIE 1931 standard observer and CIE XYZ tristimulus values are characterised mainly for the colour matching functions that are all positive and the Y channel that directly corresponds to luminance measurements [100]. Wyman et al. [114] provide analytical fits for the CIE 1931 standard observer, $\tilde{x}_{31}(\lambda)$, $\tilde{y}_{31}(\lambda)$ and $\tilde{z}_{31}(\lambda)$. These colour matching functions, obtained through multiple piecewise continuous Gaussian fits for each spectral curve (amongst other fit options available in [114]), are less error-prone than the widely available tabulated values. If taken as sensitivity functions to the spectral radiant power from any stimulus, and integrated over the spectrum, the colour matching functions provide the tristimulus colour representation in the XYZ space.

The 1931 standard observer multi-lobe piecewise Gaussian fit is given by [114] (see also Table 3.1)

$$\tilde{x}_{31}(\lambda) = \sum_{i=0}^{2} \alpha_{\tilde{x}_i} exp\left(-\frac{1}{2}\left[(\lambda - \beta_{\tilde{x}_i})\mathbf{S}_{\tilde{x}_i}\right]^2\right) \tag{3.5a}$$

$$\tilde{y}_{31}(\lambda) = \sum_{i=0}^{1} \alpha_{\tilde{y}_i} exp\left(-\frac{1}{2}\left[(\lambda - \beta_{\tilde{y}_i})\mathbf{S}_{\tilde{y}_i}\right]^2\right) \tag{3.5b}$$

$$\tilde{z}_{31}(\lambda) = \sum_{i=0}^{1} \alpha_{\tilde{z}_i} exp\left(-\frac{1}{2}\left[(\lambda - \beta_{\tilde{z}_i})\mathbf{S}_{\tilde{z}_i}\right]^2\right) \tag{3.5c}$$

$$\mathbf{S}_{\tilde{x}_i,\tilde{y}_i,\tilde{z}_i} = \begin{cases} \gamma_{\tilde{x}_i,\tilde{y}_i,\tilde{z}_i} \textit{ (see Table 3.1)}, & \text{if } \lambda < \beta_{\tilde{x}_i,\tilde{y}_i,\tilde{z}_i} \\ \delta_{\tilde{x}_i,\tilde{y}_i,\tilde{z}_i} \textit{ (see Table 3.1)}, & \text{otherwise} \end{cases} - \tag{3.5d}$$

Table 3.1: Coefficients for the analytical fits of the CIE XYZ colour matching functions. Data reproduced from [114].

	$\tilde{x}_0$	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{y}_0$	$\tilde{y}_1$	$\tilde{z}_0$	$\tilde{z}_0$
α	0.362	1.056	-0.065	0.821	0.286	1.217	0.681
β	442.0	599.8	501.1	568.8	530.9	437.0	459.0
γ	0.0624	0.0264	0.0490	0.0213	0.0613	0.0845	0.0385
δ	0.0374	0.0323	0.0382	0.0247	0.0322	0.0278	0.0725

The luminance quantity embedded in the Y channel is the photometric equivalent of the radiometric radiance. Radiance (L) is the radiant flux per unit projected area ($A^{\perp}$) per unit solid angle (ω), as defined in Eq. 3.6 [104]. It represents the amount of radiant power at a point in space, arriving from a certain direction in a surface area perpendicular to this direction [104]. It is the quantity of interest for rendering algorithms, as it will be described in Section 3.3, and understanding its relationship to vision is therefore important. Similar to the trichromatic approach to colour representation, luminance therefore addresses the psychological and physiological aspects of radiance with respect to the human visual system [100].

$$L = \frac{d^2\Phi}{dA^{\perp}d\omega} \tag{3.6}$$

The all-positive characteristic of the colour matching functions means that the CIE XYZ colour space cannot be recreated in a device. The space is viewed as device-independent, since some tristimulus values in this space are not even physically realisable and do not correspond to any physical colour. Therefore, the CIE XYZ colour space is normally considered unsuitable for most imaging synthesis computations. It is rather a computational artifice [100], or a standard intermediate between spectral radiance and any other trichromatic device-dependent colour space under which displays tend to operate. These are usually efficiently encoded and customised for specific hardware and software [100]. The XYZ triplets are thus commonly used in physically-based renderers (e.g. spectral path tracer) [114].

Therefore, to convert between the device-independent XYZ colour space and any other, a simple $3x3$ matrix can perform the transformation, according to Grassmann's Law. The conversion matrix, specific to each pair of colour spaces under analysis, depends on the specification of the three primaries and the white point, i.e. the vectors $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ and $(1,1,1)$, respectively [100]. The recommendation ITU-R BT.709 [115] provides those specifications for HDTV, typically used in video and computer monitors, and they are exactly the same as that of the sRGB colour space, a standard designed to specifically represent computer displays. It is important to note that there are many classes of RGB colour spaces, each encoded according to an specific display device, and the sRGB is just an attempt to provide a standard for computational activities [100].

The conversion matrices between the sRGB and CIE XYZ colour spaces are given as [100]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.7a}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2405 & -1.5371 & -0.4985 \\ -0.9693 & 1.8706 & 0.0416 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.7b}$$

Brightness is the attribute according to which different visual stimulus or colours can be judged in terms of how intense they are or how much light (photons) they contain [100]. It is perceived by the human visual system in a way that differences between darker colours are much more noticeable [116]. Physiologically, a power relationship with an exponent smaller than one compress the high values of intensity and increase the sensitivity to lower values. For this reason, display devices normally operate according to the inverse of this relationship [100, 116], which means that variations in electrical voltage inside the hardware of the apparatus will result in the same variation of colour perceived by the spectator. This way the correct physical brightness is mapped and displayed, although internally to each of these systems the signal processing is nonlinear. In terms of colours, signals are dimmed by displays and brightened by the human visual system.

Therefore, when working with digital imaging, before submitting the radiometric quantities calculated in each pixel to the rendering pipeline for the image to be synthesised, the colour triplets have to be prepared by applying the inverse of the monitor's nonlinear power law, a process called gamma correction [100, 116]. This process is represented in Figure 3.4, where I is the sRGB/XYZ representing linear luminance or radiance, II is its gamma corrected value sent for rendering and display and III is the signal presented to the observer by the computer display.

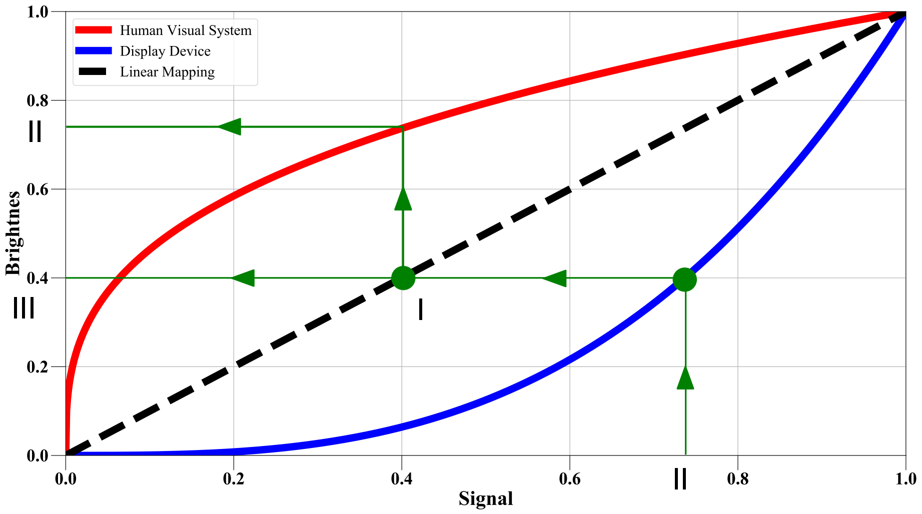


Figure 3.4: The gamma correction process.

The nonlinear power law is given by

$$L_{nonlinear} = L_{linear}^{1/\gamma} \tag{3.8}$$

and although each display device operates according to its own specific gamma value, $\gamma = 2.2$ is usually employed in computer graphics.

Analogously, the 8-bit depth images computed in the sRGB colour space containing 256 colour shades per channel are usually encoded according to a nonlinear relationship between the radiance or luminance values they represent and the actual colour values that they store [100]. This allocates more memory to the darker colours, which are more perceptible to the human visual system [100]. The actual conversion between the linear sRGB triplets obtained via Eq. 3.7 and their respective nonlinear values, a process called gamma encoding, is given by [100]

$$R_{nonlinear} = \begin{cases} (1+f)R_{linear}^{\gamma} - f, & t < R_{linear} < 1 \\ sR_{linear}, & 0 \leq R_{linear} \leq t \end{cases} \tag{3.9a}$$

$$G_{nonlinear} = \begin{cases} (1+f)G_{linear}^{\gamma} - f, & t < G_{linear} < 1 \\ sG_{linear}, & 0 \leq G_{linear} \leq t \end{cases} \tag{3.9b}$$

$$B_{nonlinear} = \begin{cases} (1+f)B_{linear}^{\gamma} - f, & t < B_{linear} < 1 \\ sB_{linear}, & 0 \leq B_{linear} \leq t \end{cases} \tag{3.9c}$$

where the constants are given as: $f = 0.55$, $\gamma = 1/2.4$, $s = 12.92$ and $t = 0.0031308$ [100].

3.1.3 Directional Light

In computer graphics, a directional light source concerns parallel light rays that are emitted all in the same global direction [104]. This is an acceptable approximation for distant lights or point lights at infinity [103], such as the sun illuminating a scene located on earth [102, 103]. This approximation therefore is a simple mathematical abstraction [104] and does not have any physics basis [102], as it neglects the geometric blur that occurs due to the finite size of the sun [1]. Nevertheless, it is still very often employed in rendering. This is done by directing light rays based on positions outside and inside the scene, and by projecting the area of the scene that is illuminated and specifying the power of the light source in terms of radiant exitance or irradiance. The irradiance arriving at the illuminated objects decouples the light source from the dimensions of the scene [102].

The direction and position of light together compose the five dimensions necessary to describe radiance [102], or the restriction of radiant energy in time, space and direction [104]. Since this energy transported by photons along a solid angle towards a surface area is closely related to the colour sensed by a light detection device [102], and a light ‘ray is a line drawn in space corresponding to the direction of flow of radiant energy’ [13], it is logical to compute this radiometric quantity in ray tracing algorithms [102, 104]. In a vacuum, radiance is constant in both directions along a *linear* light ray and it can be defined anywhere in space [102, 104], making it plausible to be evaluated at a surface reflecting light and at a sensor receiving light. In nonlinear media where the refractive index varies continuously so does radiance [102]. However, in compressible flow even with the presence of inhomogeneities, e.g. shock waves, the refractive index of air is still approximately close to one. Therefore this variation of radiance due to nonlinearities is not taken into account in this study.

3.1.4 Light Source Model

In the initial CGAL light simulation framework sunlight is modelled as a simple white RGB colour ($[1.0, 1.0, 1.0]$ or $[255, 255, 255]$). However, whilst this was acceptable for the development of the basic nonlinear ray tracing methodology, this is a non-physical simplification of the real situation. In the final VTK light simulation framework the aim was to implement a higher fidelity model of the sun. Whilst any of the spectra presented in Section 3.1.1 are options to model sunlight, the final VTK implementation uses Planck's blackbody radiation equation evaluated at 5777 $[K]$ and at the mean sun-earth distance, i.e. Eq. 3.3. Although the behaviour of the sun's terrestrial radiation does not completely agree with the emission from a blackbody, this spectral distribution is able to model the total amount of energy from the sun and its wavelength dependence with a reasonable level of accuracy. Since these are the two primary factors that enable the physically correct representation of the sun as a light source in the computational shockshadow generation, this is a reasonable simplification.

However, the spectral irradiance distribution presented in Figure 3.2 cannot be employed directly in the global illumination algorithm. Instead, it has to be integrated over the visible spectrum in order to derive the total radiant flux density arriving at the mean sun-earth distance. During this integration, the spectral distribution is also multiplied by the analytical fits for the CIE XYZ colour matching functions provided by Wyman et al. [114], described in Section 3.1.2, thus representing the irradiance in the XYZ colour space.

Light rays are then originated at evenly distributed positions above the surface to be illuminated and emitted in a single direction, according to the directional light source approximation mentioned in Section 3.1.3. In the initial implementation this direction is determined solely based on the global coordinate directions implicit in the fluid flow solution of the cases analysed and only planar angles are used. In the final implementation an approach more in agreement with the flight experimental analysis performed by Fisher et al. [16] is taken, with the quarter-chord line of the wing used as a reference. From this reference both an azimuth and an elevation angle are measured in order to determine this global direction of light, as depicted in Figure 3.5.

These light rays represent not only a direction normal to the light wavefront [13] that will be traced through the solution of the Ray Equation during the illumination stage, but also represent photons, the carriers of the light radiant flux that are used in Photon Mapping during the visibility stage. Each photon is therefore assigned with an equal fraction of the total power arriving from the sun at the wing surface. In the initial implementation, this is simply the RGB value assigned to the source divided by the number of emitted photons. In the final implementation, the tristimulus XYZ values of the total power are evaluated by projecting the wing surface in

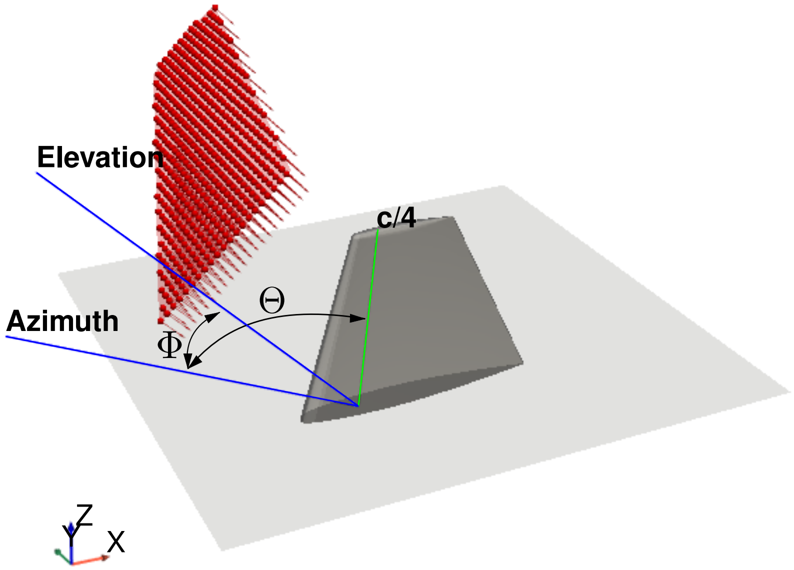


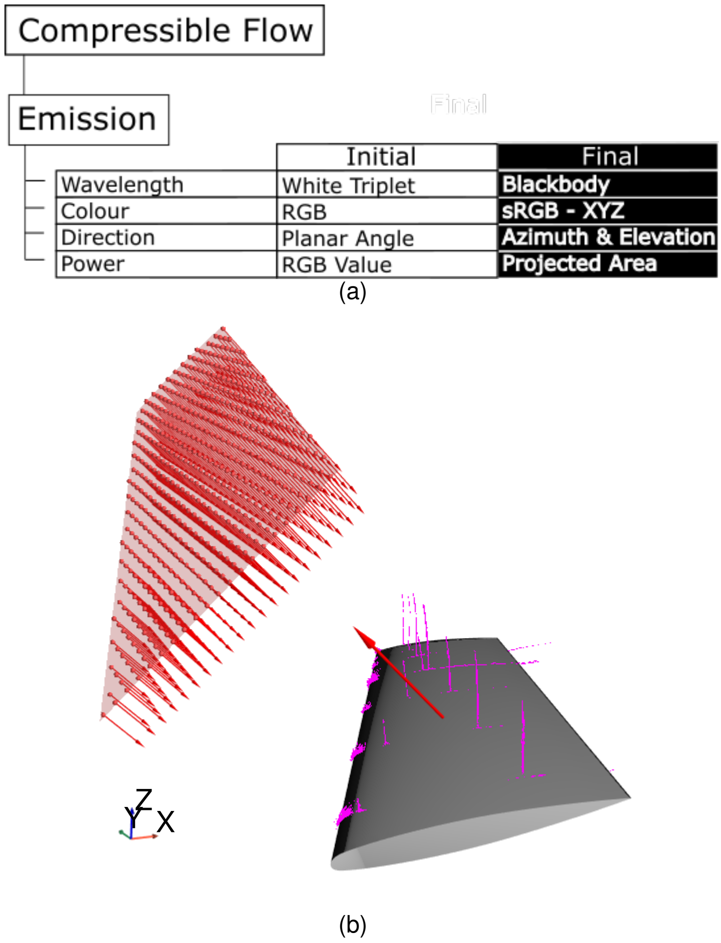
Figure 3.5: Layout of the azimuth (Θ) and elevation (Φ) angles with respect to the wing quarter-chord line ($c/4$), used to determine the sun illumination direction.

the desired direction of light emission, as shown in Figure 3.5, and multiplying the area of this projection surface by the previously calculated total irradiance. Finally, the colour triplet is converted to the sRGB colour space, as described in Section 3.1.2, which is more appropriate to be displayed in pixels on a screen.

Figure 3.6 summarises these decisions related to the modelling of light emission in the initial and final implementations. Figure 3.6(a) shows the development of the light simulation framework along the roadmap of this Chapter and Figure 3.6(b) illustrates the final process of light emission employed.

3.2 Light Propagation and Refraction

The idea behind ray tracing has been used for a long time in optics to design lenses [117] through the analysis and evaluation of the paths followed by light rays in optical systems. Following the same principles, a computer graphics technique that adopts the same name has found its niche in the generation of digital or synthetic two-dimensional pictures of the three-dimensional world [118]. This process has traditionally embedded the assumption that light rays propagate in linear trajectories. When it comes to refraction this phenomena has been modelled simply by applying Snell's Law [13] at the boundaries of objects, or between different media of different but constant refractive indices [119, 120].



	Initial	Final
Wavelength	White Triplet	Blackbody
Colour	RGB	sRGB - XYZ
Direction	Planar Angle	Azimuth & Elevation
Power	RGB Value	Projected Area

(a)

(b)

Figure 3.6: (a) Roadmap of the current progress in terms of light modelling, and (b) graphic representation of the light emission model used in the final implementation.

However, in a gradient-index media the index-of-refraction is not constant or homogeneous, but varies continuously in space [120]. This situation arises in compressible aerodynamics and light rays propagate along nonlinear trajectories (curved paths) [120, 121]. In practice, it is usually impossible to accurately predict the profile of the refractive index, hence the reason why the mathematical equation that governs the behaviour of light in optical media has to be solved in order to obtain the light path, at least approximately.

3.2.1 Nonlinear Ray Tracing

Some authors attempt to simplify the ray tracing process by classifying distinctive types of optical property distributions [117, 122, 123] or deriving analytical light ray paths [119, 120]. However, these studies make assumptions regarding the spatially varying media that are too restrictive. In more general physical simulations, the alternative is to obtain the nonlinear light ray path via integration of the light ray equation coming from the geometrical optics domain. A number of different numerical schemes appear in the literature [119], however most of the approaches involve a piecewise linear approximation of the light rays [120]. This concerns an assumption of a locally constant index-of-refraction within each integration step, which constrains the step size required for accurate solution and can lead to high computational costs [119].

3.2.2 Ray Equation of Geometrical Optics

The mathematical equation that governs the behaviour of light in optical media is derived from Fermat's principle [24]. The principle states that light rays travel between two points in a certain medium following a path that establishes the minimum traversal time. This can be posed as a Calculus of Variations problem by considering the three possible types of optical media and the definition of an optical path length (OPL).

Homogeneous media are described by a constant index of refraction; isotropic inhomogeneous media are those where the index of refraction is a function of the location in space; and anisotropic media are the ones where the refractive index also depends on the light ray direction [26]. An OPL is the distance traversed by the light in a vacuum over the same time (t_{travel}) it takes to travel a limited distance in a medium (l_{medium}) with a certain index of refraction (n), defined as [124]

$$l_{medium} = \int_1^2 ds \tag{3.10a}$$

$$t_{travel} = \frac{l_{medium}}{v} = \frac{l_{medium}}{c/n} \tag{3.10b}$$

$$OPL = ct_{travel} = nl_{medium} = \int_1^2 nds \tag{3.10c}$$

where c and v are the speeds of light in a vacuum and in the homogeneous medium, respectively, and ds is an infinitesimal distance along the light ray path. This definition can also be expressed in terms of the time of transit between two points (T) [26]

$$T = \int_1^2 dt = \int_1^2 \frac{1}{v} ds = \int_1^2 \frac{n}{c} ds = \frac{OPL}{c} \tag{3.11}$$

In an isotropic inhomogeneous medium a variational integral (I) corresponding to the OPL is written as [26]

$$I = \int n(x,y,x)ds = \int n(x,y,x)\frac{ds}{dt}dt = \int n(x,y,x)\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}dt \tag{3.12}$$

Fermat's principle is then a statement of minimisation of the optical path length [124] or a mathematical problem of finding the curve in space for which the variational integral is an extreme [26]. A general and parametric form of the variational integral is [26]

$$I = \int_{p_1}^{p_2} f\left(x(t), y(t), z(t), \frac{dx(t)}{dt}, \frac{dy(t)}{dt}, \frac{dz(t)}{dt}\right)dt \tag{3.13}$$

where f is a function determined by the nature of the problem to be treated by Calculus of Variations. This can be solved by first converting it to the Euler equations, and although the mathematical treatment is outside the scope of this work, it is available in [26]. The respective Euler equations are

$$\frac{d}{dt}\nabla_t f = \nabla f \tag{3.14}$$

where ∇f is the gradient vector of the function f and $\nabla_t f = \langle \frac{\partial f}{\partial(dx/dt)}, \frac{\partial f}{\partial(dy/dt)}, \frac{\partial f}{\partial(dz/dt)} \rangle$. This is a system of coupled ordinary differential equations whose solution or extreme arc is dependent on the main function, and on the boundary conditions. These boundary conditions are the fixed initial and terminal points connected by the respective spatial curve. Therefore, comparing Eq. 3.12 and 3.13, the paths adopted by the light rays in inhomogeneous optical media are the particular parametric curves, $\overrightarrow{r}(t) = \langle x(t), y(t), z(t) \rangle$, that minimize the optical path length or, analogously, solve the variational calculus problem by satisfying the Euler equations. Considering

$$f = n(\overrightarrow{r})|\overrightarrow{r}'(t)| = n(\overrightarrow{r})\sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2 + \left(\frac{dz(t)}{dt}\right)^2} \tag{3.15a}$$

$$\nabla_t f = n(\overrightarrow{r})\frac{\overrightarrow{r}'(t)}{|\overrightarrow{r}'(t)|} \tag{3.15b}$$

$$\nabla f = \nabla n(\overrightarrow{r})|\overrightarrow{r}'(t)| \tag{3.15c}$$

The Euler equations then become

$$\frac{d}{dt}n(\overrightarrow{r})\frac{\overrightarrow{r}'(t)}{|\overrightarrow{r}'(t)|} = \nabla n(\overrightarrow{r})|\overrightarrow{r}'(t)| \tag{3.16}$$

Taking the norm of the tangent vector of the arc length at a point as

$$\frac{ds}{dt} = |\overrightarrow{r}'(t)| \tag{3.17}$$

and rearranging the Euler equations in terms of the arc length parameter (ds), the so-called Ray Equation for an inhomogeneous medium then is

$$\frac{d}{ds}\left(n(\vec{r})\frac{d\vec{r}}{ds}\right) = \nabla n(\vec{r}) \tag{3.18}$$

3.2.3 Solution of the Ray Equation

The solutions of the Ray Equation (Eq. 3.18) for an isotropic inhomogeneous medium provide the paths followed by light rays in nonlinear media. A compressible fluid flow has a varying fluid density and thus can be characterised as a field of this type [22]. Given the Gladstone-Dale relation that relates the density to the refractive index of air, presented in Section 1.3, the propagation of light through this fluid flow can be captured by solving the ray equation.

As detailed by Sakamoto [125], algorithms to approximately solve the Ray Equation can be developed based on a transformation of the independent variable. This transformation is written in terms of the arc length along the ray or simply ray path, s, and a power of the refractive index, n^k. One of the main references in ray tracing methods for gradient-index media [121] concerns the case in which $k = -1$ and the new independent variable is called the quasiray path [125].

Following Sharma et al. [121] the equation for t is

$$t = \int \frac{1}{n} ds \tag{3.19}$$

and hence dt is given by

$$dt = \frac{ds}{n} \tag{3.20}$$

Then using these substitutions in Eq. 3.18 and the definition of the unit tangent vector $\hat{T} = d\vec{r}/ds$ yields a new system of two ordinary differential equations

$$\frac{d^2\vec{r}}{dt^2} = n\nabla(n) = \frac{1}{2}\nabla\left(n^2\right) \tag{3.21a}$$

$$\frac{d\vec{r}}{dt} = \vec{T} = n\cos(\alpha)\vec{i} + n\cos\left(\beta\right)\vec{j} + n\cos\left(\gamma\right)\vec{k} \tag{3.21b}$$

These equations can be solved iteratively to give the position of the light ray $\vec{r}$ and the tangent vector $\vec{T}$ along its incremental path ds (parameters represented in Figure 3.7), given the angles α, β and γ between the tangent and the coordinate axis directions and the refractive index distribution.

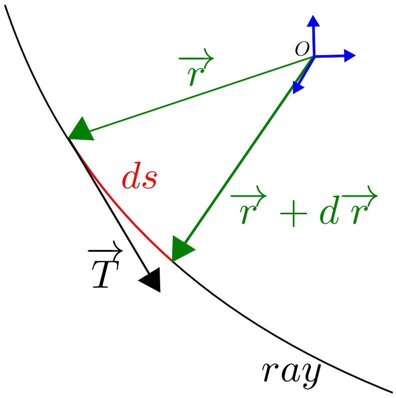


Figure 3.7: Variables that describe light rays in the Ray Equation.

3.2.3.1 Adaptive Integration

Runge-Kutta (RK) methods have been used extensively in optics to solve the ray equation due to their computational efficiency. Two approaches are possible since the ray equation (Eq.3.21a) is a second-order ordinary differential equation (ODE) [126, 127]. The first approach reduces the problem to a system of two first-order differential equations by introducing a new variable corresponding to the first derivative of the dependent variable ($\vec{T}$ in Eq. 3.21b). Each new first-order ODE, one for $d\vec{r}/dt$ and one for $d\vec{T}/dt$, can then be solved approximately using standard RK methods, resulting in interdependent numerical formulae at each integration step.

This approach has been used in the widely cited paper by Sharma et al. [121], who uses a simplified form of the traditional fourth-order RK method. This simplification, which reduces the number of function evaluations in the traditional algorithm, is only possible because there are no first order derivatives on the right hand side of Eq.3.21a.

The second approach directly integrates the second-order equation employing an alternative RK process, called Runge-Kutta-Nyström (RKN) [128] methods. Because of the absence of the first derivatives on the right hand side, the derivation of the coefficients that compose RKN methods is also simplified [128]. This approach has advantages in terms of computation time and memory requirements. There are a multitude of numerical integration schemes using the RKN approach [129–131], with different formulae, orders of accuracy and advantages depending on the target application.

In the initial CGAL implementation of the light simulation framework the second approach has been used. The RKN method used is a traditional and general method described by Fehlberg [128] and is termed the Runge-Kutta-Nyström 4(5) method. This is referred to in the rest of this thesis as the Runge-Kutta-Fehlberg (RKF) method. Described as embedded formulas, the method provides two approximations for the light ray position vector with fourth ($\overrightarrow{r_4}$) and fifth ($\overrightarrow{r_5}$) orders of accuracy, and an approximation for the light ray direction vector also with fourth order of accuracy ($\overrightarrow{T_4}$). These approximations use the same function evaluations (f_0, f_1, f_2, f_3 and f_4 in Eq. 3.22) for computational efficiency. The higher-order approximation is used to advance the subsequent steps, a technique known as local extrapolation. The difference between the two approximations is interpreted as an estimation of the local error (TE) in the position of the ray at each step. This error can then be used to establish an adaptive integration scheme with an automatic step size (h) control and adjustment procedure, based on a preset error tolerance [132]. The RKF 4(5) formulas are then written as [128]

$$\overrightarrow{r_4} = \overrightarrow{r_0} + \overrightarrow{T_0}h + h^2\left(\frac{13}{120}f_0 + \frac{3}{10}f_1 + \frac{3}{40}f_2 + \frac{1}{60}f_3\right) + O\left(h^5\right) \tag{3.22a}$$

$$\overrightarrow{r_5} = \overrightarrow{r_0} + \overrightarrow{T_0}h + h^2\left(\frac{13}{120}f_0 + \frac{3}{10}f_1 + \frac{3}{40}f_2 + 0f_3 + \frac{1}{60}f_4\right) + O\left(h^6\right) \tag{3.22b}$$

$$\overrightarrow{T_4} = \overrightarrow{T_0} + h\left(\frac{1}{8}f_0 + \frac{3}{8}f_1 + \frac{3}{8}f_2 + \frac{1}{8}f_3\right) + O\left(h^5\right) \tag{3.22c}$$

$$TE = \overrightarrow{r_5} - \overrightarrow{r_4} = \frac{1}{60}(f_3 - f_4)h^2 \tag{3.22d}$$

$$f_0 = \frac{\nabla n^2\left(\overrightarrow{r_0}\right)}{2} \tag{3.22e}$$

$$f_1 = \frac{\nabla n^2\left(\overrightarrow{r_1}\right)}{2} \rightarrow \overrightarrow{r_1} = \overrightarrow{r_0} + \frac{1}{3}\overrightarrow{T_0}h + h^2\left(\frac{1}{18}f_0\right) \tag{3.22f}$$

$$f_2 = \frac{\nabla n^2\left(\overrightarrow{r_2}\right)}{2} \rightarrow \overrightarrow{r_2} = \overrightarrow{r_0} + \frac{2}{3}\overrightarrow{T_0}h + h^2\left(0f_0 + \frac{2}{9}f_1\right) \tag{3.22g}$$

$$f_3 = \frac{\nabla n^2\left(\overrightarrow{r_3}\right)}{2} \rightarrow \overrightarrow{r_3} = \overrightarrow{r_0} + 1\overrightarrow{T_0}h + h^2\left(\frac{1}{3}f_0 + 0f_1 + \frac{1}{6}f_2\right) \tag{3.22h}$$

$$f_4 = \frac{\nabla n^2\left(\overrightarrow{r_4}\right)}{2} \rightarrow \overrightarrow{r_4} = \overrightarrow{r_0} + 1\overrightarrow{T_0}h + h^2\left(\frac{13}{120}f_0 + \frac{3}{10}f_1 + \frac{3}{40}f_2 + \frac{1}{60}f_3\right) \tag{3.22i}$$

Alongside these formulas, the adaptive integration setup also consists of a prescribed tolerance ϵ for comparison with the maximum component of the local error TE_{max}. From a starting step size h_{old}, proportional to the domain discretisation, the step size control procedure automatically adjusts it to h_{new} based on the ratio between the tolerance and this maximum error. The automatic step size control and adjustment procedure is mathematically written as [133]

$$h_{new} = h_{old} min\left(\delta_{max}, max\left(\delta_{min}, \delta\left(\frac{\epsilon}{TE_{max}}\right)^{\gamma}\right)\right) \tag{3.23}$$

where the exponent is $\gamma = 1/5$, the safety factor $\delta = 0.9$ is used to increase the probability of acceptance of the error (and consequently the size) on the next step, and control factors $\delta_{min} = 0.1$ and $\delta_{max} = 1.5$ contain the rate of decrease and increase of the size, respectively, following the recommendations of Hairer et al. [133]. Moreover, as advised in [133], after every step size rejection the control factor δ_{max} is returned to 1, considering that the size has to be decreased.

In the final VTK implementation of the light simulation framework the first approach has been used to integrate the ray equation. The system of two first order ODEs are written in matrix form (Eq.3.24) as

$$\overrightarrow{y} = \begin{bmatrix} \overrightarrow{r} \\ \overrightarrow{T} \end{bmatrix} = \begin{bmatrix} x & y & z \\ \frac{dx}{dt} & \frac{dy}{dt} & \frac{dz}{dt} \end{bmatrix} \tag{3.24a}$$

$$\frac{d\overrightarrow{y}}{dt} = \begin{bmatrix} \dot{\overrightarrow{r}} \\ \dot{\overrightarrow{T}} \end{bmatrix} = \begin{bmatrix} \frac{dx}{dt} & \frac{dy}{dt} & \frac{dz}{dt} \\ \frac{d^2x}{dt^2} & \frac{d^2y}{dt^2} & \frac{d^2z}{dt^2} \end{bmatrix} = \begin{bmatrix} n\cos(\alpha) & n\cos(\beta) & \cos(\gamma) \\ \frac{1}{2}\nabla\left(n^2\right)_x & \frac{1}{2}\nabla\left(n^2\right)_y & \frac{1}{2}\nabla\left(n^2\right)_z \end{bmatrix} \tag{3.24b}$$

and higher order RK integration methods than those used by Sharma et al. [121] are considered. An embedded formulae method, analogous to the RKF 4(5), is the RK 8(7) scheme due to

Prince and Dormand [134]. This method is referred to as RK8(7)13M in [134] and DOPRI8 in [133]. It is composed of approximations of the state variables with eighth ($\vec{y}_8$) and seventh ($\vec{y}_7$) orders of accuracy. It was constructed with the goal of minimising the error coefficients of the higher order approximation [133], thus allowing the setup of the local extrapolation mode. The higher-order RKF methods derived by Fehlberg [128] are restricted in terms of local extrapolation, as they could, in certain cases, result in inaccurate error computations if the mode was used [133]. The DOPRI8 scheme also incorporates the FSAL (First Same As Last) idea [133], where the last computation of each step coincides with the initial function evaluation of the next accepted step, saving some computational effort.

The original truncation error estimator of DOPRI8 is of sixth order of accuracy. However, this algorithm misbehaved in some practical applications, presenting loss of accuracy particularly when applied to problems containing discontinuities [133]. Therefore, Hairer et al. [133] modified the formulae and constructed two additional approximations, of third and fifth orders of accuracy, in order to create a composite error estimator whose convergence followed that of the method and was in general more robust [133]. The modified algorithm was originally written in Fortran and given the name "DOP853". Given the higher order aspect of the scheme and the modification introduced, 'there is no easy way to present the coefficients in a tableau' [133] or in the form of equations, but implementations of the code in different programming languages are available in [135].

The DOP853 scheme was shown to have superior performance compared to other lower order and classical RK methods, including the RKF 4(5), when applied to a range of mathematical problems with analytical solutions. It has been shown to be highly effective and especially useful for cases requiring high precision [133]. The method has been widely adopted in multiple programming languages, platforms and libraries and it is also available as one the options of the SciPy's integration sub-package for solving initial value problems of ODEs [136]. This integration method and Python package have been used in the final implementation of the light simulation framework.

In the Hairer's modified method, the automatic step size control and adjustment is also slightly different to that described by Eq. 3.23. It is based on two prescribed tolerances, an absolute ϵ_{abs} and a relative ϵ_{rel}, instead of a single one. These preset values are used to establish a control parameter of the form [133]

$$cp_i = \epsilon_{abs_i} + max\left(|\vec{y}_{0_i}|, |\vec{y}_{8_i}|\right)\epsilon_{rel_i} \tag{3.25}$$

Then, instead of simply using the relationship between the tolerance and maximum component of the truncation error (ϵ/TE_{max}) in Eq.3.23, this expression is substituted by $1/error$, where the following measure of the error [133] is used

$$error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{TE_i}{cp_i}\right)^2} \tag{3.26}$$

and the exponent is also changed to $\gamma = 1/8$ to calculate the adjusted step size.

3.2.3.2 Gradient Vector Estimation

When using the RK methods to solve the ray equation, for each integration step the gradient of the refractive index of the air squared, ∇n^2, has to be evaluated at multiple interdependent points along the light ray path. However, the calculation of ∇n^2 has not been addressed directly with respect to ray tracing algorithms [122], but rather the general numerical estimation of the gradient vector has been approached in terms of specific applications in the fields of CFD and computer graphics. In ray tracing, accuracy and efficiency are imperative in the gradient vector reconstruction and interpolation processes in order to achieve a satisfactory solution of light behaviour within the gradient-index domain. Gradient evaluation can directly affect the speed and complexity of the integration scheme. This is especially so when the refractive index distribution is provided as a discrete data set [122], which is the case in the present research. If this optical property of the fluid is available as a set of vertex-based mesh values then the gradient field is estimated in two stages: first, ∇n^2 is evaluated based on the embedded mesh and, secondly, the value of ∇n^2 at every point required to trace a light ray is then found via interpolation.

The most cited methods for gradient vector estimation in the CFD field are variants of the Green-Gauss (GG) [137, 138] and Least Squares (LSQR) [139–141] methods. Therefore, in the initial implementation the unweighted LSQR quadratic vertical regression (QVR) has been used to provide the gradient. This choice of method is due to its higher order of accuracy in comparison to the GG and the LSQR linear vertical regression (LVR). The least squares approach exploits the connectivity information of the underlying mesh to perform an approximation of the discrete function values stored at the vertices of the grid, in this case $f(x,y,z) = n(x,y,z)^2$. Since using the vertices of a grid cell as the stencil can lead to an ill-conditioned linear system [140], the origin of the coordinate system is usually translated to the vertex where the gradient vector is required and a vertex-centred formulation is derived. A quadratic approximation to the function of the form

$$\begin{aligned} f(x,y,z) - f(x_0,y_0,z_0) \approx \tilde{f}(x,y,z) = \frac{A}{2}(x-x_0)^2 + \frac{B}{2}(y-y_0)^2 + \frac{C}{2}(z-z_0)^2 + \\ D(x-x_0)(y-y_0) + E(x-x_0)(z-z_0) + F(y-y_0)(z-z_0) + \\ G(x-x_0) + H(y-y_0) + I(z-z_0) + J \end{aligned} \tag{3.27}$$

is assumed, where the subscript 0 indicates the main vertex. The gradient vector within the stencil then has a linear form and is calculated from

$$\nabla f(x,y,z) \approx \nabla \tilde{f}(x,y,z) = \begin{bmatrix} (Ax+Dy+Ez) - (Ax_0+Dy_0+Ez_0) + G \\ (By+Dx+Fz) - (By_0+Dx_0+Fz_0) + H \\ (Cz+Ex+Fy) - (Cz_0+Ex_0+Fy_0) + I \end{bmatrix} \tag{3.28}$$

whereas the gradient vector for the central vertex is simply

$$\nabla f(x_0,y_0,z_0) = \begin{bmatrix} G & H & I \end{bmatrix}^T \tag{3.29}$$

The LSQR method then works by finding the $(n-1)$-dimensional flat subspace in $\mathbb{R}^n$, or hyperplane in 3D, that best fits the data contained within the stencil. This is done via the concept of minimisation of the mean square error induced by the fitting, based on regression distances measured normally to the proposed hyperplane. This minimisation problem can be presented on a matrix formulation as

$$[X][N] = [Y] \tag{3.30}$$

where

$$[X] = \begin{bmatrix} \Delta x_k^2 & \Delta y_k^2 & \Delta z_k^2 & \Delta x_k \Delta y_k & \Delta x_k \Delta z_k & \Delta y_k \Delta z_k & \Delta x_k & \Delta y_k & \Delta z_k & 1 \\ \vdots & \vdots & \vdots & \vdots & & & & & & \\ \Delta x_n^2 & \Delta y_n^2 & \Delta z_n^2 & \Delta x_n \Delta y_n & \Delta x_n \Delta z_n & \Delta y_n \Delta z_n & \Delta x_n & \Delta y_n & \Delta z_n & 1 \end{bmatrix} \tag{3.31a}$$

$$[N] = \begin{Bmatrix} A/2 & B/2 & C/2 & D & E & F & G & H & I & J \end{Bmatrix}^T \tag{3.31b}$$

$$[Y]=\begin{Bmatrix} f(x_k,y_k,z_k)-f(x_0,y_0,z_0) \\ \vdots \\ f(x_n,y_n,z_n)-f(x_0,y_0,z_0) \end{Bmatrix} \tag{3.31c}$$

This determinate linear system of equations can then be solved through

$$[X]^T[X][N]=[X]^T[Y] \tag{3.32a}$$

$$[N]=\left([X]^T[X]\right)^{-1}[X]^T[Y] \tag{3.32b}$$

thus providing as part of the solution the gradient vector at the central vertex. The formulae contained in the previous equations can be found in [139].

In the final implementation an alternative method based on parametric coordinates and shape functions is used to estimate the gradient vector at the vertices of the mesh. This method is available as the function *compute derivative* [142] provided in the PyVista package, which is a Python wrapper for the class vtkGradientFilter [143] from VTK. Since the foundation of the final implementation is constructed based on VTK, and the algorithms contained in the library are computationally optimised for intended applications, the substitution was a sensible choice.

The gradient estimation algorithm implemented in the VTK filter works based on normalised or parametric coordinate systems and shape functions, defined locally for each mesh cell or tetrahedron. The coordinate systems are analogous to barycentric coordinates (a local coordinate system that describes the position of a point inside a simplex, e.g. triangle, based on the positions of the vertices of this simplex) [144]. Whilst commonly described for triangles in two-dimensional space they are also applicable to tetrahedrons. The coordinate transformation relationship between the global (x,y,z) and local (ξ,η,ζ) coordinate systems is given by [145]

$$x=x_0+(x_1-x_0)\xi+(x_2-x_0)\eta+(x_3-x_0)\zeta \tag{3.33a}$$

$$y=y_0+(y_1-y_0)\xi+(y_2-y_0)\eta+(y_3-y_0)\zeta \tag{3.33b}$$

$$z=z_0+(z_1-z_0)\xi+(z_2-z_0)\eta+(z_3-z_0)\zeta \tag{3.33c}$$

or

$$\left\{\begin{matrix} x \\ y \\ x \end{matrix}\right\} = \left\{\begin{matrix} x_0 \\ y_0 \\ x_0 \end{matrix}\right\} + \begin{bmatrix} (x_1-x_0) & (x_2-x_0) & (x_3-x_0) \\ (y_1-y_0) & (y_2-y_0) & (y_3-y_0) \\ (z_1-z_0) & (z_2-z_0) & (z_3-z_0) \end{bmatrix} \left\{\begin{matrix} \xi \\ \eta \\ \zeta \end{matrix}\right\} \tag{3.34a}$$

$$r = r_0 + \mathbf{J}\delta \tag{3.34b}$$

where **J** is the Jacobian matrix.

The shape functions are continuous polynomials, including at the interface between elements. They represent an assumption of linearity to approximate the solution, which in turn will render constant gradients inside each tetrahedron. In the normalised coordinate system, these functions assume the form

$$N_0(\xi,\eta,\zeta) = 1 - \xi - \eta - \zeta \tag{3.35a}$$

$$N_1(\xi,\eta,\zeta) = \xi \tag{3.35b}$$

$$N_2(\xi,\eta,\zeta) = \eta \tag{3.35c}$$

$$N_3(\xi,\eta,\zeta) = \zeta \tag{3.35d}$$

Therefore, if the solution at the vertices of the tetrahedron is represented by a combination of its value and the respective shape function, i.e. $f_i = N_i n_i^2$ for $i = 0,1,2,3$, then the gradient can be approximated as [145]

$$\nabla f_i = \left\{\begin{matrix} \frac{\partial f_i}{\partial x} \\ \frac{\partial f_i}{\partial y} \\ \frac{\partial f_i}{\partial z} \end{matrix}\right\} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \left\{\begin{matrix} \frac{\partial f_i}{\partial \xi} \\ \frac{\partial f_i}{\partial \eta} \\ \frac{\partial f_i}{\partial \zeta} \end{matrix}\right\} = \left(\mathbf{J}^{-1}\right)^T \left\{\begin{matrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{matrix}\right\} n_i^2 = \left(\mathbf{J}^{-1}\right)^T \nabla^n f \tag{3.36}$$

The local gradient vector is computed at each node of each tetrahedron and subsequently averaged based on the respective values contained on cells incident to each node, in order to avoid discontinuities at the interface between cells [146].

3.2.3.3 Gradient Vector Interpolation

The requirement for the refractive index gradient vector at any spatial location in the domain, not just at the CFD mesh vertices or cell centres, is a functional scattered data interpolation problem. This is very common in computer graphics and scientific visualisation [147]. The selection of an interpolation method is usually based on the specific application, approaches commonly employed in the field and, more importantly, on the distribution of the data points [147]. In this research, the data to be interpolated stems from another simulation (CFD) and it always has an underlying mesh. However, all of the aspects of this grid and of the solution itself can vary greatly, depending on the source of the simulation. In order to construct a general light simulation framework, an interpolation method that does not rely on the CFD mesh nor makes any assumptions about it is ideal. Radial basis functions (RBFs) methods fit exactly this meshfree or meshless interpolation category [147].

An RBF is usually a strictly positive definite and radially symmetric basis function centred around a point, referred to as the centre. The centre normally coincides with the locations where data is available [147, 148]. This function depends solely on a distance metric in between the centre and the remaining data points [148]. The interpolation problem of finding a continuous function that fits the available data, i.e. $P(\mathbf{x_i}) = f(\mathbf{x_i})$ for $i = 1,...,N$, is then solved by assuming that this function is a linear combination of radial basis functions, in the form [147–149]

$$P(\mathbf{x}) = \sum_{k=1}^{N} w_k \phi(\mathbf{r}) \tag{3.37}$$

where w_k are the unknown weight coefficients of the linear combination, ϕ is the RBF, $\mathbf{r} = r(\mathbf{x}, \mathbf{x_k})$ is the distance metric, $\mathbf{x_k}$ are the centres and $\mathbf{x}$ are the points where the interpolated function value is requested. The interpolation condition then requires the fitting function evaluated at the input data points to be equal to the input function values, which results in the following system of linear equations [147–149]

$$[A]\{w\} = \{y\} \tag{3.38}$$

where the vector of coefficients $w = [w_1,...,w_N]$ is the solution and the vector $y = [f(\mathbf{x_1}),...,f(\mathbf{x_N})]$ is the data available in the scattered points, i.e. $f(\mathbf{x_i}) = \nabla n^2(\mathbf{x_i})$, $i = 1,...,N$. A distance matrix based on the Euclidean distance between the set of distinct points, $\mathbf{A_{ik}} = \phi(||\mathbf{x_i} - \mathbf{x_k}||_2)$, $i, k = 1,...,N$, is always non-singular and renders a well-posed scattered data interpolation problem [148]. Interpolation is then achieved via solution of the linear system in Eq. 3.38 and substitution of the coefficients obtained in Eq. 3.37.

RBFs can be classified in two different groups based on the values that they assume at any distance from the centre. RBFs with global support are those which never vanish on any interval and an example is the Gaussian, written mathematically as $\phi(\mathbf{r}) = \exp\left(-\epsilon^2 \mathbf{r}^2\right)$, where ϵ

is a positive shape parameter. Conversely, RBFs with compact support become equal to zero a certain distance away from the centre.

Computational libraries tend to provide optimised options of both linear system solvers and interpolation methods [149]. The ALGLIB C++ library [150] is an example, which makes available to users an RBF multilayer (RBF-ML) algorithm implementation, intended for spatial scattered multidimensional interpolation of vector functions. In order to achieve a higher performance comparatively to other available packages (e.g. SciPy), it employs compact Gaussian RBFs of the form

$$\phi(r) = \exp\left(-\frac{r^2}{R_{Base}^2}\right) \tag{3.39}$$

where R_{Base} is a tunable parameter that controls the 'size of the spheres of influence of the points with known function values' [150], i.e. the scale in which the RBF assumes significant values. Small radii result in sharp RBFs and thus degrade the quality of the model in terms of reproducing the function behaviour between data points, whereas large radii tend to result in ill-conditioned matrices, slowing the model construction and evaluation. Because of this a compact, rather than global, RBF is used, which results in sparse distance matrices. The convergence, speed and precision of the linear system solution is then greatly improved.

Global RBF approaches use a dense solver to solve the whole system of equations for all available data points (the interpolation condition), which is expensive in terms of computational costs and storage requirements [147]. The multilayer algorithm builds a true hierarchy of RBF models through iterative updates. In this case the data is only fitted with a least-squares (LSQR) approach (a sparse solver) and the residuals, i.e. the differences between approximated values and function values at nodes, are passed to subsequent iterations. Starting from the initial radius input, after each iteration the radius is halved. Although the models are only inexact approximations of the function, they are robust and correct some of the errors through each layer. The multilayer and LSQR approximation aspects also enable the algorithm to start with a large radius, which is usually difficult to interpolate directly due to ill-conditioning issues. Smoothing is then controlled through the number of layers and the final comparatively smaller radius allows the capture of higher frequency features or finer details of the function [150].

This RBF-ML algorithm within the ALGLIB library is used in the initial implementation of the light simulation framework to interpolate the gradient vector of the refractive index. The setup of the algorithm then involves the tuning of two parameters: the initial radius of the Gaussian and the number of layers through which the radius will be successively decreased [150]. Taking into consideration the reproduction of the input data with a low level of noise, the radius is then selected to be equal to the global average of the mean distance between each vertex and respective nearest neighbours. This is determined to provide a good coverage of the interpolation area. Also, five layers are established as an appropriate trade-off between sufficient levels of smoothing in the interpolated solution and relatively fast RBF model construction and interpolation evaluation. Although not exactly following the recommendations in terms of radius and layers available in [150], previous interpolation tests performed with the proposed setup proved to be sufficiently accurate for the current analysis. This is a per data sample tuning, performed before each and every analysis involving a different flow case.

In the final implementation, to interpolate the gradient of refractive index use is made of algorithms available within the Visualization Toolkit [74]. VTK provides an interpolation filter [151] that is very similar to the Gaussian RBF. It works by probing a scattered point cloud where the function values are stored with another point data set and, for each of the points of the latter, finding a number of nearest nodes or all nodes within a certain radius. A kernel of choice is then applied to this neighbourhood information, the Gaussian being one of the options, generating the interpolated function values [143]. The spherical Gaussian interpolation kernel has the form [152]

$$k(\mathbf{r}) = \exp\left(-\left(\frac{s\mathbf{r}}{R}\right)^2\right) \tag{3.40}$$

where, as before, r is the Euclidean distance between the probing point and respective neighbouring nodes within the radius R and s is a sharpness parameter that controls the decay of the kernel [152]. This point interpolator filter with the Gaussian kernel is initially employed in the final implementation.

The fact that the vtkPointInterpolator filter invariably uses a point locator and that the vtkGradientFilter can be set to calculate the gradient vector on each mesh cell, instead of on each vertex, suggested that a different approach could be used for interpolation, as an attempt to improve the computational efficiency of the ray equation solution process. The times required to compute the gradient field on all of the cells or vertices of the mesh are of the same order of magnitude. If the gradient is assumed to be constant inside each tetrahedron, interpolation then becomes a simpler cell location query, which also has approximately the same computational costs as the point location already performed in the VTK interpolation filter. Therefore, using gradient computation and interpolation on a cell basis reduces considerably the time required to evaluate the refractive index function throughout the light ray path, since no Gaussian RBFs or kernels need to be constructed. Given that the vtkGradientFilter based on parametric coordinates and shape function already contains this assumption of linearity within cells, the final implementation is then modified to implement gradient estimation and interpolation per cell.

The cell location requirement is typical of problems involving continuous vector fields stored in unstructured grids that need to be numerically integrated to obtain particles trajectories [153]. Since the integration is usually based on interpolation requests at arbitrary points within the data set, the cells containing these points need to be located via spatial data structures and assessed for confirmation of point inclusion [153]. An optimised computational performance is then indispensable, since the vector field data, the cell candidates searching range and the point interpolation set can all be very large. Garth and Joy [153] presented a *celltree* spatial data structure, implemented in VTK as the vtkCellTreeLocator class [154], that outperformed the other cell locators available in VTK ([155–158]) in terms of build time and memory overhead [153]. These built-in locators are based on octree or kd-tree subdivision approaches, whereas vtkCellTreeLocator is based on the bounding interval hierarchy space partitioning scheme [153]. It can be applied to general unstructured meshes composed of any cell types and also supports CPU/GPU parallel implementations [153]. As such, it is ideal for the task of global interpolation of variables over the mesh domain via cell location and local cell-based interpolation. This is a common setting used in vector-field visualization from unstructured data stemming from simulation codes [153]. The vtkCellTreeLocator is then employed with the vtkGradientFilter to perform the refractive index interpolation in the final implementation of the light simulation framework.

3.2.3.4 Automatic Discontinuity Detection and Integration Refinement

The focus of this research is on tracing light rays passing through a shock wave or waves that form in high-speed flow over aerodynamic bodies. For much of the flow field the variations in density are negligible and the high density gradient region is confined to a thin region around the shock. This specific characteristic of high speed compressible flow is challenging for ray tracing solvers. The RK-type numerical integration schemes and their automatic step size control and adjustment procedures only handle local domain information. This can cause an issue when two adjacent integration points fall in the regions upstream and downstream of the shock wave where local variations in density at each point are very low, despite the difference between the density values at the two points being high. The zero gradient of density condition in the regions around each integration point leads to an integration truncation error in the RK scheme that is below the preset tolerance. Thus the tracing continues without reduction of the step size, missing the shock discontinuity and, consequently, the physical light ray bending.

In these applications where the function defining the differential equation (∇n^2) is not continuous everywhere, the numerical integration solution invariably crosses the surface of discontinuity at various points [133]. The user can then treat these cases with an additional code that detects the singularity and handles it appropriately [133]. Therefore, in the initial implementation of the light simulation, a modified methodology is established to ensure that the dramatic density change that occurs across a shock wave [22] is detected and the step size reduced accordingly. The air density is checked continuously along a light ray path via interpolation at the beginning and end of every step, with the percentage change between the two values compared to a preset threshold of density variation. If the percentage change exceeds the threshold, the shock density discontinuity has been clearly missed within the step. It is now detected and the current step is rejected as it is too large. A bisection approach is taken to reduce the step so that a suitable smaller step size is identified, which yields a density percentage change across the step that lies within the threshold. The logic of the approach is described in Figure 3.8.

This corrects the integration so that both ends of this step are on the same side of the shock as the origin of the cast light ray. The new smaller step size is then used to advance the numerical integration and provides a suitable level of refinement as the light ray crosses the shock wave. The interpolation of the air density along the integration path is performed using a hyperplane fitting approach, similar to the Least Squares method presented in Section 3.2.3.2 but without requiring an estimation of the mean squared error. This process is efficient even on unstructured polygonal meshes, which is a key requirement due to the number of times this interpolation is performed.

In the final implementation the DOP853 algorithm in SciPy handles the discontinuity detection and integration refinement for all of analyses. In this case the step size control mechanism within the algorithm performs the detection automatically and adjusts the step size accordingly [133].

3.2.3.5 Intersection Computation

An important aspect of the ray tracing is to be able to correctly capture the intersection with solid surfaces. In the initial implementation, the point at which a light ray leaves the flow domain and enters the solid surface during a step of the RKF 4(5) scheme is detected using an internal function of the CGAL library [99]. This function queries if a certain point is located inside or outside the constrained tetrahedralisation. In order to obtain accurate results for the position of the intersection between a single light ray and the solid surface, the nonlinearity of the light ray path has to be accounted for since the density gradients in the boundary layer adjacent to the surface are usually high.

Following the technique devised by Fine [126] for Runge-Kutta-Nyström methods, the solution

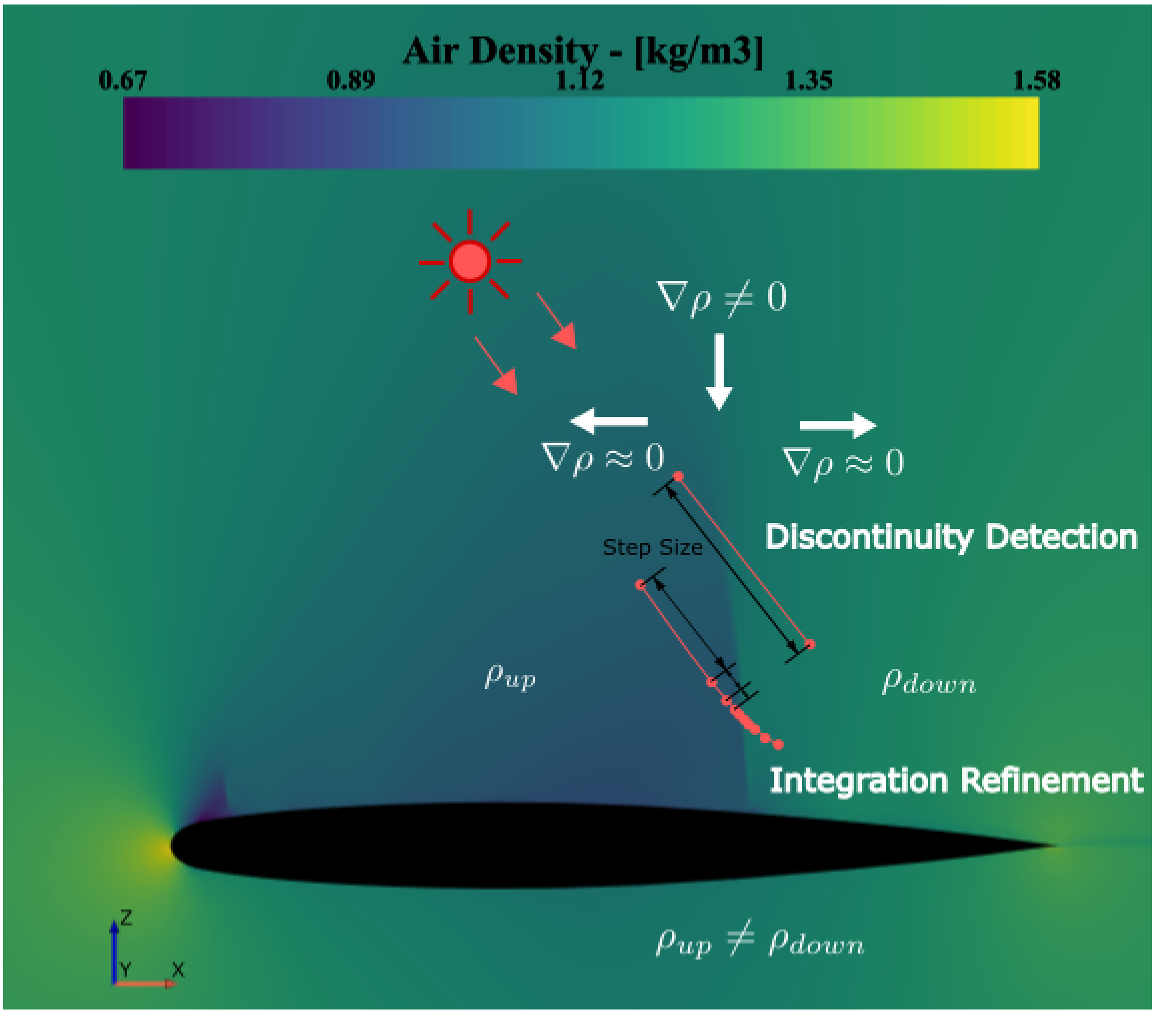


Figure 3.8: Schematic of the automatic discontinuity detection and integration refinement approach.

and its first and second-order derivatives at the beginning $(\vec{r}_n, \dot{\vec{r}}_n, \ddot{\vec{r}}_n)$ and end $(\vec{r}_{n+1}, \dot{\vec{r}}_{n+1}, \ddot{\vec{r}}_{n+1})$ of an integration step are used to construct interpolation polynomials, or Hermite interpolants. These are continuous intermediate trajectories between integration points of the form [126]

$$\vec{r}(t_n+\xi h) = d_1\vec{r}_n + d_2\dot{\vec{r}}_n + d_3\ddot{\vec{r}}_n + d_4\ddot{\vec{r}}_{n+1} + d_5\dot{\vec{r}}_{n+1} + d_6\vec{r}_{n+1} = \vec{r}(t_n+\xi h) + O\left(h^6\right) \tag{3.41a}$$

$$\dot{\vec{r}}(t_n+\xi h) = d_1^{'}\vec{r}_n + d_2^{'}\dot{\vec{r}}_n + d_3^{'}\ddot{\vec{r}}_n + d_4^{'}\ddot{\vec{r}}_{n+1} + d_5^{'}\dot{\vec{r}}_{n+1} + d_6^{'}\vec{r}_{n+1} = \dot{\vec{r}}(t_n+\xi h) + O\left(h^5\right) \tag{3.41b}$$

where

$$d_1 = 1 - 10\xi^3 + 15\xi^4 - 6\xi^5 \tag{3.42a}$$

$$d_2 = (\xi - 6\xi^3 + 8\xi 64 - 3\xi^5)h \tag{3.42b}$$

$$d_3 = (\xi^2 - 3\xi^3 + 3\xi^4 - \xi^5)h^2/2 \tag{3.42c}$$

$$d_4 = (\xi^3 - 2\xi^4 + \xi^5)h^2/2 \tag{3.42d}$$

$$d_5 = (-4\xi^3 + 7\xi^4 - 3\xi^5)h \tag{3.42e}$$

$$d_6 = 10\xi^3 - 15\xi^4 + 6\xi^5 \tag{3.42f}$$

$$d_1^{'} = (-30\xi^2 + 60\xi^3 - 30\xi^4)/h \tag{3.42g}$$

$$d_2^{'} = 1 - 18\xi^2 + 32\xi^3 - 15\xi^4 \tag{3.42h}$$

$$d_3^{'} = (2\xi - 9\xi^2 + 12\xi^3 - 5\xi^4)h/2 \tag{3.42i}$$

$$d_4^{'} = (3\xi^2 - 8\xi^3 + 5\xi^4)h/2 \tag{3.42j}$$

$$d_5^{'} = -12\xi^2 + 28\xi^3 - 15\xi^4 \tag{3.42k}$$

$$d_6^{'} = (30\xi^2 - 60\xi^3 + 30\xi^4)/h \tag{3.42l}$$

The ray position variable can then be evaluated backwards (from $\xi = 1$ and $t = t_n + h$ to $\xi = 0$ and $t = t_n$), starting from the last computed integration point until it reaches the boundary between the aerodynamic body and the fluid flow. When this condition is established, the point is then considered to be inside the constrained tetrahedralisation. The intersection between the light ray and the object is thus determined, along with the ray direction variable and the mesh cell in which the light ray hit the surface. This technique does not result in additional expensive computations, since all the information needed to compose the coefficients of the polynomials derive from the integration step, and the accuracy of the interpolants is relatively close to that of the inputs used in its derivation [126].

In the final implementation using the DOP853 scheme, the intersection is detected by a built-in filter of the VTK [74] library. The filter assumes a closed solid surface, and employs a ray casting statistical methodology. Through the method, multiple random rays are shot to accurately find the mesh cells crossed in their paths, and ultimately determine if the queried point is inside or outside the surface. The returning in/out mask is then employed as the integration termination event of the SciPy integration function [136], when the former occurs.

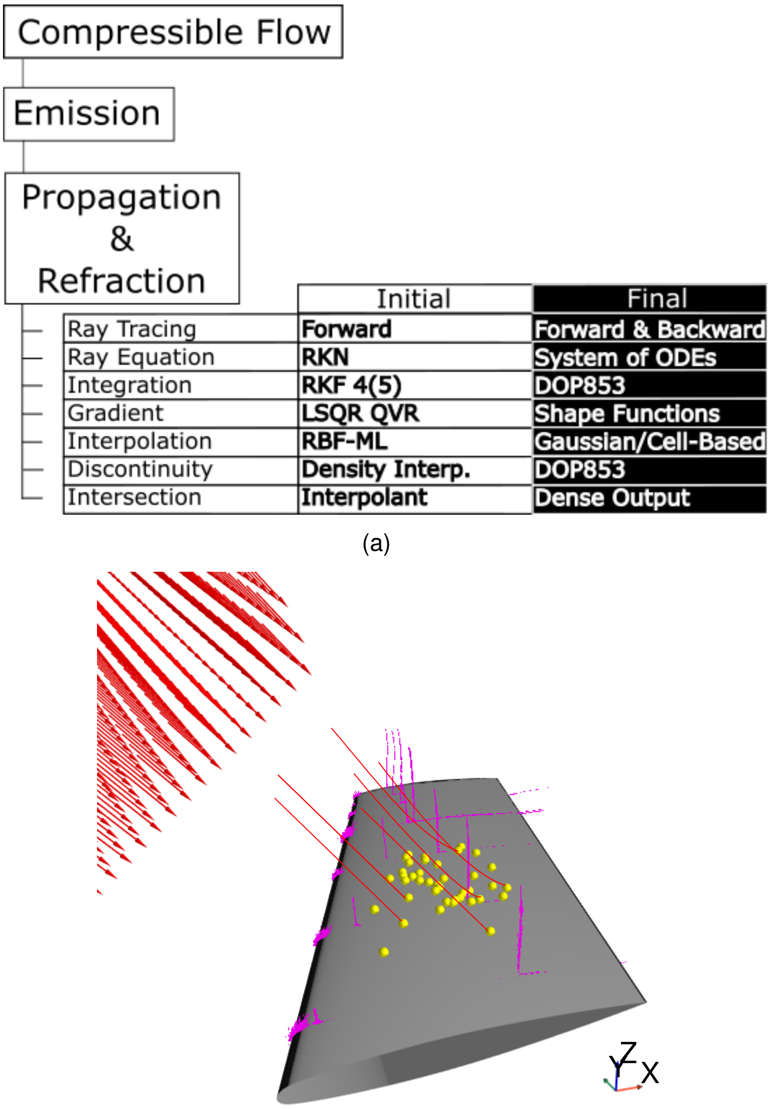
Similar to the method of Fine [126], the RK method of the DOP853 scheme is characterised as 'dense output formulas'. This means that in addition to the solution itself, each integration step of the scheme provides a computationally cheap Hermite interpolation for the whole integration interval [133]. Thus, as before, continuous differentiable approximations of the form $y_n(t_n + \xi h)$ in $0 \leq \xi \leq 1$ for all the state variables (light ray position and direction) are available, requiring none or minor additional function evaluations. In the DOP853 specifically, a simple boot-strapping three-stage process is added to the original algorithm [133], resulting in a seventh order interpolant. The intersection between the polynomial and a surface cell is then easily computed using simple geometry calculations and it is stored along with the ray direction variable and the cell normal at the hit location.

3.2.4 Light Propagation and Refraction Model

In the light simulation framework light propagation and refraction are modelled via solution of the Ray Equation from Geometrical Optics in both phases of implementation. In the initial implementation, a traditional RKN method (the RKF 4(5) [128]) is used to integrate directly this second order ODE during forward (only) nonlinear ray tracing. The right-hand side of this equation, i.e. the gradient of the refractive index medium, is estimated by the unweighted LSQR quadratic vertical regression [139, 140] method at the vertices of the underlying CFD mesh. These gradient vectors are then interpolated along each light ray path using the RBF-ML algorithm from the ALGLIB C++ library [150]. To ensure that the discontinuous gradient typical of a shock wave surface is not missed during integration, the density field is interpolated using a hyperplane fitting approach. When density jumps are detected along the integration, the step size is automatically reduced and the light ray path is forced to cross the shock wave again, but with a higher level of refinement. Finally, intersections of light rays with any solid object present in the scene are detected by internal function of the CGAL library [99] and the point of intersection is calculated using computationally cheap interpolation polynomials constructed from the RKF solution.

In the final implementation, a RK method of higher order of accuracy (DOP853 [133, 136]) performs the numerical integration both forward and backward by transforming the ray equation into a system of two ODEs. The gradient of refractive index is computed at the vertices of the mesh through a built-in class from VTK and function from PyVista [142, 143]. The class is based on local parametric coordinates and shape functions for each mesh cell. Similarly, interpolation of gradient vectors was originally achieved using an internal filter of VTK [151]. This is replaced with a more computationally efficient cell locator [153, 154], also contained in VTK. Discontinuity detection and integration refinement is handled solely by the DOP853 algorithm, which adjusts the step size automatically. Finally, intersections are detected by another built-in filter of point location from the VTK library and, similar to the previous interpolants, the DOP853 scheme provides cheap dense output formulas that are used to determine the intersection points.

Figure 3.9 summarises these decisions related to the modelling of light propagation and refraction in the initial and final implementations. Figure 3.9(a) shows the development of the light simulation framework along the roadmap of this Chapter and Figure 3.9(b) illustrates the final process of light propagation employed.



	Initial	Final
Ray Tracing	**Forward**	**Forward & Backward**
Ray Equation	**RKN**	**System of ODEs**
Integration	**RKF 4(5)**	**DOP853**
Gradient	**LSQR QVR**	**Shape Functions**
Interpolation	**RBF-ML**	**Gaussian/Cell-Based**
Discontinuity	**Density Interp.**	**DOP853**
Intersection	**Interpolant**	**Dense Output**

(a)

(b)

Figure 3.9: (a) Roadmap of the current progress in terms of light modelling, and (b) graphic representation of the light propagation and refraction models used in the final implementation.

3.3 Light Reflection

After being emitted from the directional light source and propagated and refracted through the gradient-index medium, the photons interact with the wing and are either reflected or absorbed

[102]. Conductors - metals - 'don't transmit light, but some of the incident light is absorbed by the material and turned into heat' [103]. In order to perform a physically-based simulation of light reflection from the wing and accurately predict a realistic synthetic image that represents the scene containing the shadow of a shock wave, radiometry and respective electromagnetic radiation variables and principles are again employed.

Forming the basis of rendering and global illumination algorithms [102–104], the radiance (energy arriving at or leaving a point on a surface per unit time per unit surface area and per unit solid angle) is the variable of interest to be computed. Just as cameras record radiance, i.e. light from multiple directions measured during an exposure time and stored in an array of pixels distributed over a small surface area, and human eyes sense a compatible quantity [100], so does the light framework of this research.

As explained in Sections 3.1.3 and 3.1.4, although the radiance embodied by light rays changes during refraction [102], this light scattering effect is not captured by the nonlinear ray tracing process used in this work. Since in compressible flow the refractive index is close to one, the effect is assumed to be small. Instead, the source emits radiant flux, which is propagated via the photons. As radiance can be defined anywhere in the domain [104], it can then be evaluated at the wing surface from the energy distribution produced from the photons that were showered on the scene, and at the image plane of the virtual camera from the reflection that takes place at the wing. These two directions of analysis, from sun to the wing and also from the camera to the wing, explain the existence of both the forward and backward ray tracing techniques, performed in each respective direction.

By referring to the definition of radiance, Eq. 3.6 presented in Section 3.1.2, it becomes evident that radiance depends not only on the point where it is evaluated (p), but also on the direction ($\overrightarrow{\omega}$) and on the wavelength of light (λ), i.e. $L = L(p, \overrightarrow{\omega}, \lambda)$. Both the directional dependence, which determines the solid angle ($d\omega$) and projected area perpendicular to it ($dA^{\perp} = cos(\theta)dA$), and the spectral dependence of radiance, affect how an objects reflects light and its appearance. This includes the perceived colour and the presence of highlights for example [103].

As in the illumination stage, the nature of light is again treated from the geometrical optics perspective. This way, the radiant energy behaviour at reflection is assumed to be linear, energy conserving and in the steady state [103]. Furthermore, diffraction and interference effects are neglected. Therefore, the (differential amount of) radiance arriving at a point on the surface (dL_i) from an incoming direction ($\overrightarrow{\omega_i}$) is directly proportional to the radiance reflected (dL_o) in another direction ($\overrightarrow{\omega_o}$) [104]

$$dL_o(p,\overrightarrow{\omega_o},\lambda) = f_r(p,\overrightarrow{\omega_i},\overrightarrow{\omega_o},\lambda)L_i(p,\overrightarrow{\omega_i},\lambda)cos(\theta_i)d\omega_i \tag{3.43}$$

where the Bidirectional Reflectance Distribution Function or simply BRDF (f_r) is the constant of proportionality between the incident and exitant radiances. It is a formal abstraction that describes the mechanisms of light reflection from a surface made of a certain material [103, 104]. BRDFs neglect the effect of subsurface light transport that occurs in translucent materials, a suitable abstraction for metals [103]. Figure 3.10 depicts the angular dependence of incoming and reflected radiances, as represented in Eq. 3.43.

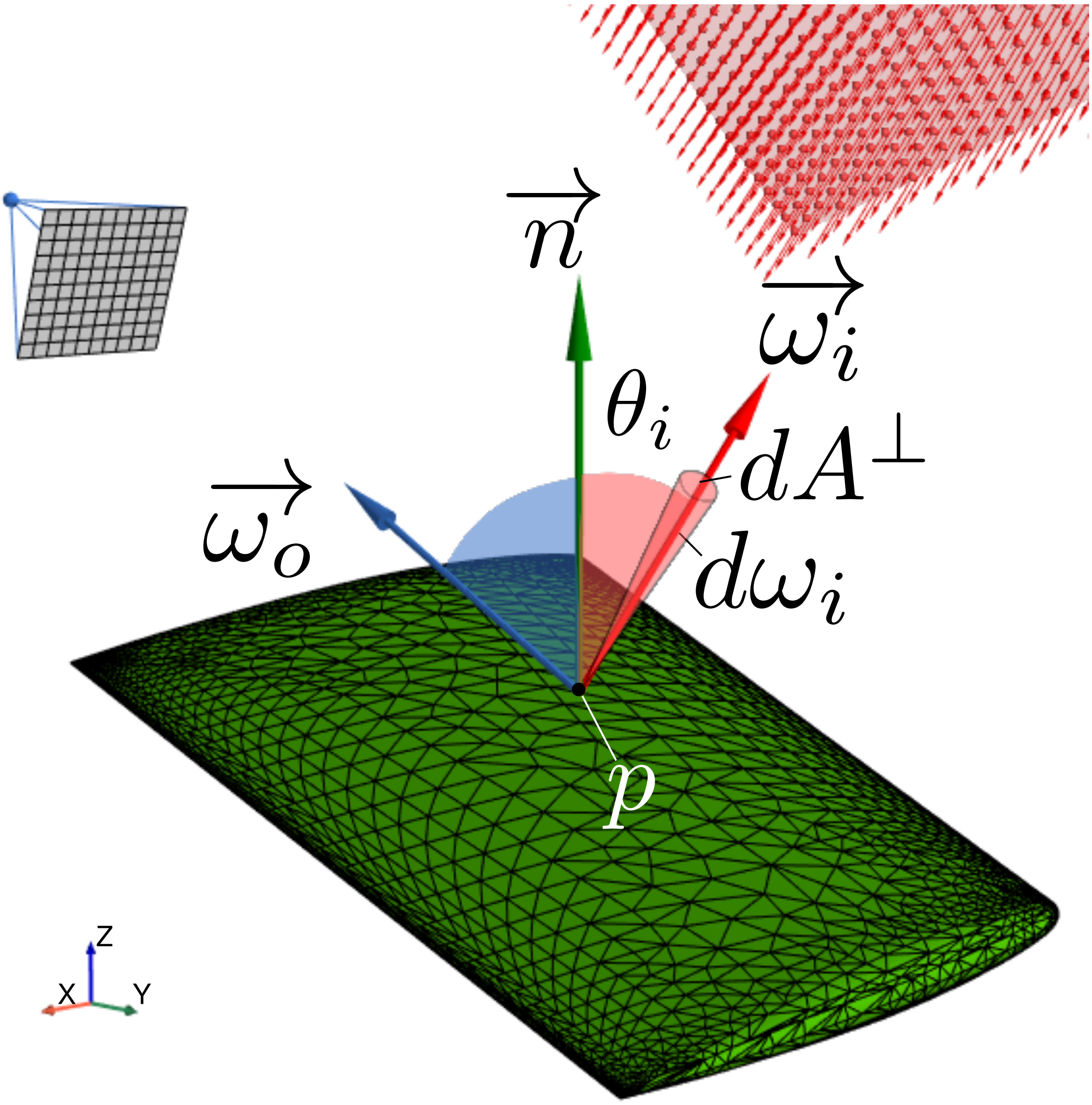


Figure 3.10: Angular dependence of incoming and reflected radiances evaluated at a point.

3.3.1 Physically-Based Rendering

BRDF formulae are devised from various types of sources, including: measured data of real materials obtained in experimental facilities; pure qualitative analysis and descriptions of the scattering phenomenon; detailed physical or geometric optics models; or even from microgeometry simulations [103]. Thus the formulas range in the level of complexity from tabular and readily available data, to approximate values or complex equations and respective solution procedures [103]. However, in order to be considered an appropriate representation of real materials, the BRDF must not be composed of only an ideal diffuse or perfect specular components [102]. The BRDF is therefore usually a combination of these two, glossy specular and retro-reflection, and possibly consider effects of anisotropy, common in brushed metallic surfaces [103]. Figure 3.11 represent the reflection behaviour of the four types of materials that are most commonly modelled.

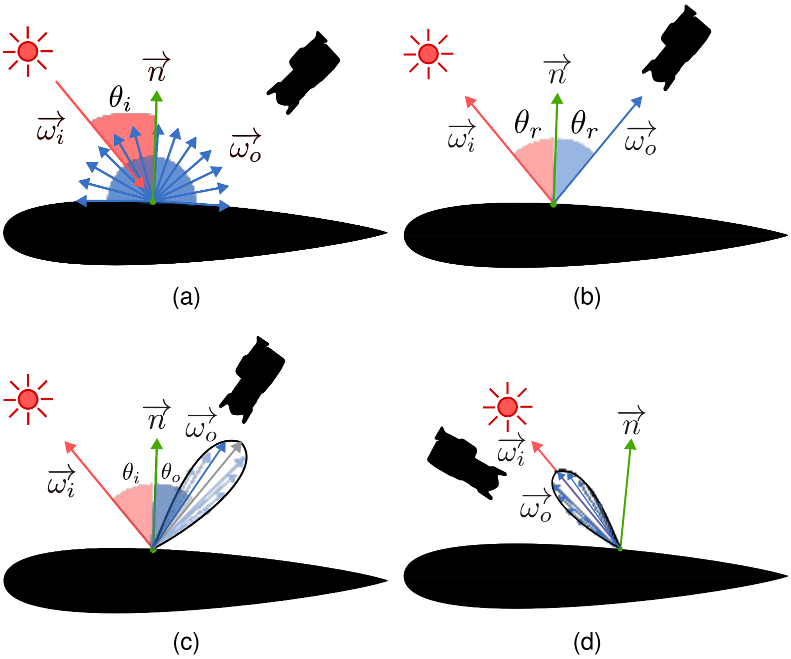

$\overrightarrow{n}$
θ_i
$\overrightarrow{\omega_i}$
$\overrightarrow{\omega_o}$
(a)
$\overrightarrow{n}$
θ_r
θ_r
$\overrightarrow{\omega_i}$
$\overrightarrow{\omega_o}$
(b)
$\overrightarrow{n}$
$\overrightarrow{\omega_o}$
θ_i
θ_o
$\overrightarrow{\omega_i}$
(c)
$\overrightarrow{\omega_i}$
$\overrightarrow{n}$
$\overrightarrow{\omega_o}$
(d)

Figure 3.11: Representation of (a) ideal diffuse, (b) perfect specular, (c) glossy specular and (d) retro-reflective light distributions.

Perhaps the simplest BRDF is that of a perfect diffuse or Lambertian reflecting surface, which scatters the incident light equally in all directions [103, 104]. To model this type of material, which is not necessarily realistic but a good approximation for matt surfaces such as those covered with a flat paint [103, 104], the concept of reflectance is used. Reflectance (ρ) is defined as the ratio between reflected and incident radiant fluxes, written as [104]

$$\rho(p,\Omega_i,\Omega_o) = \frac{d\Phi_o}{d\Phi_i} = \frac{\displaystyle\int_{\Omega_o}\int_{\Omega_i} f_r(p,\overrightarrow{\omega_i},\overrightarrow{\omega_o})L_i(p,\overrightarrow{\omega_i})cos(\theta_i)cos(\theta_o)\,d\omega_i d\omega_o}{\displaystyle\int_{\Omega_i} f_r(p,\overrightarrow{\omega_i},\overrightarrow{\omega_o},\lambda)L_i(p,\overrightarrow{\omega_i})cos(\theta_i)\,d\omega_i} \tag{3.44}$$

where $\Omega_{i,o}$ refers to the hemisphere above p. Since a diffuse BRDF is independent of direction, the following relationship is established after integrating Eq. 3.44 [104]

$$f_{r,d}(p) = \frac{\rho_d(p)}{\pi} \tag{3.45}$$

Since the radiant power of the light source has been integrated over the visible spectrum and is represented as a triplet on the standard RGB or CIE XYZ colour spaces, the wavelength dependence of the reflected light is also embedded in the representation of the diffuse reflectance (and respective BRDF) as tristimulus values. These values are then simply the portion ($0 \leq \rho_d(p) \leq 1$) of the incident radiant energy that is reflected back to the environment, independently of the direction that it is being visualised. This principle is applied to the three bands of the visible spectrum covered by each of the colour channels.

BRDFs of real materials are usually a linear combination of multiple models, e.g. of diffuse and specular reflection [104]. Moreover, in order for the BRDF to be considered physically-based, these models must embed the energy conservation principle and are usually based on the concept of microfacets [102, 103, 159]. An approach based on the geometric optics field of study, the microfacet model, as the name suggests, consider a surface with arbitrary roughness to be composed of a statistical distribution of microscopic perfectly (specular) reflective mirrors or facets [103, 159]. The individual behaviour of each facet then contributes to the overall surface light scattering. This type of model depends, primarily, on the facet collection alignment or, in other words, the microscopic distribution and orientation of microfacets [103, 159]. It also depends on local lighting effects, such as occlusion, shadowing and interreflection between facets [103]. Occlusion and shadowing occur when a microfacet is effectively masked from the viewing system and from the light source, respectively, by another neighbouring facet [103]. These models also take into account the amount of the incoming energy that is actually reflected, instead of absorbed or transmitted, thus distinguishing dielectric and conductor materials [159].

One of the most recent microfacet models was developed by Walter et al. [160], as an extension of the classical microfacet models due to Torrance and Sparrow [161] and Cook and Torrance [162] for metallic reflective surfaces. Variants of the classical models are now widely used in computer graphics to model different characteristics of real surfaces [160]. These models are given as [103, 116, 160]

$$f_r(p,\overrightarrow{\omega_o},\overrightarrow{\omega_i}) = \frac{\mathbf{D}(\overrightarrow{\omega_h})\mathbf{F_r}(\overrightarrow{\omega_o})\mathbf{G}(\overrightarrow{\omega_o},\overrightarrow{\omega_i})}{4cos(\theta_o)cos(\theta_i)} \tag{3.46}$$

where $\mathbf{D}$ is the *distribution* function that describes statistically the distribution of microfacet normals with respect to the macrosurface normal; $\mathbf{F_r}$ is the *Fresnel* term that determines the ratio of light reflected by each microfacet; $\mathbf{G}$ is the *geometric attenuation* term which accounts for occlusion and shadowing between adjacent microfacets; and ω_h is the half-angle vector in between $\overrightarrow{\omega_i}$ and $\overrightarrow{\omega_o}$ that the normals of the microfacets have to be aligned with in order for them to reflect light as perfect specular microsurfaces [103]. An extensively applied formula [160] for the Fresnel term is the Schlick's approximation, written as [159]

$$\mathbf{F_{Schlick}}(\overrightarrow{\omega_h},\overrightarrow{\omega_o},F_0) = F_0 + (1-F_0)(1-(\overrightarrow{\omega_h}\cdot\overrightarrow{\omega_o}))^5 \tag{3.47}$$

where F_0 is the surface reflectivity at normal incidence, data available from measurements performed with a wide range of materials and, for computational purposes, presented in the sRGB space as a colour triplet. For aluminium, $F_0 = (0.96, 0.96, 0.97)$ [159].

One proposed distribution function is the Trowbridge-Reitz GGX, defined by [159]

$$\mathbf{D_{GGXTR}}(\overrightarrow{n},\overrightarrow{\omega_h},\alpha) = \frac{\alpha^2}{\pi\left(\left(\overrightarrow{n}\cdot\overrightarrow{\omega_h}\right)^2\left(\alpha^2-1\right)+1\right)^2} \tag{3.48}$$

where $\overrightarrow{n}$ is the surface normal and $\alpha = [0,1]$ is a measure of the surface roughness.

An existing geometric attenuation term is a combination of GGX, Schlick-Beckmann and Smith approximations, given as [159]

$$\mathbf{G}(\overrightarrow{n},\overrightarrow{\omega_o},\overrightarrow{\omega_i},k) = \mathbf{G_{SchlickGGX}}(\overrightarrow{n},\overrightarrow{\omega_o},k)\mathbf{G_{SchlickGGX}}(\overrightarrow{n},\overrightarrow{\omega_i},k) \tag{3.49a}$$

$$\mathbf{G_{SchlickGGX}}(\overrightarrow{n},\overrightarrow{\omega_{i,o}},k) = \frac{\overrightarrow{n}\cdot\overrightarrow{\omega_{i,o}}}{\left(\overrightarrow{n}\cdot\overrightarrow{\omega_{i,o}}\right)(1-k)+k} \tag{3.49b}$$

$$k = \frac{(\alpha+1)^2}{8} \tag{3.49c}$$

The specific details of these approximations and other variants of the microfacet model due to Cook and Torrance [162] are thoroughly explained in the above-mentioned references but are outside the scope of this thesis.

3.3.2 Light Reflection Model

In the initial implementation of the light simulation framework, the microfacet BRDF is used to model light reflection in all of the images generated. In the final implementation, a perfect diffuse material is assumed for all of the direct rendering analyses involving simplifications of the physical problem, and the microfacet BRDF is again employed in the remaining simulations. Figure 3.12 summarises these decisions related to the modelling of light reflection in the initial and final implementations, and shows the development of the light simulation framework along the roadmap of this Chapter.

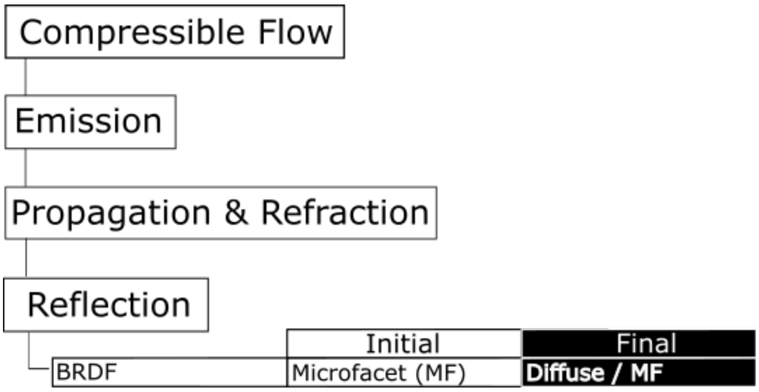


Figure 3.12: Roadmap of the current progress in terms of light modelling.

3.4 Image Synthesis

Image synthesis is the process of creating a 2-D picture of a 3-D world [118]. The computational simulation of this process is entirely based on the so-called Radiance Equation [163] and its solution as a means to achieve the complete description of the light distribution in a scene. The equation stems from transport theory viewed from the radiometry perspective [163], with analogous mathematical treatments in the radiative heat transfer literature [164] and many other different fields such as automobile traffic flow and gas dynamics [163]. It models the light energy flow in a particle-based approach. According to this approach, energy is transported through a medium by the photons and thus the energy at a certain location and at a certain time depends on the number of particles passing through that region. This information can be calculated in every direction and wavelength, and for any point in the environment, including the location in the computational model where a photorealistic image is to be formed [163].

The theory assumes that particles travel at the same speed and do not interact with each other, i.e. light is treated from the Geometrical Optics perspective [163].

3.4.1 Rendering Equation

In computer graphics, all of the rendering methods attempt to model the same physical phenomena [164], i.e. light transport and scattering from surfaces, by finding efficient and accurate approximate solutions to an approximate form of the more general Radiance Equation [163]. This form neglects polarisation, fluorescence and phosphorescence effects, as well as volumetric emission, scattering and absorption. Polarisation is of importance for coherent light, such as lasers. Phosphorescent materials briefly trap the incident light before re-emitting it to the environment diffusely. Similarly, fluorescent materials absorb light of a certain wavelength and quickly re-emit it at another frequency in a diffuse way. Therefore for unpolarised light and standard materials, neglecting these phenomenon is acceptable. These assumptions simplify the solution process and mean that unpolarised monochromatic solutions can be simulated independently and linearly combined a posteriori through digital image processing. Time, in turn, does not influence these solutions and the radiant energy modelled is considered to be time-averaged. The image synthesis is assumed to occur in vacuum, or in a homogeneous medium which does not participate in the light transport [163, 164]. The treatment of the refractive index of the medium, which depends on the wavelength of light, and its possible inhomogeneous distribution is left for path integral techniques [164] and scattering functions [163], such as the Ray Equation described in Section 3.2.2.

The approximate Radiance Equation, also referred to as the Vacuum Time-Invariant Gray Radiance Equation (VTIGRE) [163] or simply Rendering [102, 164] or Reflection [103, 104] Equation, is expressed as

$$L_o(p,\vec{\omega_o}) = \int_{2\pi^+} f_r(p,\vec{\omega_i},\vec{\omega_o})\left[L_i(p,\vec{\omega_i})cos\theta_i d\omega_i\right] \tag{3.50}$$

where the radiance reflected (L_o) in a certain direction ($\vec{\omega_o}$) from a point (p) in the scene depends on the radiance incoming (L_i) from all the directions ($\theta_i,\vec{\omega_i}$) in the hemisphere above it ($2\pi^+$) and the reflective properties of the material that composes the surface itself (f_r). This equation describes the relationship between incident and outgoing distributions of light at a point [103] and comprises the quantity computed in ray tracing [104].

3.4.2 Global Illumination

For a digital image to be considered realistic it has to be a faithful representation of what it models, a photograph of the real world, within computational limitations [102]. In the computer graphics industry, realism is achieved through global illumination algorithms [102] that perform physically based rendering [163]. These algorithms simulate all the light scattering occurring within a computational environment based on (some) physics theories and principles, and render images according to an accurate prediction of the resultant light distribution everywhere in the model. The model includes specifications of geometric objects, their reflective materials, light sources illuminating the scene and the viewing system recording its image. In global illumination, not only light arriving at a point from the source (*direct illumination*), but also light that has undergone single or multiple reflections, refractions or transmissions before reaching the point (*indirect illumination*) is taken into account in the computation of the radiance. These scattering events take place during interactions with objects contained in the scene and invariably reduce the amount of energy that will reach the point. Nevertheless, indirect illumination can still make a comparatively substantial contribution to the final colour rendered at an image pixel and to the reproduction of visual effects necessary to achieve realism [102].

The global illumination algorithms considered in this research render an image through an implicit sampling approach, as opposed to the explicit approximation approach of radiosity algorithms [163]. Radiosity techniques compute an explicit function of the unknown radiance distribution [163], in a condition of equilibrium in terms of light transport [102]. This is done using a finite element mesh, or tessellated representation of the geometric model composed of small patches, upon which this function is evaluated independently of the viewing conditions to obtain the final light distribution in the scene. The evaluation is performed by solving a set of linear equations. This class of algorithms can efficiently handle simple models with purely diffuse reflective surfaces. In more complex models, they become computationally expensive and have difficulty in reproducing sharp features in the illumination, and are thus impractical [102]. Radiosity algorithms are therefore not considered in this research.

Instead, by evaluating point samples of radiance at locations and in directions relevant to the image synthesis process, e.g. at the surface of an object in the scene and/or on the virtual image plane, a radiance function that satisfies the Rendering Equation is reconstructed for each viewing direction [163]. The term *virtual* refers to a fictitious image plane construct that does not influence the scene or the light distribution within it [163]. Ray-tracing-based view-dependent point-sampling techniques are extremely advantageous considering the flexibility in terms of geometry, physics of reflection and illumination models that can be simulated by these algorithms.

The Classic Ray Tracing [165] algorithm works by tracing linear infinitesimal beams of light from the viewing system backwards towards the scene (backward ray tracing). It then finds the nearest object present in the model intersected by this primary ray, as it is commonly called, and then traces a second ray, referred to as the shadow ray, from the location of intersection towards the light sources defined to illuminate the scene. If the shadow ray hits an object, light is blocked from the intersection point. In other words, if the visibility test returns a negative, the pixel in the image plane where the primary ray was assigned from is rendered black. Otherwise, the radiance along the line of sight is calculated based on the viewing and illumination direction, on the geometrical characteristics of the object intersected and the scattering properties of its material [102].

In the nonlinear light propagation phenomenon concerning the formation of a caustic due to the presence of shock waves in compressible flow, the drawback of employing only the ray tracing technique to solve the rendering equation is the complexity or possibly even the inability of connecting the two stages of light propagation to perform reflection, shading and shadowing computations. Employing either the forward or the backward approach, one would be faced with the same challenge of ’Given two points in space, find a curved ray that obeys the underlying laws of nonlinearity and passes through the two defining points’ [166]. In backward linear ray tracing, for example, the concept of shadow rays is a useful visibility test to determine if a hit point is in shadow or if it should be appropriately shaded. Given the nonlinearity, the shadow ray direction, usually from hit point to light source in linear ray tracing, is not easily determined anymore, and the shadow ray may not even exist, since the shadowing effect can be due to light shift, instead of simple light blockage [166].

Arvo [167] refined the classical ray tracing methodology to try and address exactly this difficulty in computing indirect illumination and rendering caustic effects [102]. This difficulty is due to the fact that reflected and refracted shadow rays are not defined by a single direction, and thus the determination of their origin or if they actually come from the light source and contribute to the indirect illumination is not straightforward. The solution adopted in [167] was the construction of an illumination map by means of forward ray tracing (from the light source(s) towards the scene) before actually using backward ray tracing to perform the shading calculations and generate a computational image. The *light source pass* or *illumination phase* concerned the tracing of multiple rays from the source and, as they reflected and refracted within the scene, the energy carried by these rays were accumulated at the locations where they hit diffuse surfaces. The subsequent *eye pass* or *rendering phase* then dealt with the tracing of rays from the camera until the same surfaces were hit [167].

The so-called illumination map linked the two passes as the numerical structure to handle the energy information. This information was stored in a rectangular array of RGB data connected to the illuminated surface via a parameterisation function, which distributed light hit points and respective energies across points defining a cell through weighted bilinear partitioning. A particular portion of this energy was then retrieved for a point being shaded via bilinear interpolation and transformed into light intensity by considering the area that the hit point covered. Being based on point sampling, this ray tracing extension required sample sizes considerably denser than the illumination map grid in order to render accurately the regions where global illumination was required [167].

This simple classic ray tracing technique is capable of rendering shadows and specular reflections originated from direct illumination. However, similarly to what was done by Arvo [167], it has to be extended in order to form the basis of a more complete global illumination algorithms [102]. Two main extensions are worth mentioning. Path Tracing [164] is an algorithm devised from the idea of distributed ray tracing [168] and Monte Carlo integration theory that additionally computes indirect illumination in diffuse surfaces via stochastic sampling of all the possible light paths [102]. Similarly, Bidirectional Path Tracing [169, 170] advances ray tracing further and performs both backward and forward path tracing, given that sampling straight from light sources has its own advantages in terms of capturing specific effects such as caustics [102]. Although unbiased, all of these algorithms invariably suffer from random errors or variance, visible as noise on the final rendered images. The relationship between noise and sample size then directly translates to the high computational cost involved in noise reduction, both in terms of rendering time and memory consumption [102].

3.4.3 Photon Mapping

The reduction of high-frequency noise is exactly the motivation behind the creation of the Photon Mapping technique. While still grounded on bidirectional point-sampling ray tracing this approach now decouples the computed light distribution from the geometry and stores this data in a separate data structure, called the Photon Map. Containing not only information of points of intersection of traced photons with the geometry, but also the incoming direction of these photons and the fraction of radiant power carried by each one of them. The photon map is a compact representation of cached light paths that is able to handle global illumination and different types of reflective surfaces. The two photon tracing passes are then connected during the rendering pass through a density estimation method that describes the illumination statistics everywhere in the scene, hence the description of Photon Mapping as a two-pass algorithm. Although bias is introduced by the statistical evaluation and the results may still suffer from low-frequency noise, the technique provides a consistent solution that will converge if the number of photons being traced is increased [102].

The photon map data represents samples of the energy distribution at any surface location within the virtual environment, e.g. over the aircraft wing. Therefore, the construction of an estimate of the photon density from the sampled data points can be used as an element of the Rendering Equation to subsequently indicate how much radiance is reflected back in a certain direction [102, 171]. The photon density is exactly the radiant flux arriving at a unit area of the surface, or the irradiance (E_i), calculated as [104]

$$E_i(p) = \int_{2\pi^+} L_i(p, \vec{\omega_i}) cos\theta_i d\omega_i \tag{3.51}$$

Instead of using the simpler histogram density estimator, photon mapping utilises the generalised nearest neighbours estimator. The histogram approach is characterised by inevitable discontinuities, depending on the choices of position and size of the bins, which makes it difficult to graphically present multivariate data [171]. The generalised nearest neighbours estimator works by locating in the photon map the k photons located within a sphere of radius h centred around the query point x. It then considers the flux, $\Delta\Phi_p$, contribution of each of those photons on the reflected radiance, given the BRDF f_r of the surface's material. The Rendering Equation rewritten in these terms is given as [102, 172]

$$L_r(x, \vec{\omega_o}) \approx \frac{1}{h(x)^2} \sum_{p=1}^{k} K(y) f_r(x, \vec{\omega_i}, \vec{\omega_o}) \Delta\Phi_p(x, \vec{\omega_i}) \tag{3.52a}$$

$$y = \frac{\left(x - x_p\right)^T \left(x - x_p\right)}{h(x)^2} = \frac{r^2}{h(x)^2} \tag{3.52b}$$

where y is the distance between each photon and the query point normalised by the radius of search. This formulation is normally used to define the filters $K(y)$.

This formulation assumes that the photons are located on a surface that is locally flat around the query points. Therefore, the filters are radially symmetric two-dimensional normalised functions that integrate to unity within the searching radius [102, 171, 172]. If the kernel has exactly the same value for every photon found inside the search radius, i.e. the cube filter [173], the energy of each of these photons contributes equally to the reflected radiance computation. This corresponds to the usual nearest neighbour estimator, which is prone to noise if the global photon map sample size or local photon density are too low. This noise invariably hinders the capture of sharp illumination features, such as caustics.

3.4.3.1 Image Filtering

Much like image filters, it is preferable to choose a kernel that weights the photon contribution in the density estimation proportionally to its distance to the query point. This represents a generalisation of the nearest neighbours method that establishes a connection with the kernel density estimation method, an approach commonly employed for pattern recognition [171]. In its generalised form, the *local* degree of smoothing of the estimator is controlled not only by the radius of the sphere and the respective number of near photons contained inside it, but it is also dependent on the choice of kernel function. The radius is commonly referred to as window width or bandwidth in statistics [171]. The kernel then determines the shape of the photon density function placed at the samples and thus it governs the overall characteristics of the radiance distribution computed at the query points [171].

Jensen [102] suggested the cone filter, Pavicic [173] presented the Gaussian kernel and Silverman [171] also introduced the Epanechnikov and another useful kernel simply called "auxiliar" in this research [102, 171, 173]. These kernel functions are respectively presented as

$$K(y)_{cone} = \begin{cases} \frac{1-\frac{\sqrt{y}}{\kappa}}{(1-\frac{2}{3\kappa})\pi}, & \text{if } \sqrt{y} < 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.53a}$$

$$K(y)_{Gaussian} = \alpha \left[1 - \left(\frac{1-\exp\left(-\beta y\right)}{1-\exp\left(-\beta\right)}\right)\right] \tag{3.53b}$$

$$K(y)_{Epanechnikov} = \begin{cases} \frac{2}{\pi}(1-y), & \text{if } y < 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.53c}$$

$$K(y)_{auxiliar} = \begin{cases} \frac{3}{\pi}(1-y)^2, & \text{if } y < 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.53d}$$

where, for the Gaussian filter the values of $\alpha = 0.918$ and $\beta = 1.953$ are given as optimal by Pavicic [173], while still satisfying the criteria of unit-volume and $K(1) = 0$ [173]. The former requirement, formally defined in Eq. 3.54a, presents a way to vary the values of α and β if the previously mentioned optimal Gaussian, intended for image anti-aliasing purposes, does not perform as efficiently when filtering photon map data. From Eq. 3.54a, through solution of the definite integral and rearrangement, an expression relating the two parameters that control the kernel shape is established, as presented in Eq. 3.54b. The parameter α is the height of the Gaussian and β determines its sharpness.

$$\int_0^1 \int_0^{2\pi} K(r) r \, d\theta \, dr = 1 \tag{3.54a}$$

$$\alpha = \frac{1}{\pi \left[\left(\frac{1}{1-e^{\beta}} \right) + \left(\frac{1}{\beta} \right) \right]} \tag{3.54b}$$

As Jensen [102] highlights, the accuracy of the rendering pass will depend on the global number of light rays traced from the light source, the resulting photon map data structure size and the local number of photons used in the radiance estimation [102]. In statics, Silverman [171] explained that the resulting bias in density estimation depends on the choice of kernel and it is directly proportional to the bandwidth. The bandwidth in turn is usually chosen as a function of the sample size, and variance is inversely proportional to it. This results in the fundamental trade-off between bias and variance, or systematic and random errors committed when estimating density from sampled data [171]. This effect is easily seem as a fitting problem, where as the window width reduces the samples become overfitted and the density function tends to a sum of sharp kernels at the observations. However, with larger windows, the samples are underfitted by flat kernels and the resulting distribution is not general enough [171].

Similarly, Silverman [171] points out another issue and trade-off in density estimation. In a distribution comprising of samples located across the whole surface, if the bandwidth is kept fixed for all the query points distributed, two scenarios will alternatively take place. Either a considerable amount of artificial noise will appear in some regions, which will be undersmoothed due to a choice of small window to appropriately resolve specific features present in other regions, or, in order to reduce this noise, the main details of the distribution will become oversmoothed by a larger window size [171].

Density estimation, as in the photon mapping approach, aims at discerning features and/or drawing conclusions from the available data set [171]. However, both the nearest neighbours and the kernel methods are very susceptible to noise, which can mask features or hinder the judgement of the overall distribution. Therefore, a common approach is to employ an adaptive kernel method, usually multi-stage as presented in [171], which adapts the bandwidth and related degree of smoothing to local characteristics of the data, solving the previously described trade-off concerns.

3.4.3.2 Adaptive Kernels

Schjøth et al. [174] effectively proposed a variable-bandwidth kernel estimator, but based on the concept of photon differentials in connection with the photon map. The differentials exploit the coherence of neighbouring photons following very similar if not the same path and consider clusters of particles forming a beam of light. These will have their shape altered during propagation, due to reflections and refractions. These geometric effects are then directly linked to the ray footprints, areas used in the density estimation that, in contrast to classical photon mapping, adapt to the illumination structure in the scene. The variation of the footprint is tracked through a parametrisation of the light beam and the respective derivatives, simple operations related to the light scattering. The footprint indicates the spread of the rays when intersecting a geometry. Two images containing caustics of different complexity were rendered using photon differentials and compared against solely photon mapping. The proposed method resulted in images that revealed the structure of the caustics with far better quality and, although the photon tracing takes longer to run when using photon differentials, it provided visually acceptable results even with a lower photon sample size and thus the overall rendering time was reduced.

Herzog et al. [175] suggested another viable alternative to the classical photon mapping technique, similar to the concept of photon differentials. Still based on bidirectional path tracing and density estimation, their method utilised an alternative photon splatting pass. This replaced the traditional photon gathering employed in the radiance estimation stage. This pass computed the contribution of each light ray to various pixels at the same time. The "splatting" considered that the energy that each photon carries is distributed across query points located within a conical frustum centred around the respective light ray and shaped by a two-dimensional kernel perpendicular to the direction of this ray. By using splat footprints, the method actually estimated photon density in the light ray space, performed nearest neighbours search in the opposite sense by looking for query points in the zone of influence of photons, and finally projected the irradiance back onto the surface of geometries present in the scene. Radiance was then directly estimated from these neighbouring splats, considering the reflective properties of the surface. Compared against photon mapping, the photon splatting method provided images in which boundary bias was eliminated and topological bias was considerably reduced. The former is an underestimation error of illumination at boundaries of the scene and the latter an overestimation error in complex geometries, partly associated to density estimation at surfaces. The resulting images were rendered comparatively fast and also exhibited better visual quality.

Spencer and Jones [176], instead of focusing on kernel shape and bandwidth optimisation to reduce the noise of synthetic images, proposed a reconfiguration of the photon map through a diffusion operator as an intermediate stage. Photon relaxation was therefore achieved by a point repulsion method, which similarly to the classical photon mapping density estimation is also based on a nearest neighbours search. Due to the migration of photons in the direction of the photon density gradient, diffusion bias was invariably introduced, consequently blurring out high frequency details. These features were then preserved by employing constraints on the photon repulsive force and respective migration based on irradiance differentials detected in the previously mentioned search, a very costly operation. This redistribution of the photon samples located over the surface of an intersecting geometry enabled the use of very compact kernel sizes, perhaps the main advantage of the photon relaxation method. Images rendered using photon relaxation were compared against the fixed bandwidth photon mapping and the method had proven to be effective in detecting and preserving discontinuities and details, simultaneously removing noise in areas of uniform illumination. Moreover, the method operated considerably faster when rendering caustics. The relaxed photon maps were more equidistributed and exhibited a blue noise spectral signature, which was linked to the resulting lower noise in the radiance estimates.

Hachisuka et al. [177] also reorganised the classical photon mapping technique and exploited its consistency to devise a method to progressively improve the accuracy of the estimated radiance, without the memory overhead associated to the storage of full photon maps. By using multiple photon tracing passes, the illumination of scenes dominated by complex caustics lighting effects could be refined up to a desirable extent. Each sequential pass was composed of a small number of photons and it was cumulative, meaning that all of the traced photons would contribute to the energy in the reflected radiance, which could be rendered after any of the passes. Once the contribution of a photon pass was taken into account, the current photon map could be discarded, reducing considerably the computational memory requirement of such an operation. Radiance was reflected from a visibility point towards the viewing system, direction that was traced a priori, and it was computed using the same nearest neighbours density estimation technique. The assumption that the photon density is constant within a certain radius then allowed the reduction of this search radius in subsequent passes, considering that the number of photons inside this region increased as the contributions of photons were consecutively added. The progressive increase in the number of photons and decrease in search radius conditioned the behaviour of the photon density to grow to infinity as more photon passes were traced. The convergence of the radiance solution to an arbitrary accuracy then depended on the number of selected passes. With lower memory requirements, progressive photon mapping rendered complicated scenes using greater number of photons, which invariably reduced the noise of the images whilst capturing detailed illumination features otherwise missed by photon mapping, all at comparative rendering times.

Jensen [102] also mentioned an extension to the classical photon mapping technique called differential checking. This concerns an approach to determine the smallest bandwidth and respective number of photons inside the searching sphere necessary to reduce bias without introducing too much noise to the radiance estimate. Differential checking worked by progressively increasing the radius of the sphere until a drastic increase or decrease in radiance was observed, which indicated that the bandwidth had crossed a boundary between two distinct lighting features or an inhomogeneity, for example the edge of a caustic. Upon this observation the radius variation would be stopped and its previous value would be used for rendering.

In this research, a simpler adaptive approach to the radiant energy density estimation in Eq. 3.52 is constructed. It is based on the information already available from the illumination stage. This avoids major modifications of the classical photon mapping algorithm or the introduction of new numerical parameters. The kernel bandwidth is automatically adapted based on a measure of the scale of refraction that occurred in the compressible flow medium. This measure is the amount of light deflection undergone after propagation from the light source, a quantity embedded in the photon map itself, given the direction that light rays intersect the solid surface compared to the direction of emission.

3.4.4 Viewing System

As described in Section 3.4.2, when using nonlinear ray tracing to solve the rendering equation, the process of connecting the illumination and the visibility stages of light propagation to perform reflection and shading calculations is complex. Gröller [166] therefore suggested two strategies to overcome the complexities. The first strategy was the simple omission of shading and shadowing calculations [166], thus skipping the visibility or rendering stage. Indeed, the photon map constructed after a (nonlinear) illumination or photon tracing stage and its radiant energy distribution data already present relevant information regarding the global illumination of the scene: an indication of the existence of a shadow pattern on the wing surface given by the shift in concentration of particles in certain adjacent regions. However, the map lacks the phenomenological and perceptual aspects of radiometry and photometry. It is neither the real, physical quantity that contains the visual information measured by recording devices such as cameras, nor the psycho-physical signal transmitted to the human visual system and thereby interpreted to construct a response suitable to the respective optical information [100]. Therefore, it does not represent the shockshadow as an observable feature.

Similarly, the much easier estimation of irradiance compared to radiance, due to the absence of the light direction dependence, still would only be an indication of the caustic formation but not a true visual pattern. By regarding the wing surface as a simple light sensor, irradiance could then be computed, but it would not be possible to reproduce this image in reality and, consequently, the virtual experiment would not be validated. Employing only forward ray tracing from the sun towards the wing, the radiance could also be estimated by considering a recording film shaped exactly as the wing and positioned right outside of its boundary. This way, all the reflection would be assumed to occur in the directions normal to the surface and the reflected light rays would only traverse an infinitesimal distance until they reached the film. This case is analogous to the wing regarded as a light sensor previously mentioned, although now it is considered to be sensitive to reflected radiance. However, the degree of light deflection after undergoing refraction through the compressible flow and the resulting photon concentration shift are very small and insufficient to produce a synthetic shockshadow without considering real reflections and further light propagation. Moreover, the noise in the resulting images, inherent to the classic ray tracing and photon mapping techniques, is usually of the same order of magnitude as the expected disturbances in the radiance distribution when viewed from this perspective.

The second strategy proposed by Gröller [166] was the assumption that light propagates in linear paths from the point of intersection at a surface present in the scene to the viewing system model [166]. The assumption therefore ignores the inhomogenous optical medium (compressible flow) during the visibility or rendering stage, and thus it greatly simplifies the ray tracing process, as the standard visibility test to perform shading and shadowing computation can again be employed. This is the same as the direct visualisation of the photon map proposed by Jensen [102], through which the classic ray tracing technique is applied to compute reflected radiance from diffuse materials using the density estimation explained in Section 3.4.3. According to Jensen [102], 'this approach can be used to render caustic efficiently' [102] and this simplified visualisation is still a complete solution to the approximate rendering equation (remembering that the equation assumes image synthesis in vacuum - see Section 3.4.1). Given the linearity properties of Eq. 3.50, by splitting the incoming radiance ($L_i(p,\overrightarrow{\omega_i})$) as the sum of a component related to direct illumination from the light source and a component associated to indirect illumination due to single or multiple refractions, the latter is considered to be responsible for the caustic formation and can therefore be rendered independently [102]. A *caustic photon map* can then be constructed, stored and its rendered radiance convergence analysed separately from the remaining illumination, for efficiency purposes [102].

A simplification associated with this strategy is the positioning of a recording film, now appropriately defined as an image plane. This is placed a certain distance away from the boundaries of the domain, either the object's solid surface or the inhomogeneous optical medium itself. Inspired by the way sensitivity is controlled via adjustments of the screen position in the shadowgraphy experimental technique, by allowing the reflected light rays to travel linearly further from the intersection locations the geometrical displacement of photons due to refraction and light deviation becomes much more pronounced. Then, measuring the contrast in the pixels, i.e. the difference between radiance values computed considering light propagation in air free of discontinuities and in an inhomogeneous refractive-index field, a characteristic shadowgram pattern is then observable in the computational image. This linear light traversal after crossing the compressible flow is true in the case of reproductions of shadowgraphs [22], as with the supersonic axis-aligned cylinder simulated during verification of the initial implementation of the light simulation framework (see Section 4.2). In shadowgraphy, light indeed tends to travel outside the wind tunnel in homogeneous air. However, as mentioned, this is only an approximation in terms of the natural shadowgraph formed in flight, where secondary refractions would also occur. Nevertheless, if the recording film is oriented in the perfect specular or mirror-like reflection, as in the transonic NACA 0012 aerofoil simulation (see Section 5.1), it senses any slight variations in the photon distribution and the sensitivity of the final shadowgraph analogous image can easily be adjusted.

3.4.4.1 Virtual Pinhole Camera

In the computational simulation of a camera, a convenient simplification involves the emulation of the principles according to which pinhole cameras operate. Being possibly the most basic existing photography device, pinhole cameras take snapshots of a scene by allowing light to enter through a small aperture contained in a sealed box, over long exposure times and without the influence of optical systems, and then recording this light in a film located inside the box but on the opposite side to the hole [103]. The resulting picture is thus inverted in both directions that define the plane of the recording film [104]. Although not realistic, the abstract configuration in which the recording film is placed in front of the pinhole (see Figure 3.13) avoids the inversion of the final image [103, 104]. Even with the inversion of positions between the *view plane* and the *eye*, names used to refer to the film and the pinhole in the virtual camera, respectively, the *view frustum* or pyramid-shaped volume of the scene that will be imaged on the film is nevertheless maintained [103].

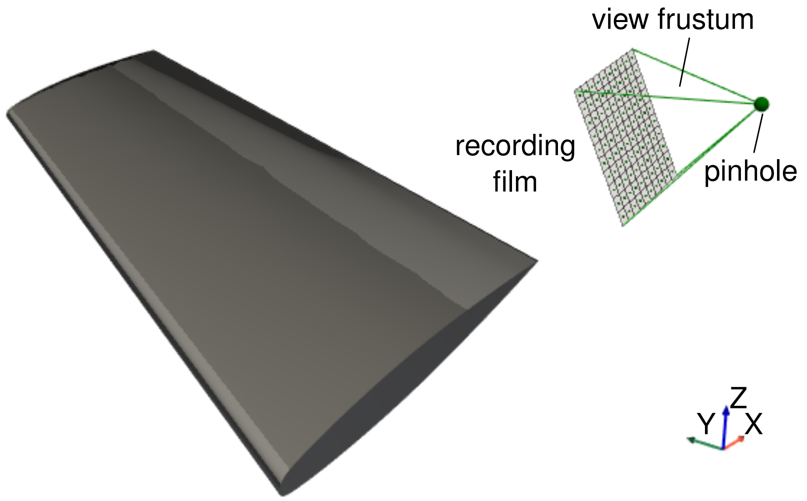


Figure 3.13: Viewing system model: a virtual pinhole camera.

The main task of the rendering engine (implementation) and of the camera model itself is then the determination of the light rays that contribute to the image formation, and from the illumination information gathered from these light rays the selection of the colour to display in the image pixels [103]. For the virtual pinhole camera, these light rays of importance are easily defined as those that pass through the picture elements and reach the eye, analogous to a real pinhole camera [103]. This solves the 3D viewing problem of projecting the three-dimensional geometrical objects that are present in the scene, and visible inside the view frustum, into the two-dimensional view plane [103, 104]. The solution is then a perspective type of projection, where all the light rays are traced from the same location and their directions are determined based on the pixel that they will pass through when cast during the backward ray tracing process [104], see Figure 3.14. Perspective projection is closely related to how the human visual system or camera systems generate images [103, 104].

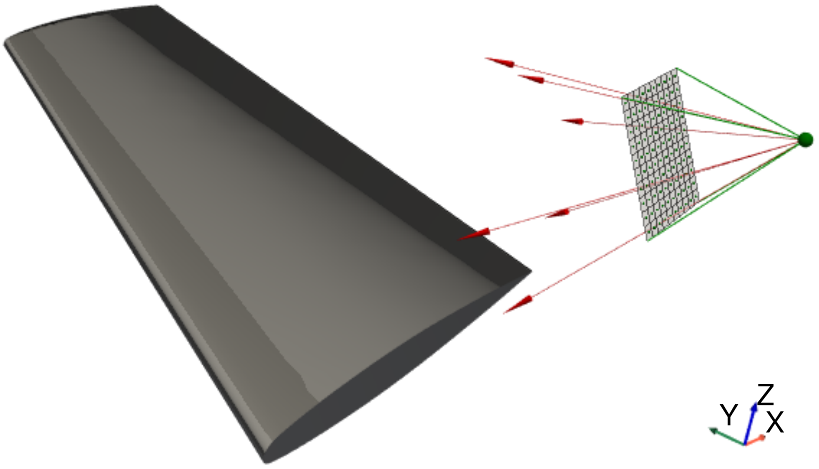


Figure 3.14: Primary light rays origin and directions in the viewing system model.

This camera model is completely defined by four parameters: the eye point ($\mathbf{e}$), the look-at point ($\mathbf{la}$) towards which the camera is directed, the $\overrightarrow{up}$ vector and the distance d between the eye point and the view plane [104], all of which are represented in Figure 3.15. These parameters are directly related to the transformations of coordinate spaces that take place within the camera model. The geometric object (wing) and inhomogeneous optical medium (compressible flow) are assumed to be provided in the *world space*, the main reference for other local systems of coordinates. This global system coincides with the geometric definitions of the CFD grid and solution. The camera model is set up in the scene based on the location of its centre of projection in world coordinates (the eye point), its particular viewing direction determined from the eye to the look-at point, also defined in world coordinates, and its orientation with respect to this direction determined by the up vector. The *camera space* or viewing coordinates are then created according to an orthonormal basis (u,v,w) with origin at the eye. The negative w-coordinate is mapped to the viewing direction and the positive v-coordinate is mapped to the up vector direction [103, 104], as depicted in Figure 3.16. Another useful local system is the two-dimensional *screen space* or image coordinates, with origin at the centre of the view plane and x_v and y_v directions defining the plane [103, 104], as also shown in Figure 3.16.

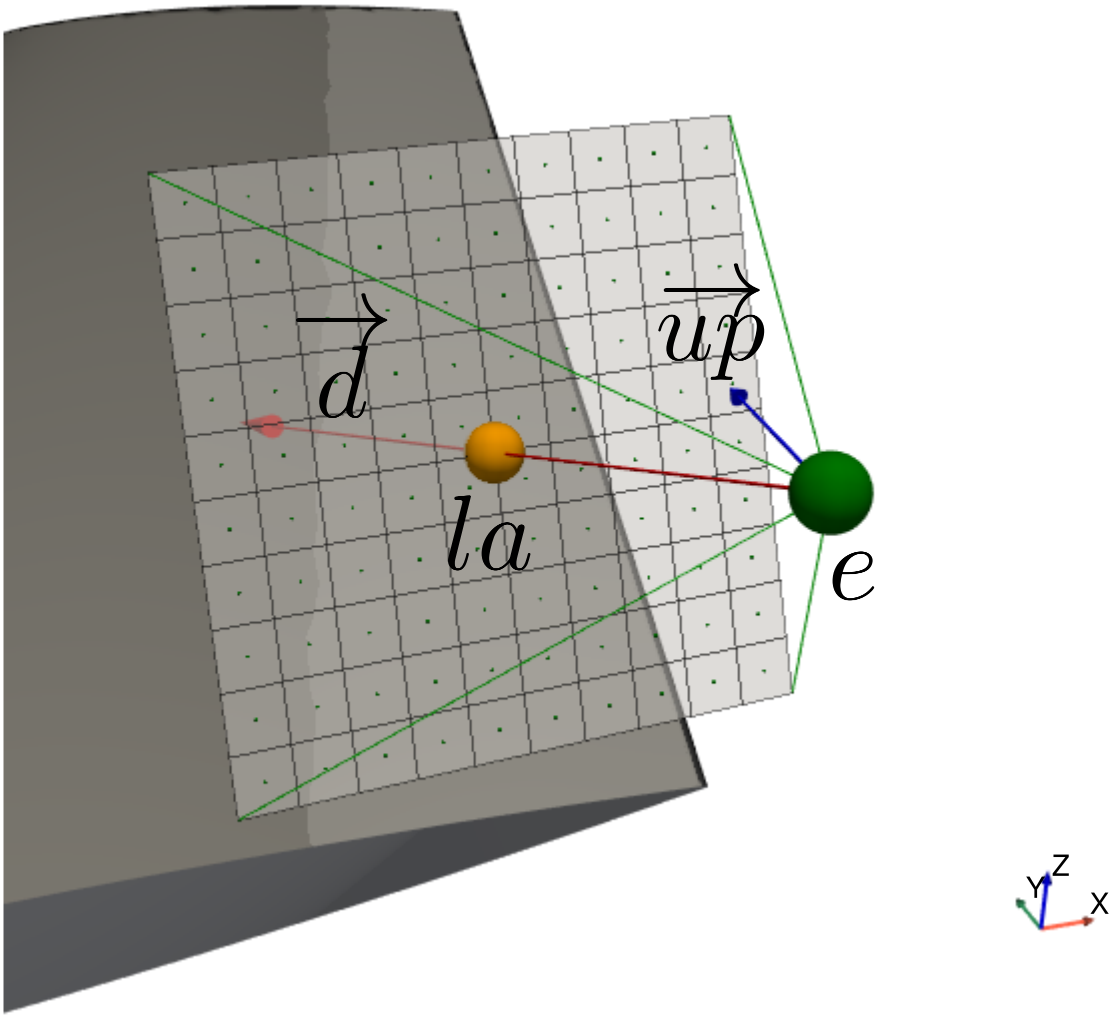


Figure 3.15: Virtual camera model definition parameters.

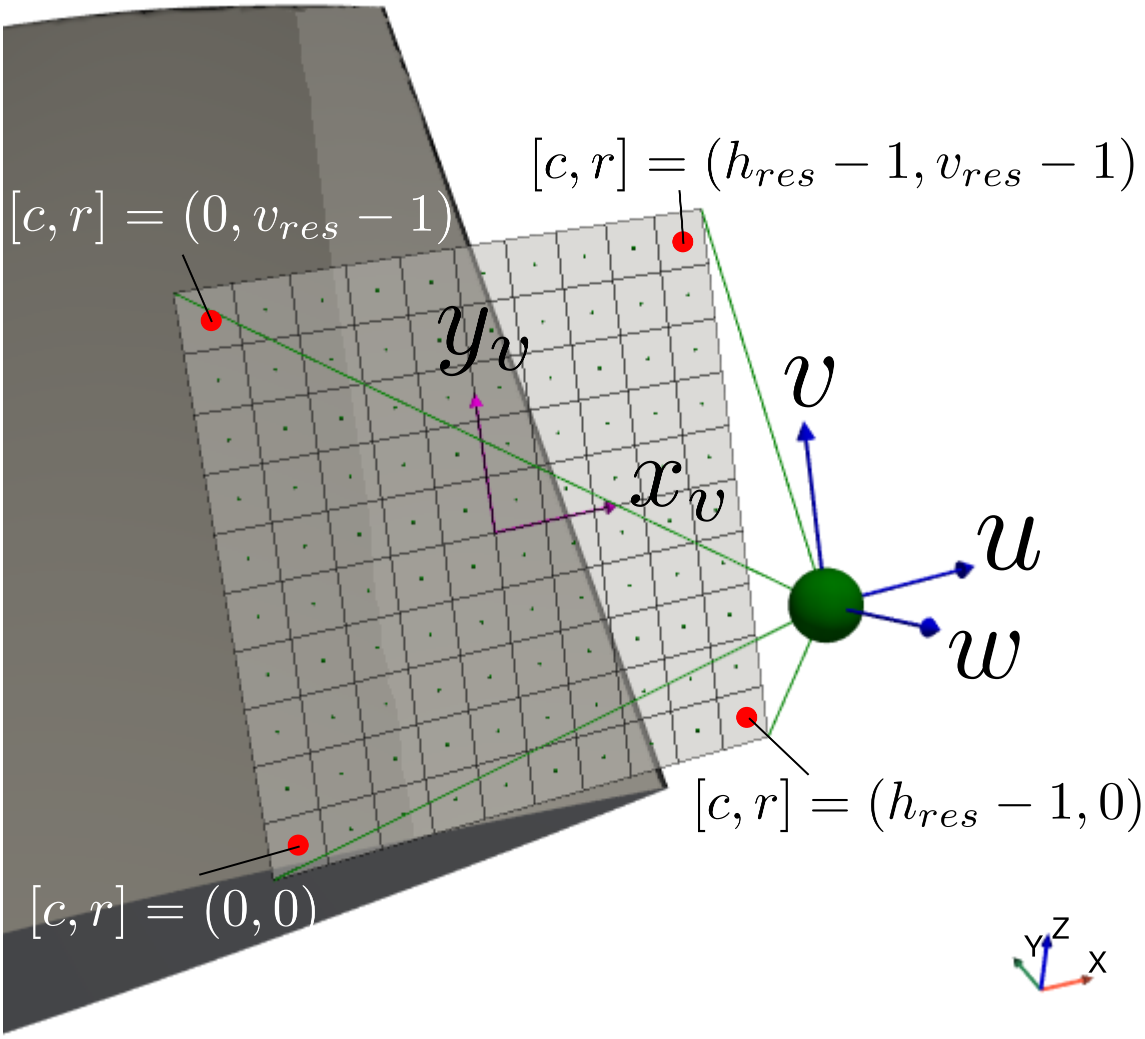


Figure 3.16: Camera and screen space, or viewing and image local coordinates.

The camera and screen spaces can then be used together to calculate the direction of light rays to be cast from the virtual camera as [104]

$$\vec{d} = x_v\vec{u} + y_v\vec{v} - d\vec{w} \tag{3.55}$$

where

$$\vec{w} = \frac{(\mathbf{e} - \mathbf{la})}{\|\mathbf{e} - \mathbf{la}\|} \tag{3.56a}$$

$$\vec{u} = \frac{\vec{up} \times \vec{w}}{\|\vec{up} \times \vec{w}\|} \tag{3.56b}$$

$$\vec{v} = \vec{w} \times \vec{u} \tag{3.56c}$$

The pixel coordinates (x_v, y_v), in turn, depend on the horizontal and vertical resolutions (or number of pixels in each direction - hor_{res} and ver_{res}) and the pixel size (s) used in the image, as given by [104]

$$x_v = s\left(c - \frac{h_{res}}{2.0} + 0.5\right), \quad c = 0, 1, ..., h_{res} - 1 \tag{3.57a}$$

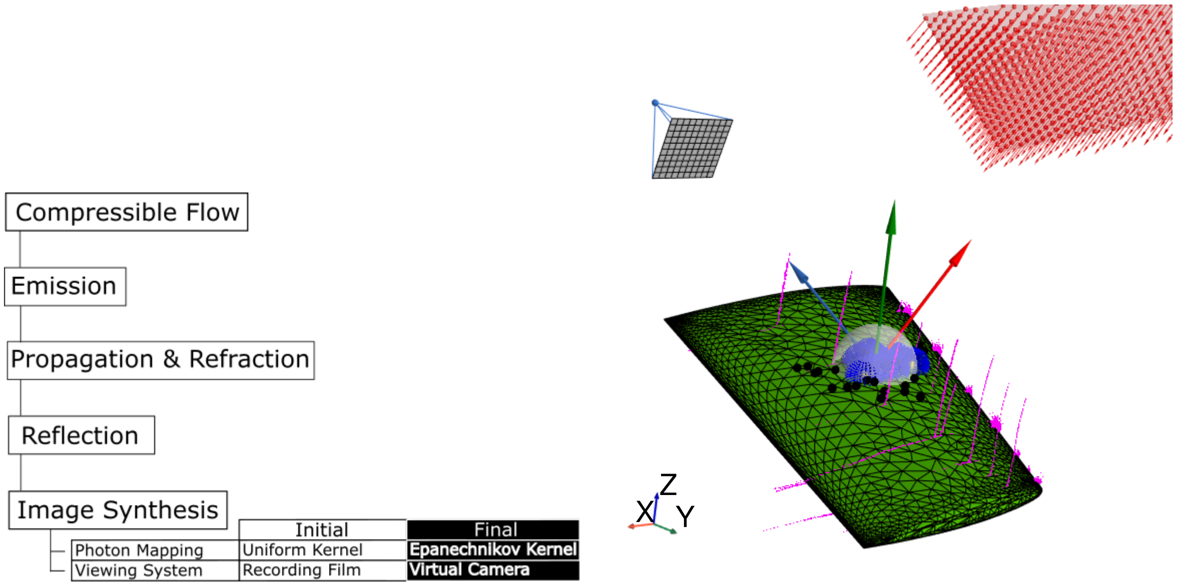
$$y_v = s\left(r - \frac{v_{res}}{2.0} + 0.5\right), \quad r = 0, 1, ..., v_{res} - 1 \tag{3.57b}$$

3.4.5 Image Synthesis Model

In the initial implementation of the light simulation framework a simpler plane-shaped recording film is positioned in strategic locations within the scene being imaged, similar to a light sensor. It is also specifically oriented in order to appropriately capture the photon displacements achieved during the illumination stage. The Rendering Equation is then solved at the recording film itself using the photon mapping technique together with a uniform kernel, i.e. an unfiltered nearest neighbour density estimation approach.

In the final implementation a virtual camera model (pinhole) is constructed and, during the additional visibility stage, light rays are also traced from the eye point towards the scene. Both stages of light propagation are then connected in the Rendering Equation. This is solved at the points where these camera rays intersect solid surfaces, employing the Epanechnikov kernel and an adaptive bandwidth approach based on the light deviation from its original direction.

Figure 3.17 summarises these decisions related to the modelling of image synthesis in the initial and final implementations. Figure 3.17(a) shows the development of the light simulation framework along the roadmap of this Chapter and Figure 3.17(b) illustrates the final process of image synthesis employed.



(a)

(b)

Figure 3.17: (a) Roadmap of the current progress in terms of light modelling, and (b) graphic representation of the image synthesis model used in the final implementation.

CHAPTER 4

Verification and Validation

A prerequisite for the implementation of a reliable shockshadow simulation framework for practical analysis of aerospace applications, based on the computational models described in Chapter 3, is an assessment of the approach on simple test cases. First a direct comparison is made between simulation results and exact analytical solutions [178] for light refraction. This ensures that the implemented method is consistent with expected results and that the mathematical equations are correctly written in computational code and solved [179]. A further validation assessment is subsequently made to check that the modelling appropriately represents the physical reality of the direct natural shadowgraph formation [178]. It also evaluates the imagery from the perspective of the intended application, i.e. shock wave visualisation and possible aerodynamic data extraction [180]. A direct comparison between the simulation results and experimental data for problems of the same nature (i.e. shadowgraphy) enables an assessment of whether the chosen equations are reasonable [180], ultimately demonstrating the overall accuracy and credibility of the simulation. Chapter 4, therefore, details the verification and validation assessments performed in this research.

4.1 Verification of Light Propagation and Refraction

Discontinuities in the propagation of light, whether at the interface between different homogeneous media or corresponding to inhomogeneities in density present in a medium, will cause the scattering of the electromagnetic radiant energy. From a submicroscopic perspective, this will involve the interaction between photons and the atoms that form molecules in substances such as air. Macroscopically, there will be a change in the direction of light rays, associated with the phase shift in the light field and respective variation in speed that a light wave experiences. This is the refraction phenomenon [13]. The index-of-refraction (IOR) of a medium (n), defined in Eq. 4.1 [13], represents this apparent shift when its value is different than one, as the speed of light in the medium (v) will be different compared to the magnitude developed in a vacuum (c).

$$n = \frac{c}{v} \tag{4.1}$$

Since light scattering is the phenomenon being modelled in this research, an initial verification assessment of the Nonlinear Ray Tracing implementation is performed using two simple test cases. The first case is for a single refraction between two media with different refractive indices and the second case involves multiple refractions and transmission through a dispersing prism. These cases are selected as they have exact analytical solutions given by the well-known Snell's Law of Refraction [13]. This Law of Refraction is given in Eq. 4.2 as

$$n_i sin\theta_i = n_t sin\theta_t \tag{4.2}$$

where the incidence angle is θ_i, the transmission angle is θ_t, and n_i & n_t are the refractive indices of the two media involved in the scattering. A representative example of the test case is given in Figure 4.1.

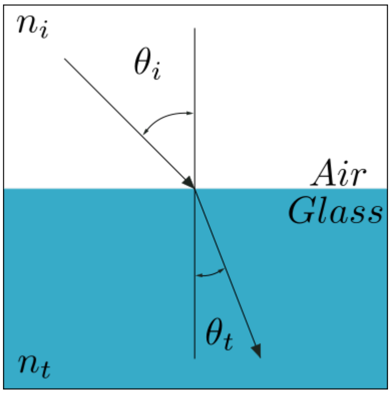


Figure 4.1: Schematic diagram of single refraction through two homogeneous media.

4.1.1 Single Refraction

The first scenario considers light emission from the sun on the yellow part of the visible spectrum, where the wavelength ranges from approximately $570\ nm$ to $590\ nm$. The decision to use only a smaller range of the visible spectrum was made to check the consistency of the methods used to calculate the Gladstone-Dale constant, and consequently the refractive index of the medium. It was also used to confirm if monochromatic solutions could be simulated independently and combined a posteriori. This is the linearity assumption embedded in the approximate Radiance Equation or Rendering Equation (see Section 3.4.1). The Gladstone-Dale constant for air at $288\ K$ with the density of $\rho = 1.225\ kg/m^3$ is then obtained from the values presented in Table 1.1 in Section 1.3. At wavelength bins of $1\ nm$, the respective constant is linearly interpolated from the reference tabular values and, to obtain a single constant for the entire medium, these interpolated values are averaged over the considered wavelength spectrum. The Gladstone-Dale relation (Eq. 1.1) subsequently provides the index-of-refraction of air.

From emission, light is transmitted through air to a second medium composed of light flint glass [13] (similar to the situation depicted in Figure 4.1). The dependence of the refractive index of this material on the wavelength of light is described in [13]. This optical property of the glass is then estimated from the spectral curve shown on Figure 4.2 [13], through cubic spline interpolation from the curve in the assumed wavelength bins and again final averaging in the emission wavelength spectrum. The averaging operations performed when estimating both refractive indexes are assumed to represent the overall spectral behaviour of light in the media within the range of wavelengths in the visible spectrum characterised by a yellow hue, i.e. a yellow monochromatic light beam.

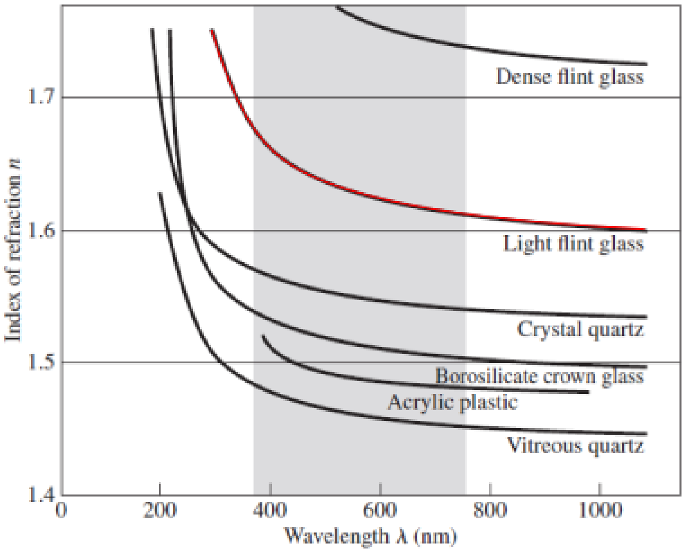


Figure 4.2: Spectral refractive index of different materials, including light flint glass (highlighted in red). Figure reproduced from [13].

Originating from air where $n \approx 1.0003$ and directed towards the glass where $n \approx 1.6244$, several light rays are cast at different incidence angles, ranging from $0°$ to $45°$ of orientation with respect to the y-coordinate direction. These rays are then traced through both media, as illustrated in Figure 4.3, by solving the Ray Equation from Geometrical Optics (see Section

3.2.3). When crossing the interface between media, the numerical approach is responsible for sensing the inhomogeneity, i.e. the gradient in refractive index (shown in the clipped domain of Figure 4.4), and thus it captures the corresponding deflection of the rays. To obtain an accurate representation of the discontinuity, the optical domain mesh has to be appropriately refined in that region, which is also shown in Figure 4.4.

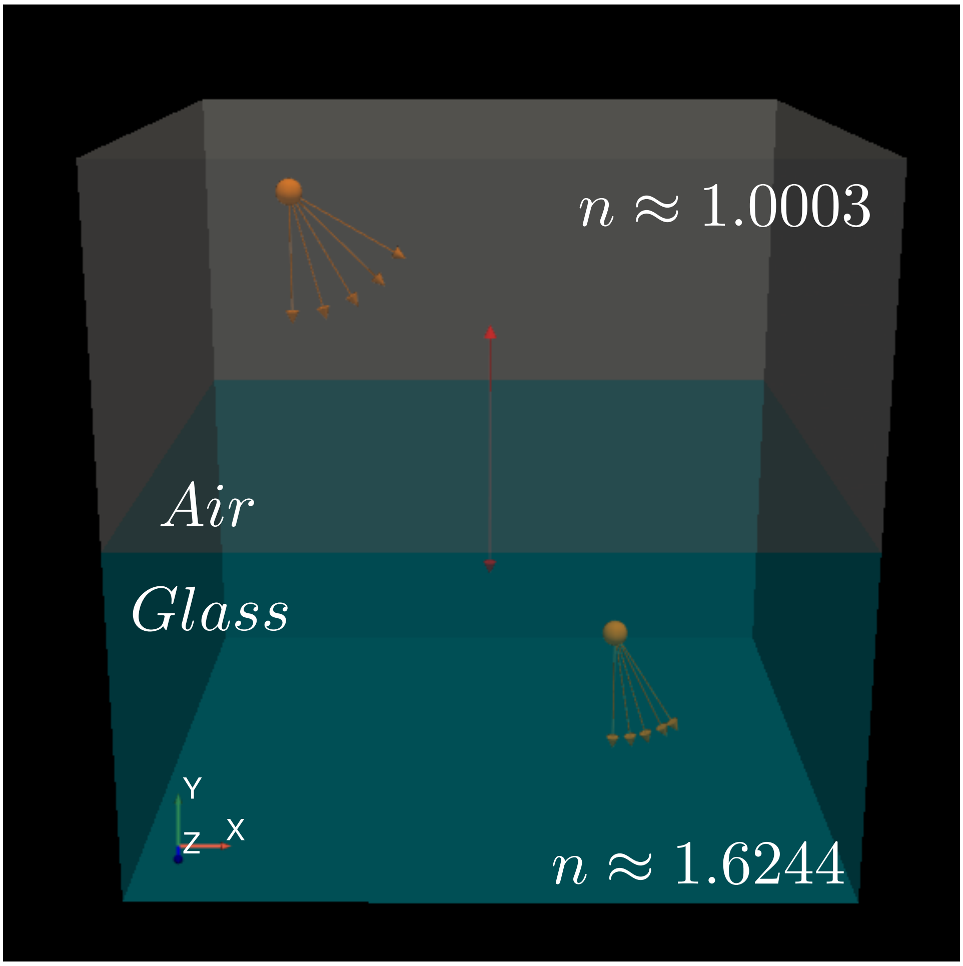


Figure 4.3: Index-of-refraction distribution of the two homogeneous media involved in the single refraction verification case.

The behaviour of the light rays in the described simulation is depicted in Figure 4.5. It can be seen in the picture that the higher the angle of incidence θ_i that the light rays reach the interface between air and glass the more they deflect from their original direction, or the stronger is the refraction effect. This is the expected result based on the application of the refractive index values and selected incident angles to the mathematical Law of Refraction. The percentage difference between the transmission angle values (θ_t) calculated analytically using Eq. 4.2 and predicted computationally within the Ray Equation solution process are then presented in Table 4.1.

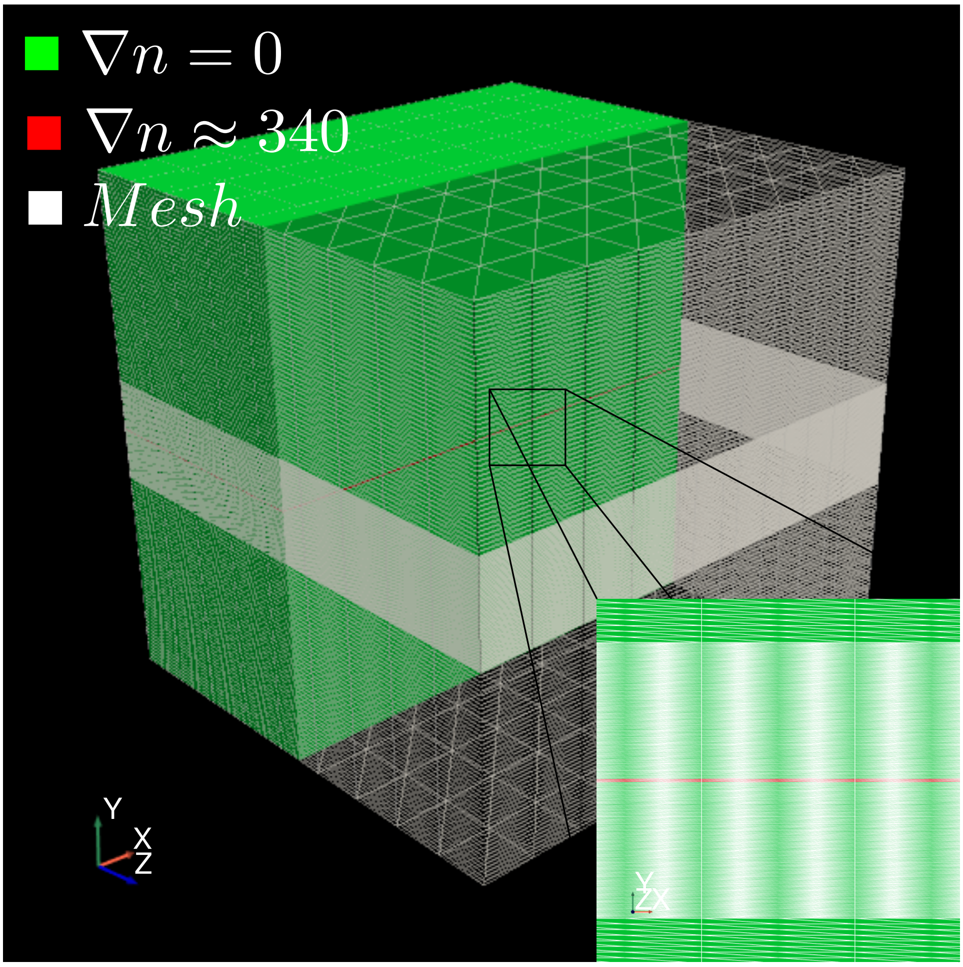


Figure 4.4: Details of the gradient of refractive index and respective domain discretisation employed in the single refraction verification case.

Table 4.1: Numerical results of the single refraction verification case.

θ_i [°]	θ_t [°]		Percentage Difference [%]
	Computational	Analytical	
1	0.6292	0.6298	0.0945
5	3.1473	3.1463	0.0313
10	6.2570	6.2771	0.3199
15	9.3786	9.3762	0.0259
20	12.4407	12.4271	0.1094
25	15.4434	15.4124	0.2004
30	18.3063	18.3138	0.0409
35	21.0857	21.1109	0.1194
40	23.7776	23.7821	0.0188

It can be seen that the maximum error level is 0.32% which arises for an incident angle of 10^o and confirms that the developed methodology can correctly predict the abrupt change in light direction at the interface of two homogeneous, but different, media.

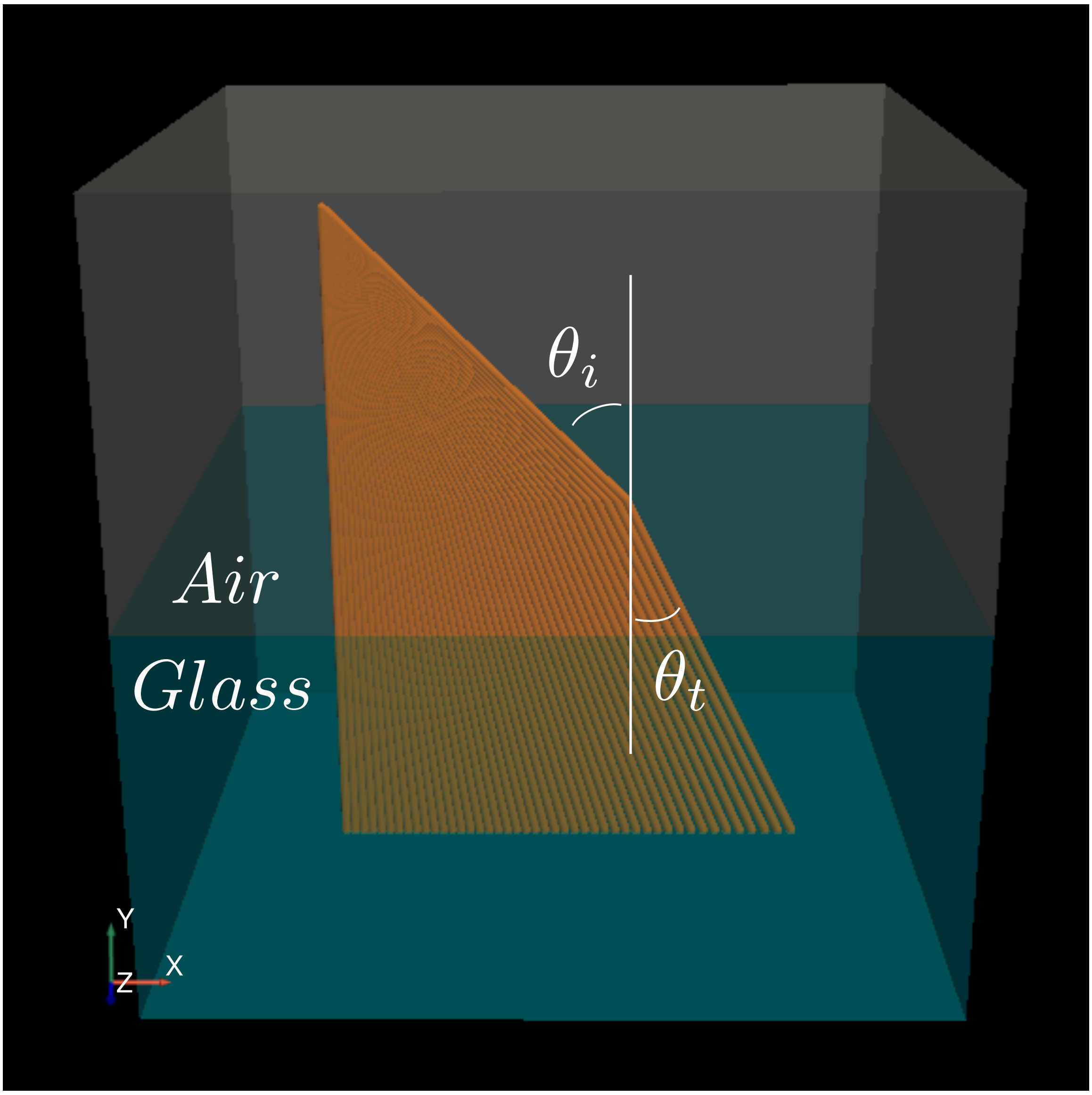


Figure 4.5: Traced light rays emitted from air, refracted and transmitted through glass.

4.1.2 Multiple Refraction Through a Dispersing Prism

The single refraction test case considered in the previous section verified the computational solution obtained by solving the Ray Equation using the selected numerical approach. It also confirmed the possibility of tracing approximate monochromatic light rays, where sunlight is split into a number of ranges within the visible spectrum. Therefore, this wavelength dependency of light propagation and refraction is further verified and checked for consistency by means of a scene containing a dispersing prism, an optical device employed to split the colour (or frequency) components of a polychromatic light beam that passes through it [13].

A polychromatic light ray and respective spectral characteristics are not handled directly by the simulation, but are rather embedded in the index-of-refraction of the medium, and how it varies according to the wavelength of emission. Thus, monochromatic light rays representative of seven (approximate) colour bands in the visible spectrum (shown in Table 4.2 alongside the respective wavelength bin values) are traced. The individual behaviour of each colour is then captured within the dispersing prism. It is assumed that the colours combine in a polychromatic beam before reaching the prism. The prism is again assumed to be made of light flint glass and immersed in the air at $288\ K$. The indexes-of-refraction of the media are again estimated from data gathered from the literature [13, 22] and respective averaging processes (see Table 4.2).

Table 4.2: Wavelength bands and refractive index of the media in the visible spectrum employed in the multiple refraction verification case.

Colour	λ_{min}	λ_{max}	n_{air}	n_{glass}
Violet	380	420	1.000283	1.657438
Indigo	420	450	1.000281	1.647791
Blue	450	500	1.000280	1.638844
Green	500	570	1.000278	1.629896
Yellow	570	590	1.000277	1.624369
Orange	590	630	1.000276	1.621575
Red	630	740	1.000275	1.615639

It can be seen that the refractive index of air is approximately one across the spectrum, whereas there is a significant variation for glass. The overall effect of the prism on the light rays that are transmitted through it can be evaluated through successive applications of Snell's Law of Refraction (Eq. 4.2) on each of its boundaries. Alternatively, a formula for the angular deviation of a light ray (δ) from its original emission direction can be derived as a function of the prism apex angle α, the angle of incidence of the light ray θ_i and the refractive index n_{glass} of the triangular prism (see Figure 4.6) [13]:

$$\delta = \theta_i + sin^{-1}\left[(sin\alpha)\left(n_{glass}^2 - sin^2\theta_i\right)^{1/2} - sin\theta_i cos\alpha\right] - \alpha \tag{4.3}$$

Considering an equilateral triangular prism with apex angle $\alpha = 60°$, and the refractive indexes of light flint glass in the visible spectrum, there are two limiting factors for this verification assessment. The first concerns the conditions where the light rays at either end of the spectrum emerge from the prism parallel to its surface (i.e. $\theta_e = \theta_{t_2} = 90°$ in Figure 4.6). If the angle of incidence is slightly decreased from this value, these monochromatic light rays do not reach the opposite surface of the prism surface at all. This is known as the light grazing emergence condition, and it will occur at different incidence angles for each wavelength band, as shown in Figure 4.7(a). The second is in terms of the difference in angular deviation of light rays

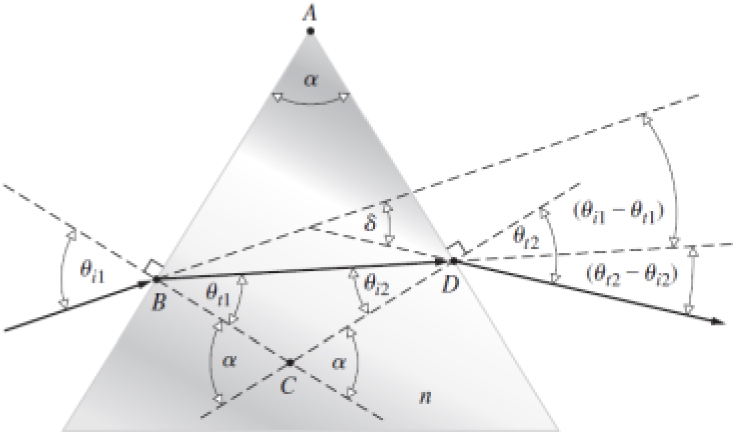
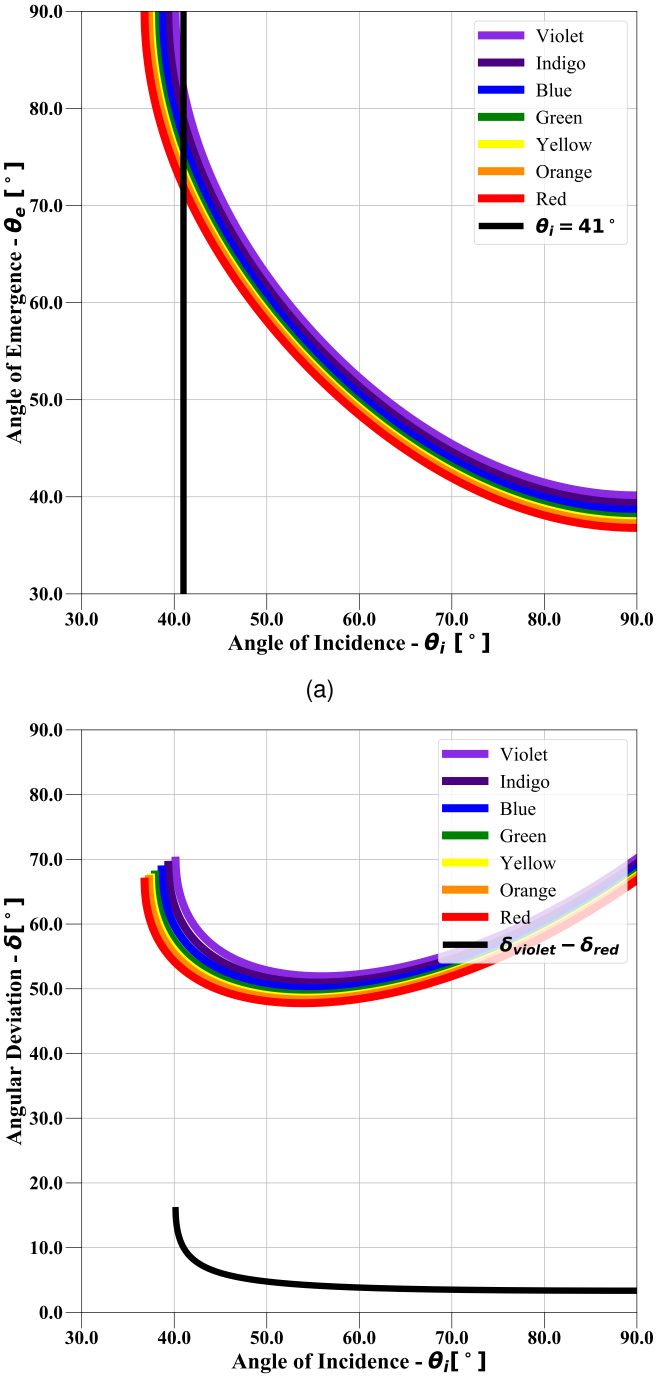


Figure 4.6: Schematic diagram of multiple refractions through a dispersing prism. Figure reproduced from [13].

across the spectrum, which can be minimal for certain values of angle of incidence (higher than $60°$, as seen in Figure 4.7(b)), thus hindering a simple visualisation of the different levels of refraction. Therefore, the angle of incidence for the verification assessment is chosen to be $41°$, corresponding to a near grazing emergence in the shorter wavelengths, where the splitting of colours is at its maximum.

Given these conditions, individual light rays are emitted from one of the sides of the prism, as depicted in Figure 4.8 and the resulting deviation is measured at the opposite side, with respect to the change in the direction of light rays when passing through the prism. The mesh size at the boundaries between air and glass, presented as a coarse example in Figure 4.9, is refined until the gradient of refractive index is appropriately captured and the errors in angular deviation fall within an acceptable level.

The behaviour of the differently coloured light rays in the simulation is depicted in Figure 4.10. The adaptive numerical integration method employed to solve the Ray Equation and trace the nonlinear paths of light rays is characterised as embedded formulae (see Section 3.2.3.1). Because of that, the function evaluations used to approximate the paths are sensitive to the gradients in the refractive index even before the integration points have reached the discontinuity. Therefore, light rays seem to progressively bend before and after crossing the boundaries of the prism. Nevertheless, the cumulative behaviour of light refraction is accurate and the deviation angle is measured based on the transmission angle obtained after the completion of light ray bending on the right-hand side boundary of the prism. The percentage difference between deviation angle values calculated analytically using Eq. 4.3 and predicted computationally within the Ray Equation solution process are then presented in Table 4.3.



(a)

(b)

Figure 4.7: (a) Angle of emergence and (b) angular deviation dependence on the angle of incidence of light rays in the wavelength bands of the visible spectrum.

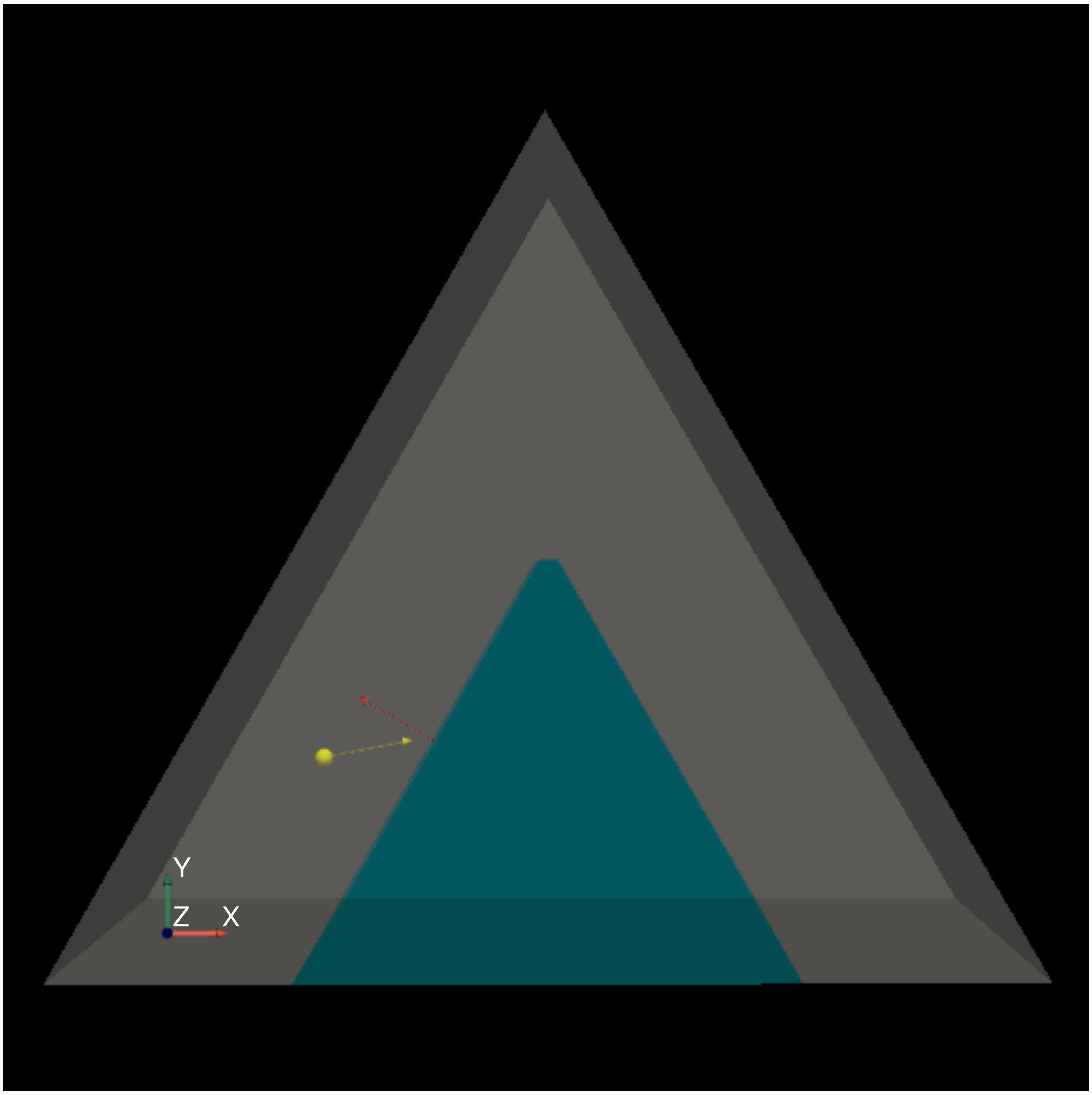


Figure 4.8: Index-of-refraction distribution on the glass dispersing prism immersed in air, the multiple refraction verification case.

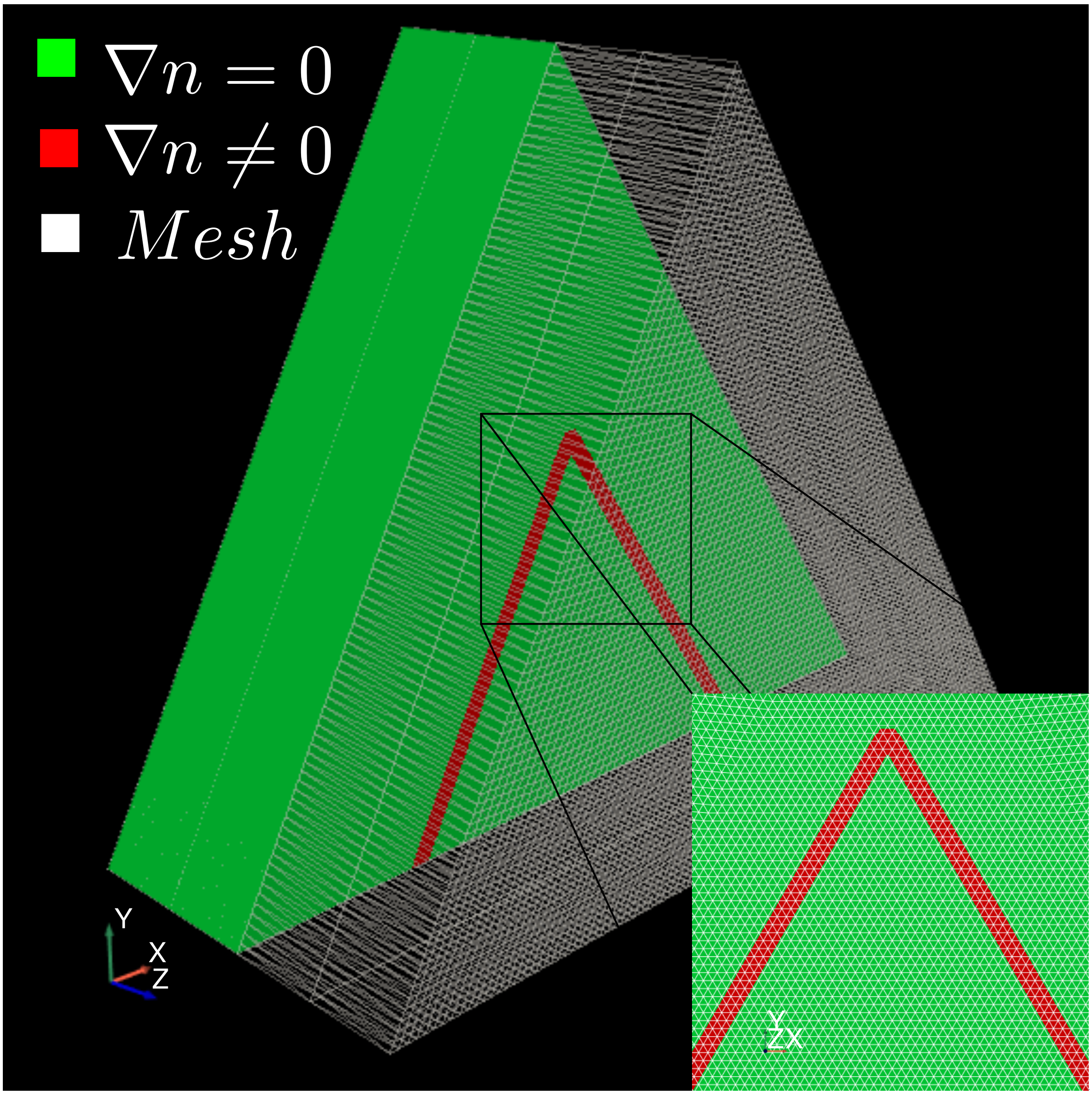


Figure 4.9: Details of the mesh and gradient of refractive index distribution in the glass dispersing prism immersed in air, the multiple refraction verification case.

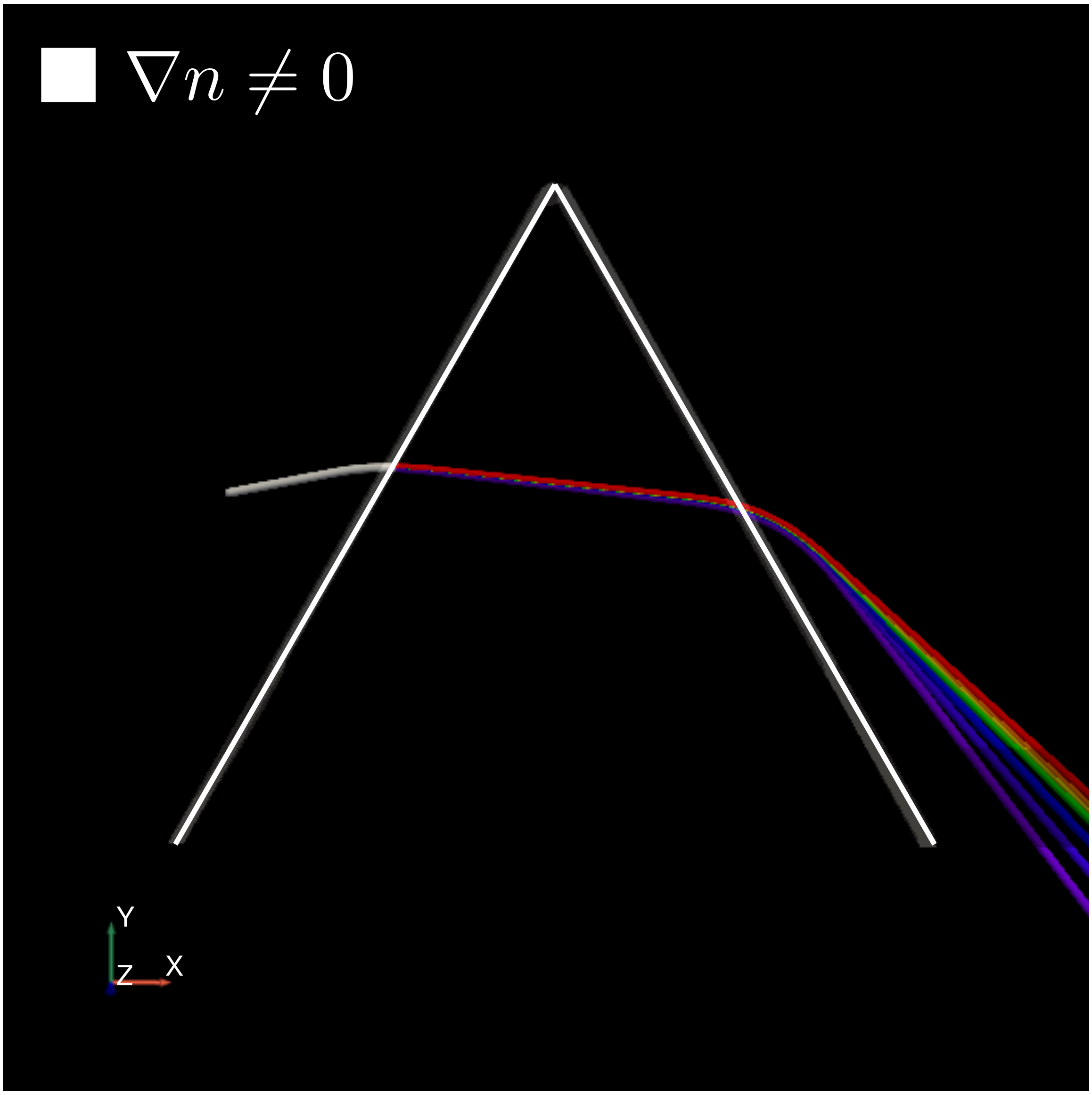


Figure 4.10: Traced light rays emitted from air, refracted and transmitted through the glass dispersing prism.

Table 4.3: Numerical results of the multiple refraction verification case.

Colour	$\delta\,[^\circ]$		Percentage Difference [%]
	Computational	Analytical	
Violet	63.1654	62.9388	0.3600
Indigo	60.0292	59.8182	0.3528
Blue	57.4135	57.5481	0.2340
Green	55.1565	55.6020	0.8013
Yellow	54.2838	54.5110	0.4170
Orange	53.8078	53.9851	0.3286
Red	52.8722	52.9165	0.0836

It can be seen that the errors in predicting the dispersion angles are small, with the highest error of 0.8% for the green light rays.

4.2 Validation of Light Reflection and Image Synthesis

The verification assessments described so far only check that the mathematical modelling of light propagation and refraction through different homogeneous media is accurate. This indicates that tracing light from the sun to an aircraft wing and also from a wing to a camera will be modelled correctly and that the refractions due to density variations in the air will be appropriately captured. However, there still remains another two important elements of the simulation of shockshadow images seen by an observer or a camera that need to be assessed. These are the light reflection at the wing surface after it is illuminated and the image synthesis or rendering process that occurs in the pixels of the viewing system after reflection. These elements link the two light ray passes through the solution of the Rendering Equation (see Section 3.4.1) using physically-based properties of materials and the photon mapping technique.

The physical reality of the in-flight shockshadow formation is still unexplored from an experimental, wind tunnel perspective. However, traditional shock wave visualisation techniques, such as shadowgraphy and schlieren imaging, are based on the same behaviour of light within a compressible flow field and share many similarities with the current study. There is a variety of image data related to these traditional methods available in the literature that can be compared to computational results to check the mathematical modelling of light reflection and image synthesis.

The work of Panda and Adamovsky [181] represents a suitable alternative to support the computational imagery validation assessment, since it contains both qualitative and quantitative experimental data concerning light scattering by shock waves. The tests detailed in the paper involve the shadowgraphy technique, performed at a range of different wind tunnel conditions, for the case of supersonic flow over an axis-aligned cylinder. The available data includes raw and processed shadowgraph images of a steady shock wave formed in front of the test model, apart from light intensity distributions relative to the latter. This type of data is ideal to be compared against computational results of light simulations set under conditions as close as possible to the experimental reality, in order to determine if the chosen numerical models from the field of computer graphics are suitable for the purpose of reproducing shockshadow images.

4.2.1 Summary Overview of Test Case

The test case from their paper that is considered in this study is summarised as follows, including some of the specifications of the experimental facility where the tests were performed:

- $127mm$ x $127mm$ wind tunnel test section;
- supersonic freestream Mach number of $M\infty = 1.5$;
- freestream total pressure of $P_0 = 134kPa$;
- freestream total temperature of $T_0 = 294K$;
- a cylindrical bluff body with diameter of $12.7mm$;
- a $4mW$ He-Ne laser ($\lambda = 632.8nm$) as a coherent light source;
- test section with two plexiglass windows with thickness of $t = 25.4mm$ each;
- a semi-transparent screen placed normal to the laser path at a distance of $685mm$ away from the shock location to capture the scattered light; and
- a low light $12-bit$ 576 x 384 pixel CCD camera to photograph the resulting patterns.

The supersonic flow over a bluff or blunt body is a classical aerodynamic problem. Characteristically, a strong bow (curved) shock wave forms in front of the body, detached from it and located at a certain distance upstream from its nose. This type of compressible flow has been thoroughly studied, both from the analytical, numerical and experimental perspectives, given its extreme importance for, for example, missile ballistics and the atmospheric re-entry of high-speed vehicles [60].

However, the solution of this flow field is not at all straightforward [60]. In fact, the general curved shape of the shock wave and its location with respect to the body could not be theoretically predicted [182]. Attempted analytical treatments of this aerodynamic problem involved too many assumptions and approximations and were not applicable to any arbitrary blunt body and flow combination, therefore proving to be inadequate [183]. Early numerical techniques were in turn limited by the then available computational power and, although somewhat more accurate than the existing analytical methods, they were still in general unsuccessful in addressing general supersonic blunt bodies [60].

This rather three-dimensional flow contains complex non-uniform regions of mixed supersonic and subsonic flow between the bow shock wave and the axisymmetric body. This changes the nature of the flow governing equations from hyperbolic to elliptic, across the so-called sonic line that divides the previously mentioned regions [60]. Due to the subsonic region, the shock wave not only depends on the freestream, but also on the far-downstream conditions (geometry and flow) [182], and a technique that encompasses both regions simultaneously becomes mathematically and numerically cumbersome [60]. Additionally, other dominant features that further complicate the flow, such as expansion waves and embedded shocks, also arise as a result of the sharp corner formed by the face of the body and the separation of the boundary layer from the body surface [184]. In fact, the onset, evolution, interaction and impact of these intricate compressible and viscous effects, particularly flow separation, are still a major challenge even to state-of-the-art computational models [185].

For a truly satisfactory analysis one must then rely on the solution of the full fluid flow governing equations [60, 183]. Although there exist some specific algorithms for the calculation of supersonic flow over bluff bodies [60, 183, 184], the implementation of such an algorithm is outside the scope of this research, which is rather built on standard computational fluid dynamics solvers. Therefore, simulations of the fluid flow scenario from [181] are performed using the OpenFoam CFD code described in Section 2.5.1.

4.2.2 Domain Discretisation

The fluid flow domain discretisation is simply performed using the *blockMesh* mesh generation utility supplied with OpenFoam [186]. In this utility, boundary faces are organised into sets of *patches* or regions [186], which aid the application of boundary conditions. Assuming that both the cylinder and the shock wave are axisymmetric three-dimensional surfaces of revolution [181], a basic patch type that is adequate to the physical arrangement being reproduced is the *wedge*, represented in Figure 4.11. This patch type is used specifically for axisymmetric cases [186–188]. This greatly simplifies the simulation, as it enforces an axial symmetry constraint and means that the entire computational domain does not need to be meshed, thus reducing the computational cost of the simulation.

The whole computational domain is specified as a wedge of small angle and only one cell thick [186–188]. The size of the cell in the tangential direction is exaggerated in Figure 4.11 only to

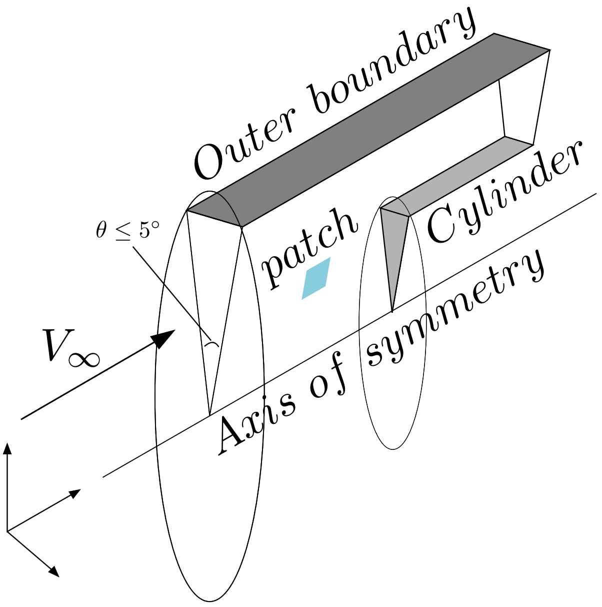


Figure 4.11: Axial symmetry constraint with the *wedge* patch type in OpenFOAM.

aid visualisation. In reality, the small angle used to construct the wedge defines a very small arc length, thus appropriately representing both the cylinder and the wind tunnel boundaries even though a straight line is used to join the two vertices. The cylinder is mounted with its axis along the flow direction [181]. As seen in Figure 4.12, the flow field is decomposed into three different three-dimensional hexahedral blocks [186]. Although no specific shock fitting or capture techniques in grid adaptation approaches are employed in the simulation, a different number of cells is assigned to each block and the uniform expansion rate parameters available in the utility are differently tuned for each direction of the cross-sectional plane of the patch (see Figure 4.12). This ensures the refinement of the very small region in front of the cylinder along its axis where a bow shock is expected. The refinement is progressively adjusted until a suitable shock wave solution is obtained. The final mesh contains approximately 3.2×10^5 points and 1.6×10^5 cells.

4.2.3 Details of the CFD Simulation

Panda and Adamovsky [181] give minimal information about the wind tunnel, however they direct the reader to the work of Willis et al. [189] and a detailed description of the 1×1-ft supersonic wind tunnel at the NASA Glenn Research Center can easily be found in the respective user manual [190]. Oddly, in [190] the tunnel is said to be able to operate only in the discrete freestream Mach numbers of 1.3, 1.6 and higher, by changing the nozzle installation at

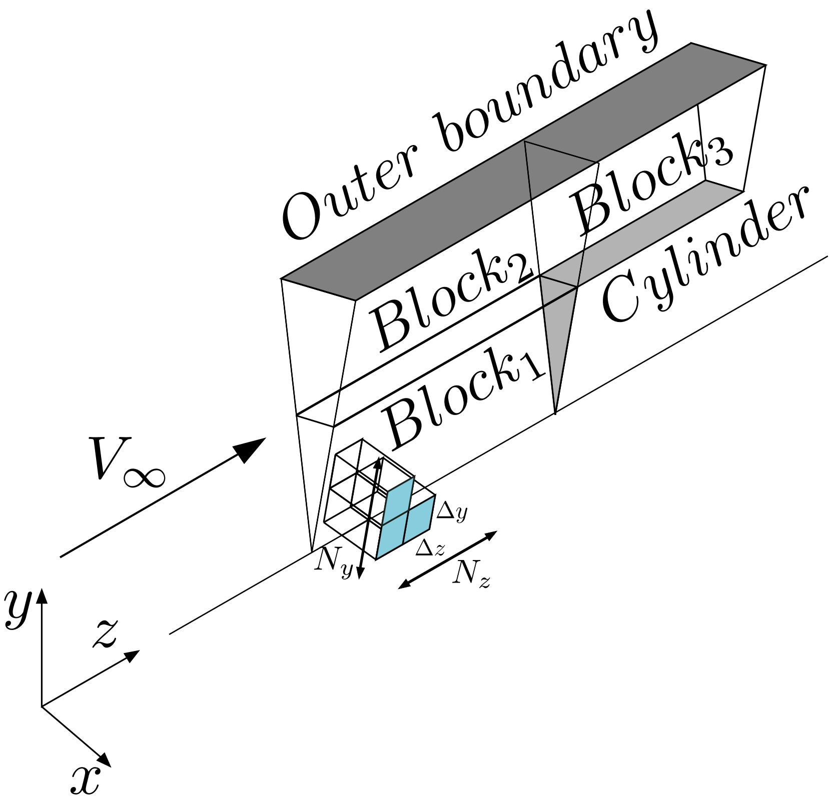


Figure 4.12: Details of the *wedge* type of boundary condition used in the CFD simulation.

the test section, but it is not reported to operate on 1.5. Similarly, in [189] the closest upstream reference conditions presented are of $M_{ref} = 1.27$ and 1.58, which are associated to the nominal values presented in the manual, but disagree with the specification in [181]. Therefore, it is very difficult to know the exact conditions at the wind tunnel when Panda and Adamovsky [181] performed the shadowgraphy experiments and to accurately model the experimental facility. Nevertheless, it is assumed that the specifications presented in [181] are accurate.

For a simplified simulation the circular outer boundary is specified such that the cross-sectional area of the flow domain is the same as the cross-sectional area of the wind tunnel ($A_{wt} = 0.016129\ m^2$). The outer boundary cross-section is circular, whereas the tunnel has a square cross-section. Making the cross-sectional area the same makes the outer boundary radius $0.07165\ m$, whereas the cylinder has a radius of $0.00635\ m$. The outer boundary location based on this approach is therefore located at more than 10 cylinder radii from the cylinder surface. Thus the flow at the boundary is simulated as a parallel flow on the simplifying assumption that as the model is small compared to the size of the wind tunnel, the effects of the walls and associated boundary layer can be simplified. This is a form of inviscid boundary condition and it significantly reduces the cost of a RANS simulation of the flow over the cylinder, which requires a fine mesh on all solid surfaces.

From the given freestream total conditions and Mach number [181], isentropic flow equations (Eq. 4.4 [60]) are then applied to find the static pressure ($P_\infty = 36\ kPa$), static temperature ($T_\infty = 202\,K$) and velocity ($V_\infty = 427\ m/s$) at the inflow, where R is the specific gas constant and γ is the heat capacity ratio of air. These values are then employed as the initial conditions of the CFD simulation, which result in a representative converged solution containing a stationary bow shock wave formed in front of the cylinder.

$$V_\infty = M_\infty \sqrt{\gamma R T} \tag{4.4a}$$

$$\frac{T_0}{T_\infty} = 1 + \frac{\gamma - 1}{2} M^2 \tag{4.4b}$$

$$\frac{P_0}{P_\infty} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}} \tag{4.4c}$$

Due to the wedge type of flow domain used in OpenFOAM, the converged stationary density solution is essentially two-dimensional, as shown in Figure 4.13. Based on the initial assumption of symmetry along an axis, the solution is simply replicated via $360°$ rotation around this axis in order to compose a 3D solution, as represented in Figure 4.14. This is an approximation, as the wind tunnel is not cylindrical shaped and some three-dimensional effects are lost. The use of a circular, rather than a square boundary, and the simplification of the effects that the boundary layers that develop on the real wind tunnel walls exert over the internal flow can have a more significant influence than it is assumed. However, this allows the quick comparison between real and simulated shadowgraphs without having to attempt to model every aspect of the wind tunnel experiment. The true wind tunnel wall shape and boundary layers would probably have to be accounted for in the solution if the bow shock was to be reproduced with better accuracy.

4.2.4 Light Source, Light Sensor and Rendering

In the light simulation, a Gaussian signal is used to represent the spectral power distribution of the source, i.e. that of a red gas Helium-Neon laser beam. The signal is centred around the wavelength of emission and has the power amplitude specified in [181], as shown in Figure 4.15. The maximum spectral power value slightly decreases due to the deviation of the curve from a Dirac delta function, but upon integration the total power of the laser is conserved.

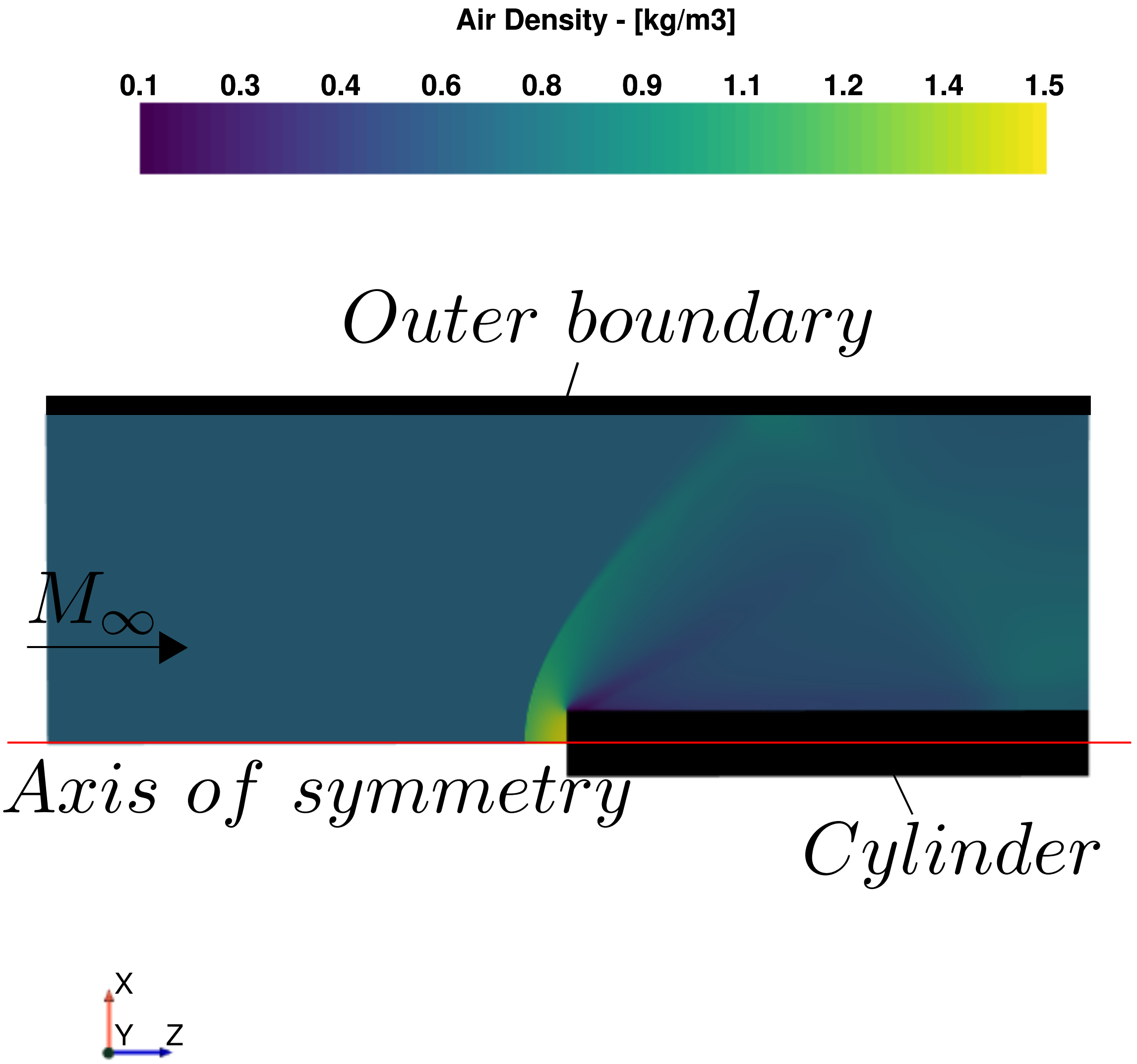


Figure 4.13: CFD two-dimensional air density solution of supersonic flow over an axis-aligned cylindrical bluff body.

Light rays are emitted in the same global direction, parallel to the cross-sectional plane, analogous to the large diameter collimated beam obtained with the expander shadowgraph system employed in [181]. These light rays are then traced through the compressible aerodynamic flow. The windows in each side of the wind tunnel test section [181] and the corresponding refractions that would occur when the laser entered and exited the section are not considered. A plane-shaped sensor is appropriately positioned at the opposite side of the flow domain, capturing the deflected light rays in a similar manner to the screen used in [181] and at the same location relative to the cylinder. Figure 4.16 depicts the whole scene.

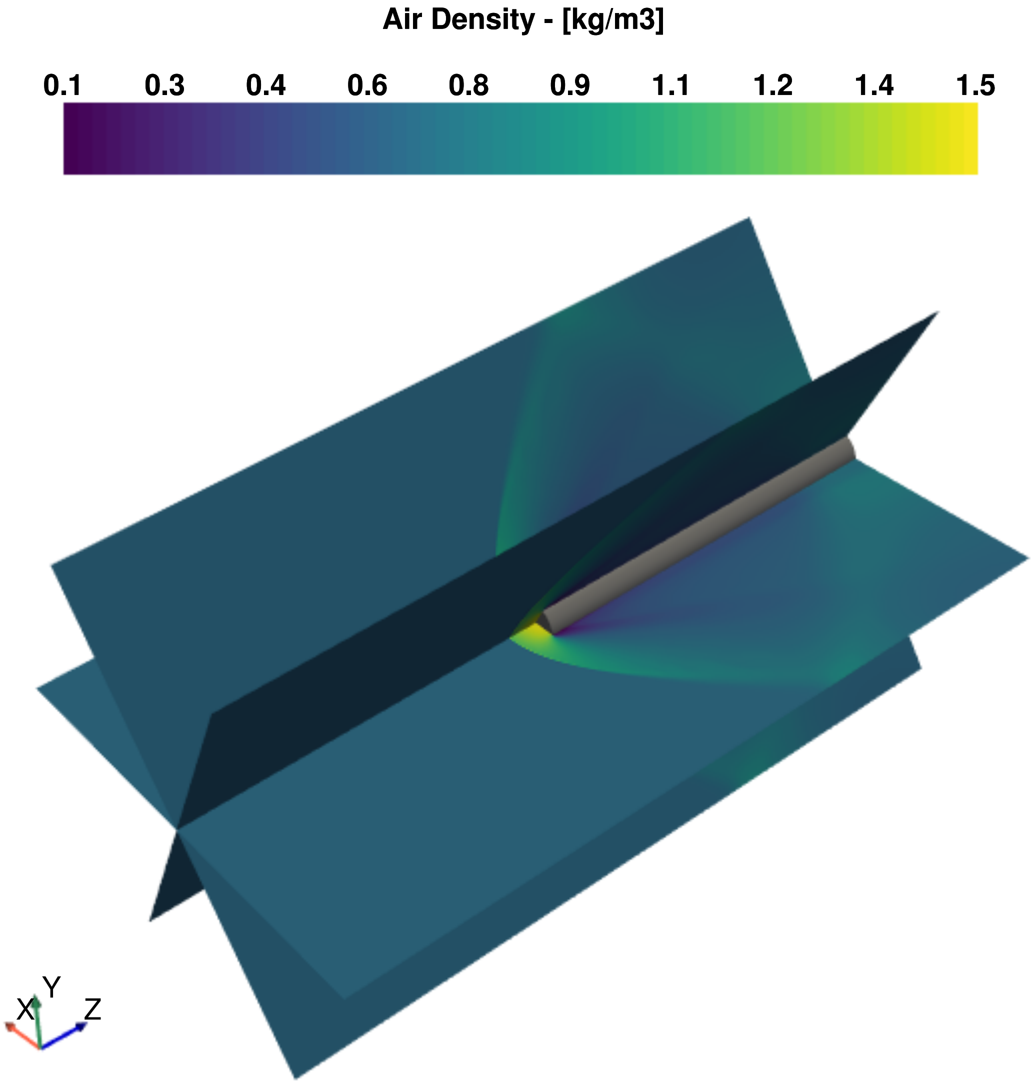


Figure 4.14: CFD three-dimensional axis-symmetric air density solution of supersonic flow over an axis-aligned cylindrical bluff body.

The sensor is then considered to behave as a perfect diffuse material, reflecting light equally in all directions. A virtual pinhole camera is positioned aligned with, and directed towards its plane, mimicking a photograph being taken from the screen, as it is usually done in real shadowgraphy. Subsequently, employing the backward nonlinear Ray Equation solution algorithm to trace shadow rays from the viewing system to the sensor and the photon mapping technique without any filtering, i.e. using a uniform kernel, the Rendering Equation is solved to estimate the radiance values of the pixels that compose the image plane.

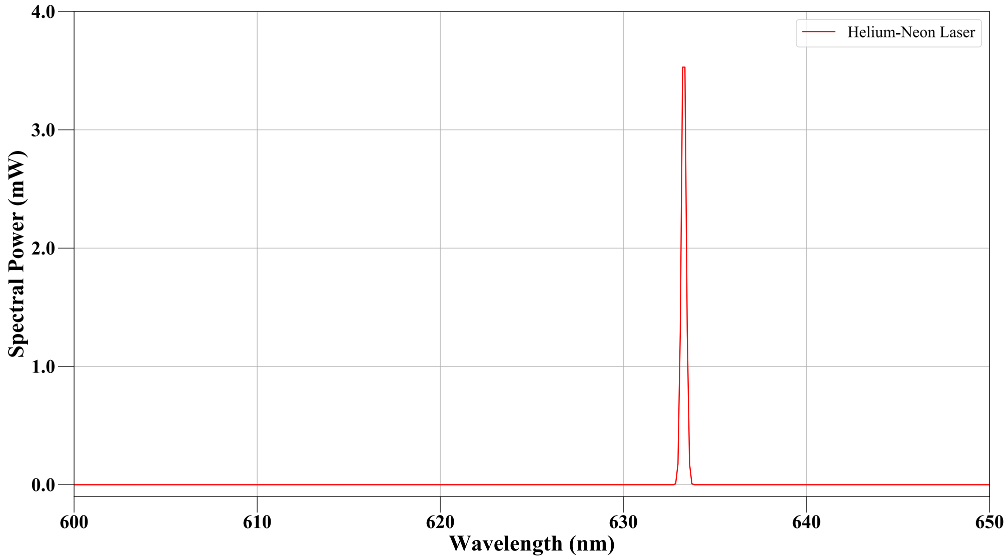


Figure 4.15: Red gas He-Ne laser spectral power distribution representation.

Additionally, a second image of the same scene is generated in the same manner as before but, instead of considering the aerodynamic flow, a homogeneous medium with no density nor index-of-refraction variation is assumed instead. This is the "no-flow" image mentioned in [181], as opposed to the previous "flow" [181] image. In both images, the sRGB radiance values estimated are converted to the CIE XYZ colour space and the Y channels, which corresponds to luminance photometric measurements, are used to generate the results. Before being rendered, the resulting luminance distribution is also gamma corrected and normalised to 8-bit values.

At last, the processed shadowgraphs differ from the raw images regarding the "no-flow" image being subtracted from the "flow" image, in order to highlight the effects that the shock wave exerts over light [181]. Furthermore, while the raw image visualises approximately three cylinder radii in the radial direction, including a part of the cylinder itself, as in [181] the processed image is focused on a very narrow region in front of the cylindrical body along its axis, where the tip of the bow shock wave is situated.

4.2.5 Data Comparison

The shock shape and its detachment distance (better known as the standoff distance) are the main parameters that aerodynamicists try to accurately predict [183]. These parameters depend on the geometry of the body and on the freestream Mach number [60, 182]. The standoff distance is measured along the central streamline or axis of symmetry (in axisymmetric flow/body cases), from the stagnation point at the face of the body to the normal and stronger

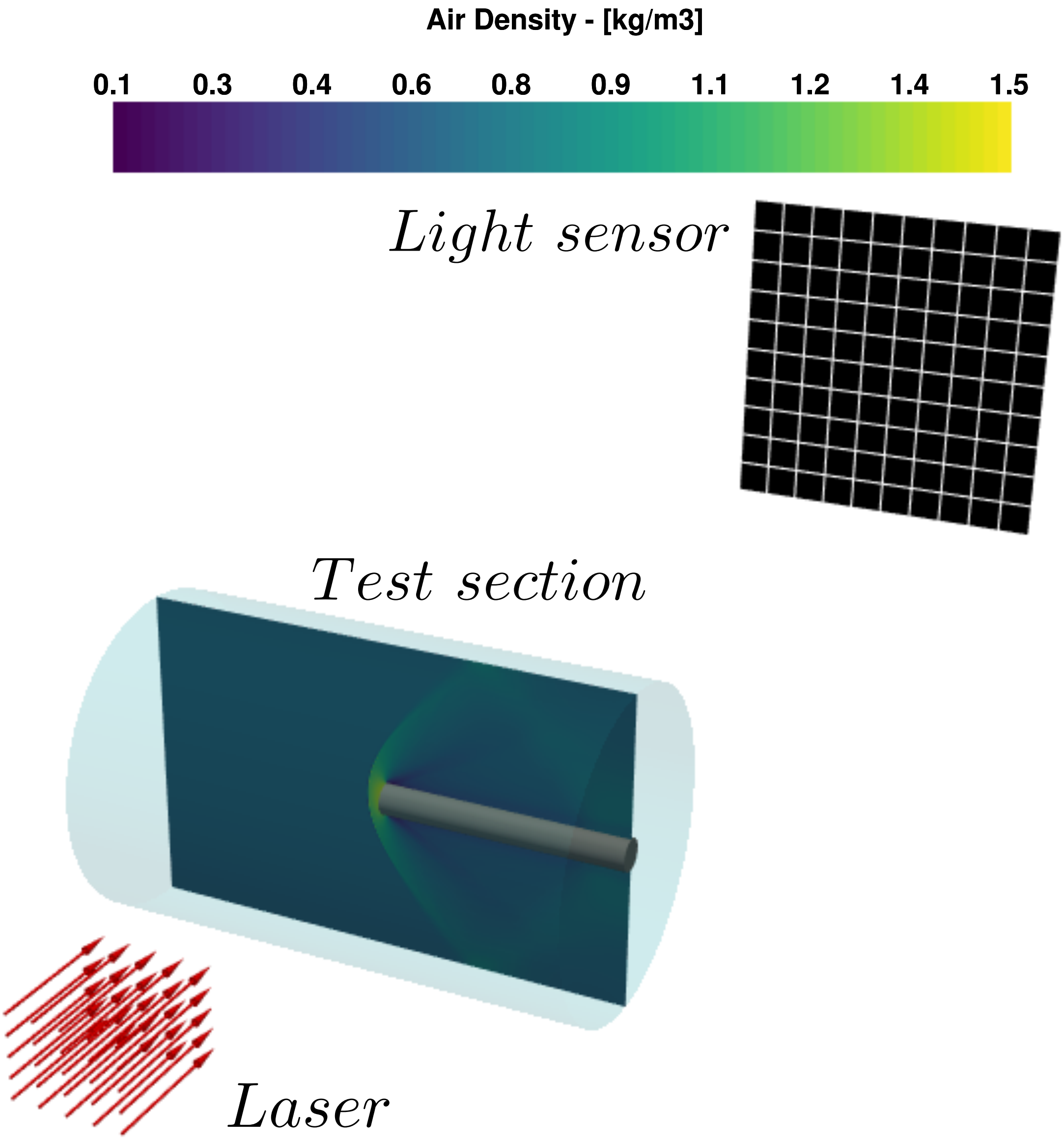


Figure 4.16: Schematic of light source and viewing system arrangement in the shadowgraphy simulation.

portion of the curved shock wave. It is then a convenient test of accuracy for any technique that attempts to predict the supersonic flow over a bluff body. This is because on either shadowgraph or schlieren photographs, it can be practically measured considering the upstream edge of the darker and wider shadow as the location of the shock front [22, 181, 183]. Since in this research the simulated direct shadowgraphs are derived from the supersonic flow prediction technique itself, i.e. the CFD, standoff distances measured on these virtual images can then be compared to values obtained from analogous experimental pictures or calculated using another already well established numerical method. The latter requires an additional implementation, which is judged unnecessary for the present purposes, and as accurate as the algorithm can be some degree of error will invariably remain.

The difficulty in using experimental data containing standoff distance information lies on the fact that the bluff body in [181] is a body of revolution with a flat face or flat nose, whereas the majority of the data available corresponds to blunt-nosed bodies or circular cylinders (not axis-aligned) and spheres [191]. Nevertheless, Roshko and Liepmann [182] provide a curve for the standoff distance of shock waves in plane and axisymmetric flow over flat-nosed bodies of revolution, for different freestream Mach numbers and body dimensions (highlighted in red in Figure 4.17). The curve is based on experimental data (e.g. interferograms from [192, 193]) for air with constant specific heat.

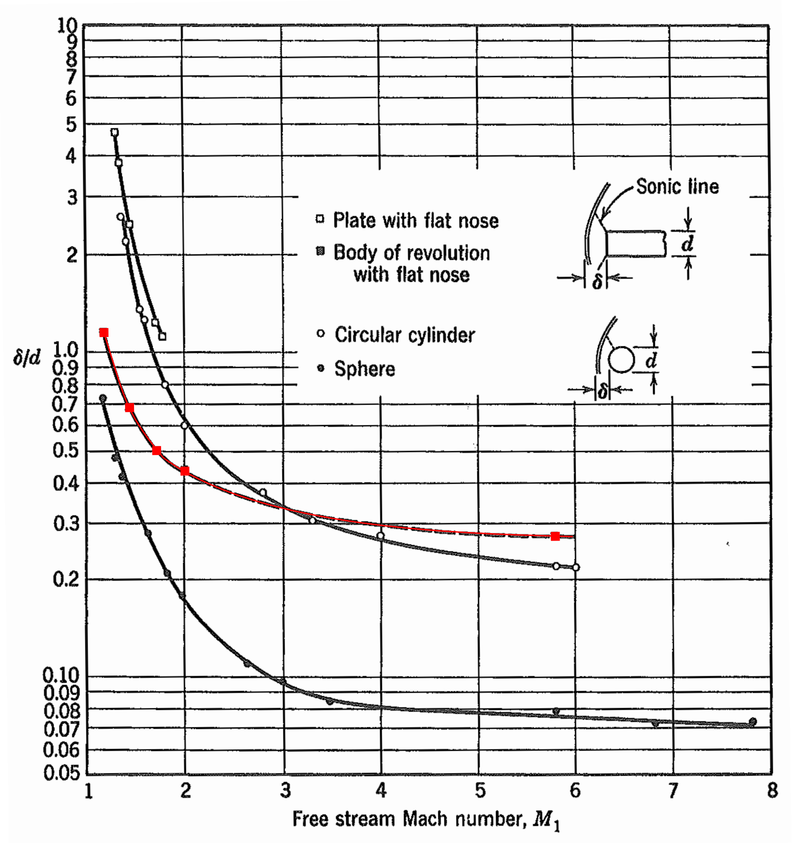


Figure 4.17: Shock wave standoff distance for plane and axisymmetric flow over bluff bodies. Figure reproduced from [182].

Considering the freestream Mach number of $M_1 = 1.5$ and cylinder diameter of $d = 12.7\ mm$ described in [181], the value of $\delta = 7.98\ mm$ for the standoff distance is (indirectly) numerically interpolated from the curve in [182]. In the raw shadowgraph for the case of $M_\infty = 1.5$ presented in [181], the standoff distance can also be measured employing appropriate software that uses the cylinder diameter, distinguishable in the picture, as the scale. As shown in Figure 4.18, the value of $\delta_{exp} = 7.81\ mm$ is measured using the outer edge of the main shadow as the reference. This very closely agrees with the value interpolated from the curve. The small difference is, most likely, due to the disparities between the shadowgraphy from [181] and the interferograms that compose the curve in [182]. These disparities are in both the air flow established at each respective wind tunnel and in the subsequent image capture and processing techniques used.

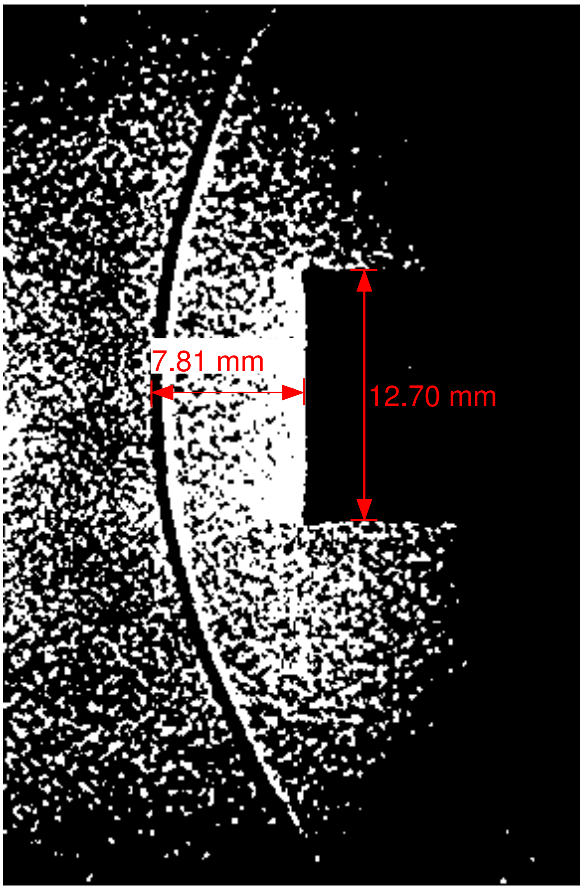


Figure 4.18: Experimental raw shadowgraph of a supersonic axis-aligned cylinder (reproduced from [181]).

A similar measurement can then be performed on the distribution of the density gradient calculated from the CFD solution during the light simulation (see Section 3.2.3.2 and Figure 4.19). In this plot, it is difficult to the determine which vertex or cell along the axis of symmetry to use as a reference to measure the standoff distance. With the level of mesh discretisation used at the shock wave region, the non-zero density gradient is spread over multiple adjacent cells, rather than a single surface of discontinuity across two cells. These cells containing a gradient different than zero lie from $7.8\ mm$ to $8.4\ mm$ ahead of the cylinder face, with the peak in gradient (at this region) at around $8.0\ mm$, as shown in Figure 4.19. The values obtained from the experimental curve-fit interpolation and shadowgraph image measurement are therefore within this range. Further refinements of the mesh and possibly by addressing the previously mentioned modelling assumptions and approximations in the CFD simulation could then improve this solution and bring the shock wave, as a thinner and more abrupt discontinuity, closer to the shorter end of this range of distances.

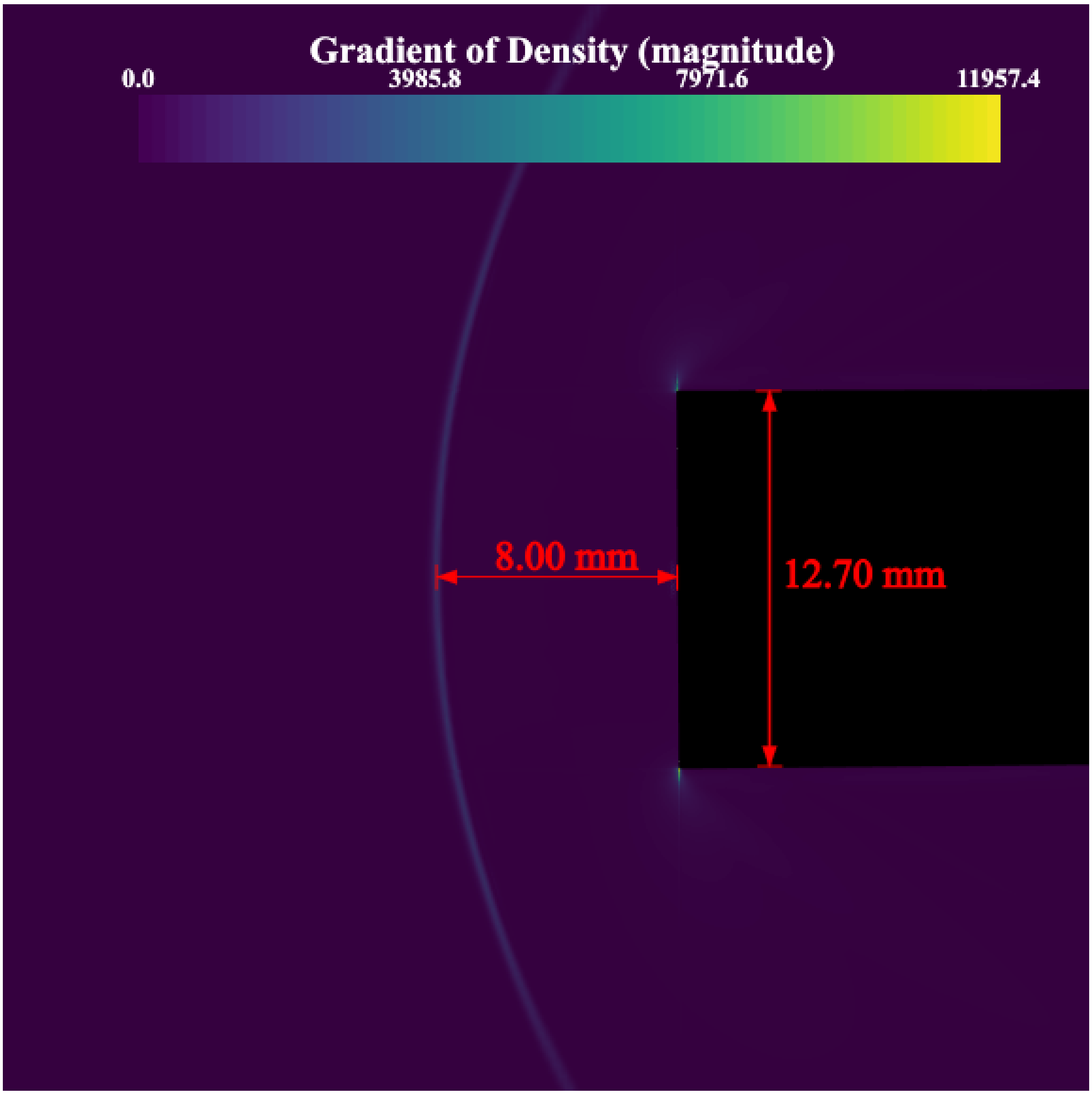


Figure 4.19: Slice of the three-dimensional density gradient distribution calculated from the CFD solution of the supersonic flow over an axis-aligned cylinder.

On the actual simulated raw shadowgraph, presented in Figure 4.20, the same difficulty associated with the density gradient distribution when measuring the standoff distance is again observed. It becomes hard to ascertain if the different levels of grey displayed in the central pixels are only a result of, for example, the choice of searching radius or bandwidth in the density estimation of photon mapping, which could have detected slightly fewer photons than in the complete homogeneous regions. In this case, these darker pixels are not necessarily related to the shock wave influence and do not indicate the upstream edge of the shadow. In any case, the distance of $\delta_{num} = 7.8\ mm$, shown in Figure 4.20, terminates at the region where, from visual inspection, the pixels are becoming closer to black rather than intermediary shades of grey. As before, this could be established as the outer edge of the shadow, which would indicate a close agreement with the curve-fit interpolation and experimental shadowgraphy measurement, although this is only subjective in this image. As previously mentioned, further improvements of the CFD solution would aid the proper location of the shock wave and subsequent measurement of the standoff distance to correctly confirm this agreement with the practice.

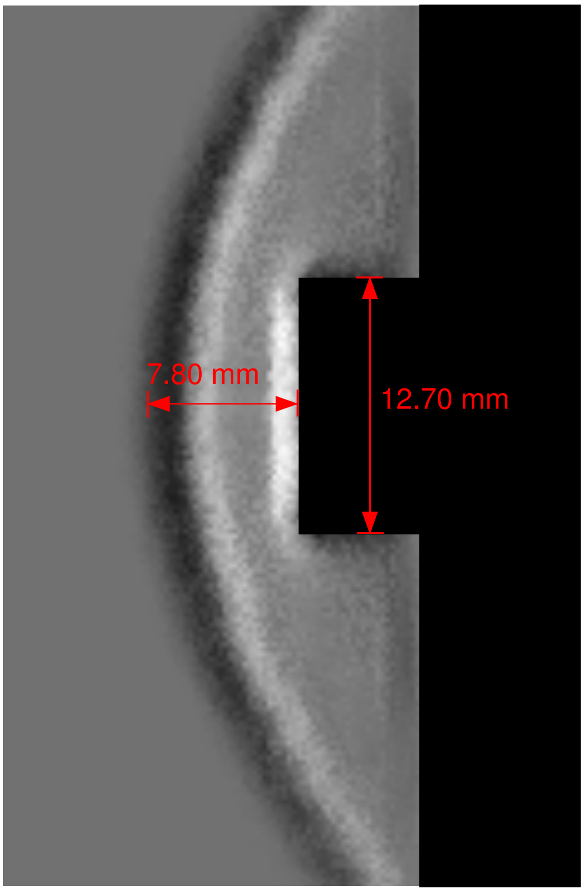


Figure 4.20: Computational raw shadowgraph of a supersonic axis-aligned cylinder, constructed with the first implementation of the present methodology.

Figure 4.21 presents the simulated raw shadowgraph compared against the experimental analogous, where, only to emphasise, flow is from left to right. It is possible to see in both images the main features of the light and compressible flow interaction, i.e. the formation of a dark band, representing the shadow of the shock wave [181], and of a light fringe downstream of the shock. However, both the shadow and the caustic bands appear thicker in the simulated image, due to the previously described limiting characteristics of the CFD solution embedded in the light simulation. Nevertheless, the outer edges of the shadows seem to approximately coincide in position and even more in shape. In fact, the curvature of the simulated bow shock wave very closely matches that of the one in the experiment. This shock shape comparison via visual inspection is believed sufficient for the present purposes, especially considering that the subsequent analysis concentrates only on the normal part of the shock. Otherwise, other techniques (e.g. [191]) could be used to also predict this shape and confirm the shock waves of both the experimental and simulated shadowgraphs.

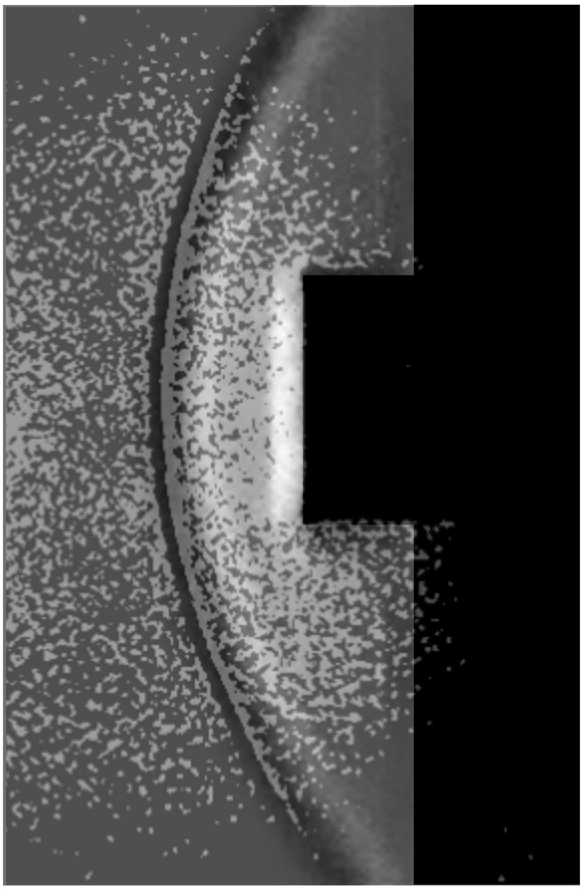


Figure 4.21: Experimental raw shadowgraph of a supersonic axis-aligned cylinder (reproduced from [181]) overlapping the analogous computational image.

As Panda and Adamovsky [181] described, for a fixed freestream Mach number the shock wave shape and location are not affected by changes in the freestream total pressure, but only the corresponding illumination patterns become weaker. This means that a weaker refractive medium (i.e. density gradient) resulting from a lower freestream total pressure produces thinner dark and light fringes with fading shades of black and white, comparatively to an image generated in a higher total pressure. Moreover, the computational raw shadowgraph simulated from the $M_\infty = 1.5$ and $P_{0_\infty} = 134\ kPa$ flow test conditions has already shown great agreement with the equivalent experimental photograph. Therefore, it is deemed unnecessary to also simulate the remaining available condition of higher freestream total pressure with the same Mach number, for which a raw shadowgraph is not presented in [181], but only a processed and very similar image. The additional condition of higher freestream Mach number is not analysed either, since the respective processed shadowgraph is not provided in [181].

Finally, Figure 4.22 presents the simulated processed shadowgraph together with its equivalent experimental photograph. Further, to replicate completely the whole experiment of Panda and Adamovsky [181], the luminance values of the "flow" image are averaged over the columns of pixels perpendicular to the centre line of the cylinder. In [181], a similar process was done to reduce background speckle noise produced by dust particles present on the wind tunnel windows. These averaged luminances are then normalised by the average luminance of the whole "no-flow" image and the resulting simulated distribution is then plotted in Figure 4.23, alongside the equivalent experimental curve obtained from [181].

In Figure 4.22, both images are appropriately scaled according to the distance from the cylinder to the point where the luminance distribution in Figure 4.23 crosses $L = 1$, i.e. the boundary between the main shadow and light fringe. The experimental shadowgraph is then placed at a position considered appropriate for the comparison, given that the initial curvature analysis demonstrated a good agreement of the simulation and, instead of being approximately normal or vertical, the illumination pattern on the experiment seems to be curving to the right from top to bottom.

Again, in these images the main illumination features arise (wider dark and bright bands), however the remaining less bright and thinner light fringe upstream of the shock is not present in the synthetic shadowgraph. This happens because the simulation is based on geometrical optics (ray tracing and photon mapping techniques), which does not take light diffraction into account. Diffraction is a wave interference phenomenon [13]. Indeed, the thicknesses of shock waves usually lie within the range of the wavelengths of visible light [2, 12] and therefore diffraction effects can be significant, as in experimental shadowgraphy [194]. This thickness is determined by the freestream Mach number and the mean free path (*m.f.p.*) of air particles [194], or the average distance travelled by one of those particles before collision with other particles [195]. Typically, for a fixed Mach number, a denser gas will present a small *m.f.p.* ahead of the shock wave and the shock will then exhibit a thickness approaching its zero value, i.e. a proper surface of discontinuity. In this case, wave optics effects play a significant role in the physical problem and diffraction is of major importance. Differently, with a larger degree of

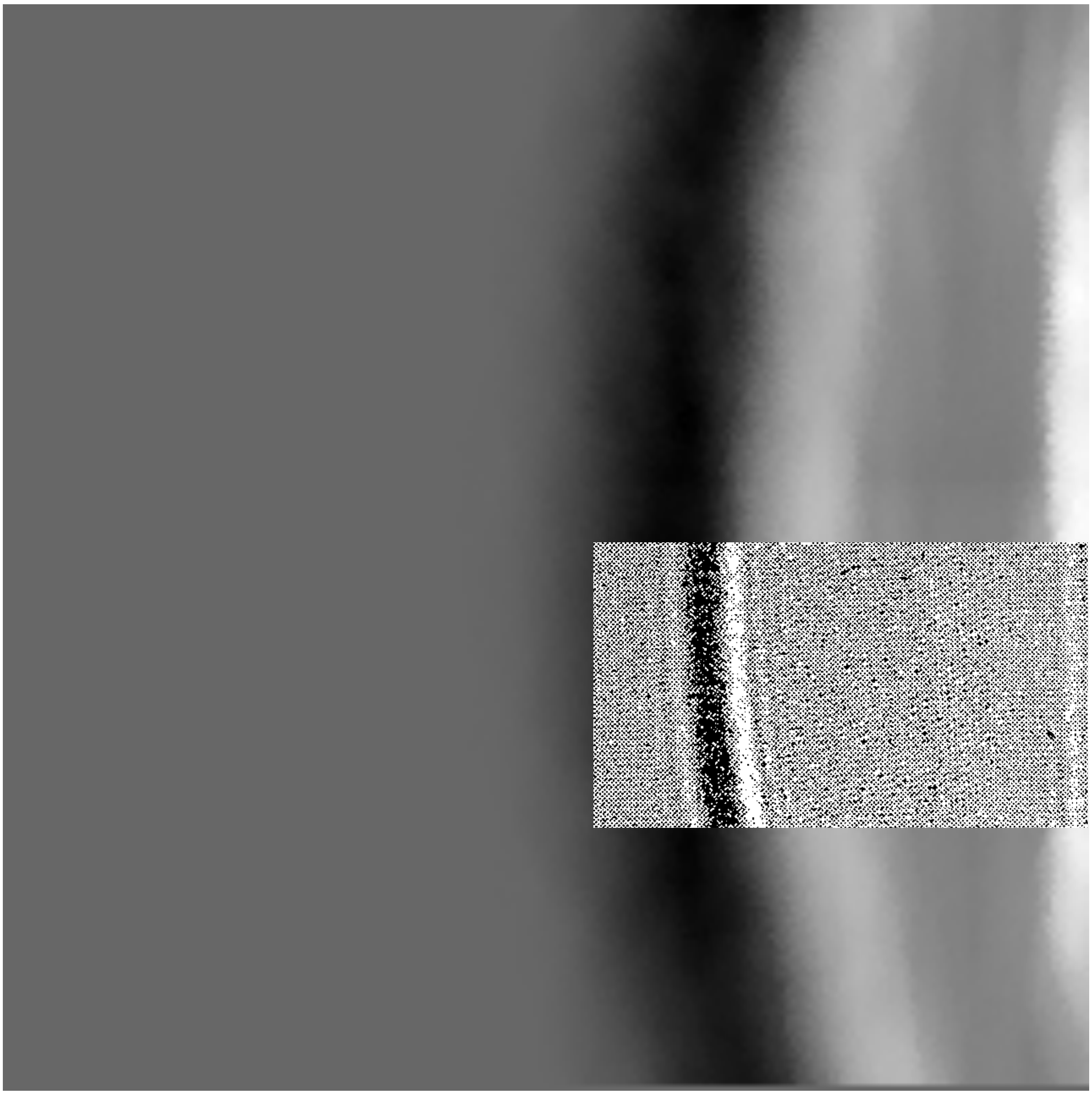


Figure 4.22: Experimental processed shadowgraph of a supersonic axis-aligned cylinder (reproduced from [181]) overlapping the analogous computational image.

gas rarefaction in front of the shock, the *m.f.p.* and the shock wave thickness increase to the order of millimetres and tens of millimetres, respectively. Geometrical optics is then appropriate to treat the physical problem and diffraction disappears [194].

At the average cruise altitude of commercial aircraft, the low atmospheric pressure determines a *m.f.p.* of the order of hundreds of micrometres, which characterises conditions of low to medium vacuum [196] and a somewhat thicker shock wave. Furthermore, CFD simulations are unlikely to converge to an infinitesimal length scale of density gradients, but rather result

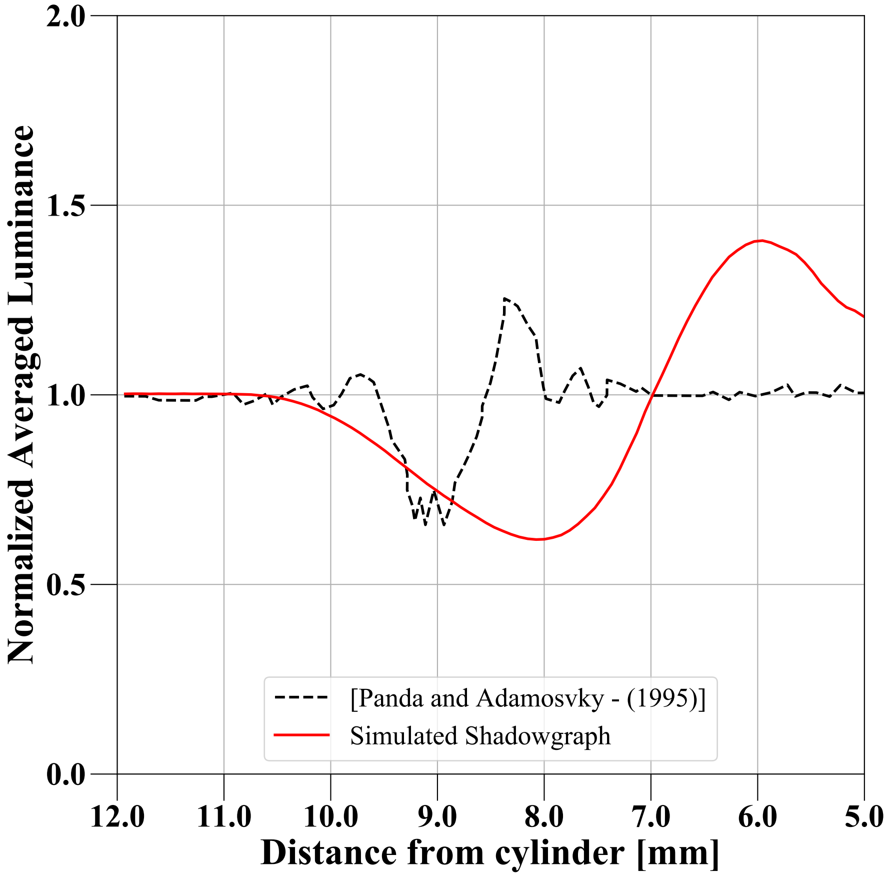


Figure 4.23: Experimental (reproduced from [181]) and computational normalised averaged luminance distributions obtained from the processed shadowgraphs of the supersonic axis-aligned cylinder.

in a shock wave density profile depicting a finite discontinuity, usually thicker than expected. In the present case, the shock wave is not even a two-dimensional plane and light rays are never exactly parallel to its surface, conditions that are known to bring about diffraction. The discontinuity is actually a three-dimensional surface of revolution with different orientations with respect to the light source direction. All these factors support the application of geometrical optics to construct the light simulation framework, thereby neglecting light diffraction. For a simple initial analysis of shock wave location and measurement, the refractive medium itself and the resulting light deflection towards a denser region are more relevant, and light diffraction, even if it happens to a certain extent, can be left from the modelling. If the more complex reconstruction of the shock wave thickness and its internal structure were required, then one would have to confirm the disappearance of the physical optics influence, possibly by ensuring a high degree of rarefaction in front of the shock wave [194]. Alternatively, if diffraction was relevant to the analysis, then one could employ hybrid models that combine geometrical and physical optics to predict light transport [197].

The dark and light bands appear in Figure 4.23 as the luminance valley and peak, respectively. However, the first apparent discrepancy is the approximately three times larger widths of these bands in the computational image. As observed in both the raw and processed shadowgraph images comparisons, and previously explained, this is directly attributed to the mesh coarseness and respective CFD solution utilised. There is also a considerable shift of the illumination features to the right, towards the cylinder, in the simulated distribution. Apart from the influence of the fluid flow domain discretisation, this is also thought to be due: inaccuracies in modelling the supersonic flow over the blunt body, especially in predicting the complex three-dimensional flow separation over the cylinder flat face; possible effects of test section boundary layer growth in the wind tunnel, though these were probably accounted for in the experiment calibration; the simplification with respect to the wind tunnel model cross-section; and differences in the procedures performed during image processing. The latter also probably affect the visible difference in terms of the main luminance peak, which is almost twice as strong in the simulation than in the real experiment. More importantly, the ray tracing sampling-based technique and the photon mapping density-estimation concept also play a considerable role in this difference. Ideally, an infinitely large number of photons would be traced from the laser and an infinitely small area would be used to map these photons into a radiance estimation for each pixel of the viewing system. This would most probably smooth out the light peak and bring it to a lower value. Since photon mapping is based on a density estimation technique, as the sample size increases and the bandwidth decreases the solution will consistently converge to a more accurate value. The luminance valley however seems to be compatible with the experiment in terms of value, resulting in similarly dark shadows in the images.

The influence of the image processing is even more evident on the shock wave standoff distance of these shadowgraphs. Although for the processed shadowgraphs the real/virtual experiments are exactly the same as the one performed to obtain the raw photographs, but simply zoomed in into the central region, the standoff distance of around $7.8-8.0\ mm$ expected for the freestream Mach number of 1.5 is actually considerably larger in these processed images. In the experimental luminance distribution in Figure 4.23, the distance from the cylinder to the upstream edge of the main stronger and wider shadow is actually $9.55\ mm$. From the curve of Roshko and Liepmann [182] this would, at first impression, suggest a lower freestream Mach number of around 1.37 and possibly the wind tunnel operation at its nominal $M = 1.3$ [190]. However, this is rather believed to be a result of the averaging and normalisation procedures performed and possibly further image processing not mentioned in [181].

4.3 Discussion

The results presented in Tables 4.1 and 4.3 show relative errors of less than one percent with respect to angles calculated via Snell's Law of Refraction, thus the light simulation model is in excellent agreement with the analytical treatment of the refraction phenomenon. It is concluded that the overall Ray Equation solution process as a Nonlinear Ray Tracing algorithm is based on a suitable methodology and refractions are adequately captured. This agreement is expected as both the Ray Equation and Snell's Law of Refraction can be derived from the same principle, Fermat's principle of the shortest optical path length. The computational implementation is not only consistent but it is also versatile, as it opens the possibility of exploring the spectral nature of light when sampling the energy from the Sun, as the dispersion in the prism clearly demonstrates. The light propagation and refraction module can therefore be employed to accurately retrieve the behaviour of light rays in compressible fluid flow.

Clearly there are differences in the shadowgraph images and luminance distributions produced from CFD and experiment for the cylinder test case. The reasons for this discrepancy can be identified and include: the difficulty in recreating the experimental wind tunnel flow in the simulation at an affordable computational cost; the difficulty in knowing the exact procedures performed during image processing in reality; and the use of a finite number of light rays and photons and the impact on radiance estimation for each pixel of the viewing system. Despite these issues, it is concluded that provided the computational fluid dynamics is an accurate representation of the fluid flow reality, so will the light simulation accurately model the refraction phenomenon and respective visualisation of its effects. From the perspective of a shadowgraph application, the Rendering Equation calculation embedded into the viewing system and based on the photon mapping technique replicates the traditional flow visualisation technique computationally in its entirety. The imaging of compressible flow features, similar to the ones produced with complex and expensive facilities & apparatus, can therefore be obtained numerically given adequate light sampling and reconstruction. For the in-flight shockshadow, as of yet no experimental data is available. However, the shadowgraphy results demonstrate that the equations being solved represent the physics of the shockshadow phenomenon. Thus the light reflection and image synthesis modules can be used to, within the extent of geometrical optics and from the perspective of computer graphics, predict the interaction of light with materials and with a camera, and the shadow formation itself.

CHAPTER 5

RESULTS

The results in Chapter 4 evaluated the light simulation framework on simple test cases with analytical solutions. This showed that the framework can be used to accurately reproduce the effect of compressible flow, or any other inhomogeneous optical media, on light scattering. Further, the simulation of a shadowgraphy experiment highlighted the dependence on the computational fluid dynamics solution and on the tuning of the algorithms to obtain accurate synthetic images. This chapter advances the investigation into the target application, the computational reproduction of the natural direct shadowgraphy phenomenon on aircraft wings. A logical strategy is then to approach the modelling of such a phenomenon in sequential stages, where at each stage more encompassing aspects of the shockshadow reality are introduced. As an initial step, a pseudo-3D wing test case is created by stacking 2D RANS simulations of flow over a NACA 0012 aerofoil to produce an artificial 3D flow, i.e. a section of a rectangular wing of infinite span. This provides a test bed to understand key issues in applying the developed framework to the light simulation involving a wing surface. The knowledge gained is then used in the final 3D test case for the Onera M6-wing, where a 3D Euler simulation provides the compressible flow field data needed to investigate the remaining aspects of shockshadow formation.

5.1 Pseudo-3D Shockshadow Simulation

In this section, an initial test case to produce synthetic shockshadows over a wing surface is described. This is a fictitious test case since the 3D flow field is built using transonic flow simulations over an aerofoil. These are stacked so that effectively a finite section of an infinite rectangular wing is considered. A shadowgraph is generated and, more importantly, an initial trial of producing synthetic shockshadows over a wing surface is made. This test case demonstrates the requirement for the coupling of the photon mapping global illumination algorithm and the traditional ray tracing technique performed through the solution of the ray equation. The first ever synthetic images of a pseudo-three-dimensional shockshadow on a wing are shown and an initial analysis of sampling and light source direction dependence is presented. These results have been successfully published in [198]. This effectively provides the necessary experience to implement the framework on a true three-dimensional test case in Section 5.2.

5.1.1 Transonic Flow Over a NACA 0012 aerofoil

The computational fluid dynamics simulation for transonic flow over a NACA 0012 aerofoil is performed using the open-source software OpenFOAM [69] to solve the RANS equations. This is a well documented uncambered symmetric aerodynamic shape and is available as a standard tutorial case within the OpenFOAM package. To simulate the 2D case a 3D mesh with a single subdivision in the span-wise (third) dimension is generated using the *blockMesh* utility. The boundary condition of type *empty* is applied on the front and back boundaries, whose normals are aligned to the spanwise direction, to ensure a 2D flow [69]. This quasi-2D mesh is composed of 32480 points and 16000 cells with curved hexahedral blocks used to define both the farfield and solid surface boundaries. The freestream Mach number is $M_\infty = 0.8$ and the Reynolds number is $Re = 1.8 \times 10^6$. The initial conditions of velocity $(U_\infty = 277m/s)$, temperature $(T_\infty = 298K)$ and pressure $\left(p_\infty = 10^5 N/m^2\right)$ are imposed at the farfield, whilst a no slip boundary condition is applied to the aerofoil positioned at at an angle-of-attack of $\alpha = 0°$. The steady state solution is calculated, and the air density values in the domain above the wing are isolated and stored, as shown in Figure 5.1. To create a pseudo-3D flow field, the air density values at the vertices of the quasi-2D mesh are replicated in six layers oriented along the remaining coordinate direction, as depicted in Figure 5.2. This new isolated pseudo-three-dimensional flow domain is composed of 41310 points.

To propagate light rays through this domain, a mesh must be created. If the flow domain and the interior of the solid pseudo wing surface were both tetrahedralised using inbuilt CGAL meshing constructors, then the 3D mesh generated invariably had additional points added for mesh refinement purposes and was unconstrained. Retaining the wing surface in this mesh and restricting the meshing of its interior is essential because of the underlying approach employed to ascertain if a light ray has fallen inside the wing during numerical integration. If the mesh is unconstrained then the point location function from CGAL does not work and is simply unable to detect the light ray intersection with the wing. This issue did not arise when the framework was initially tested in two dimensions on the aerofoil, when the inbuilt 2D CGAL utilities yielded a mesh on which the CGAL point location function could detect rays crossing the boundary.

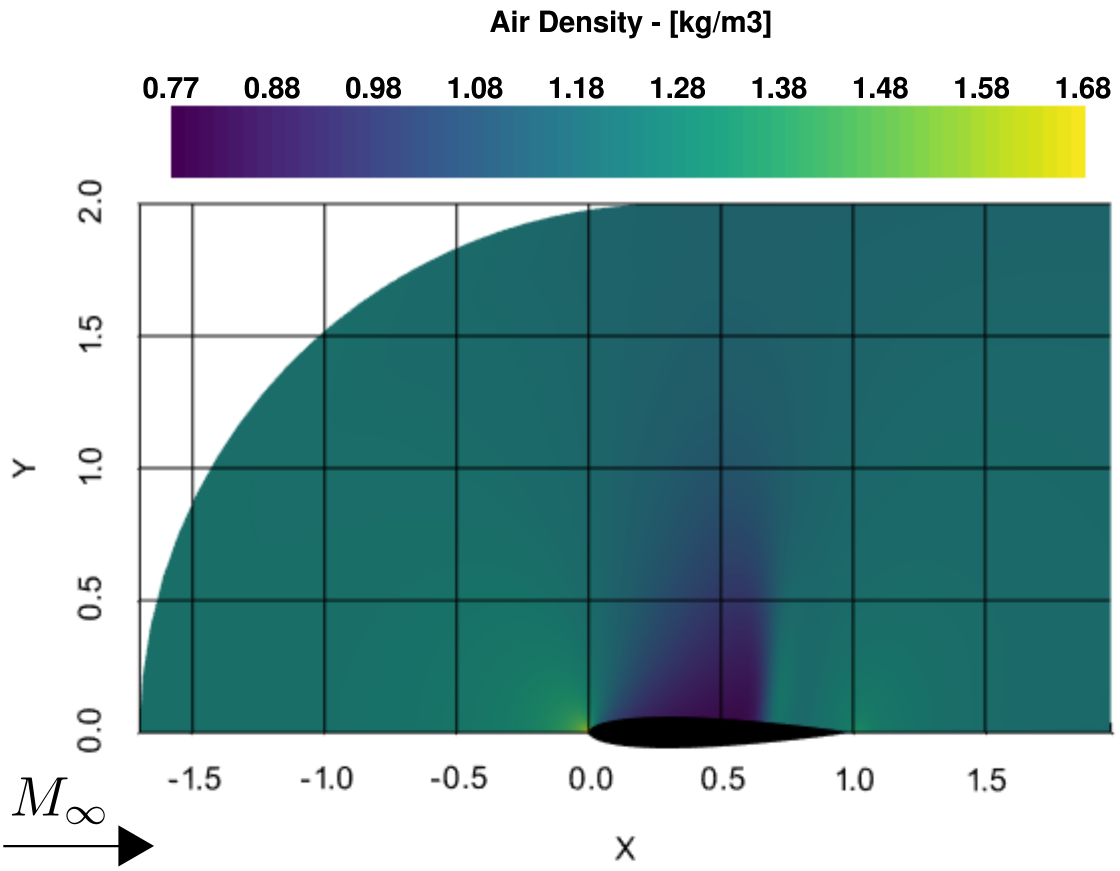


Figure 5.1: Steady state solution of transonic flow over a NACA 0012 aerofoil in terms of air density.

An alternative approach to generate a constrained tetrahedralisation was therefore needed. A survey was carried out of other C++ libraries that could generate a constrained mesh compatible with CGAL data structures, but no suitable algorithms were identified. In the field of computer graphics constrained tetrahedralisations have been reported in the literature [199], with Tetgen [200] being used because it is the only open source software available for this meshing purpose. Therefore, in this research a multi-stage Constrained Delauney Tetrahedralisation (CDT) construction process is developed involving Matlab [201], Tetgen, ParaView [202] and CGAL [99]. Matlab is used to write the input file to be used in Tetgen, containing the list of nodes with air density information, the list of boundary (farfield and solid surface) facets with identifying markers and the specification of a hole where tetrahedra will be removed in order to adhere to the constraint. The resulting CDT, presented in Figure 5.3, is then inspected in ParaView and converted back to the CGAL data structure.

An application that performs data conversion between Tetgen and CGAL has never before been made available, despite it being a question frequently asked in CGAL forums. The script used

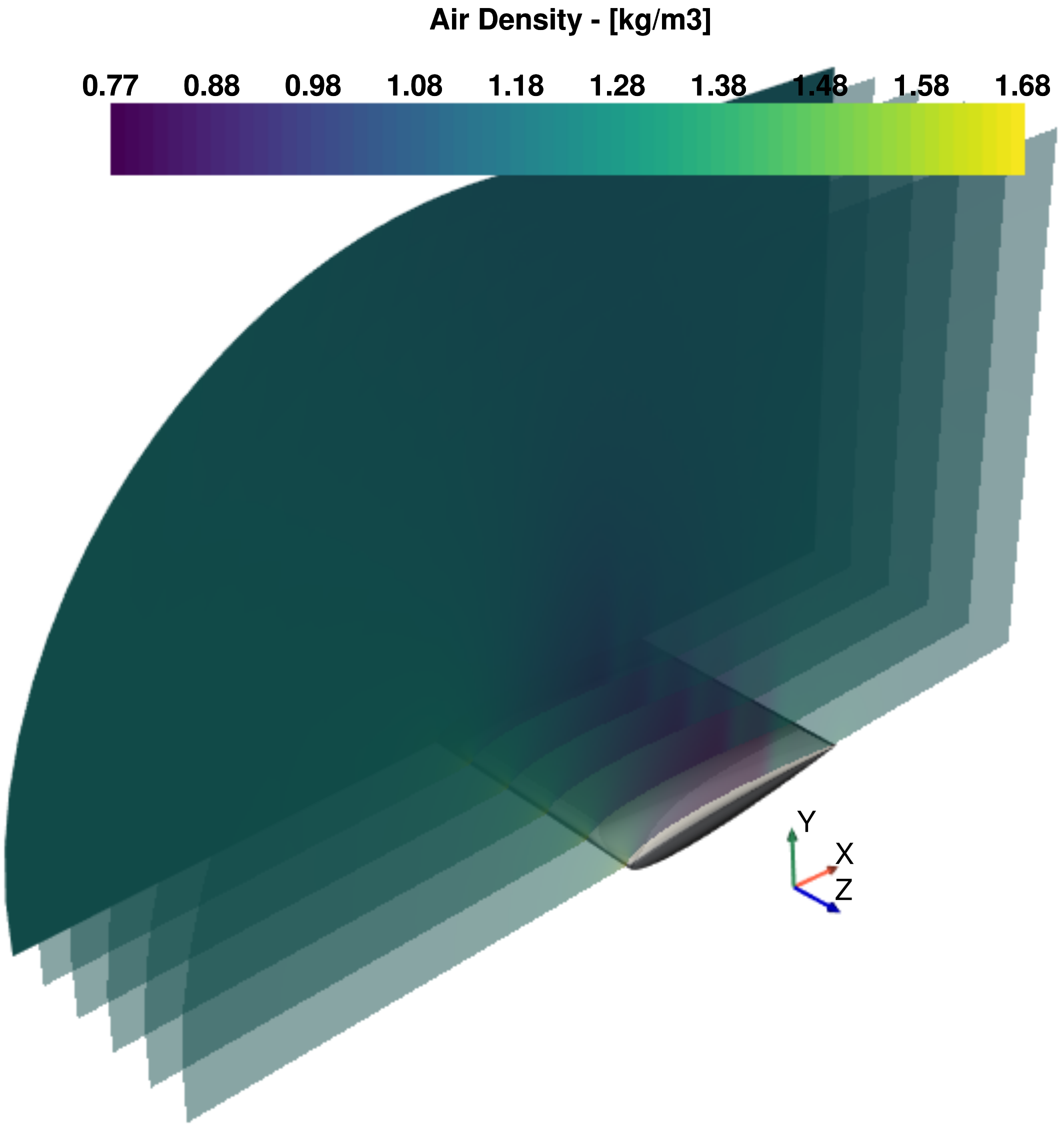


Figure 5.2: Pseudo-3D compressible flow domain constructed from the original 2D steady state solution of transonic flow over a NACA 0012 aerofoil.

in this research is presented in Appendix 1 and is an important contribution to the literature, which can be utilised by other researchers. This conversion avoided the necessity to modify the initial implementation of the light simulation framework, since the gradient estimation and interpolation algorithms were already both implemented based on CGAL data structures for the verification and validation test cases. The final CDT mesh generated using this approach contains 201600 cells.

Special consideration is needed at boundaries during gradient reconstruction to ensure the

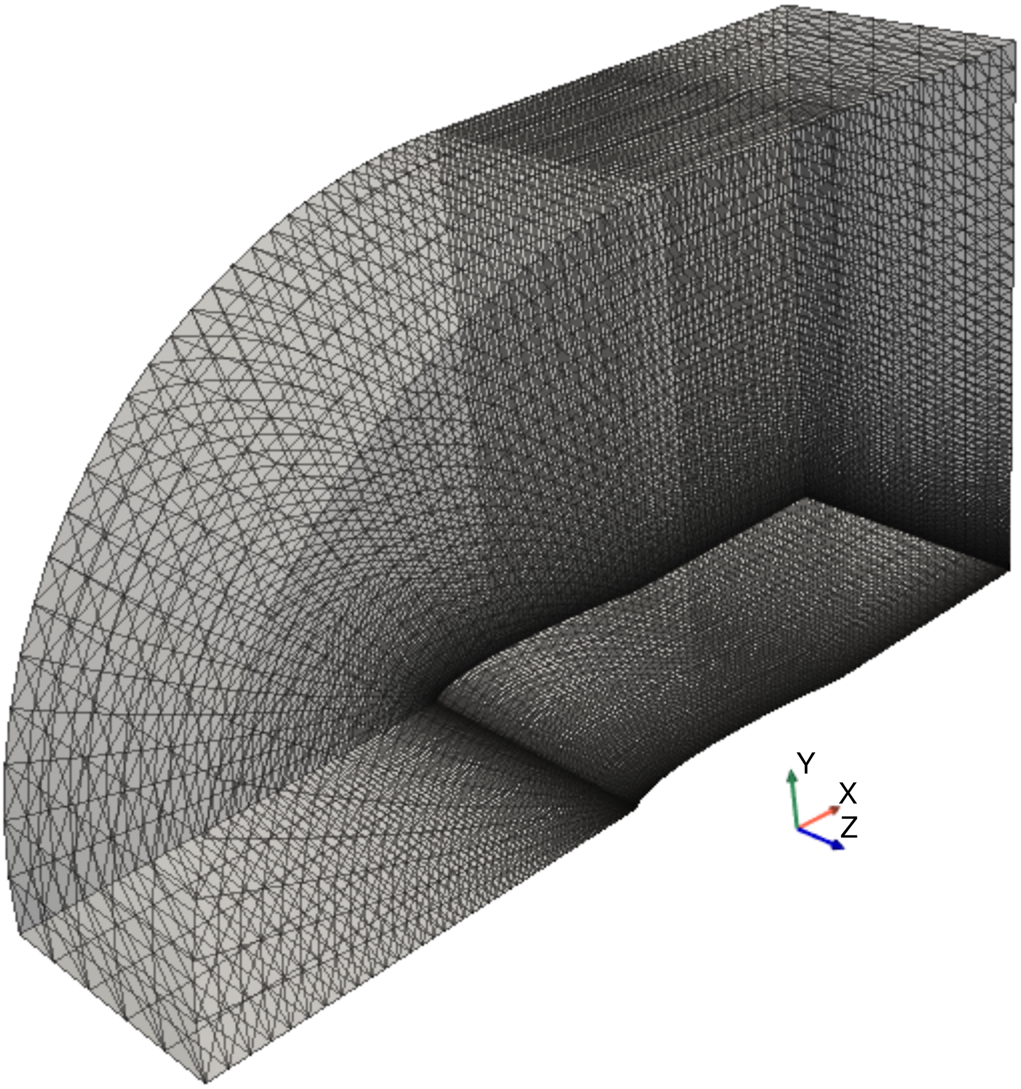


Figure 5.3: Constrained Delaunay Tetrahedralisation of the transonic flow over a NACA 0012 aerofoil constructed using Tetgen [200].

correct physical behaviour of light rays at boundaries such as the solid surface and the farfield. Flags are set to identify if vertices, facets and cells of the mesh pertain either to the interior of the domain or to one of the boundaries. For cells at the farfield, the cell has fewer neighbours and thus the number of interior data points used to estimate the gradient vector in the initial implementation is naturally lower, which can cause gradient estimation errors. This issue is overcome by increasing the number of neighbouring cells used in gradient estimation, i.e. using information from neighbours of neighbours. At a solid surface boundary it is important to identify when the integration step from the fluid domain leads to a point inside the solid surface and to treat gradient estimation differently in the vicinity of the body. This is achieved by using halo cells within the solid surface in which the density value is mirrored. A process similar to that is used in many cell-centred CFD codes to enforce boundary conditions. This enforces a zero density gradient along the normal at the solid surface. This is a reasonable assumption for the present investigations when it is assumed that there is no significant heat transfer in this region.

At this stage, the light source model does not take into account any wavelength dependence. Instead, its total power is represented as a simple white sRGB colour triple, $\phi = [1, 1, 1]$. It is nevertheless approximated as a directional type, but only different orientations in the xy plane are analysed. The wing is considered to be composed of a physically-based microfacet material modelled according to the Schlick approximation with aluminium base reflectivity at normal incidence for the Fresnel term, the Trowbridge-Reitz GGX distribution function and the Smith's method with a combination of Schlick-Beckmann and GGX approximations for the geometric attenuation term. As described in Section 3.3.1, the Fresnel term specifies the amount of incident light that is reflected back to the environment, instead of absorbed by the metallic material; the distribution function determines the statistical alignment of the microscopic facets that compose the overall roughness aspect of the macroscopic surface; and the geometric attenuation term addresses occlusion and shadowing between adjacent microfacets. The photon mapping technique without filtering and with a fixed bandwidth is used to estimate radiance in a recording film. A practical viewing system and the backward ray tracing process from the virtual camera to the wing considering nonlinearities was not implemented for this initial test case. Instead, a cheaper linear ray tracing is used to track light back towards the recording film, in a forward manner even after reflection. Finally, in order to measure the contrast in the computational images, an additional simulation considering linear light propagation in air free of discontinuities is also performed.

5.1.2 Initial Synthetic Shadowgraph

By positioning an equidistributed array of light rays on one side of the flow domain, all oriented in the same global direction normal to the original plane that defines the CFD solution, and the recording film on the opposite side with its plane also normal to the flow, as shown in Figure 5.4, a virtual experiment of shadowgraphy for the transonic NACA 0012 pseudo-3D wing is assembled.

A relatively low initial number of photons (5000) is emitted from the light source, however the effect of the geometrical displacement of light due to interaction with the shock wave is already evident in the simulated shadowgraph image presented on Figure 5.5. In this standard shadowgraphy setup, similar to the one employed with the supersonic axis-aligned cylinder

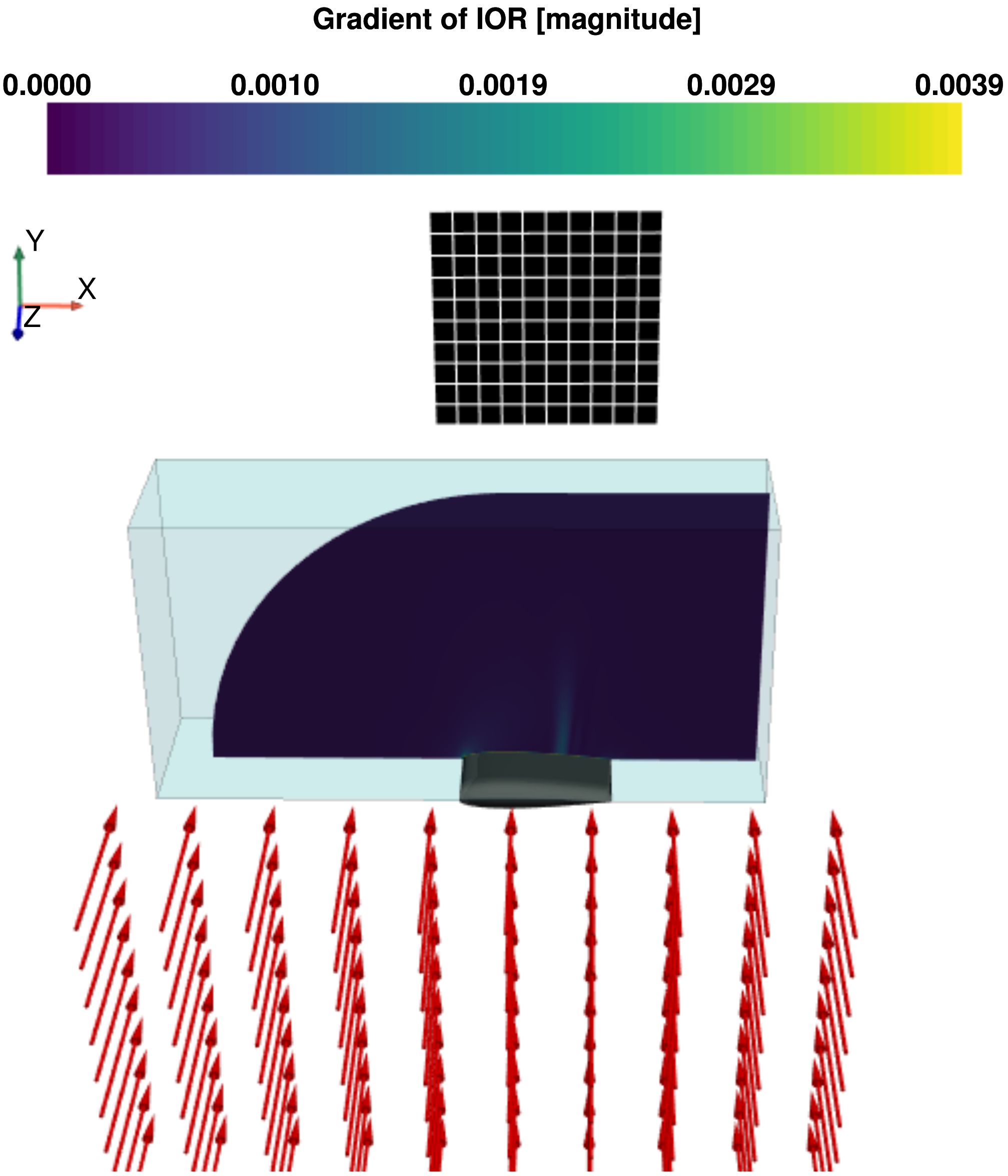


Figure 5.4: Virtual shadowgraphy of a transonic NACA 0012 aerofoil.

(see Section 4.2), the resulting visual pattern generated has the expected characteristics. A bright band or, in this case, a caustic highlight appears due to the concentration of photons at the location where they reach the recording film after being deflected by the shock wave. In consequence, a dark band arises where these deflected light rays were originally directed to. As explained in Section 1.3.1, this influence of the compressible flow over light propagation can be further confirmed by locating the foot of the shock wave in the space between the x-coordinates equal to 0.6 and 0.7, where there is a strong density and refractive index gradient (see Figures 5.1 and 5.4), and relating this position to the line of demarcation of the shadow in the image.

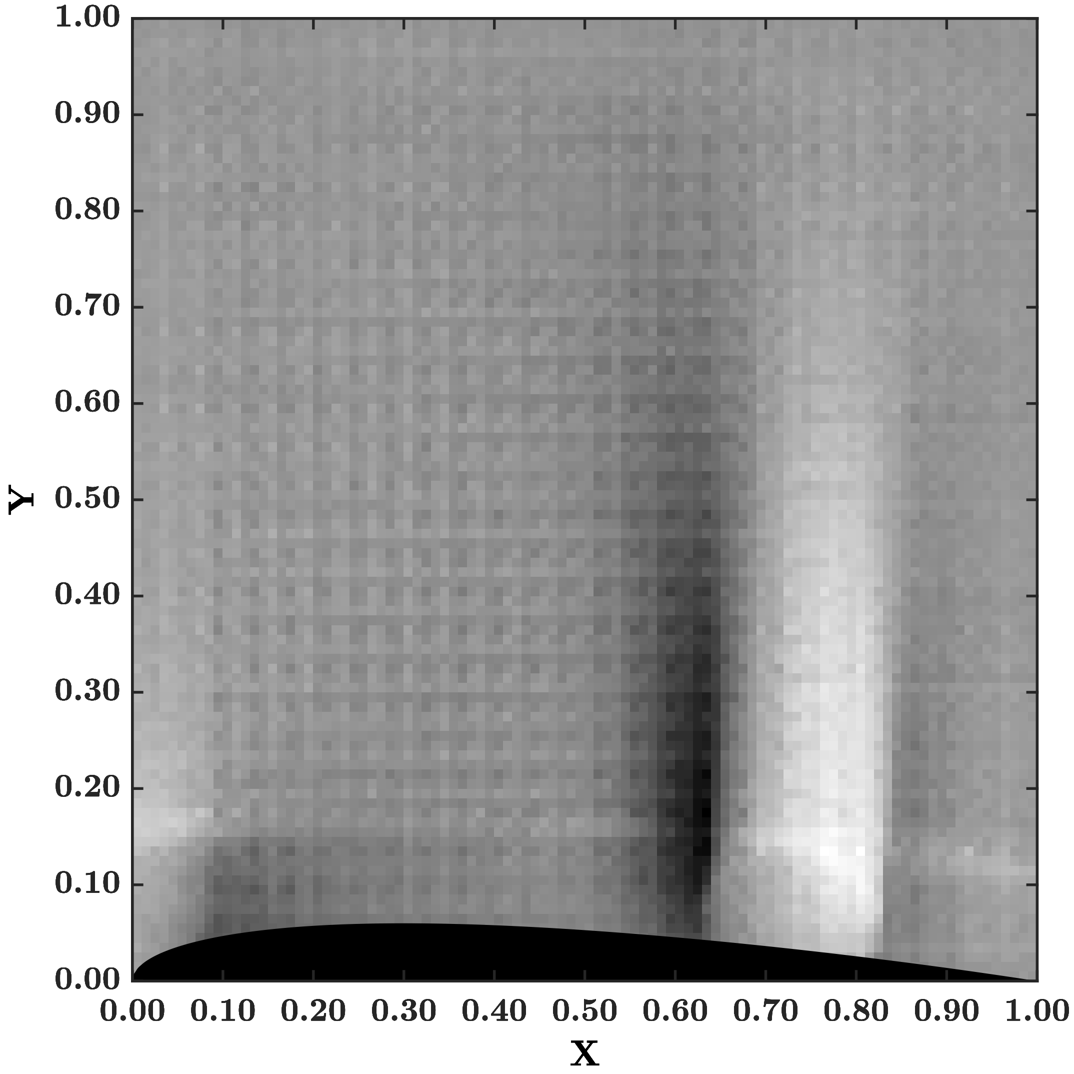


Figure 5.5: Simulated shadowgraph image of a transonic NACA 0012 aerofoil.

However, this initial test clearly indicates that the selection of the rendering parameters has a major effect over the quality of the synthesised image. The presented computational shadowgraph is heavily pixelated, to the extent that most of the picture elements boundaries are easily distinguishable. This could be corrected by increasing the image resolution, but this would increase the rendering time. Moreover, the image quality is also impacted by the small illumination sample size. Since the number of emitted photons traced in the illumination stage is still very low, the radiance statistical density estimation approach is also limited. This results in high variance or noise in the final image, which is seen as the random shades of grey in the pixels. Ideally, image filtering and a specifically tuned bandwidth would be used in this estimation to ensure that enough photons are accounted for and appropriately weighted for each pixel. The fixed radius size used for photon search results in oversmoothed visual artifacts, which are rendered as a pair of dark and bright bands spread over multiple pixel columns instead of a concentrated caustic artifact and adjacent sharp shadow. This smeared out shadowgraph is also related to the CFD mesh and solution, which were not evaluated in terms of refinement and flow resolution. Although the appearance of the synthetic shadowgraph can be improved, it still presents the expected effects of compressible flow over light propagation.

5.1.3 Analysis of the Effect of the Number of Emitted Photons on Synthetic Shockshadows

In order to generate an approximate shockshadow image of the transonic flow over the artificially created wing, the recording film is positioned a certain distance away from its surface and oriented in the perfect specular reflection direction, or the emission direction mirrored with respect to the vertical (y-coordinate direction). This way the reflected light rays are linearly traced from the locations of intersection and, thus, the geometrical displacement of photons due to refraction are accentuated. Then, in order to test the influence of the number of emitted photons on the final image, light rays evenly distributed above the wing along the x-coordinate direction are emitted. Then, the number of rays, or photons, is sequentially increased, as represented in Figure 5.6 from left to right. In this case the direction of perfect specular reflection is simply normal to the wing chord and hence the recording film just points downward.

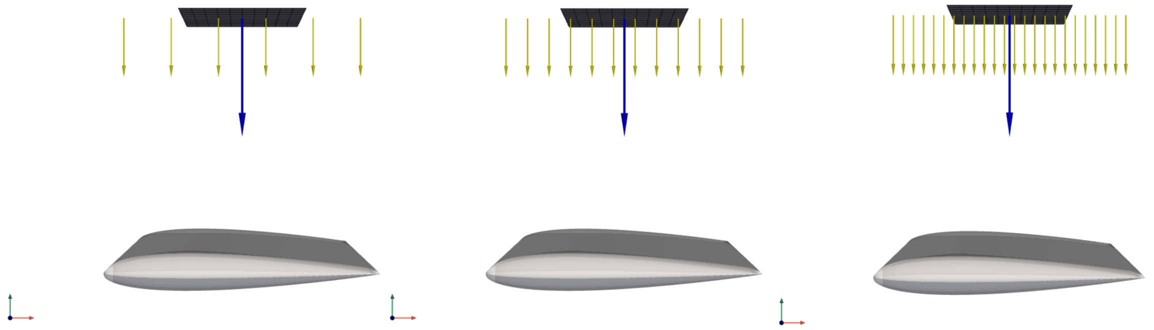


Figure 5.6: Representative schematic of the number of emitted photons analysis. The number of photons is increased from left to right.

Figure 5.7 shows the contrast at the image plane for different numbers of photons traced from the source aligned with the y-coordinate direction, i.e. vertically oriented towards the wing. The contrast is calculated by the difference between the radiance values computed in two parallel simulations, one in an homogeneous optical medium (no flow nor shock wave) and the other concerning the aerodynamic test case. On the first four (top) plots with the lowest number of photons, the progressive convergence in terms of density estimation and radiance computation is verified. It can be seen on the four (bottom) plots with the most photons that the emission of more than 10^3 photons provides only slight improvements in the colour estimation. Higher sample sizes only smooth out discontinuities, including the caustic itself which is not always desirable, and reduce numerical noise in the solution. Thus it can be seen that although a very large number of photons constituting the light source is desirable from the point of view of accurate illumination reconstruction, which would allow the use of the largest possible number of radiant energy samples and the smallest possible sampling area, rendering or radiance estimation via photon mapping is still an averaging statistical process. This means that the radiance solution inevitably converges, and the use of too large a number of photons may result in high computational costs that do not increase accuracy.

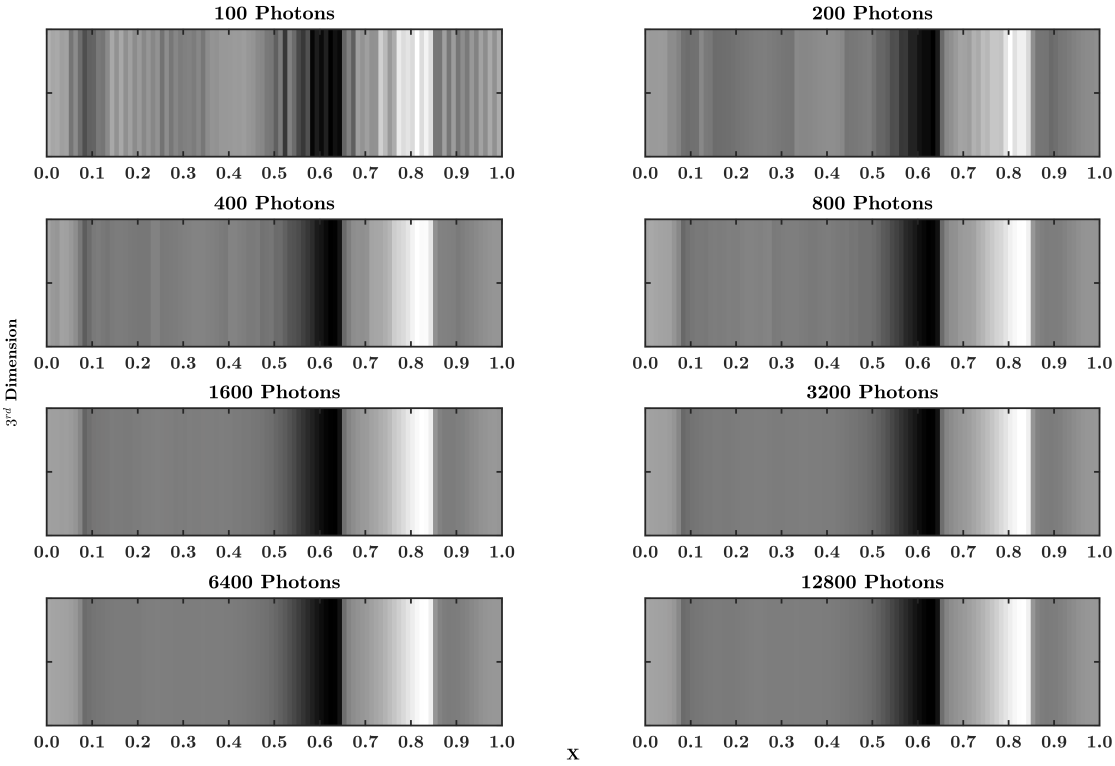


Figure 5.7: Contrast convergence of the transonic NACA 0012 aerofoil shockshadow image due to increased number of emitted photons.

5.1.4 Analysis of the Effect of the Light Source Direction on Synthetic Shockshadows

The characteristics of the shockshadow pattern are also expected to depend on the direction of emission of the photons from the light source. To evaluate such dependence, light rays are cast at angles of $\pm 20°$, $\pm 15°$, $\pm 10°$, $\pm 5°$ and $0°$ with respect to the y-coordinate direction or vertical, as represented in Figure 5.8.

The resulting contrasts at the image plane, i.e. the differences between radiance values

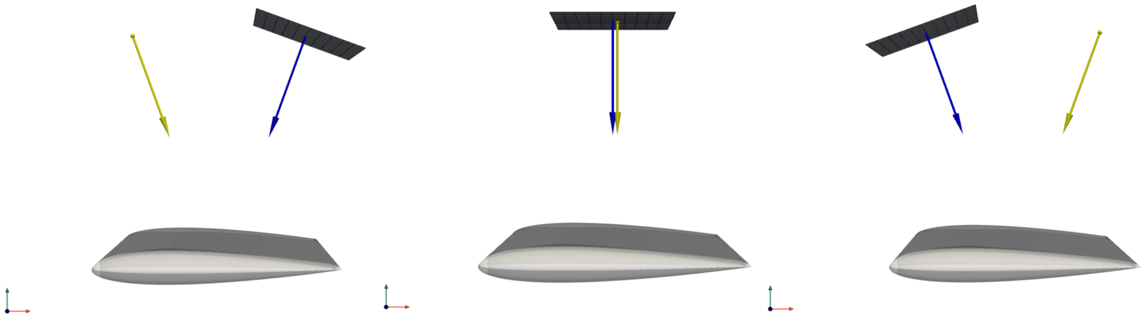


Figure 5.8: Representative schematic of the light source direction analysis. The angle of emission increases as the light source direction is rotated clockwise from left to right.

computed in an inhomogeneous and a homogeneous optical domain, are presented in Figure 5.9. Considering all the curves from $-20°$ to $+20°$, it is possible to see that light rays that reach the shock surface with a more perpendicular orientation (at the lowest negative angles) produce shallow valleys and short peaks in the contrast curve, which translates to weak dark and bright spots in the synthetic shock wave shadow image. As the light rays direction roughly aligns with the shock wave curvature (at zero and lowest positive angles), a grazing incidence occurs and the highest values of light deflection induce stronger displacement of photons and respective shockshadow patterns, or deepest valleys and tallest peaks in the curves. The depth of valleys and height of peaks then decreases as the angle increases to the highest positive angles.

5.1.5 Synthetic Shockshadow

Given the photon density convergence and the light source direction dependence studies previously presented, the resultant shockshadow image formed on the recording film assuming an emission of 1000 photons at the angle of $0°$ is shown on Figure 5.10. In this case, the image plane is simply mapped as a texture over the wing surface in order to emulate a real shockshadow. The CFD solution is also shown so that a comparison can be made between the location of the shock wave in the flow and the shockshadow in the image. This figure shows the relationship between the surface of discontinuity in the compressible flow and the edge of the shadow of the shockshadow pattern. However, it can be seen that the shock discontinuity is not sharply resolved by the CFD and consequently neither are the optical features in the synthetic shadowgraph.

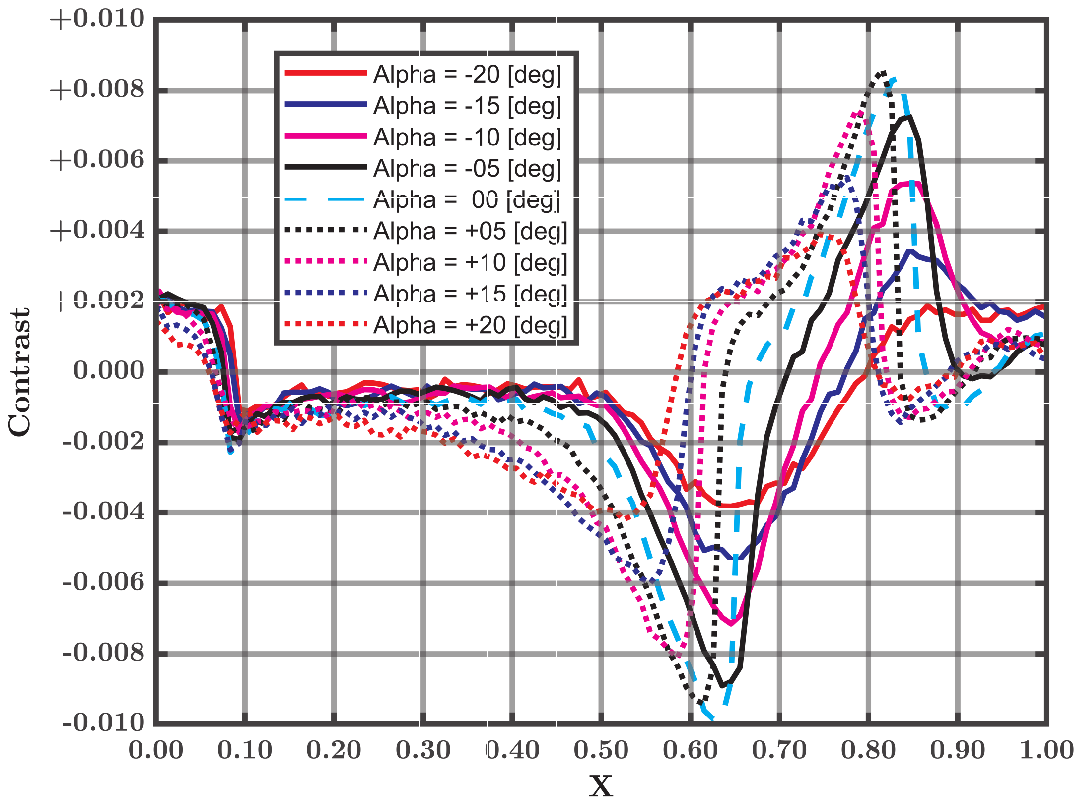


Figure 5.9: Influence of light source direction over the transonic NACA 0012 aerofoil shock-shadow image.

5.2 3D Shockshadow Simulation

Finally, shockshadow generation is studied for a well-known three-dimensional case of transonic flow over the Onera M6-wing. This provides a realistic wing and a genuinely 3D flow in which to study the developed light simulation framework. Light from the sun is sampled and propagated through the inhomogeneous compressible flow field. The reconstruction of its continuous signal in a synthetic image is then evaluated in terms of the characteristics of photon mapping and fine adjustments available to obtain realistic features. The environmental conditions that lead to shockshadow formation are subsequently analysed. This concerns the orientation of the sun with respect to the wing necessary to generate observable features. Then, a computational photograph of a shockshadow is taken from a perspective that emulates an airborne observer, in order to confirm the secondary refraction effects and, finally, to generate a virtual image of a fully three-dimensional shockshadow.

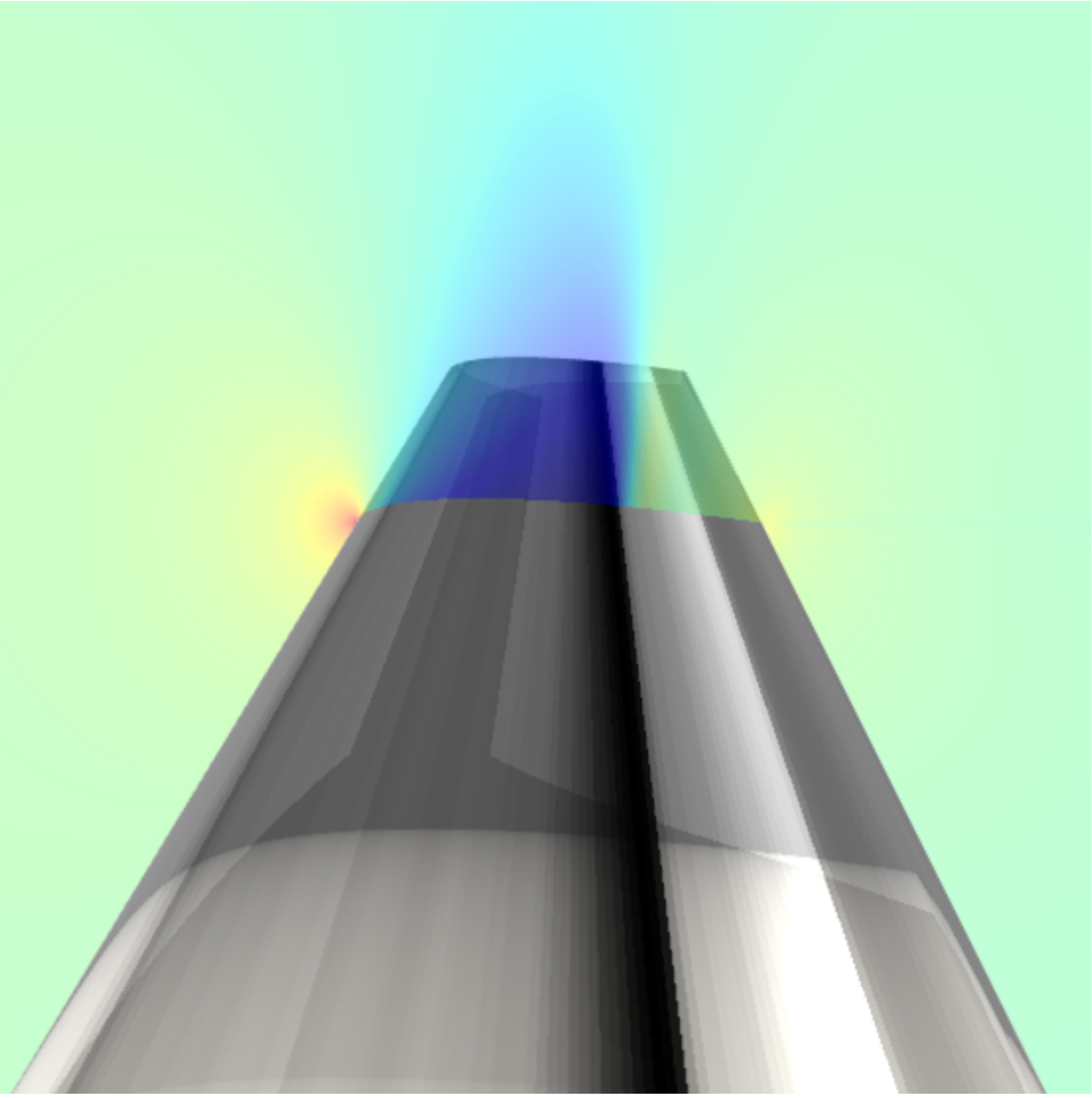
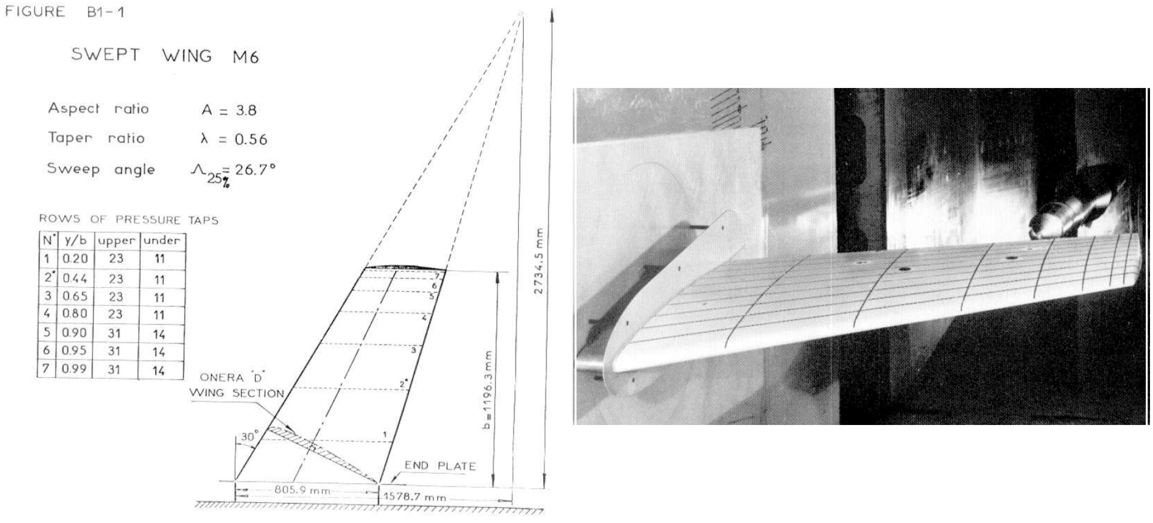


Figure 5.10: Synthetic shockshadow pattern over the NACA 0012 pseudo-3D wing.

5.2.1 Transonic Flow Over the Onera M6-Wing

The geometry, shown in Figure 5.11(a), is a tapered, untwisted, swept back three-dimensional semi-span wing devised by the ONERA Aerodynamics Department in the 1970s. It was created to provide a suitable experimental basis for studying high Reynolds number transonic aerodynamics and compressible fluid flow phenomena [203]. The model was thoroughly investigated in the ONERA S2MA wind tunnel at a variety of wind speeds and angle-of-attack configurations. Figure 5.11(b) highlights one of these wind tunnel experiments [203]. These investigations led to the generation of meaningful data that was fundamental to the validation of numerous CFD codes developed over the years [203, 204]. Specifically, the test case considered here has a lambda patterned shock wave and has been used to validate Euler solvers, since viscous effects are small and experimental results can be reasonably well-predicted by an inviscid flow model. This test case provides a realistic wing with a suitably complex shockwave structure and considerable computational savings are made since it can be run as an Euler simulation rather than a RANS simulation. The CFD data is generated using the SU2 open-source software suite [78].



N°	y/b	upper	under
1	0.20	23	11
2*	0.44	23	11
3	0.65	23	11
4	0.80	23	11
5	0.90	31	14
6	0.95	31	14
7	0.99	31	14

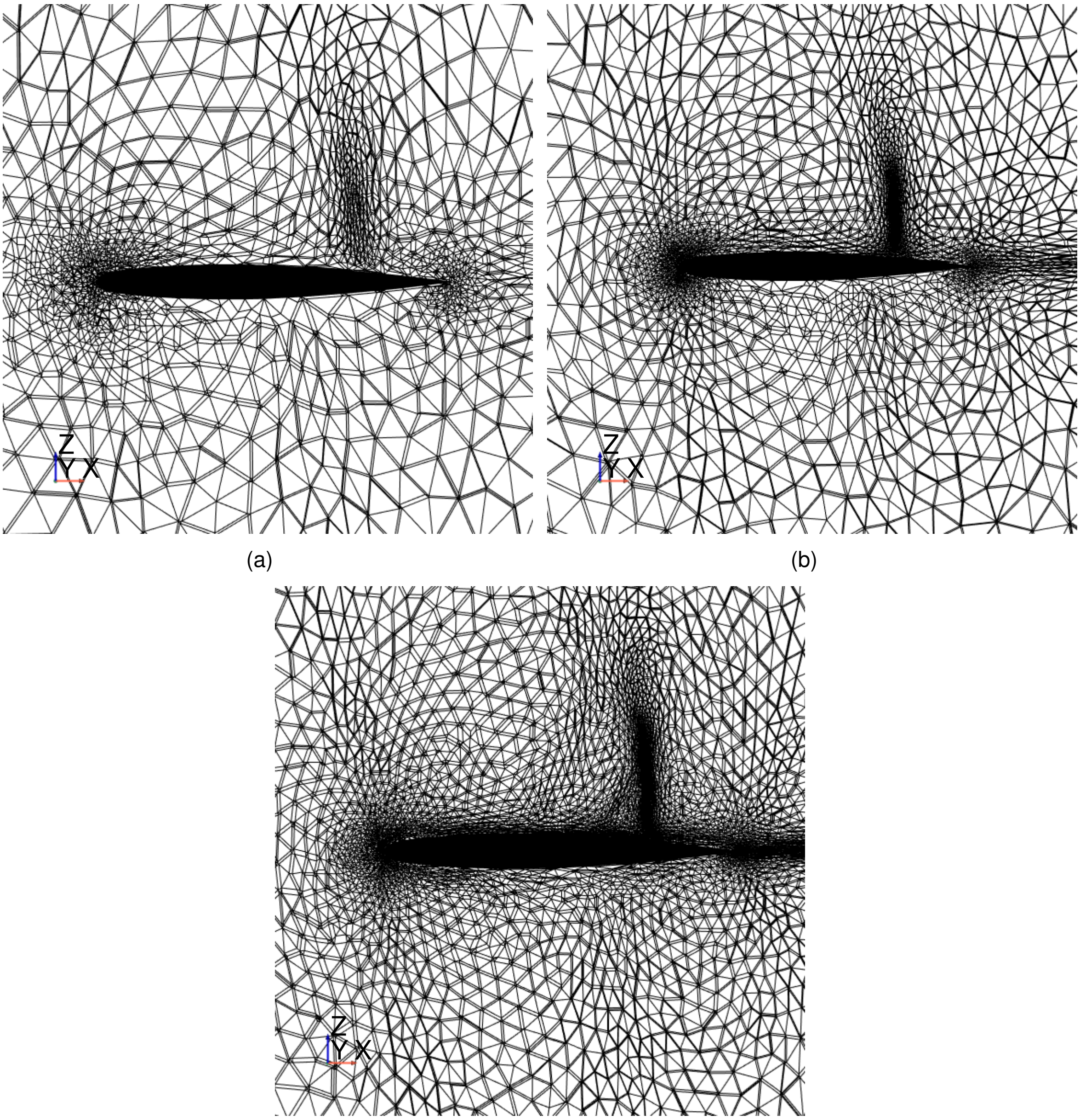
(a)

(b)

Figure 5.11: (a) Details of the Onera M6-wing model geometry and (b) respective wind tunnel experiments performed at ONERA S2MA. The images are reproduced from [203].

The wing has a curved surface and a thin trailing edge, and an unstructured mesh is used to discretise the flow domain. However, unless a high level of refinement is employed, it will not appropriately capture the characteristic lambda patterned shock wave that forms on the wing surface. The lambda shape arises due to the merging of two nearly normal shock waves from the root towards the tip of the wing. A global mesh refinement is computationally inefficient in terms of a systematic reduction of discretisation errors due to the anisotropic nature of the flow feature. Instead, the AMG library [80] is used together with SU2 to perform a goal-oriented anisotropic mesh adaption process. The transonic compressible flow over the Onera M6-wing has been verified and validated in other studies using this same mesh adaptation methodology [81, 205]. Flow results converged quicker with the adapted mesh when compared against an unmodified mesh, achieving the same values obtained with a well refined discretisation and approximately the same shock and pressure critical locations on the wing surface [81, 205].

The flow simulation is performed with three different levels of mesh density in order to analyse the influence of the spatial discretisation on the shock wave resolution, and also the effect it has on the computationally generated shockshadow images. In terms of number of vertices and cells, the coarse level contains 6×10^4 and 3.7×10^5, the intermediate level has 1.4×10^5 and 8.3×10^5, and the refined level has 4.4×10^5 and 2.6×10^6, respectively. Figure 5.12 shows these three levels of grid adaptation, where the refinement around the shock wave region is evident.



(a)

(b)

(c)

Figure 5.12: (a) Coarse, (b) intermediate and (c) refined meshes used in the CFD simulation of transonic flow around the Onera M6-wing.

The experimental test case in [203] simulated in this study has the following parameters

- Freestream Pressure = $101325.0\ N/m^2$
- Freestream Temperature = $288.15\ K$
- Freestream Mach number = 0.8395
- Angle of attack (AOA) = $3.06°$

The above values are used in the farfield boundary conditions and to initialise the solution in the flow domain. At the wing surface a slip wall boundary condition is enforced and at the root where the wing is mounted in the experiment, a symmetry wall boundary condition is used [83, 206]. The published experimental data for this test case includes the pressure coefficient distribution at seven reference spanwise stations measured via pressure taps [203], as shown in Figure 5.13.

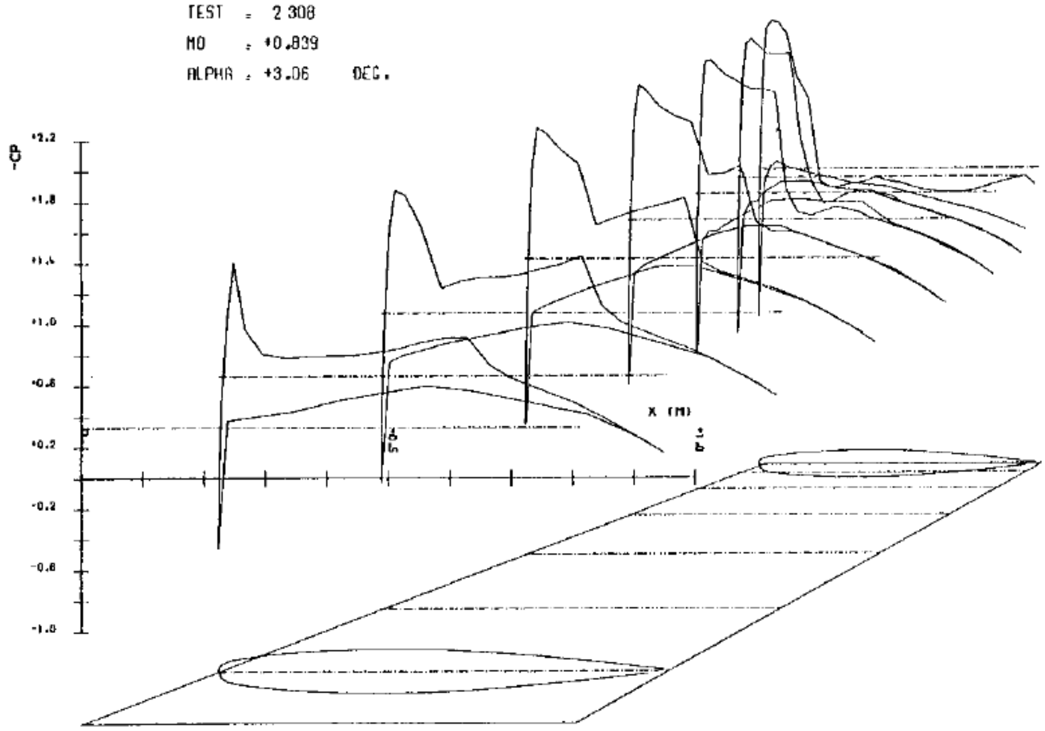


Figure 5.13: Experimental pressure coefficient distributions over the Onera M6-wing surface, measured at the seven reference spanwise stations. The image is reproduced from [203].

To check the accuracy of the solutions generated using SU2, including the appropriate shock wave capture and resolution, the calculated pressure distributions are compared to this experimental data. The distributions for the three different meshes are compared against the experimental data in Figures 5.14 and 5.15. These numerical curves show reasonable similarity with the experimental data in terms of pressure values and gradient along the different chords, with modelling discrepancies due to insufficient overall mesh density evident in the coarsest results. Discrepancies are also due to numerical interpolation errors resulting from the extraction of solution slices to construct the curves from the unstructured mesh.

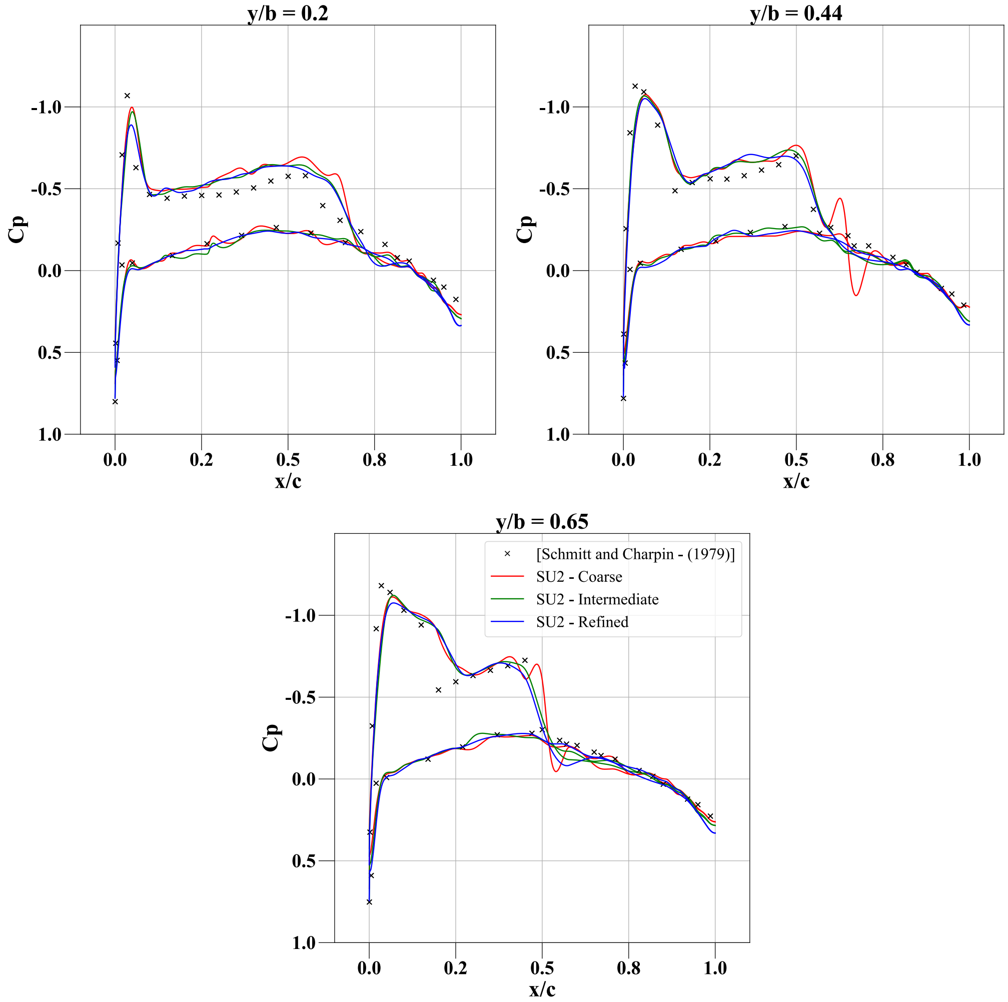


Figure 5.14: Numerical pressure coefficient distribution over the root region of the wing surface, simulated in SU2 with different levels of mesh refinement. The experimental data used for comparison is obtained from [203].

The refined mesh has a smooth pressure distribution and the lambda patterned shock wave is seen to be reasonably resolved for light simulation purposes, i.e. in terms of gradient discontinuity, hence this solution is employed for all of the subsequent analysis. Figure 5.16 shows the lambda patterned shock wave visible over the wing surface through the air density distribution and also indicates the strategic positioning of the reference spanwise stations.

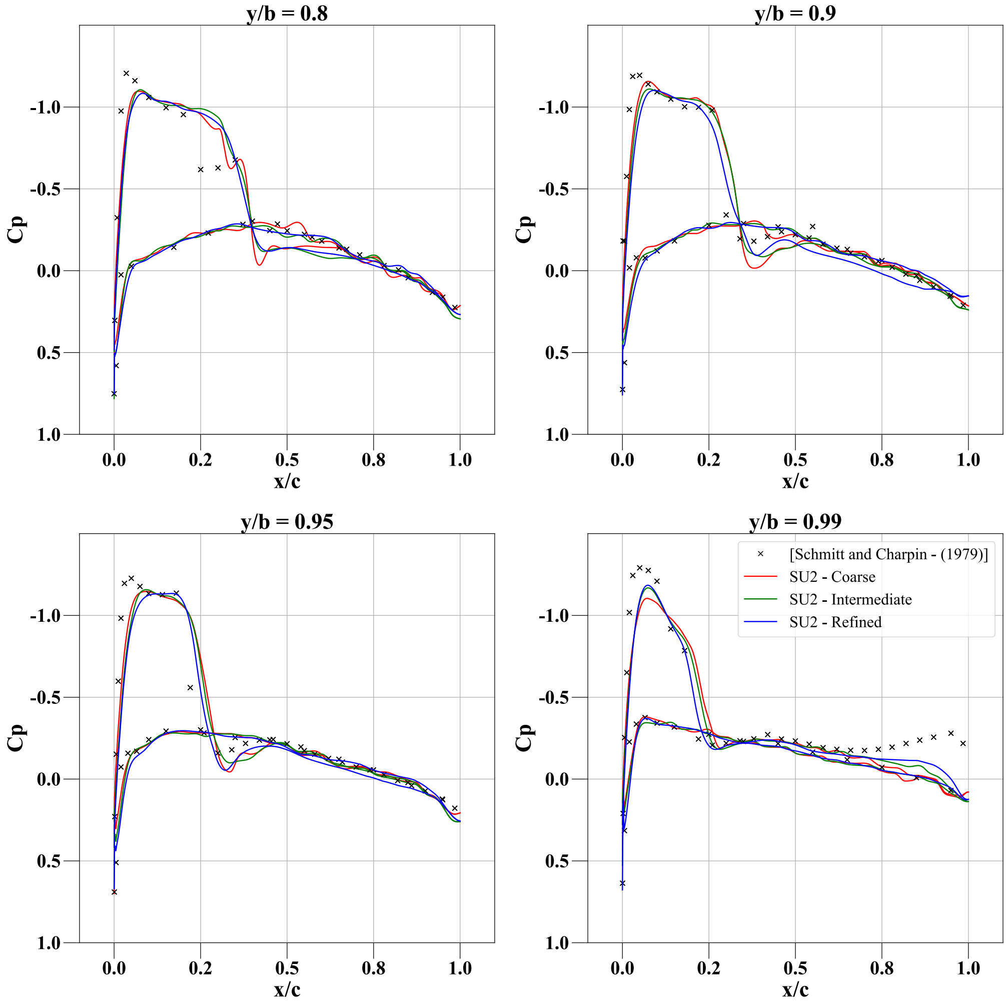


Figure 5.15: Numerical pressure coefficient distribution over the tip region of the wing surface, simulated in SU2 with different levels of mesh refinement. The experimental data used for comparison is obtained from [203].

Figures 5.17 and 5.18, in turn, present the steady state density solution in the outer spanwise stations, where it is possible to see the overall merging of the two near normal shock waves into one stronger one.

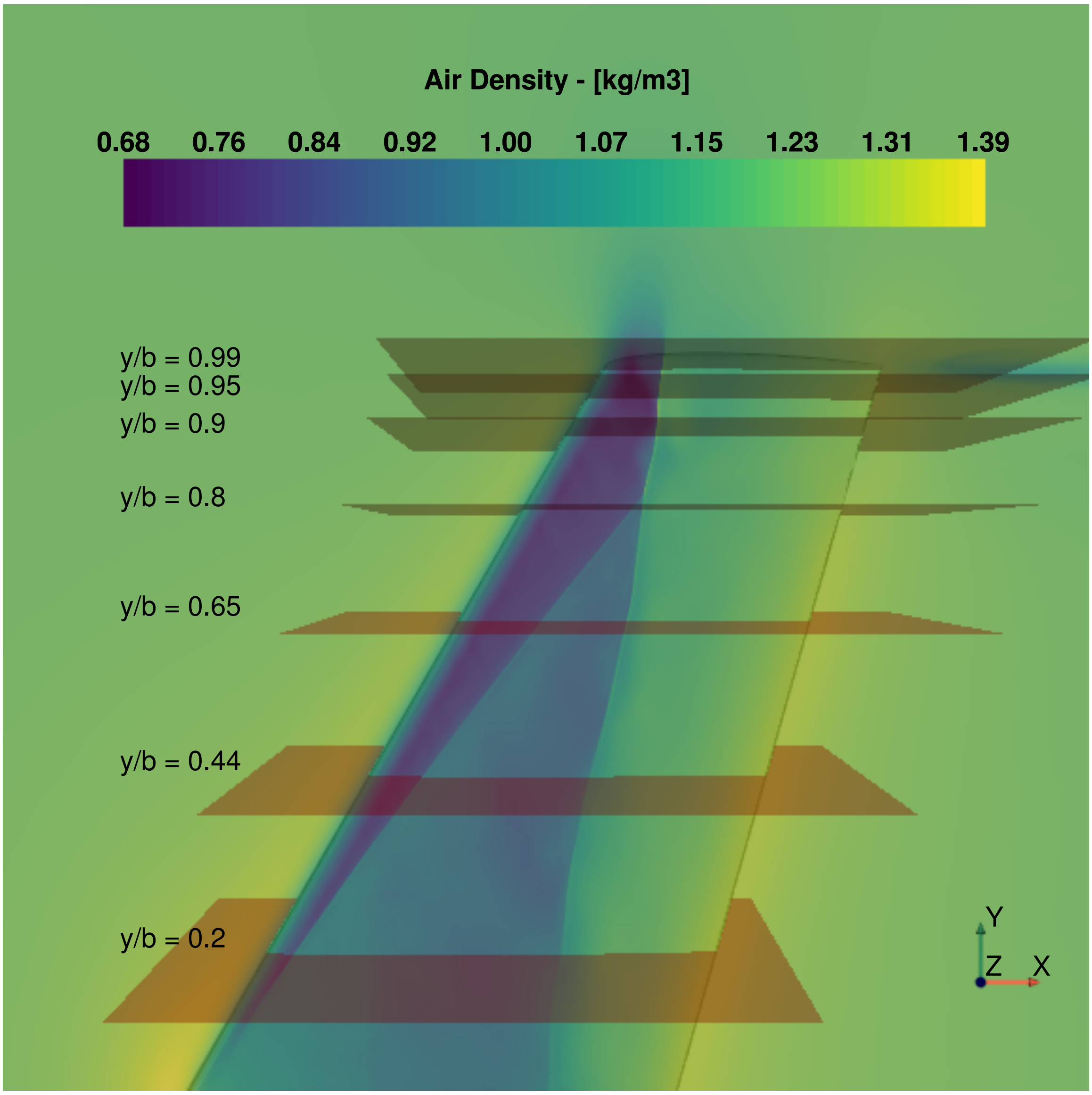


Figure 5.16: Lambda patterned shock wave and classic reference spanwise stations of the transonic Onera M6-wing.

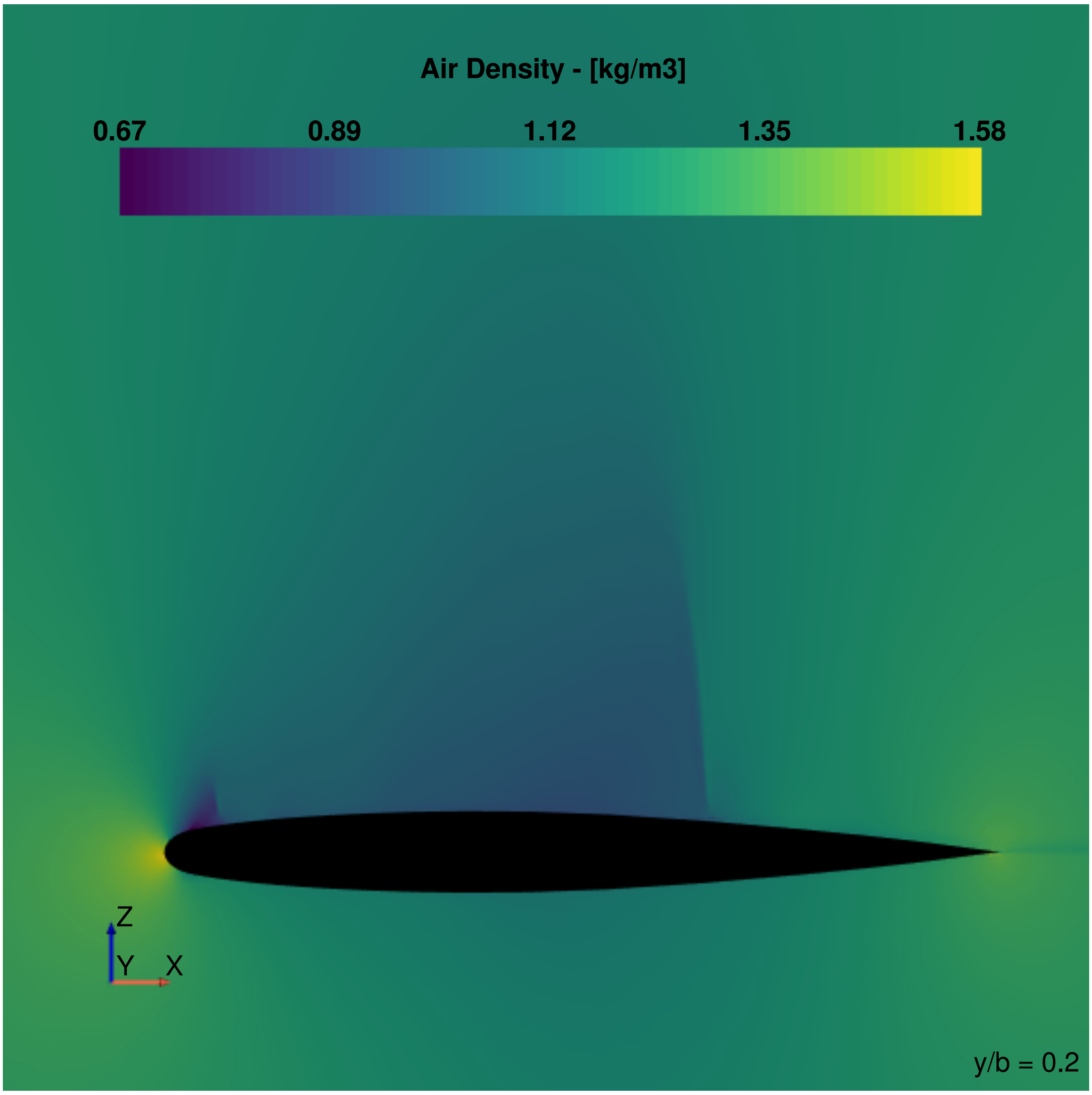


Figure 5.17: Steady state solution of transonic flow over the Onera M6-wing in terms of air density at the first (root) reference spanwise station.

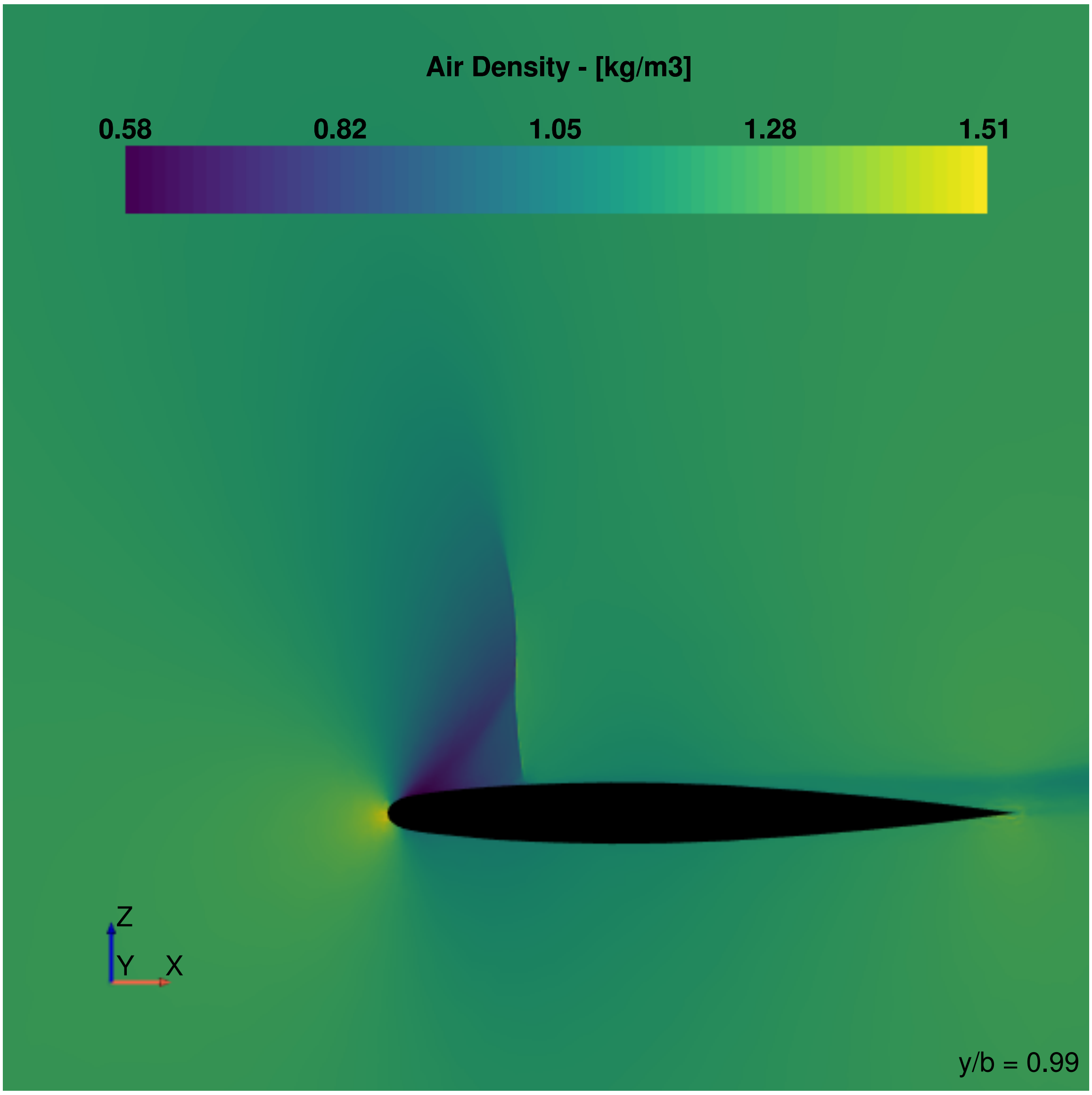


Figure 5.18: Steady state solution of transonic flow over the Onera M6-wing in terms of air density at the seventh (tip) reference spanwise station.

5.2.2 Analysis of Light Sampling and Image Reconstruction

In the methodology developed in this research, sampling occurs in two separate stages: firstly during the illumination or photon tracing stage, when infinitesimal light beams are traced forward from the Sun and the resulting photon and respective radiant flux distribution over the wing is stored in the photon map; and subsequently during the rendering or visibility stage, when light rays are traced backward from the visual system and employed to calculate the colour of any point along the wing surface or of each pixel contained in the image plane. The connection between the two sampling stages, or the reconstruction of the continuous image signal, is made via a statistical density estimation technique, which inherently suffers from random and systematic errors.

Therefore, one of the first questions that arise from the use of such method is: what is the minimum number of samples, both in the illumination and visibility stages, required to synthesise a digital image as similar as possible to the naked eyes visualisation and/or recording of the in-flight shock wave's shadow? The sample size is directly related to the computational cost of the imaging simulation, as the higher the number of samples the more light rays have to be traced across the scene.

To answer this question a simplification of the physical phenomenon is assumed. The wing is considered to be composed of a perfectly diffuse or Lambertian material, therefore scattering light equally in all directions. The visibility stage is performed in homogeneous surrounding air, i.e. a flow free of discontinuities such as shock waves. Under these conditions, the shock wave shadow and caustic effect can be rendered straight from the photon map, without requiring further backward ray tracing and camera positioning. Imagining a hypothetical situation, it is as if the picture of the wing covered with a flat white paint is taken when the aeroplane is already on the ground, the wing somehow storing a flight illumination history that records the dark and bright fringe formation. The camera is positioned somewhere far away above the wing, so far that the whole surface fits inside the view frustum, as depicted in Figure 5.19.

The effect that the sample size of each stage, i.e. illumination and visibility, has on the final image is considered separately. Initially, the number of "pixels" or, more correctly, query points distributed on the wing surface is fixed to 100 in the chordwise direction and only the seven spanwise stations used in the traditional Onera M6-Wing experimental study [203] are analysed. The sun is positioned at an elevation angle of $\Phi = 90^\circ$ and azimuth angle of $\Theta = 30^\circ$. The sun illumination conditions used here are known to generate a visible caustic from the studies of Fisher et al. [16]. Throughout this entire section light rays emitted and traced from the sun are only directed based on this positioning. The same conclusions hereby drawn apply for other sun orientations. In the radiance estimation, a uniform kernel where all the photons near to the

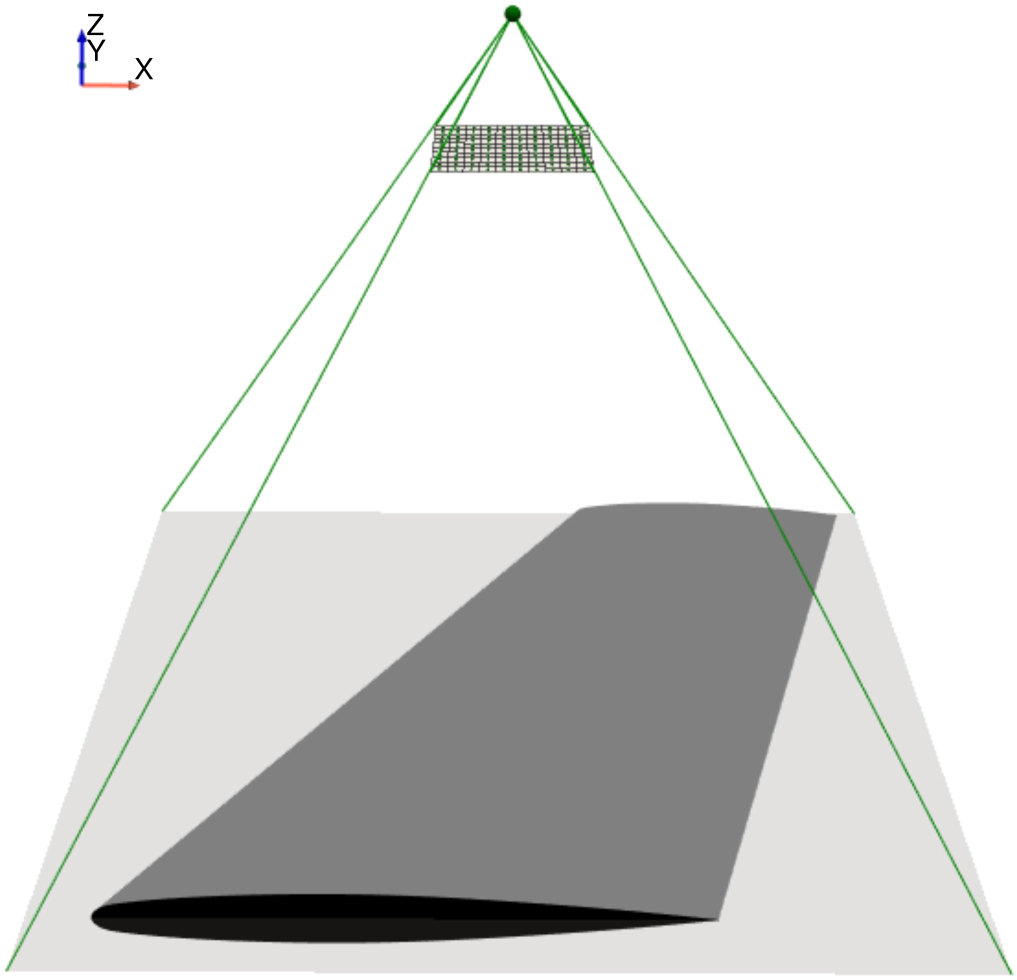


Figure 5.19: Layout of the camera positioning emulating the direct rendering of caustics over the perfectly diffuse wing surface immersed in a homogeneous optical domain.

query point and within the chosen bandwidth are weighted equally is used. The bandwidth, in turn, is varied between two sample-dependant extrema that provide a visual representation of the overall rendering behaviour.

Figure 5.20 presents the unfiltered luminance in one of the spanwise stations, $y/b = 0.9$, rendered under the above mentioned conditions. At this station there is a strong shock wave visible as a high pressure gradient in Figure 5.15, which causes a considerable deflection of the light rays passing through this region and evidence for a shockshadow is expected in the luminance distributions.

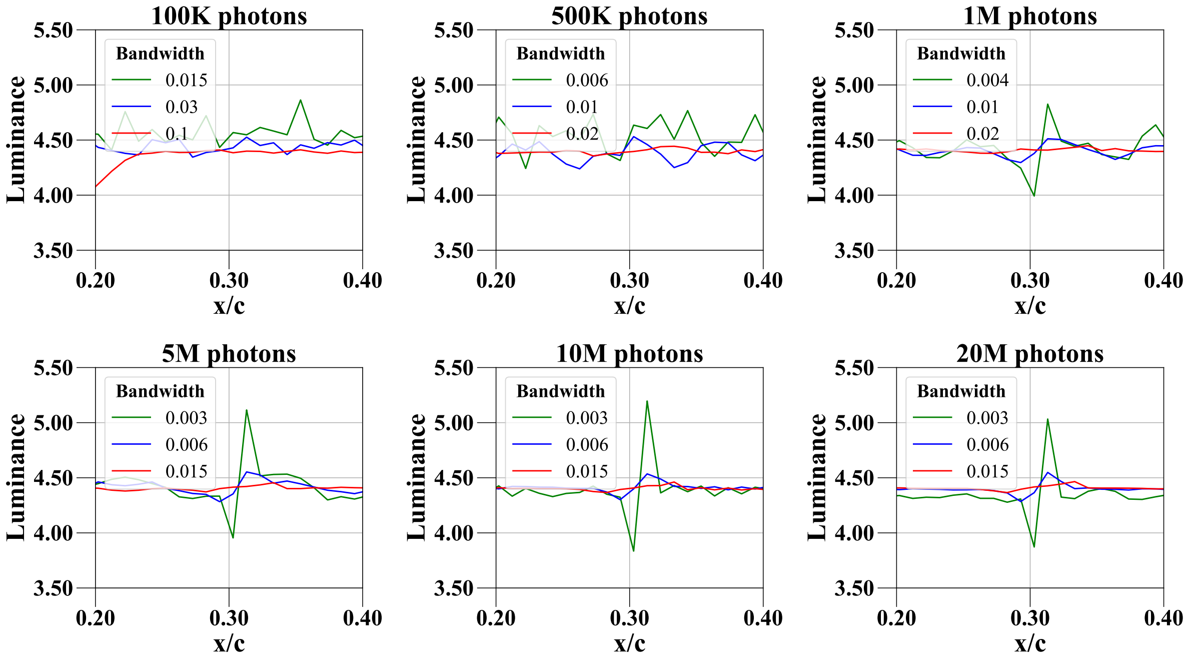


Figure 5.20: Luminance distribution at $y/b = 0.9$ rendered without filtering.

For sample sizes smaller than one million photons the light source is clearly undersampled and the number of near neighbours used in the rendered luminance is not enough, even with comparatively very large bandwidths. The luminance is either undersmoothed with a smaller window width or oversmoothed with a larger window width, and a compromise between the two cannot be achieved. The result is that the caustic is either merged with the inherent noise produced in the distribution by the ray-tracing-based technique or it is completely dampened. As the number of photons increases to the million range, the sharp edges of the shockshadow become distinguishable from the surrounding noise even for radiant energy densities estimated with small window widths. This feature is visible as a characteristic luminance valley adjacent to a luminance peak at the observation point located approximately at a third of the chord.

For the cases where more than a million photons are traced, the effects that the radius size of the sphere used to perform the near neighbours search has on the reconstructed caustic is even more evident. Larger bandwidths produce a high local degree of smoothing and consequently cause an underfit of the photon map. Noise is reduced but the resulting luminance distribution is invariably biased. The rendering process oversmooths most of the features expected to be present in the image, either completely suppressing or comparatively blurring the caustic. Conversely, as the bandwidth decreases so does the systematic error in the solution due to overfitting of the samples. However, this introduces random error in the rendered distribution. Even though the image now contains the shock wave shadow, it also exhibits a considerable amount of noise. As previously explained in Section 3.4.3.1, this is one of the most fundamental complexities in density estimation, in which the choice of bandwidth governs the trade-off between bias and variance [171].

In order to deal with these fitting and smoothing problems, an image filter or nonuniform kernel has to be used in the photon density estimation. Considering the same wing station, sun orientation, number of query points and sample sizes, Figure 5.21 then shows the luminance distributions now filtered by the optimised Gaussian kernel introduced by Pavicic [173].

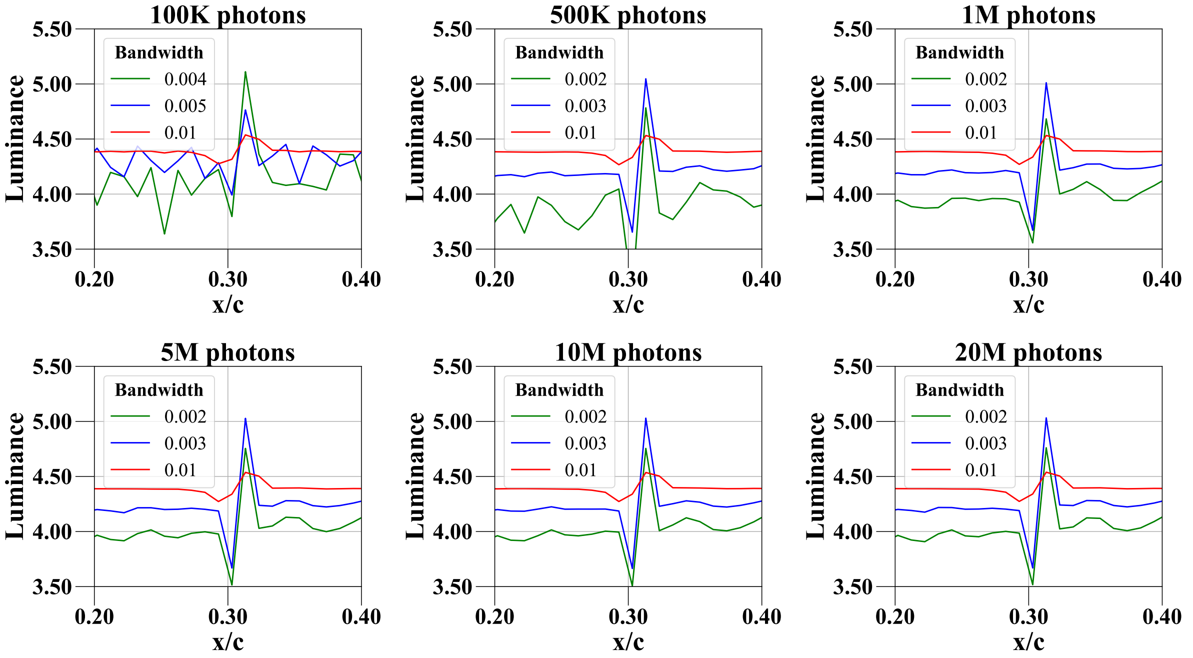


Figure 5.21: Luminance distribution at $y/b = 0.9$ rendered using an optimised Gaussian kernel.

With filtering, the bandwidth becomes less dependant on the sample size. Thus radii of equivalent orders of magnitude are selected for all of the different photon maps, generating very similar results. In fact, the distributions rendered with the larger radius (shown in red) cannot be distinguished between the sample sizes, which would indicate that only a small number of photons need to be traced from the light source and used in the density estimation. However, the shockshadow feature is considerably diminished with this choice of bandwidth. Ideally, a smaller radius would have to be employed to render a more pronounced effect in comparison to the edges of the caustic feature, which would then allow its clear identification.

With only $100K$ photons and the smaller bandwidth, for example, the caustic can be brought to magnitudes comparable to those obtained with more extensive sample sizes. However, as explained before, this proposed bandwidth adjustment comes with the cost of surrounding noise, making it necessary to actually trace more illumination light rays. In the samples with $1M$ photons and more the same convergence of luminance distribution seems to be achieved for the complete range of bandwidths analysed. In those cases, a small radius reduction does not greatly alter the caustic fringes and, similarly, slight increases in the radius also do not further smooth out the solution. The whole luminance distribution is only equally shifted to lower or higher values depending on the selected radius. Only when a somewhat larger bandwidth is used the shockshadow is compressed to an excessive extent.

Even though it appears that a sample size of $1M$ photons has reached a converged solution, this cannot be assumed to be sufficient at all points on the wing. A further analysis of the leading and trailing edge regions and of the remaining six spanwise stations exhibit different characteristics than that of this focused region of a single spanwise station. The more comprehensive luminance distributions, shown in Figures 5.22 and 5.23, are again obtained using 100 chordwise query points per spanwise station, the same optimised Gaussian kernel and one of the smaller bandwidths of $h = 0.003$.

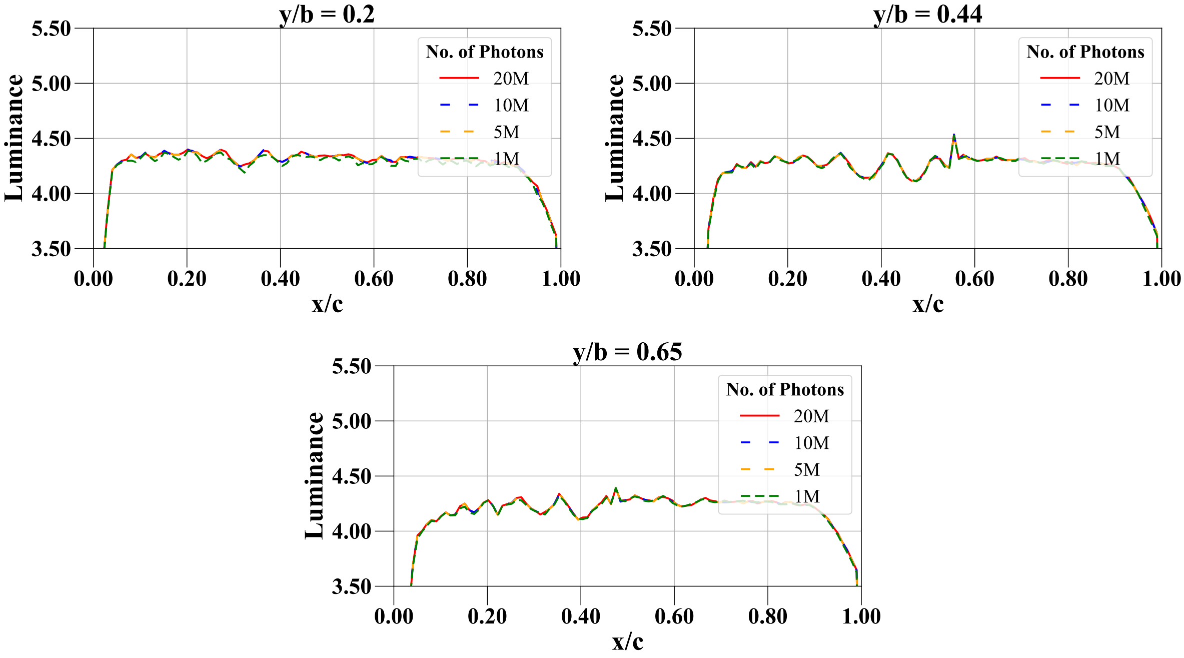


Figure 5.22: Luminance distribution over the root region of the wing surface rendered using an optimised Gaussian kernel and a fixed small bandwidth.

From these graphs, it is evident that although the solution that uses $1M$ photons has converged in terms of caustic shape in the focused region at $y/b = 0.9$, it still contains comparatively strong noise in the areas of supposedly uniform illumination across the wing. In effect, regardless of the sample size used, these oscillations of low frequency are visible in all of the spanwise

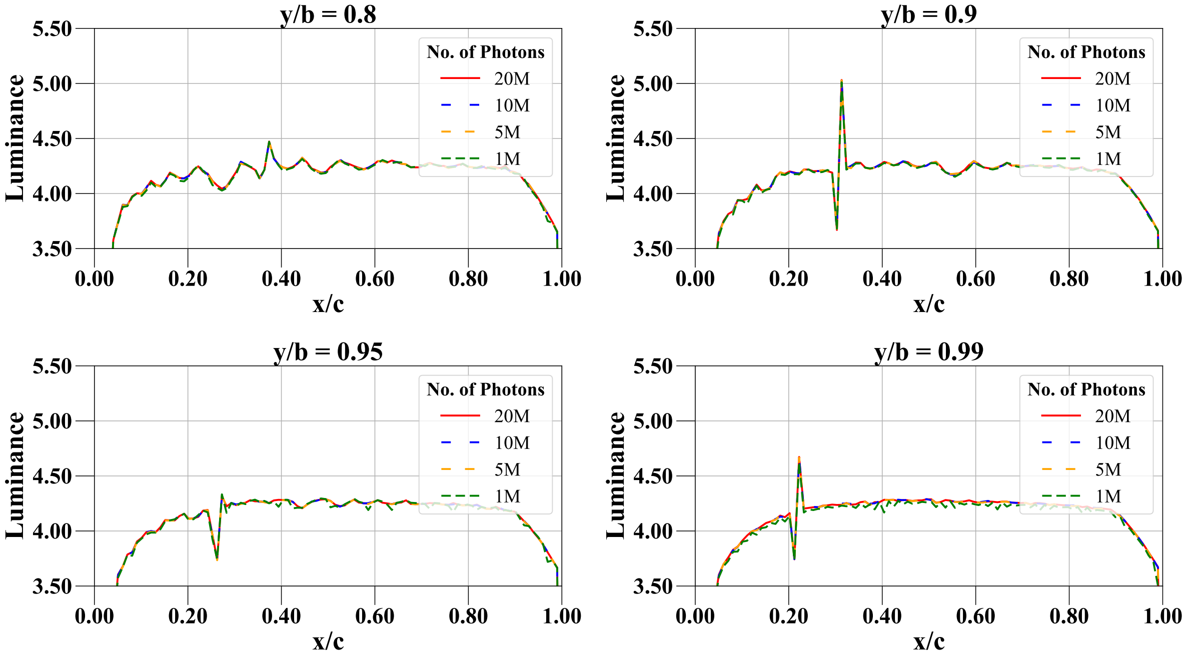


Figure 5.23: Luminance distribution over the tip region of the wing surface rendered using an optimised Gaussian kernel and a fixed small bandwidth.

stations. Although the sharp caustic feature is appropriately resolved near the tip of the wing, on the stations near the root the artificial noise has amplitudes similar to that of the shockshadow, and it is not possible to distinguish it from the noise apart from the differences in frequency.

In order to increase the level of smoothing, a larger bandwidth of $h = 0.01$ is used, resulting in the luminance distributions presented in Figures 5.24 and 5.25. The second trade-off in density estimation is now clear through these graphs. As the undesirable noise is removed to reveal the previously undersmoothed illumination details closer to the root, the shockshadow towards the tip becomes significantly oversmoothed. This is another difficulty in using density estimation to render caustics from photon maps formed by the compressible flow field scattering effects over light. Because the lambda patterned shock wave has different strengths across the wing span and even across the chord where the lambda forks, the bandwidth requirements vary along the photon map. In regions where the shock is strong a smaller window width is required to capture the consequent strong dark and bright fringes, but a larger window width is required to unmask the same light pattern where the shock is weaker. If these opposite and simultaneous requirements are not met, at the tip of the wing the shockshadow becomes blurred and at the root it cannot be visualised at all.

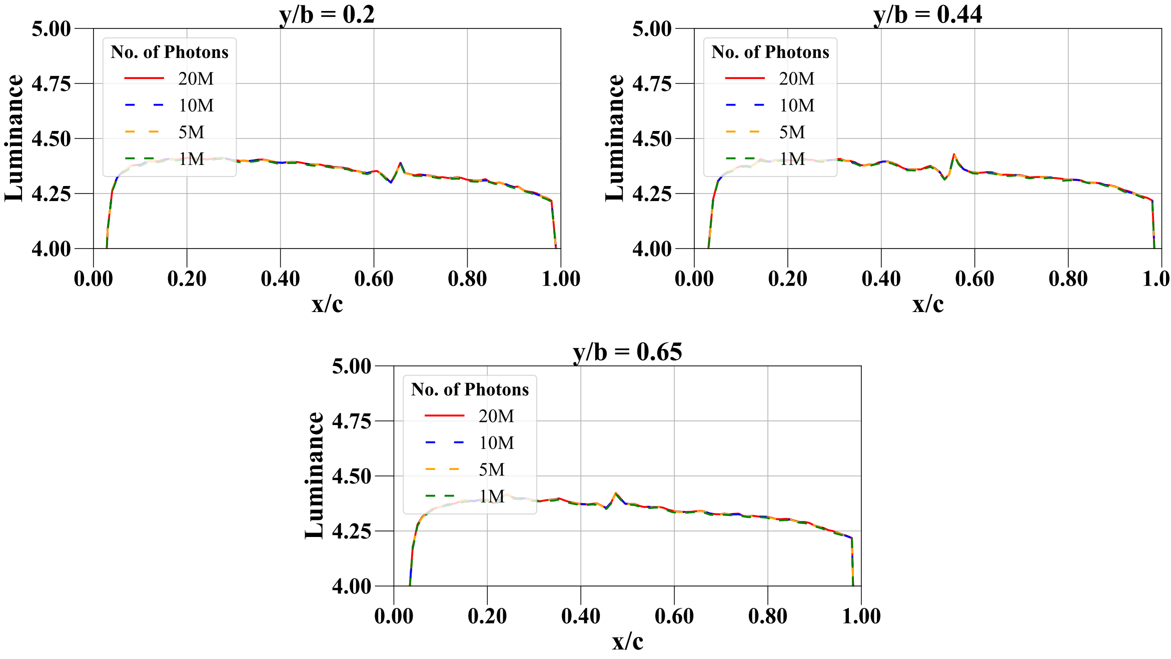


Figure 5.24: Luminance distribution over the root region of the wing surface rendered using an optimised Gaussian kernel and a fixed large bandwidth.

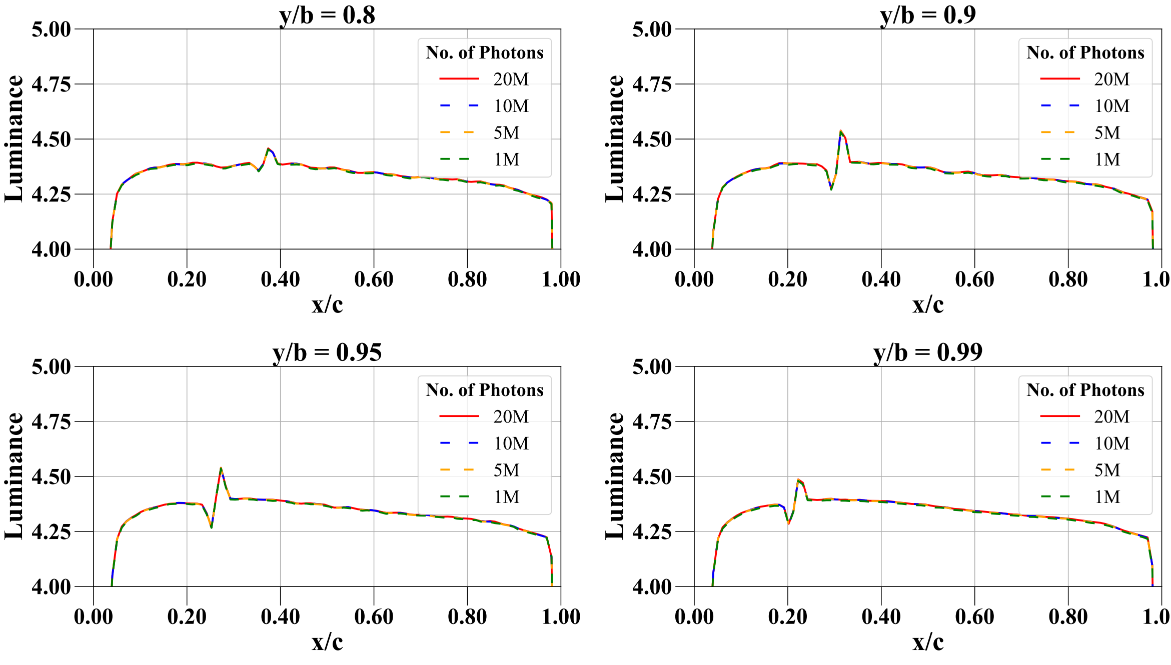


Figure 5.25: Luminance distribution over the tip region of the wing surface rendered using an optimised Gaussian kernel and a fixed large bandwidth.

Although a compromise can be accepted in terms of shockshadow pattern across the wing span, even by enabling further smoothing of the luminance distribution via radius increase some of the noise still persists in parts of the solution. Therefore, the noise source has also to be analysed. Figures 5.26 and 5.27 thus demonstrate the luminance still rendered using a sample of $1M$ photons, the optimised Gaussian kernel and large bandwidth of $h = 0.01$, but with three distinct settings: (i) a fictitious free-air solution, where density gradient is zero everywhere and consequently light refraction does not occur; (ii) the CFD solution derived from the grid with the coarser level of refinement and; (iii) 1000 chordwise query points per spanwise station. These settings are analysed separately and all of the remaining parameters are held the same as before in order to determine the possible source of noise.

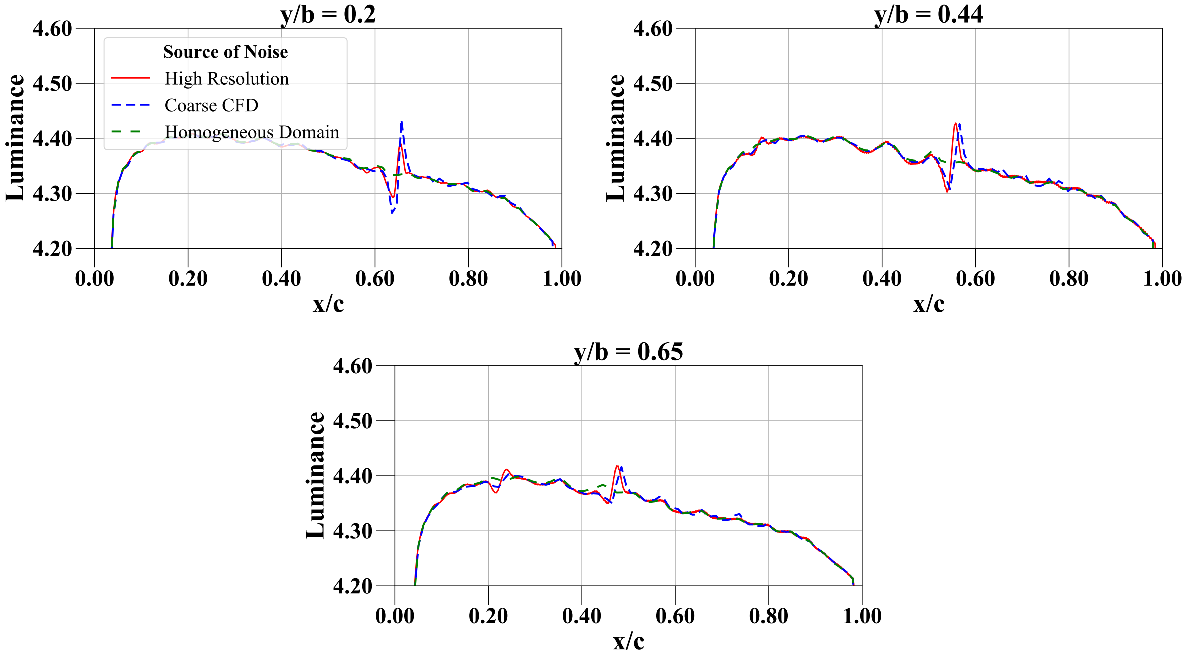


Figure 5.26: Analysis of the source of noise in the luminance distribution over the root region of the wing surface rendered using an optimised Gaussian kernel and a fixed large bandwidth.

The resolution in terms of number of query points used in the radiance estimation only influences the appearance of the luminance distribution, but not the distribution itself. With more "pixels", the details in the illumination are better resolved and the resulting valleys and peaks of the existing caustic features, for example, are smoother in comparison with a lower resolution distribution. In a picture this behaviour is seen as the transition from a jagged or pixelated aspect to a more clear visualisation of features. The noise, however, persists and similarly to the shockshadow it is only smoother but not lower with this modification.

By changing the discretisation employed in the CFD solution process, one would expect the

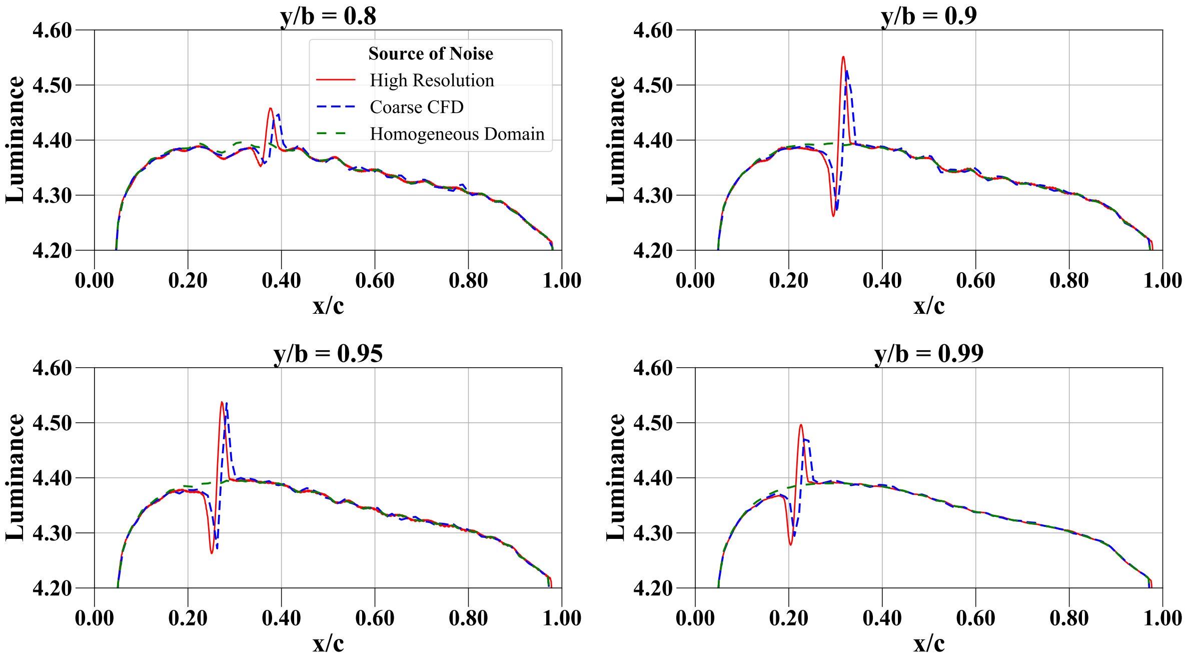


Figure 5.27: Analysis of the source of noise in the luminance distribution over the tip region of the wing surface rendered using an optimised Gaussian kernel and a fixed large bandwidth.

shapes of the illumination oscillations to vary considerably, possibly increasing the noise and/or not capturing the shockshadow when transitioning to coarser grids. Although the caustics are slightly diminished and pushed back downstream with this change of optical domain, again the noise levels are pretty much the same as when visualised with a higher resolution or using another CFD solution.

In turn, in the homogeneous optical domain light behaves linearly and travels along straight paths, which should produce a completely smooth variation of illumination across the chordwise stations, this variation only happening due to the geometry of the aerofoil that composes the wing and orientation of the sun with respect to the quarter-chord line. However, some low frequency noise is still visible in the solutions obtained without considering the aerodynamic flow at all, especially at the wing root.

Another possibility for the persistence of noise in the luminance solution is the choice of kernel to represent individual samples distributions and subsequently evaluate the photon density at query points. Therefore, Figures 5.28 and 5.29 depict the luminance evaluated from the biggest sample available of $20M$ photons, in order to remove the influence of the sample size on noise appearance. The renderer employs not only the Gaussian kernel, but also the Epanechnikov, Cone and Cone+Cosine (see Section 3.4.3.1) for comparison. The bandwidth is slightly reduced to $h = 0.008$, which provides better results when nonuniform kernels are employed, and the number of chordwise query points is returned to 100 per spanwise station.

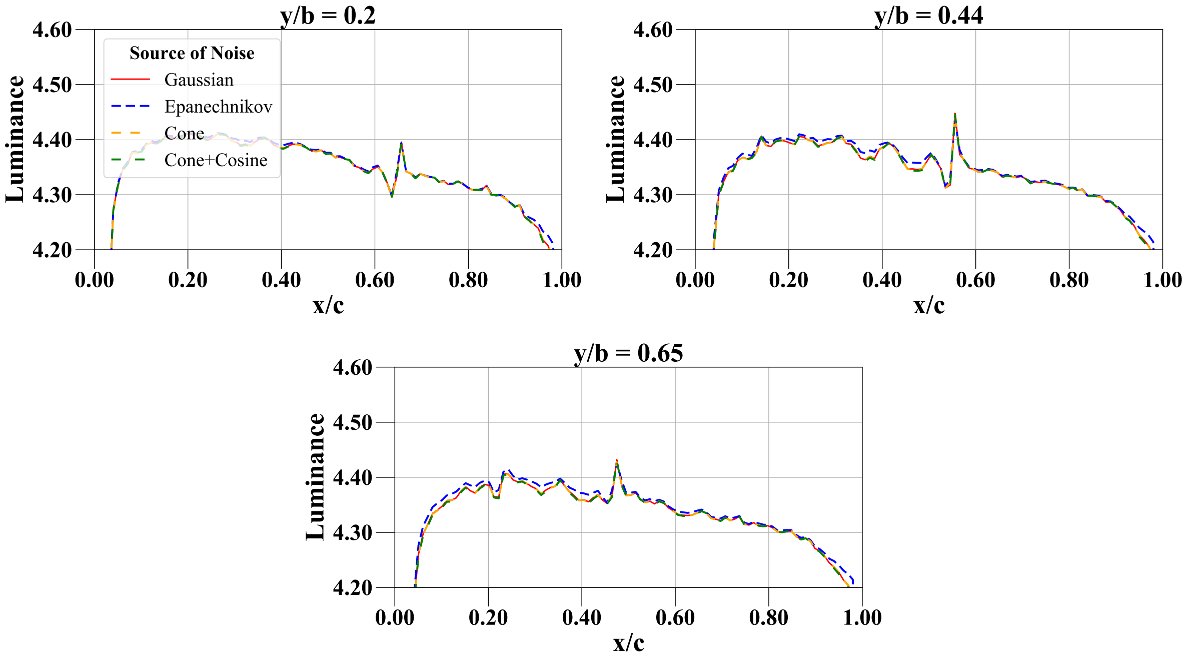


Figure 5.28: Analysis of the source of noise in the luminance distribution over the root region of the wing surface rendered using a fixed medium bandwidth.

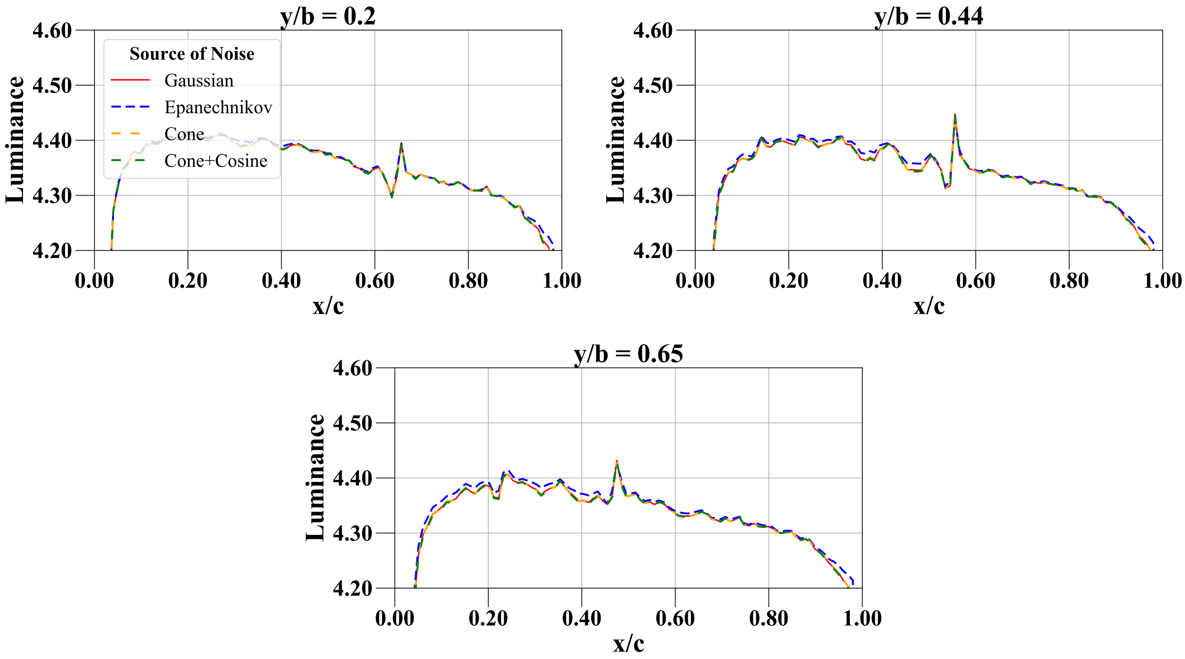


Figure 5.29: Analysis of the source of noise in the luminance distribution over the tip region of the wing surface rendered using a fixed medium bandwidth.

All of the analysed filters have similar performance and result in almost identical illuminations. The Epanechnikov is however relatively smoother across the entire illumination distribution, including the caustic details. Although the medium bandwidth utilised represents a compromise between under and oversmoothing of low and high-frequency natural and artificial features such as the shockshadows and noise, respectively, the latter is still present in the solution and, like in the previous analyses, it cannot be associated to the kernel choice.

The results of these investigations show that there remains some consistent noise structures over regions of supposedly uniform illumination regardless of the following: the underlying grid discretisation and fluid solution used to retrieve the nonlinear behaviour of light within the compressible flow domain; the number of illumination and visibility samples combined to reconstruct the illumination characteristics across the wing surface; and the choice of kernel to filter the contribution of individual photons to the local density estimation. In conclusion, therefore, since noise could not be attributed to any of the previously investigated sources, it has to be associated to a fundamental susceptibility within ray-tracing and photon mapping. The bandwidth selection does play an important role in noise control and it influences the inherent trade-offs between over and underfitting of the light energy samples and over and undersmoothing of the illumination solution in locations along the wing span where the shock wave characteristics vary considerably.

The source attributed to noise therefore suggests that a multi-stage adaptive approach to the kernel density estimation method to compute radiance is required to generate an accurate shock wave shadow. Such an approach would allow the bandwidths and degree of smoothing to vary along the wing span. A larger radius should be utilised where the shock wave strength and respective light refraction is minimal, thus smoothing out the noise and revealing the weaker lower-frequency caustic feature. A smaller radius should be utilised in regions dominated by strong shock waves to enable the resolution of the rapid changing and high magnitude of the light fringe details.

Silverman [171] suggested adapting the kernel bandwidth based on the density of photons within a query region, but the photon density does not vary extensively here, given that light ray deflections when they cross the shock wave discontinuities are small in comparison to other refraction phenomena. The resulting photon distribution in the photon map thus exhibits minimal displacements, visible only in the regions containing the strongest part of the shock wave, as shown in Figure 5.30.

Thus another new procedure is developed to guide the adaptation based on the assumption that the bandwidth size depends on the amount of light deflection. Using this assumption,

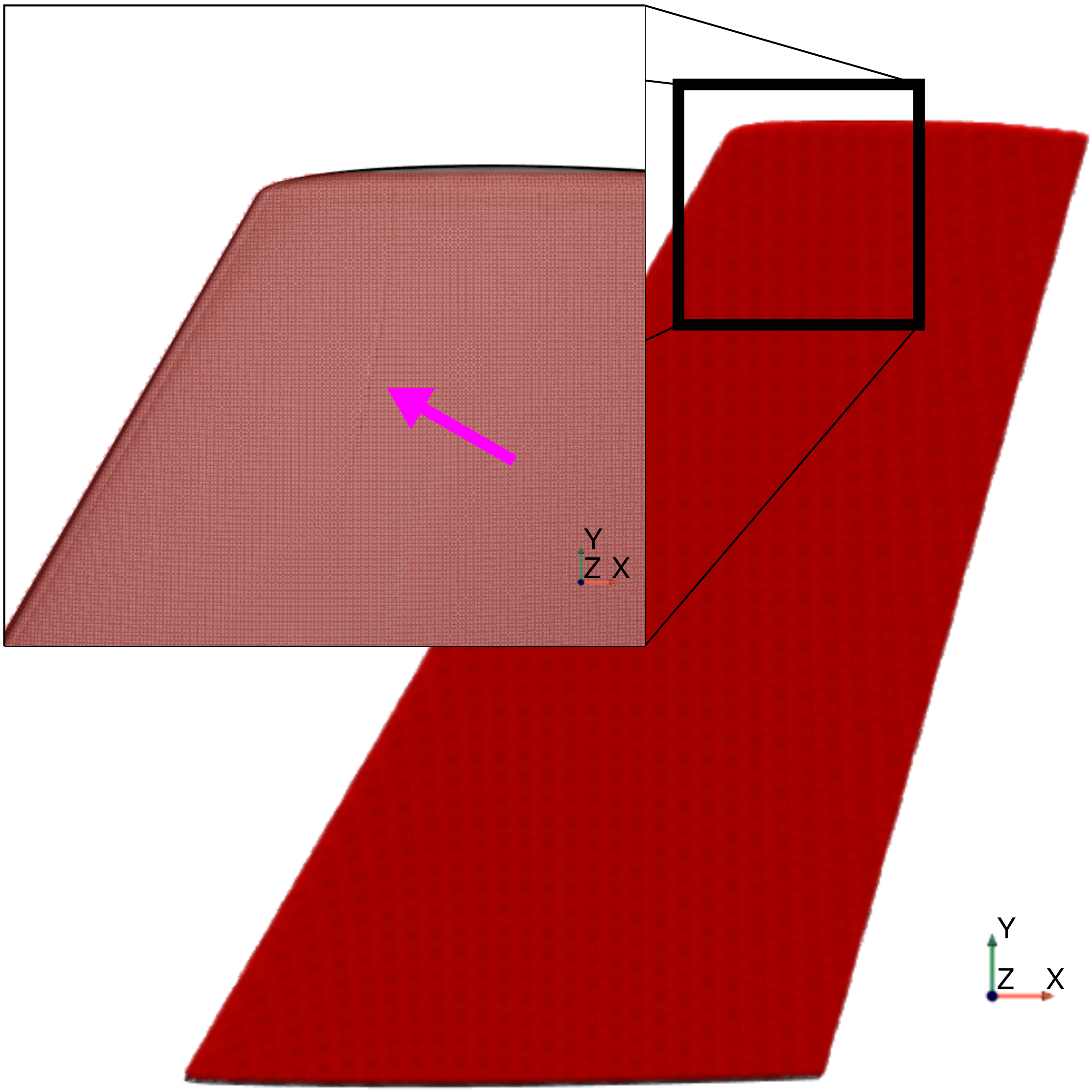


Figure 5.30: Example of the photon distribution in a certain photon map.

the bandwidth automatically adapts to the scale of refraction occurring in the domain of the compressible case under analysis. This is done by introducing a new variable, defined as

$$\theta_{z_{g/p}} = cos^{-1}(\vec{k} \cdot \vec{d}_{g/p})\frac{180}{\pi} \tag{5.1}$$

which is evaluated both for the global direction ($\vec{d}_g$) that light rays are emitted from and for the local photon direction ($\vec{d}_p$) from which they arrive at the wing surface after undergoing refraction. This new variable, θ_z, is simply the angle between the direction of light rays and the z-coordinate direction, $\overrightarrow{k} = \langle 0,0,1\rangle$. Subsequently, a quantity of the form

$$\delta\theta_{z_p} = \left|1 - \frac{\theta_{z_p}}{\theta_{z_g}}\right| \tag{5.2}$$

is also calculated to provide an indication of the normalised *deviation* - $\delta\theta_{z_p}$ - of this angle from the original θ_{z_g} established during light emission, where the higher the deviation the more light ray bending has occurred. Figure 5.31 shows the distribution of deviation angle over the wing surface, where a difference in shock strength and respective light deflection can be easily verified visually. The goal is to adapt the density estimation bandwidth based on this distribution.

The new adaptive bandwidth density estimation strategy here developed is then based on the distribution of deviation angle, $\delta\theta_{z_p}$, referred to as deviation from now on for simplicity. Lower and upper limits of radius to perform near neighbours searches, R_{min} and R_{max} are defined, and the implementation then has three stages:

(1) Approximate the deviation at a query point ($\delta\theta_{z_q}$) as the average of the deviation of near photons that are located within the minimum bandwidth:

$$\delta\theta_{z_q} = \frac{\sum_i^N \delta\theta_{z_{p_i}}}{N}\Big|_{R_{min}} \tag{5.3}$$

(2) i) Normalise the deviation at the query point by the maximum and minimum values of deviation at all other query points distributed over the wing:

$$\overline{\delta\theta}_{z_q} = \frac{\delta\theta_{z_q} - min\left(\delta\theta_{z_q}\right)\Big|_{wing}}{max\left(\delta\theta_{z_q}\right)\Big|_{wing} - min\left(\delta\theta_{z_q}\right)\Big|_{wing}} \tag{5.4}$$

ii) or normalise considering only the other query points distributed over the same spanwise station instead:

$$\overline{\delta\theta}_{z_q} = \frac{\delta\theta_{z_q} - min\left(\delta\theta_{z_q}\right)\Big|_{station}}{max\left(\delta\theta_{z_q}\right)\Big|_{station} - min\left(\delta\theta_{z_{station}}\right)\Big|_{station}} \tag{5.5}$$

iii) or normalise by the maximum and minimum values of deviation of near photons that are located within the maximum bandwidth:

$$\overline{\delta\theta}_{z_q} = \frac{\delta\theta_{z_q} - min\left(\delta\theta_{z_{photons}}\right)\Big|_{R_{max}}}{max\left(\delta\theta_{z_{photons}}\right)\Big|_{R_{max}} - min\left(\delta\theta_{z_{photons}}\right)\Big|_{R_{max}}} \tag{5.6}$$

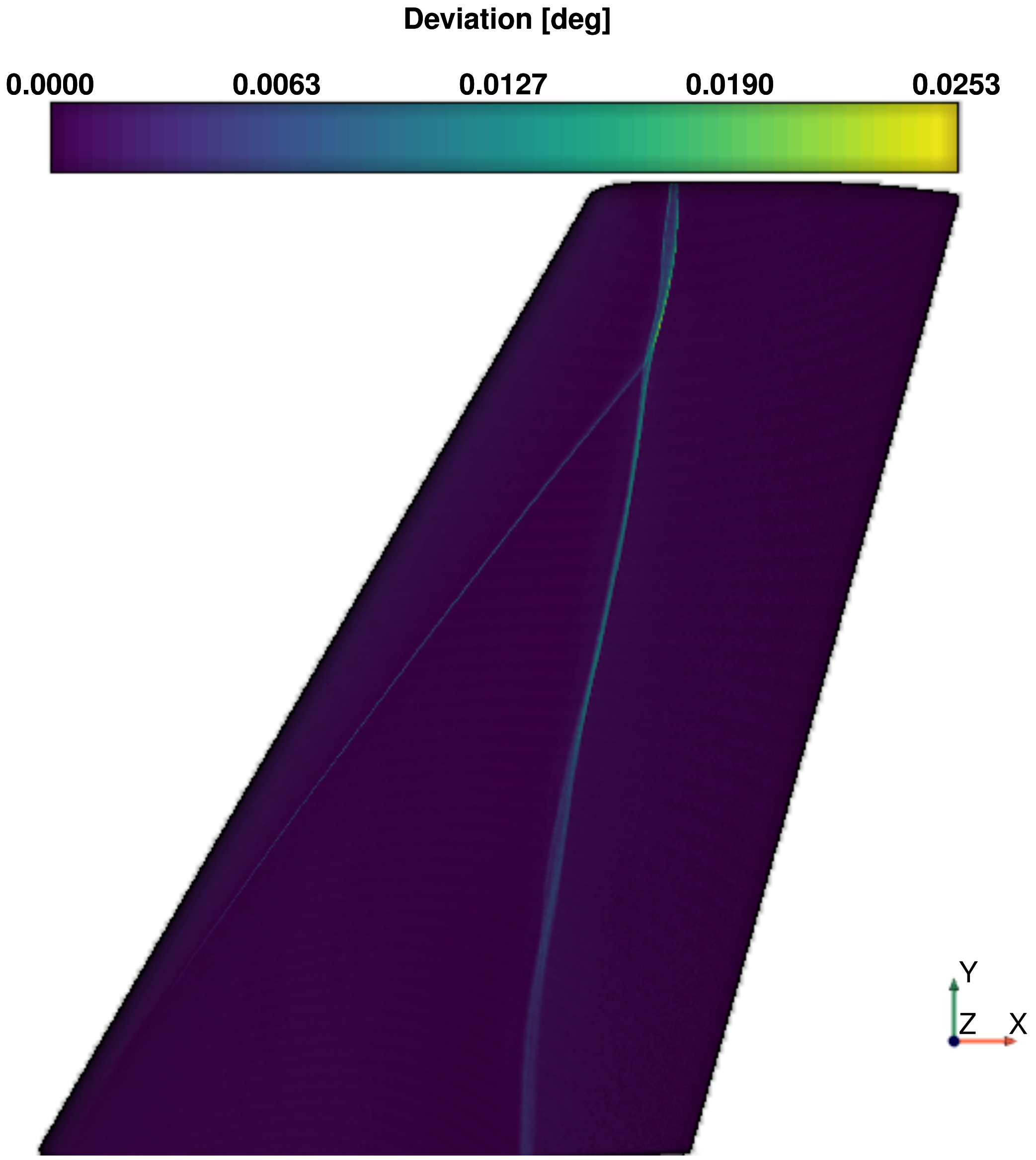


Figure 5.31: Example of normalised deviation $\delta\theta_{z_p}$ distribution calculated for a photon map.

(3) i) Assign the bandwidth at the query point bw_q linearly between the two radius extrema:

$$bw_q = (R_{min} - R_{max})\overline{\delta\theta}_{z_q} + R_{max} \tag{5.7}$$

ii) or assign the bandwidth exponentially between the two radius extrema if the maximum deviation of the photons within the maximum bandwidth is higher than a certain case-dependant threshold, otherwise just simply use the maximum radius:

$$bw_q = \begin{cases} \frac{R_{max}-R_{min}}{e^{10\overline{\delta\theta}_{zq}}} + R_{min}, & \text{if } max\left(\delta\theta_{z_p}\right)\Big|_{R_{max}} >= threshold \\ R_{max}, & \text{otherwise} \end{cases} \tag{5.8}$$

The alternative $1-2iii-3ii$ for mapping the bandwidths based on the deviation distribution is illustrated in Figures 5.32 and 5.33, as an example. In the region of uniform illumination where the deviation is zero (Figure 5.32) a large bandwidth is mapped to the query point. In the vicinity of a shock wave (Figure 5.33) the deviation is high and thus a small bandwidth is assigned to a the query point.

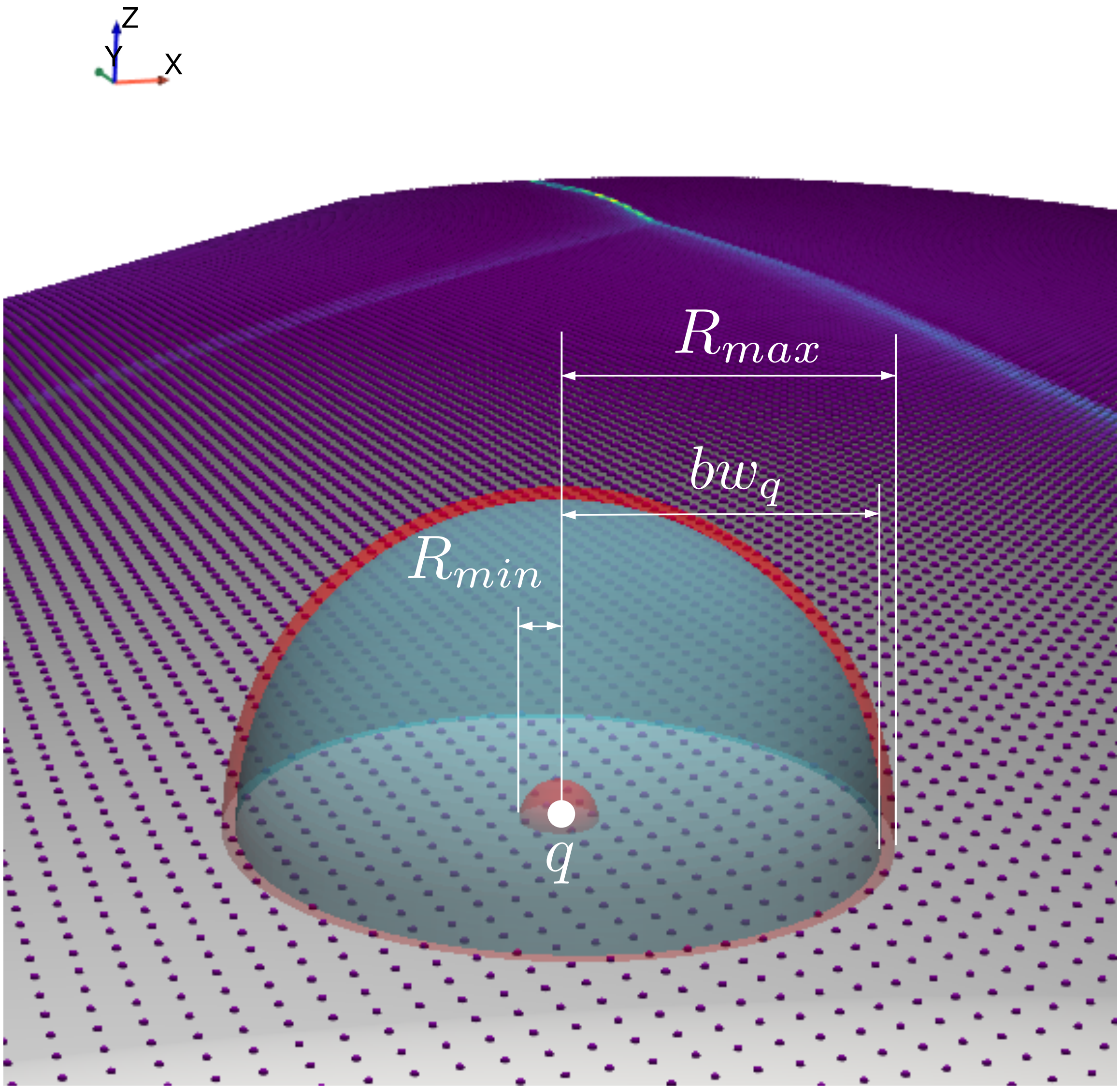


Figure 5.32: Schematic of the bandwidth mapping per window approach based on the deviation distribution on a region of uniform illumination.

The following sequences of stages are analysed: $1-2i-3i$; $1-2ii-3i$ and $1-2iii-3ii$. In effect, the approximation at stage 1 and normalisation at the different options of stage 2 represent a way to determine if the deviation at the query point is higher or lower in comparison to other

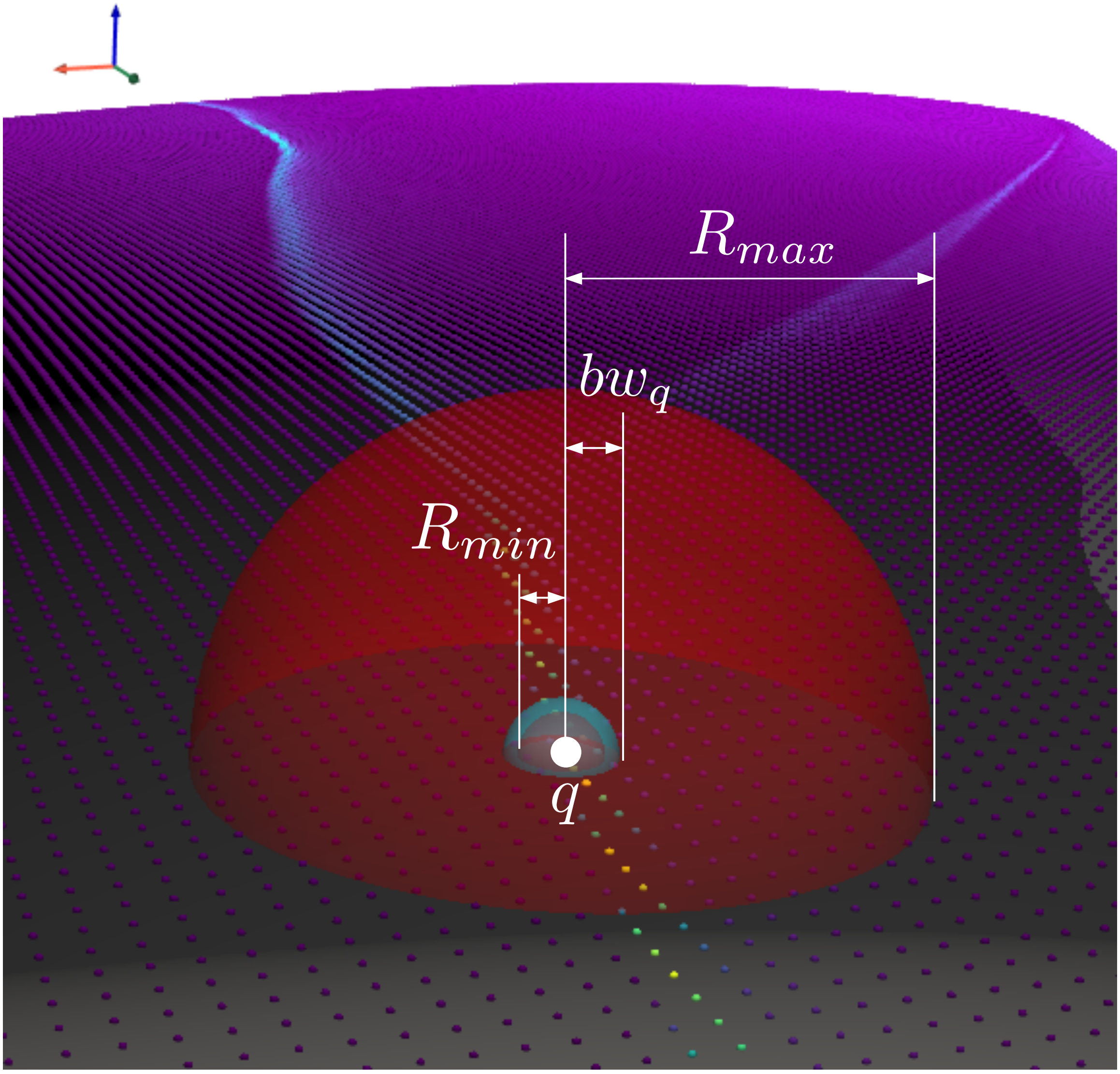


Figure 5.33: Schematic of the bandwidth mapping per window approach based on the deviation distribution on a region of a shock wave.

locations along the wing span. In $2i$ and $2ii$ the comparison is made with other query points, which in practice reduces the distribution of photon deviations to local averages, simplifying the comparison. Thus, by assessing all of the local averages, in the *mapping per span* process $(2i)$ the smaller bandwidth is assigned to the highest deviation, which is located around the wing tip where the shock wave is stronger. As the shock strength and respective deviation reduces along the other parts of the lambda patterned shock wave the bandwidth increases proportionally in a linear fashion $(3i)$. Alternatively, only the local averages situated at the same chord are assessed and the *mapping per chord* $(2ii)$ thus allows the assignment of the same small bandwidth to each local maximum deviation of different spanwise stations. Within a single station, the further away a query point is from the shock wave, the lower is its relative deviation and the larger is its assigned bandwidth, linearly proportional $(3i)$ to the rest of the chord. Finally, the remaining option focuses on even smaller neighbourhoods and assesses the query point deviation with respect to the deviation of near photons themselves, instead of comparing with other local averages. The *mapping per window* $(2iii)$ combined with the exponential $(3ii)$ assignment of bandwidths attempts to generalise the adaptation strategy. In this process, only local information is compared and the nonlinearity allows the concentration of smaller bandwidths near the shock wave and larger bandwidth everywhere else.

Independently of the strategy chosen, the process to select the minimum and maximum bandwidths for the adaptive density estimation is exactly the same. Starting from a known minimum radius that generates the strongest shockshadow at the tip region of the wing, the maximum radius is increased until a suitable caustic feature is also visible at the root region. Figures 5.34 and 5.35 present the luminance distributions obtained with the three different mapping strategies in a sample of $1M$ photons queried by 100 chordwise points per spanwise station and filtered by the Epanechnikov kernel with minimum and maximum radius of $R_{min} = 0.003$ and $R_{max} = 0.025$. These figures can be used to consider the characteristics of the three mapping strategies.

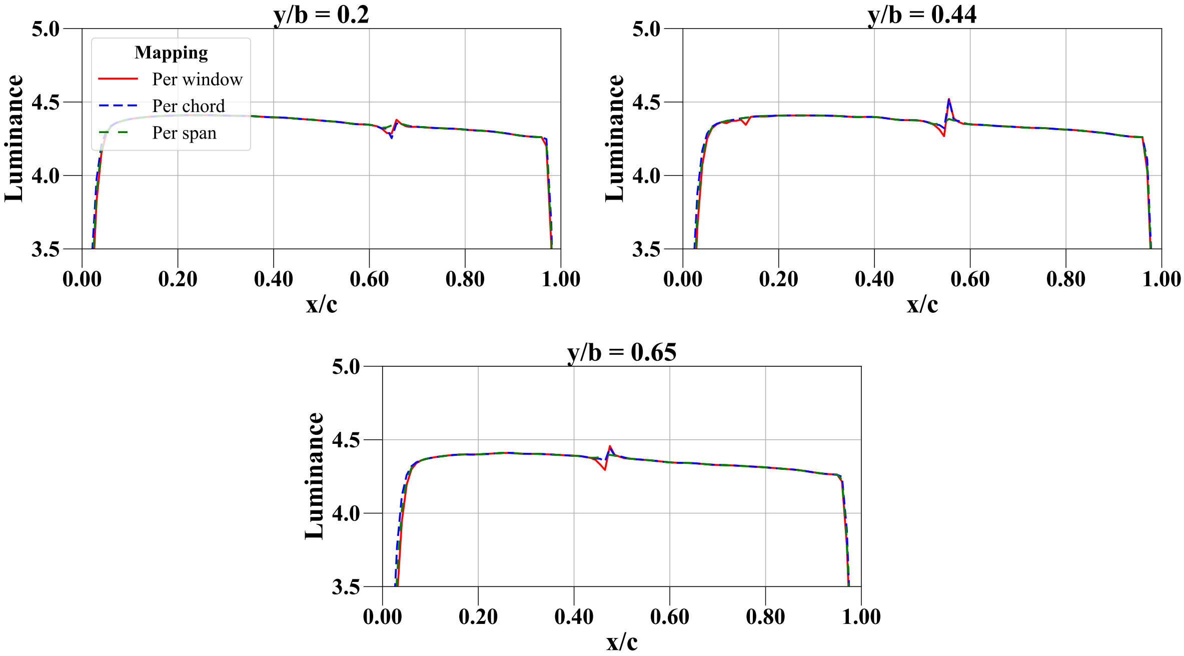


Figure 5.34: Luminance distribution over the root region of the wing surface rendered using multi-stage adaptive approaches to the kernel density estimation method.

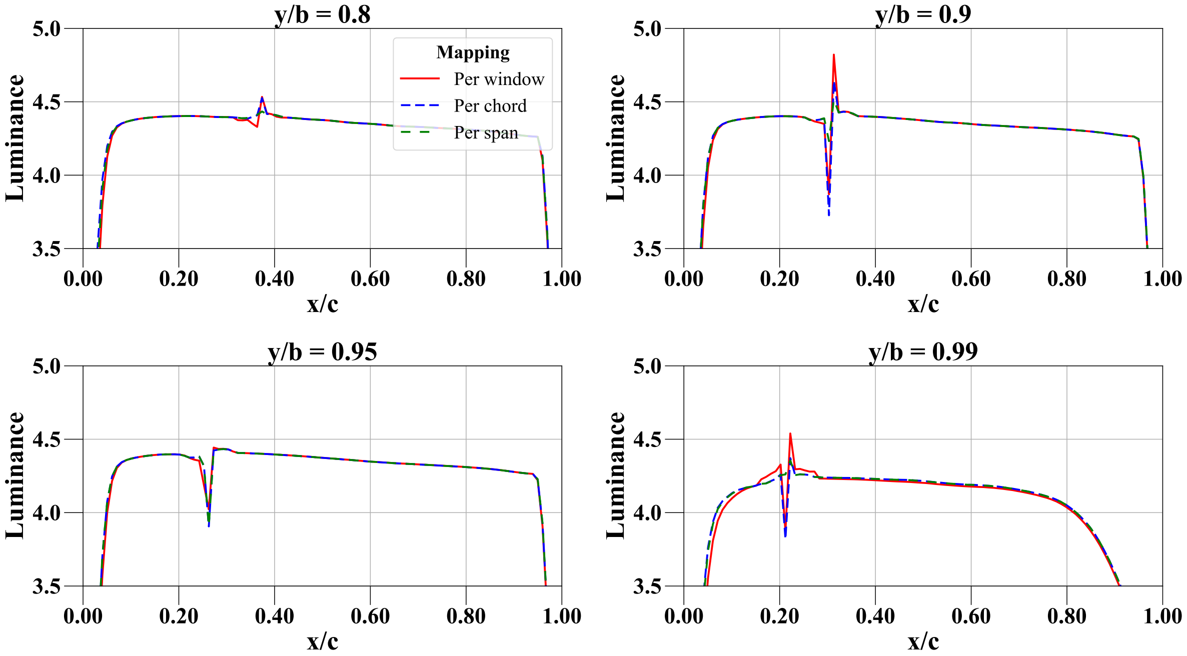


Figure 5.35: Luminance distribution over the tip region of the wing surface rendered using multi-stage adaptive approaches to the kernel density estimation method.

It can be seen that the mapping per span approach $(1-2i-3i)$ restricts the assignment of the smaller bandwidths and results in larger bandwidths being used in regions that still contain a shock wave of a moderate strength that also results in some degree of light deviation. The shock is weaker and the deviation is lower than that of the region containing the strongest part of the three-dimensional shock wave, but the larger bandwidth causes the shockshadows to be significantly blurred and almost invisible. An adjustment of the upper limit radius to further control the overall noise leads to a modification of the caustics in every single station, and at the station where the deviation is at its maximum only the bright fringe has its peak reduced. The challenge of controlling the level of smoothing across the whole wing span simultaneously is difficult to be overcome with this approach.

The figures show that the mapping per chord $(1-2ii-3i)$ approach developed in an attempt to resolve this issue only partially works for the present case because the seven reference spanwise stations invariably contain at least one portion of the lambda patterned shock wave. Nonetheless, parts of the caustic features are still oversmoothed due to the linear assignment of bandwidths at each spanwise section and therefore the strategy is not general enough. The fact that the deviation comparison is made based on query points only, which represent an average deviation over a certain chord length in this case, also makes the normalisation dependant on the distribution of query points. Those can fall in the vicinity of either a dark or a bright fringe, or both, but due to the linearity the smallest bandwidth will only be assigned to where the deviation is at its maximum. Therefore, one of those fringes will inevitably be dampened in order to preserve the other.

The mapping per window approach $(1-2iii-3ii)$ tries to generalise the adaptive kernel approach. Ideally, different upper and lower limits of search radius and one bandwidth mapping function would be specified per spanwise station, in order to locally adapt to the characteristics of the shock wave in each region. However, this introduces an excessive number of inputs that require tuning for every simulation, even if the case under analysis is the same but one of the other settings (e.g. the sun orientation) is changed. The exponential assignment of bandwidths based on the deviation information of a very local photon neighbourhood of the query point is a simplification believed to be general enough to treat different cases. It requires minimal tuning of R_{max}, R_{min} and of the *threshold*, and possibly the exponent in the denominator of the mapping function, for example. This forces small radius only in the shock wave regions and larger radius only in the uniform illumination regions. The figures show that this last strategy adequately smooths the luminance distribution and conditions an almost continuous solution at the edges of the shockshadows. Sharp caustic features are obtained in all of the spanwise stations and even a section of the leading edge portion of the shockshadow near the root appears. Mapping per window is thus determined to be the best alternative to resolve the shockshadow.

It can be seen from the luminance distributions where the adaptive bandwidth approach to density estimation was employed that some illumination features arise in other regions known to also contain parts of the shock wave or even adjacent to the shockshadow. These details are better resolved when the number of visibility samples or query points is increased. Figures 5.36 and 5.37 depict the higher resolution luminance distributions rendered from $1M$ photons filtered by the Epanechnikov kernel, whose bandwidth was mapped per window with radius extrema of $R_{min} = 0.003$ and $R_{max} = 0.025$.

Given that the sample size and the local bandwidths are kept fixed as the number of query points increases, the number of photons per query point invariably decreases and artificial noise can be seen to appear in the radiance estimation. Thus some high frequency oscillatory behaviour is rendered in the luminance curves, especially closer to the wing root. However, the underlying shape of the visual artifacts, such as for example the dark spots at $(x/c \approx 0.13, y/b = 0.44)$ - top right of Figure 5.36 and $(x/c \approx 0.32, y/b = 0.80)$ - top left of Figure 5.37, the bright spot at $(x/c \approx 0.16, y/b = 0.99)$ - bottom right of Figure 5.37 and the remaining caustics, are considered to be related to the natural illumination due to the influence of the shock wave. To investigate such artifacts, Figures 5.38 to 5.45 then show equivalent luminance distributions evaluated using more photons.

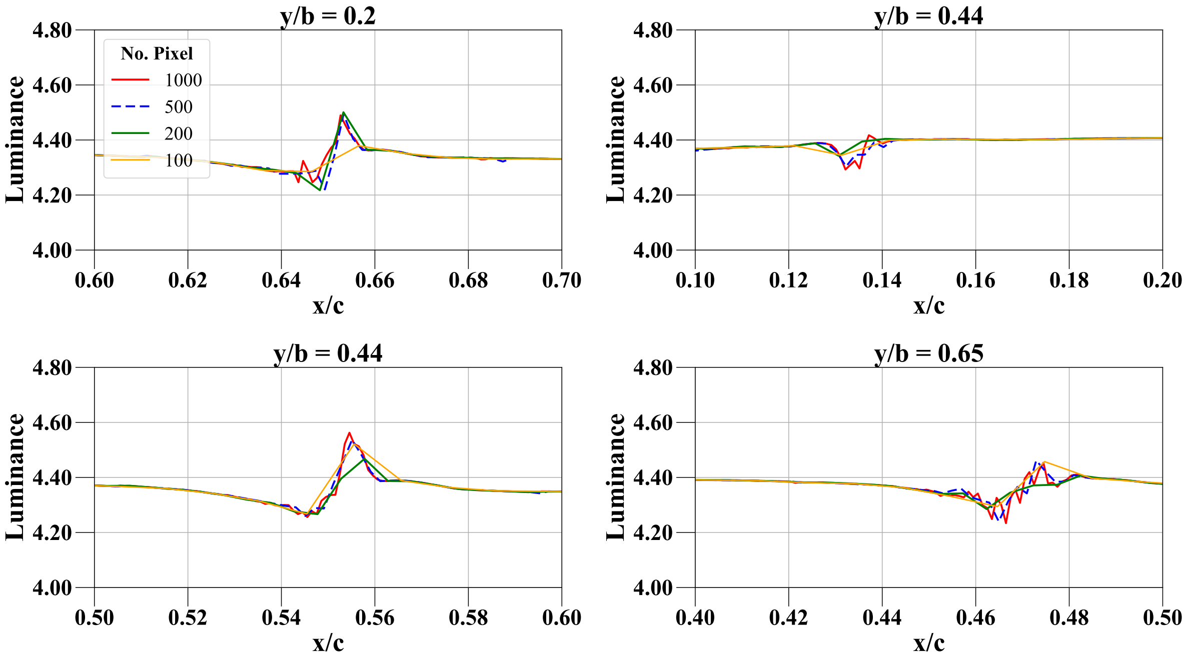


Figure 5.36: Luminance distribution over the root region of the wing surface rendered using $1M$ photons and different pixel resolutions.

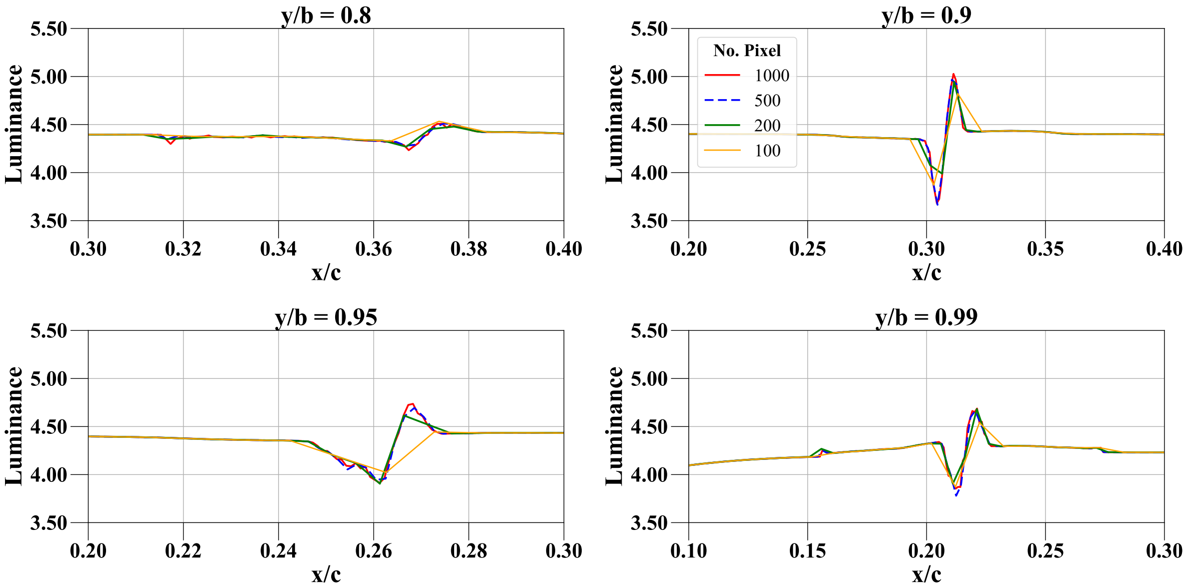


Figure 5.37: Luminance distribution over the tip region of the wing surface rendered using $1M$ photons and different pixel resolutions.

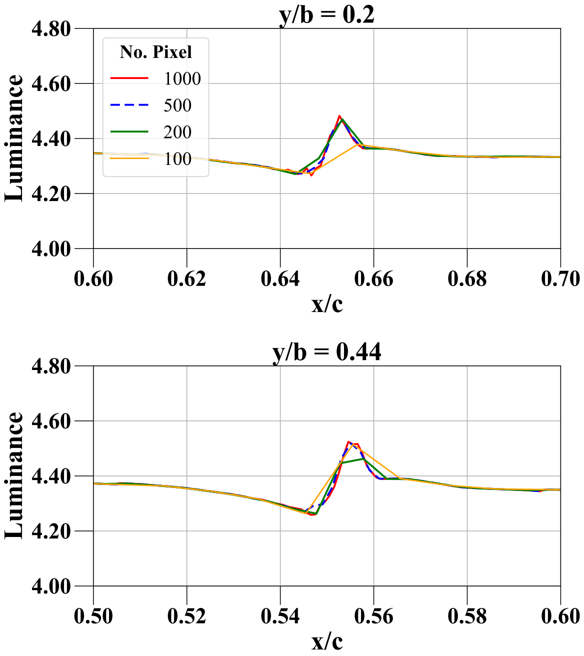


Figure 5.38: Luminance distribution over the root region of the wing surface rendered using $2M$ photons and different pixel resolutions.

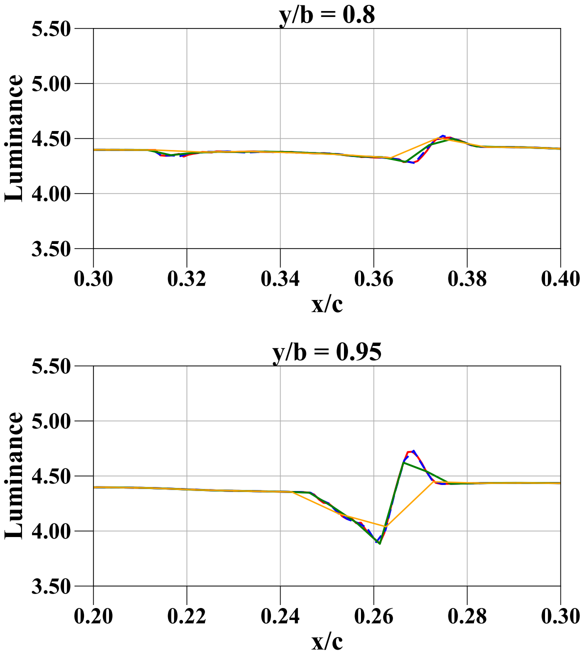


Figure 5.39: Luminance distribution over the tip region of the wing surface rendered using $2M$ photons and different pixel resolutions.

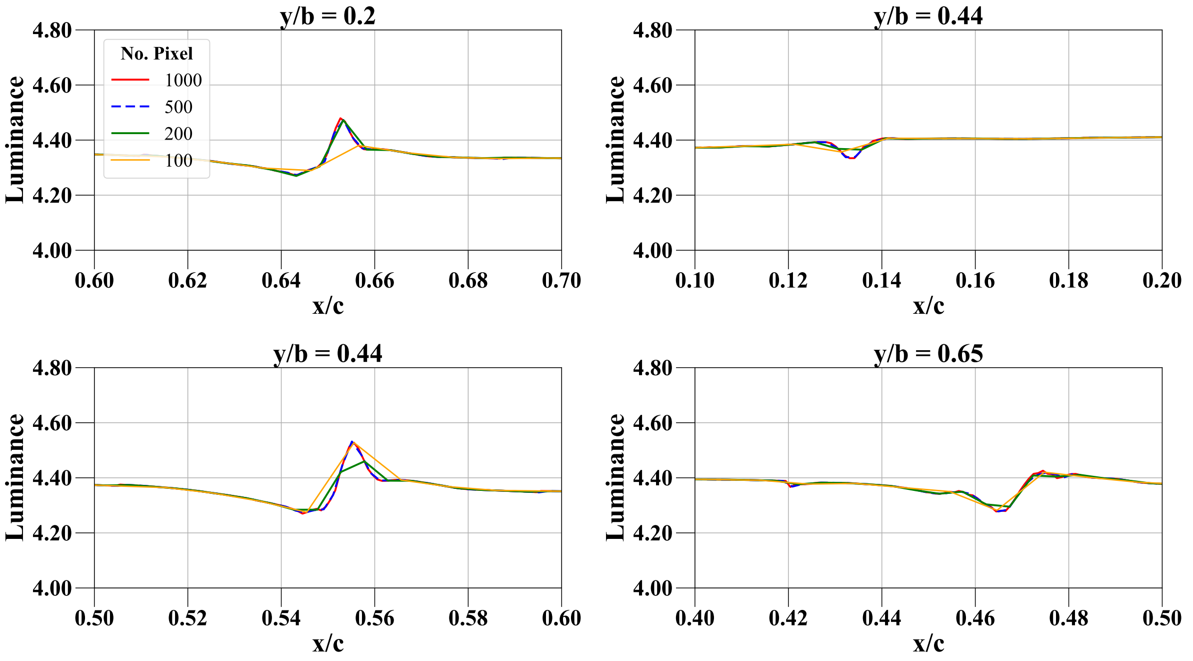


Figure 5.40: Luminance distribution over the root region of the wing surface rendered using $5M$ photons and different pixel resolutions.

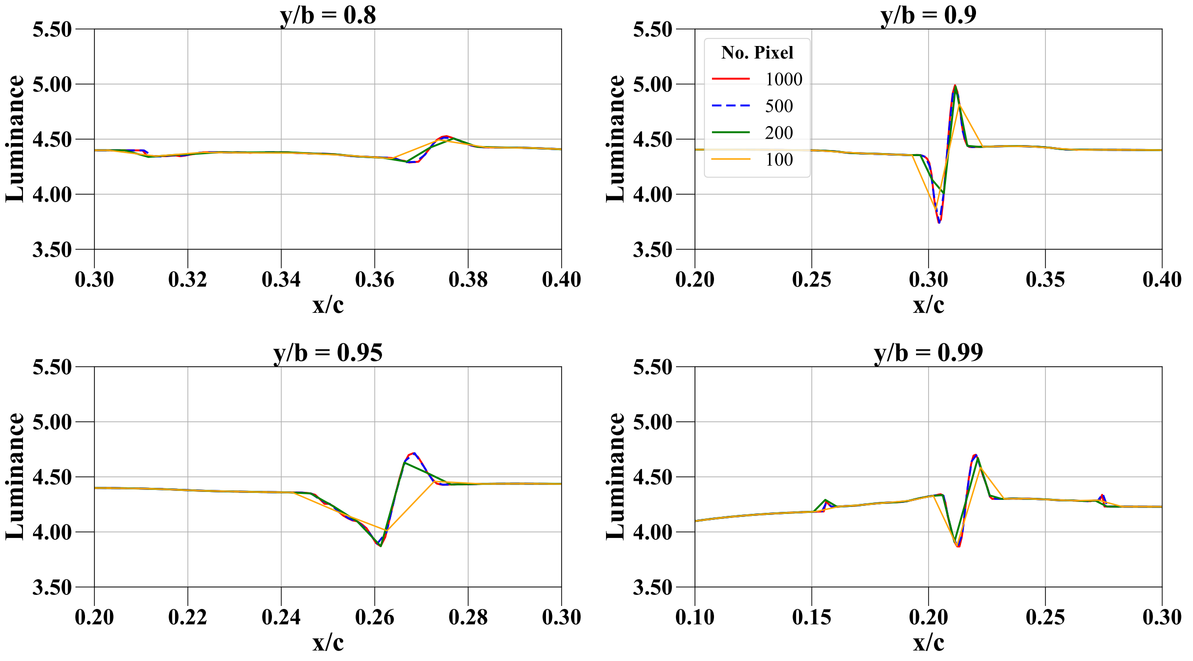


Figure 5.41: Luminance distribution over the tip region of the wing surface rendered using $5M$ photons and different pixel resolutions.

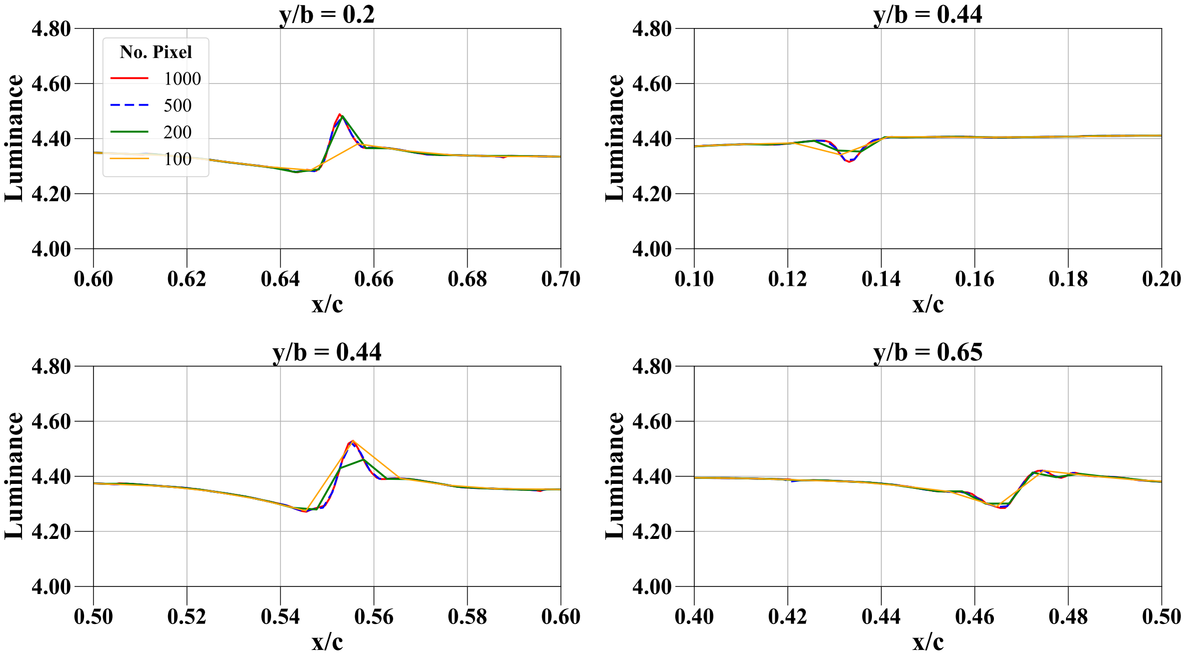


Figure 5.42: Luminance distribution over the root region of the wing surface rendered using $10M$ photons and different pixel resolutions.

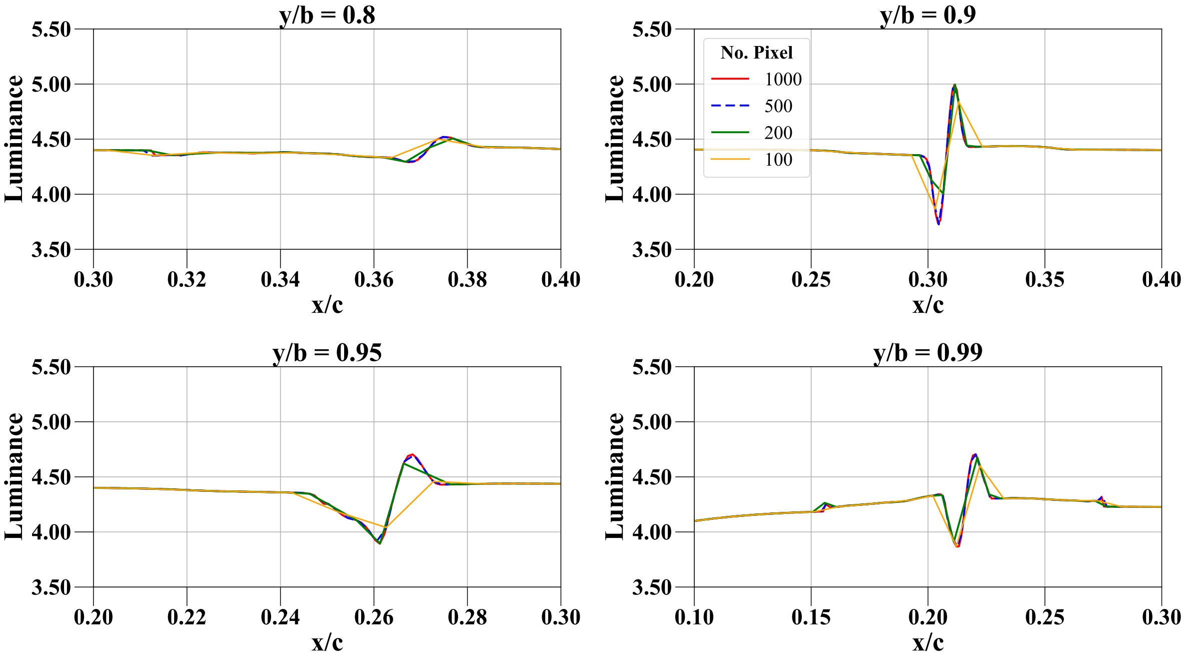


Figure 5.43: Luminance distribution over the tip region of the wing surface rendered using $10M$ photons and different pixel resolutions.

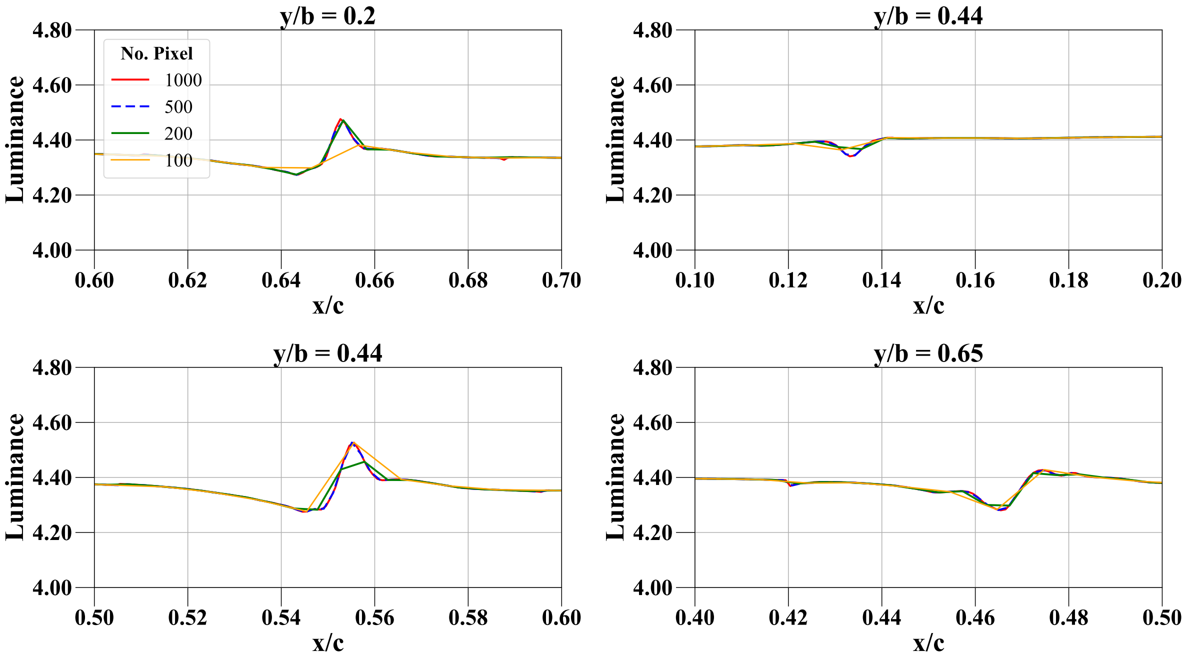


Figure 5.44: Luminance distribution over the root region of the wing surface rendered using $20M$ photons and different pixel resolutions.

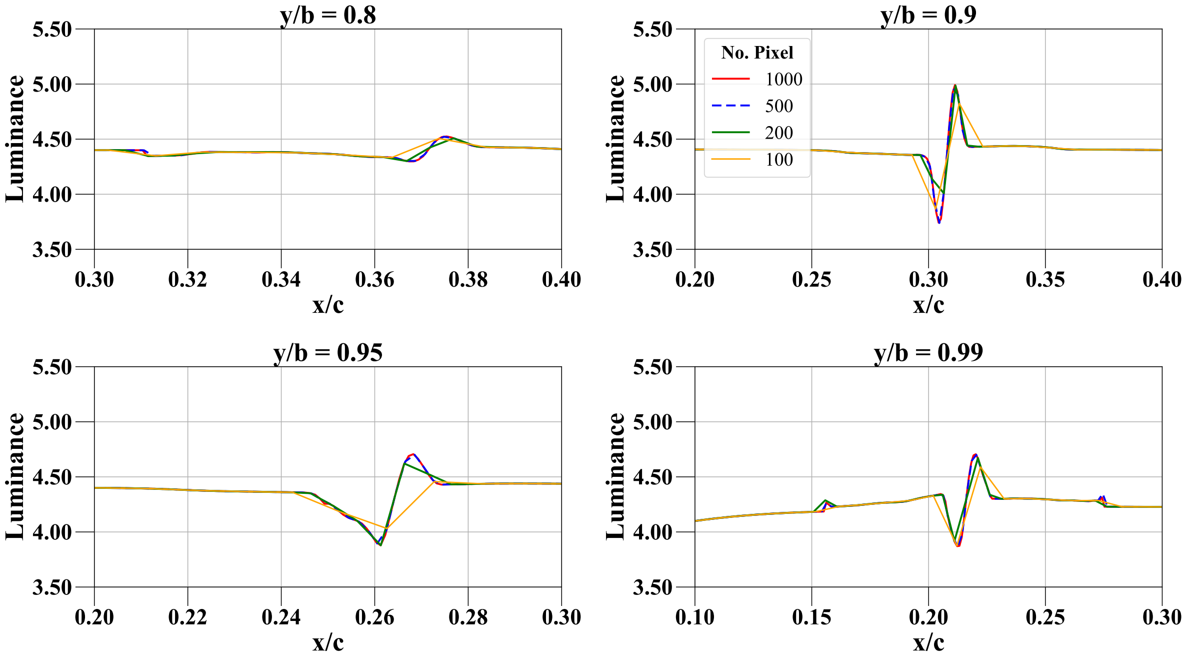


Figure 5.45: Luminance distribution over the tip region of the wing surface rendered using $20M$ photons and different pixel resolutions.

When more than $1M$ illumination samples are traced, the previously mentioned visual artifacts of interest become clearer. Noise is reduced due to the addition of more visibility samples and some other details are unmasked. At $(x/c \approx 0.42, y/b = 0.65)$ one more marginal shadow appears, at $(x/c \approx 0.27, y/b = 0.99)$ another highlight emerges downstream of the main shockshadow and at $y/b = 0.65$ and $y/b = 0.95$ the principal caustic appears to have multiple fringes, characterised by small fluctuations within the main adjacent peak and valley. Moreover, all of the samples analysed require only around 500 query points per spanwise station to resolve all of the artifacts, and additional points in effect only result in a smoother luminance solution with the same caustic shapes present.

Finally the effect of varying the number of photons is investigated, with focus on the visual artifacts, as shown in Figures 5.46, 5.47 and 5.48. In all the figures the luminance distributions are rendered with 500 query points, the Epanechnikov kernel, the mapping per window adaptive process and, again, $R_{min} = 0.003$ and $R_{max} = 0.025$.

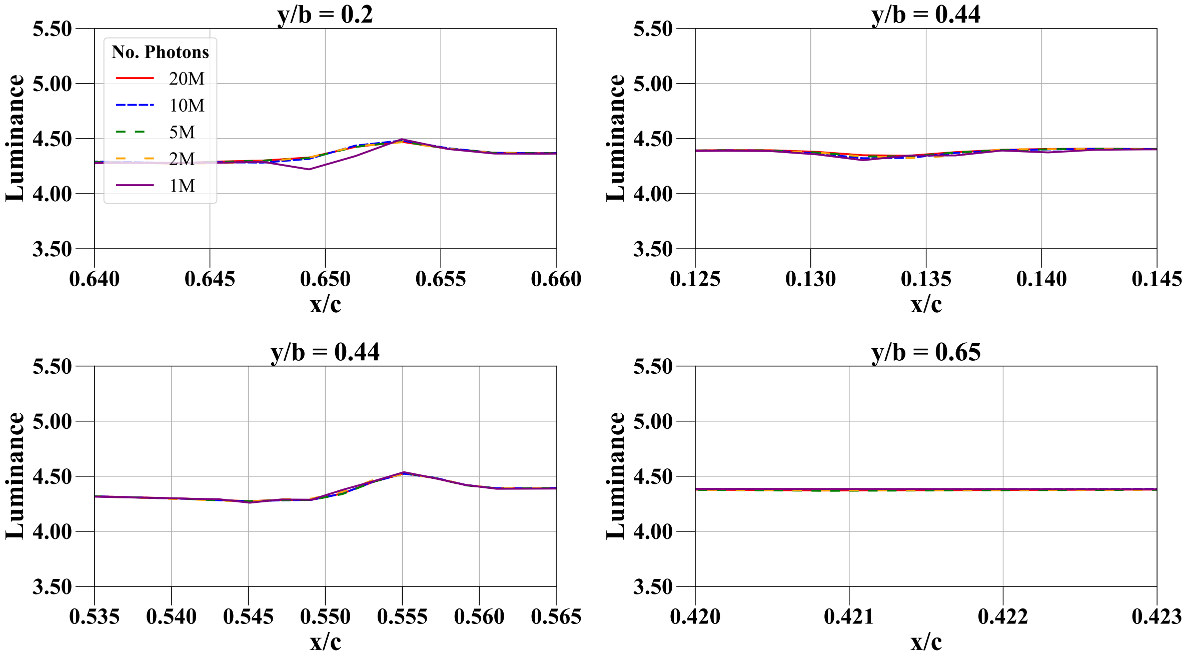


Figure 5.46: Luminance distribution over the root region of the wing surface rendered using different photon sample sizes.

From these graphs, it is possible to conclude that a sample of size $1M$ is not actually enough to accurately capture the shape of all of the caustic features present on the wing surface. With more than $2M$ the convergence of the luminance solution is achieved and, more securely with a level of $5M$ photons, the shockshadows and isolated dark and bright fringes do not alter with further increase of the number of illumination samples.

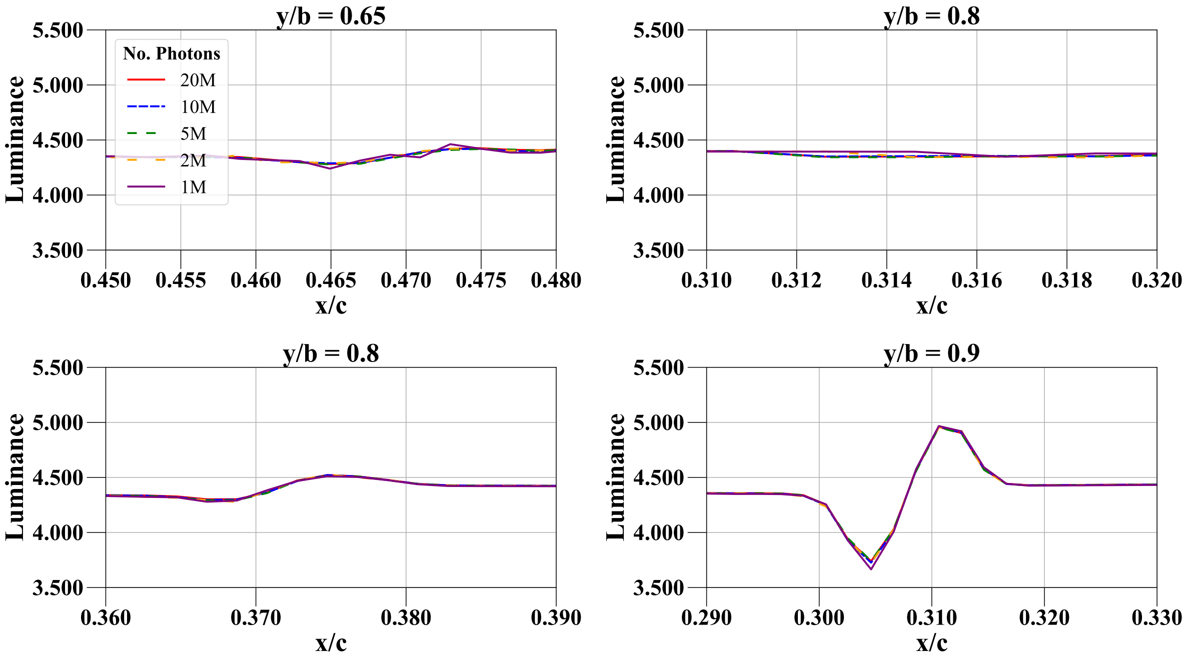


Figure 5.47: Luminance distribution over the middle region of the wing surface rendered using different photon sample sizes.

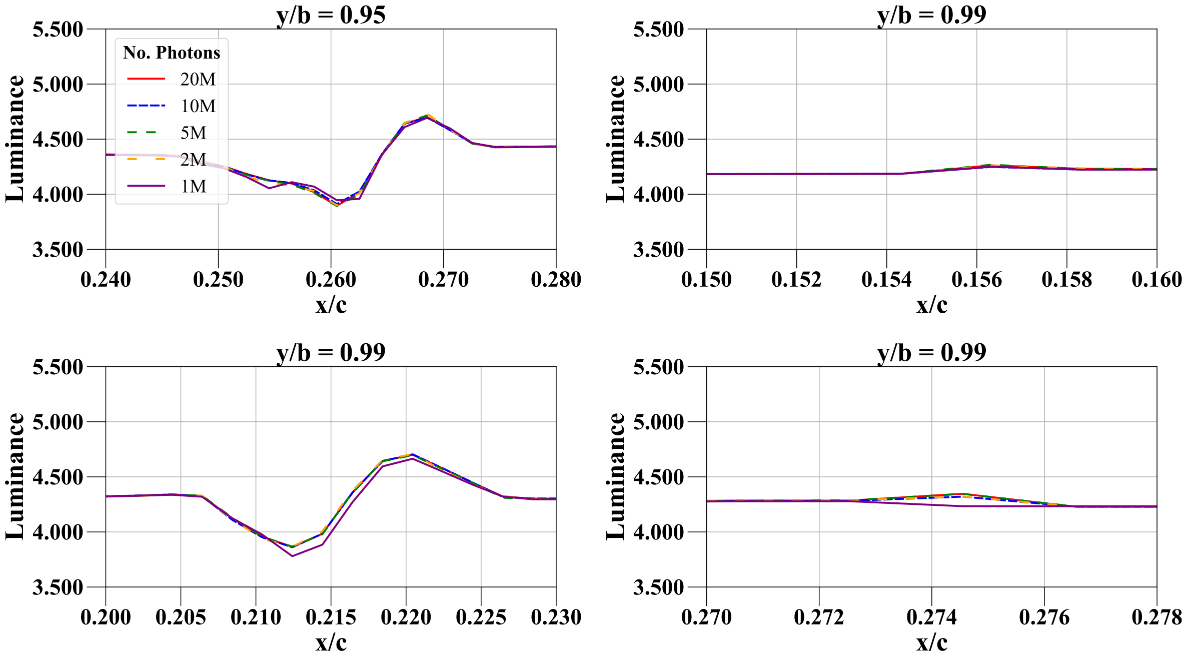


Figure 5.48: Luminance distribution over the tip region of the wing surface rendered using different photon sample sizes.

In conclusion, it is determined that the best solution to generate a faithful synthetic image of the shockshadow phenomenon is the estimation of the luminance distribution over the wing surface: from a photon map containing $5M$ photons; rendered through a mapping per window multi-stage adaptive bandwidth approach to the Epanechnikov kernel density estimation method; with radius extrema of $R_{min} = 0.003$ and $R_{max} = 0.025$; and applied in 500 chordwise query points per spanwise station.

5.2.3 Analysis of the Effect of the Light Source Direction on Synthetic Shockshadows

In Section 5.2.2, the minimum number of light rays required to be traced from the sun during the illumination stage to computationally generate an accurate shockshadow image has been determined. This sample size has to be combined with image filtering and an adaptive bandwidth approach. This approach balanced out the trade-off between bias and variance in the energy density estimation process using the photon map. It also established a compromise in the trade-off between undersmoothing of the illumination distribution at the wing root and oversmoothing at the tip, or under/overfitting of the light energy samples, thus revealing all of the expected illumination features. Moreover, the minimum number of query points or "pixels" required to appropriately resolve these features during the visibility stage was also determined.

Throughout the sampling and reconstruction analyses, however, the orientation of the sun in terms of azimuth and elevation angles with respect to the wing quarter-chord line has been kept fixed. Nevertheless, from the work of Fisher et al. [16], it is known that there is a range of sun orientations that produce visible shockshadows and that higher elevation angles generate the best visual effects. Therefore, the second analysis focuses on a confirmation of some of these in-flight experimental conclusions but now in an easily modifiable virtual environment.

5.2.3.1 Orientation of the lambda patterned shock wave

In order to analyse the effect that the direction of the light source has on the synthetic shockshadows, first the orientation of the surface of discontinuity that refracts light, i.e the shock wave itself, is measured against the quarter-chord line rotated $+90°$ around the x-coordinate direction (i.e. in the xz plane). These measurements are made at the seven characteristic spanwise stations of the Onera M6-wing [203], as they are suitable reference locations along the lambda patterned shock wave. To assess these angles, the regions where the gradient of the refractive index of air is different than zero are coloured in red to simply highlight the inhomogeneities, whereas the homogeneous regions are not displayed. These plots are then exported to a CAD software, where the rotated quarter-chord line is drawn over the horizontal chord line and the actual angles are then evaluated, as presented in Figure 5.49.

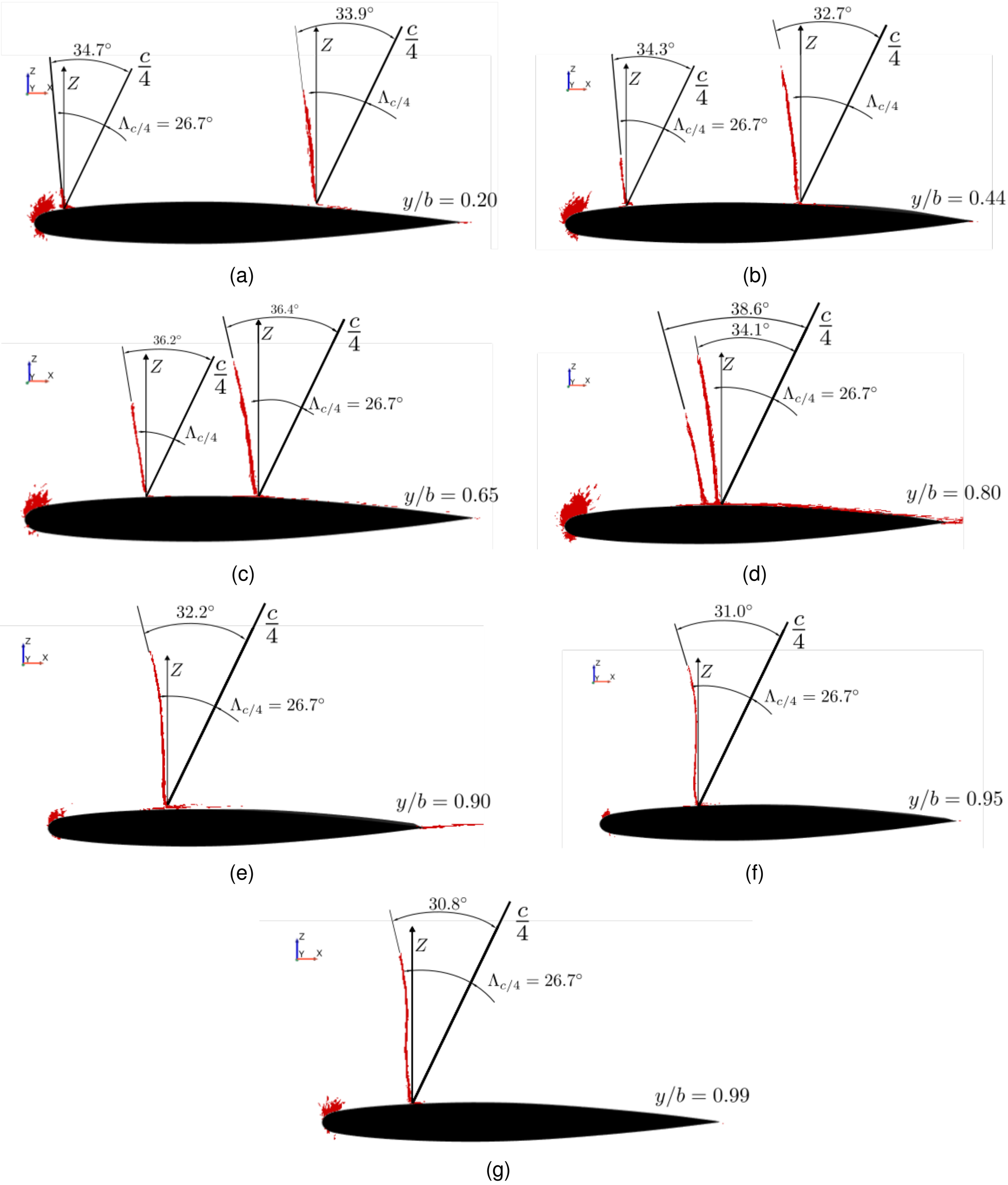


Figure 5.49: Orientation of the lambda patterned shock wave surface with respect to the wing quarter-chord line rotated around the x-coordinate direction, at the seven reference spanwise stations of the Onera M6-wing.

The angle that the shock wave makes with the rotated quarter-chord line is of particular interest because the sun, modelled as a directional light source in the light simulation framework, is parameterised with respect to this line of reference through the azimuth angle. According to Snell's Law of Refraction, which can be applied at any point along the shock wave surface, the maximum light deflection occurs when incident light rays form an angle of almost $90°$ with the normal of the surface. This involves light rays approximately aligned with the shock wave surface curvature, a condition referred to as grazing incidence in refraction phenomena. In theory, grazing incidence would produce the most pronounced shockshadow effect, or caustic pattern, considering that the surface of refraction has a variable shape along the wing span. This was previously confirmed for the pseudo-three-dimensional NACA 0012 wing (see Section 5.1.4).

The variable shape of the shock wave indicates that there will be a different optimal orientation of the sun to capture different parts of the shockshadow. In the wing inboard region (Figures 5.49(a) to 5.49(d)), the lambda patterned shock wave has two strands. The front leading edge one makes angles ranging from approximately $34.3°$ to $38.6°$ with the quarter-chord line. Analysing this orientation from the root towards the point where the two strands merge together, it is possible to see that the leading edge strand becomes more inclined with an increase of approximately $4°$. The trailing edge strand forms angles varying from $32.7°$ to $36.4°$. The single shock wave in the wing outboard region (Figures 5.49(e) to 5.49(g)), has less variation in the angle from the merging point towards the tip, with angles between $30.8°$ and $32°$.

Since the grazing incidence of light cannot be achieved at every spanwise section of the shock wave surface simultaneously, in one single illumination pass and computational image, a compromise in light source orientation has to be established. A mean value in azimuth of the sun of $35°$ is then estimated to cover almost grazing incidences for both of the shock waves in the wing inboard region. It is slightly further from the grazing condition incidence for the shock wave in the outboard wing region, but the discrepancy is still quite small. In this region the shock wave is stronger, i.e. the refractive index gradient is higher, hence the discontinuity in the optical medium itself should generate a more prominent refraction. It is also important to note that in this region the shock wave has a curved shape, instead of being simply oblique, a characteristic that will also influence the resulting shockshadow. This makes it difficult to determine a single orientation to achieve grazing incidence across its whole shape or height.

5.2.3.2 Optimal azimuth of the sun

Luminance distributions rendered considering the same simplifications used during the sampling and reconstruction analyses, i.e. a perfectly diffuse reflective surface and visibility in homogeneous optical media, are shown in Figures 5.50, 5.51, 5.52, 5.53 and 5.54. In computer graphics, this is usually called direct caustic rendering straight from the caustic photon map (the only one in the present case). The luminance is calculated by tracing $5M$ light rays during the illumination stage and using 1000×1000 query points equidistributed over the wing during the visibility stage. These samples are combined through the mapping-per-window adaptive bandwidth approach with the Epanechnikov kernel in photon mapping, with radius extrema of $R_{min} = 0.003$ and $R_{max} = 0.025$. The sun elevation is fixed at $90°$ initially and its azimuth is varied between $20°$ and $40°$, thus maintaining the light source straight above and progressively moving it slightly towards the leading edge of the wing. The elevation is kept high because the shock wave surface, if viewed from the front of the wing in the yz plane, is normal to the wing and from this perspective a grazing incidence is already achieved without inclining the sun through its elevation angle.

The luminance distributions obtained directly from the numerical solution of the rendering equation still lack the colour contrast observable in reality in devices that capture, record and display images. These calculated distributions are in fact in the "raw" or unprocessed data format. For this reason it is useful to understand how devices work and which type of processing can be performed to render sharper shockshadows. Modern digital cameras have three important components: a metal oxide semiconductor (MOS) capacitor that works as a photodiode, the fundamental light-sensing unit, and converts and stores the energy being carried by the photons arriving at the sensor to an electrical charge based on the exposure time; an on-chip multiplication gain or output amplifier that converts the charge to a proportional analog voltage signal; and an analog-to-digital converter (ADC) that performs the tone mapping in the sense that it reads the voltage, compares it against a tone curve and translates it into a digital RGB triplet with the amount of bit depth (e.g. 8, 12 or 24) encoded in the camera [207]. In order to enhance the bright and dark bands of the shockshadow patterns and thus improve the contrast of the rendered images, a step-shaped tone curve is then applied to map the calculated luminance values to the final 8-bit sRGB colours to be displayed by the VTK plotting utility. This curve simply renders black all the values below $Y = 40$, selected by visual inspection, and linearly map values above this lower limit between $[0, 255]$. In existing image editing software this contrast enhancement technique is commonly employed. An upper limit is not set with the purpose of providing visualisation of all shockshadows in all sun orientations, but the luminance values can be easily compared through the colour bars.

With a $20°$ sun azimuth (Figure 5.50), light rays arrive at the shock wave from behind, at angles of incidence around $10°$ or higher along the wing span. Although a small part of the shockshadow is generated near the strongest part of the shock wave at the very end of the wing tip, it is mostly invisible and what is observable is a rather subtle and widely spread shadow in front of the shock. Both of these effects are only visible due to the contrast enhancement processing applied to the image. In reality, the differences in the shading colours are minor and neither the shockshadow nor the front shadow would be observed as they would be

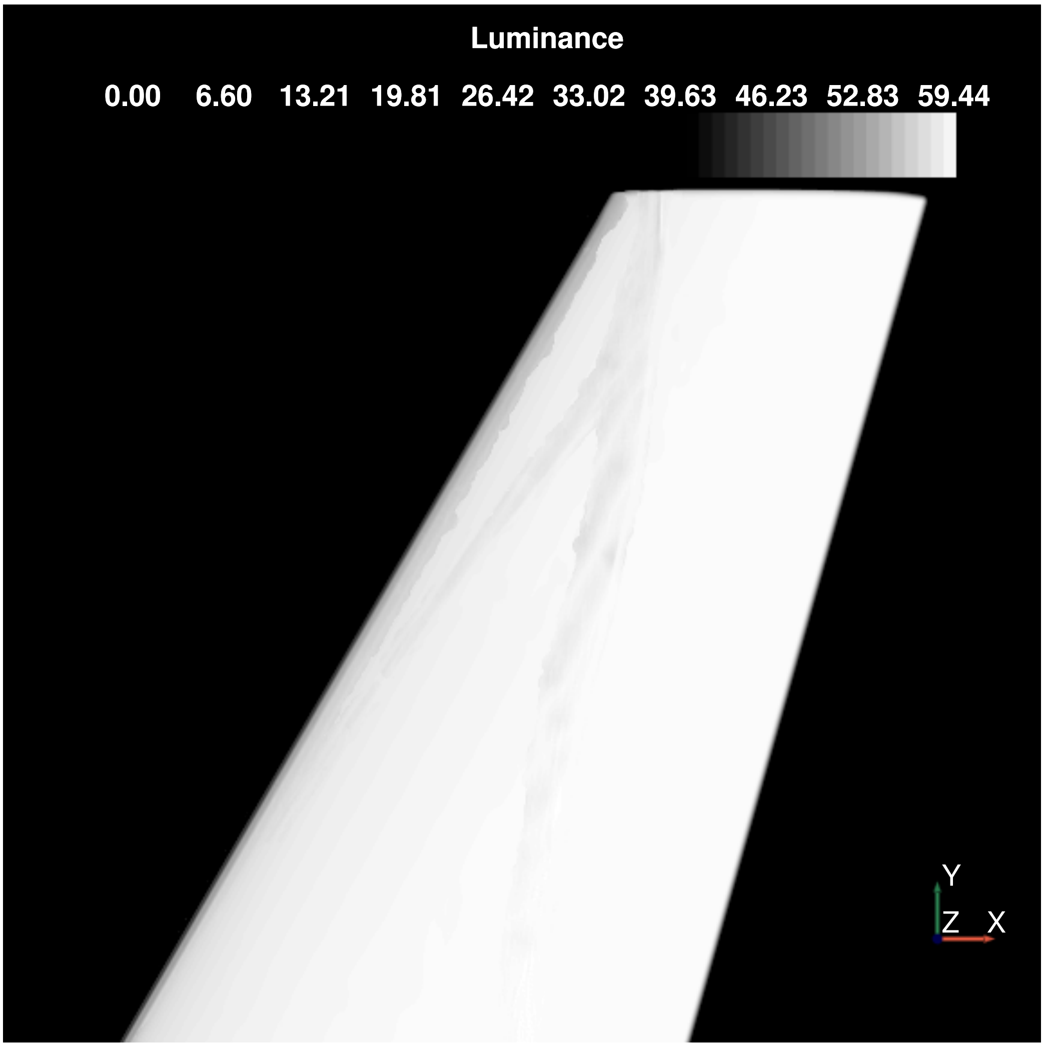


Figure 5.50: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $90°$ and an azimuth angle of $20°$.

indistinguishable form the wing surface colour. At $25°$ azimuth (Figure 5.51) the sun is almost aligned with the vertical direction and now two different parts of the shockshadow are visible, including a section in the wing inboard and another in the outboard. This happens because light rays approach the grazing incidence condition for sections of the shock wave surface in those regions. Nevertheless, the shockshadow is still incomplete.

The approach at close to a grazing incidence condition is more true at a $30°$ azimuth (Figure 5.52). At this orientation the sun generates an almost complete shockshadow of the main strand of the lambda patterned shock wave. The leading edge strand is still barely visible and so is a region at the middle of the trailing edge strand. The inclination of the shock wave at the middle of the wing is actually closer to $35°$, which explains why at this sun azimuth (Figure

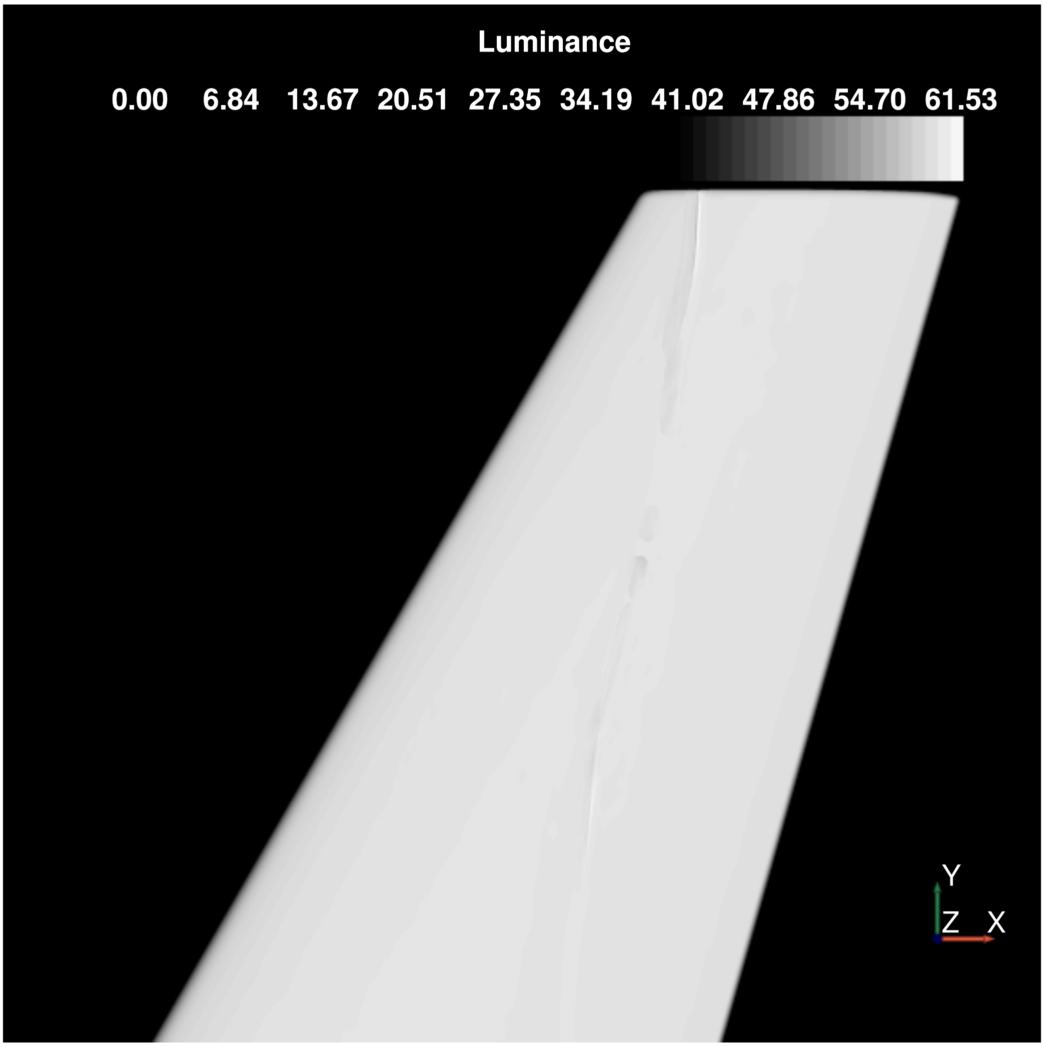


Figure 5.51: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $90°$ and an azimuth angle of $25°$.

5.53) the whole main strand of the shockshadow is distinguishable with a good visual contrast. As estimated in the previous section, at this orientation a near grazing incidence is expected for all of the weaker shock region and a close match in the stronger shock wave surface.

As the sun tilts forward with a $40°$ azimuth (Figure 5.54) the quality of the main shockshadow degrades. However, the secondary leading edge strand of the shock wave now produces an improved shockshadow, as in this region the shock wave is more inclined and light therefore only grazes the discontinuity at higher azimuths. The caustic or concentration of light behind the shadow at this front shockshadow is very subtle, and almost blends with the wing surface colour. This is because the strength of the shock wave, or refractive index gradient, is very weak in this region. Interestingly, in both Figures 5.53 and 5.54 the shock wave at the wing

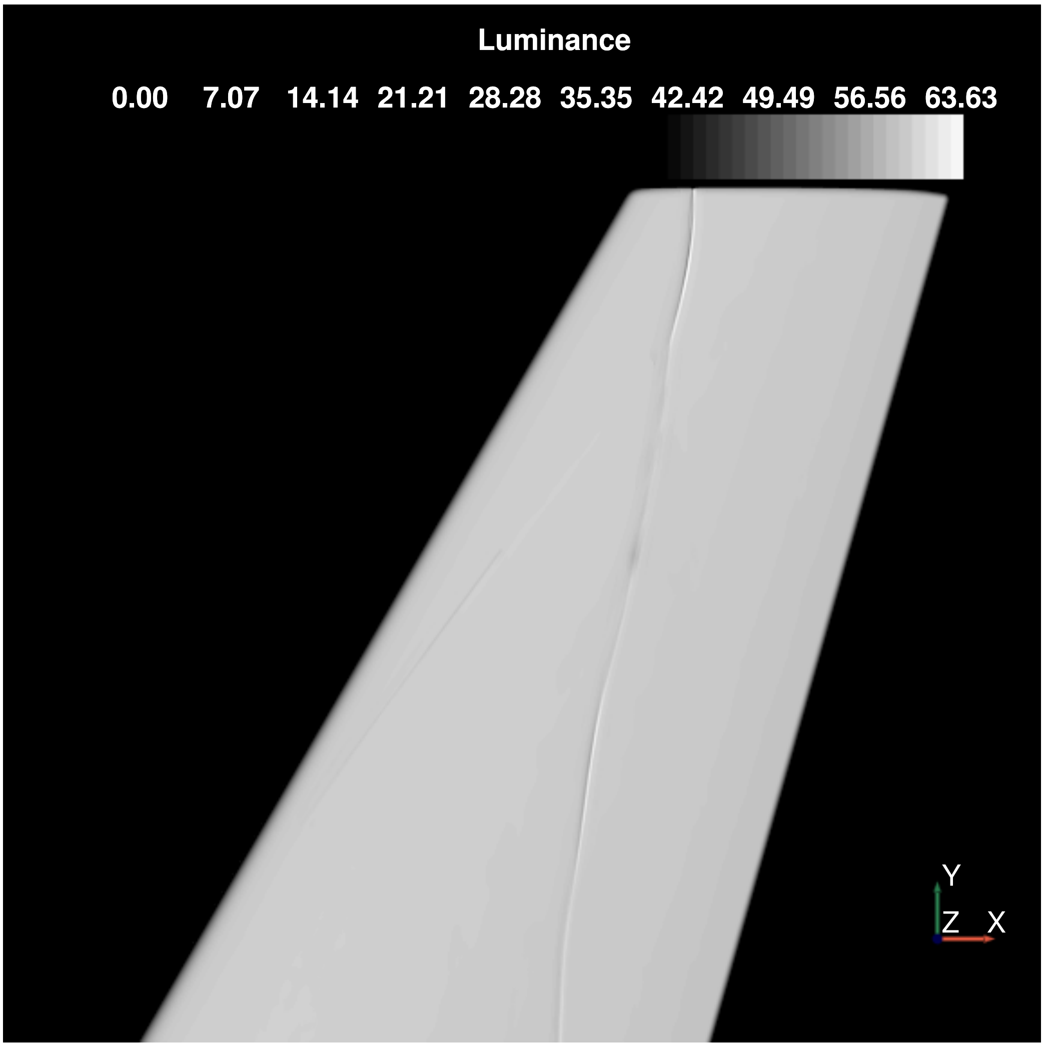


Figure 5.52: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $90°$ and an azimuth angle of $30°$.

tip seems to project a softer shadow in front of the stronger dark band of the shockshadow pattern, which spreads out further towards the leading edge as the sun moves forward. This visual effect is linked to the curved shape of the shock wave in this region, in comparison to the more oblique and near normal shapes closer to the root region. These curves in the surface of discontinuity refract light in an intrincate way and thus a different illumination pattern is formed.

As foreseen through the shock wave orientation measurements, the sun azimuth angle of $35°$ is determined to be the best solution in terms of achieving near grazing incidence of light throughout the whole lambda patterned shock wave. At this sun orientation, a high contrast and concentrated shockshadow feature is generated on the resulting image, at least for the stronger trailing edge strand of the shock wave, and luminance values at the caustic or bright band are maximised (see colour bars).

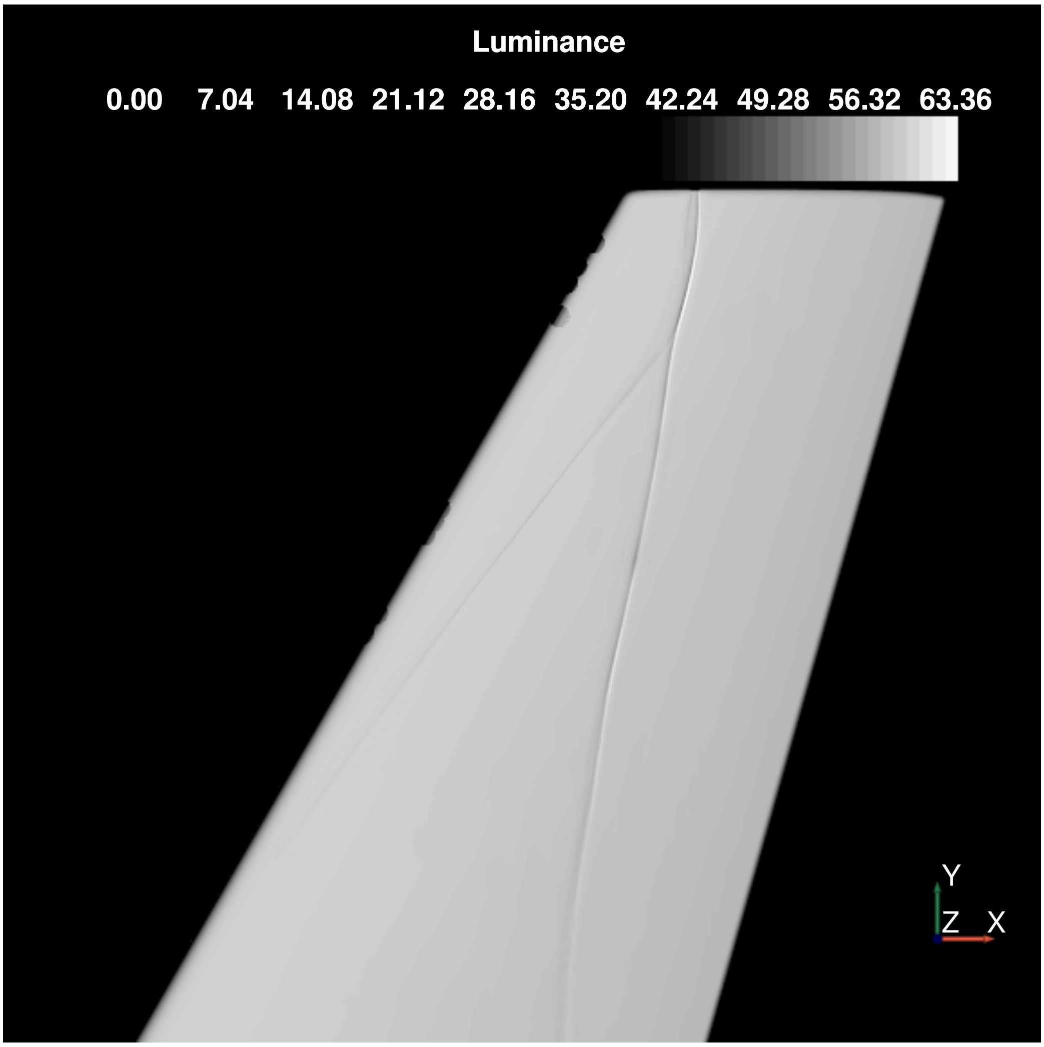


Figure 5.53: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $90°$ and an azimuth angle of $35°$.

5.2.3.3 Optimal elevation of the sun

With this azimuth fixed at $35°$, the sun elevation is now varied between $90°$ and $50°$ to confirm that high angles are indeed optimal in terms of shockshadow visualisation, as established in [16]. Figures 5.55, 5.56, 5.57 and 5.58 show the luminance distributions rendered from these orientations of the directional light source. The caustic photon map is again directly visualised at the diffuse wing surface and the previous tuning of the illumination, visualisation and radiance estimation stages are still used.

As seen in Figure 5.55, as the sun is further inclined towards the wing tip, the sharpness of the whole shockshadow is reduced. This is even more evident in Figure 5.56, where the shockshadow at the wing inboard region starts to disappear and the shockshadow due to the

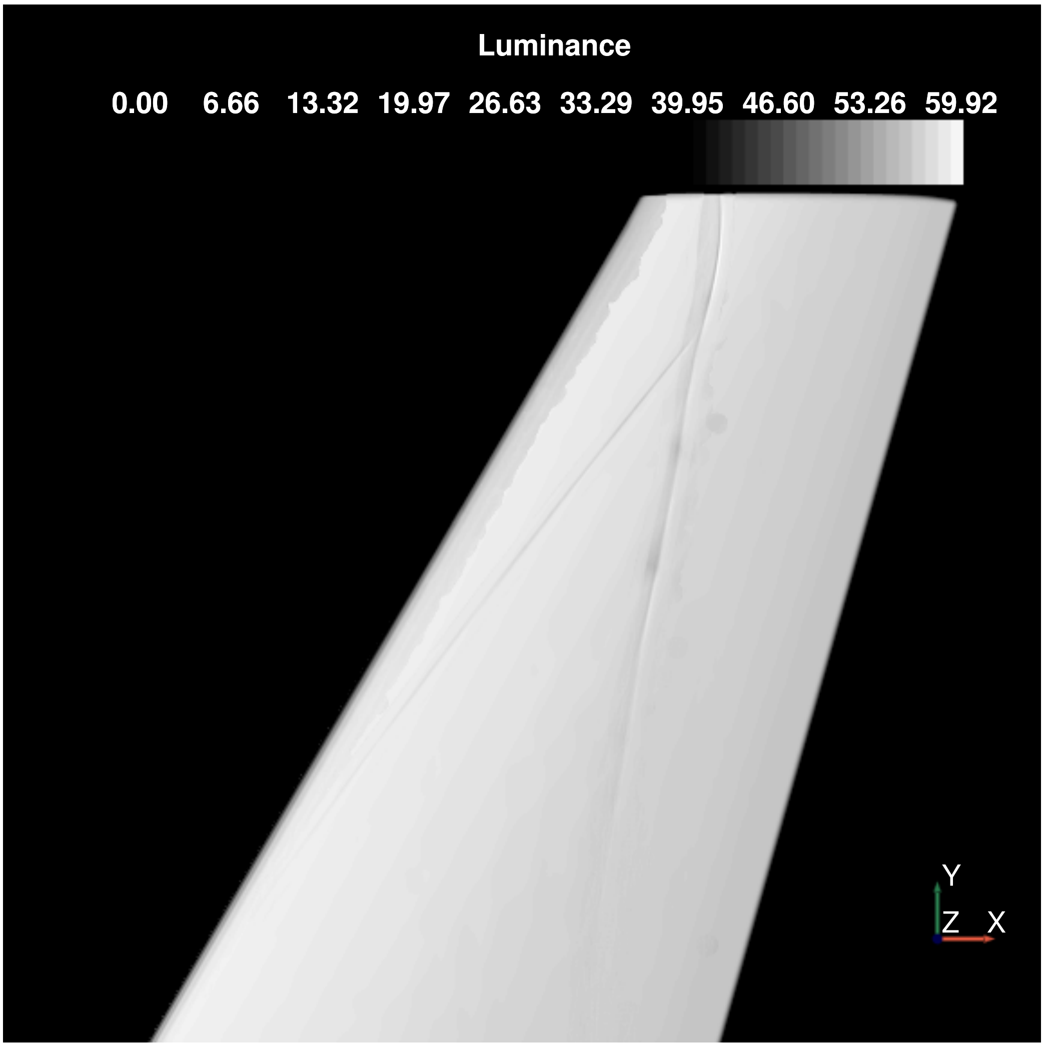


Figure 5.54: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $90°$ and an azimuth angle of $40°$.

leading edge strand of the shock wave is only a smoothed shadow at the vicinity of the point where the two shock waves strands merge. Figure 5.57 then clearly depicts the soft shadow at the wing outboard region, which forms a different shape as the sun is lowered further down towards the horizon. The figure also demonstrates that the overall illumination of the wing surface decreases, as it is evident over the trailing edge. The change of shapes as the light source direction is modified is actually true for the whole shockshadow pattern and, as seen in Figure 5.58, there is a striking difference in the feature at the tip region if compared to when the sun is positioned straight above the wing surface. These figures show that the shockshadow is projected in the direction of illumination. Therefore, the visual pattern formed over the wing surface is dependent on both the azimuth and elevation angles of the sun, although in the azimuth range analysed in the previous section these changes in shape were less noticeable.

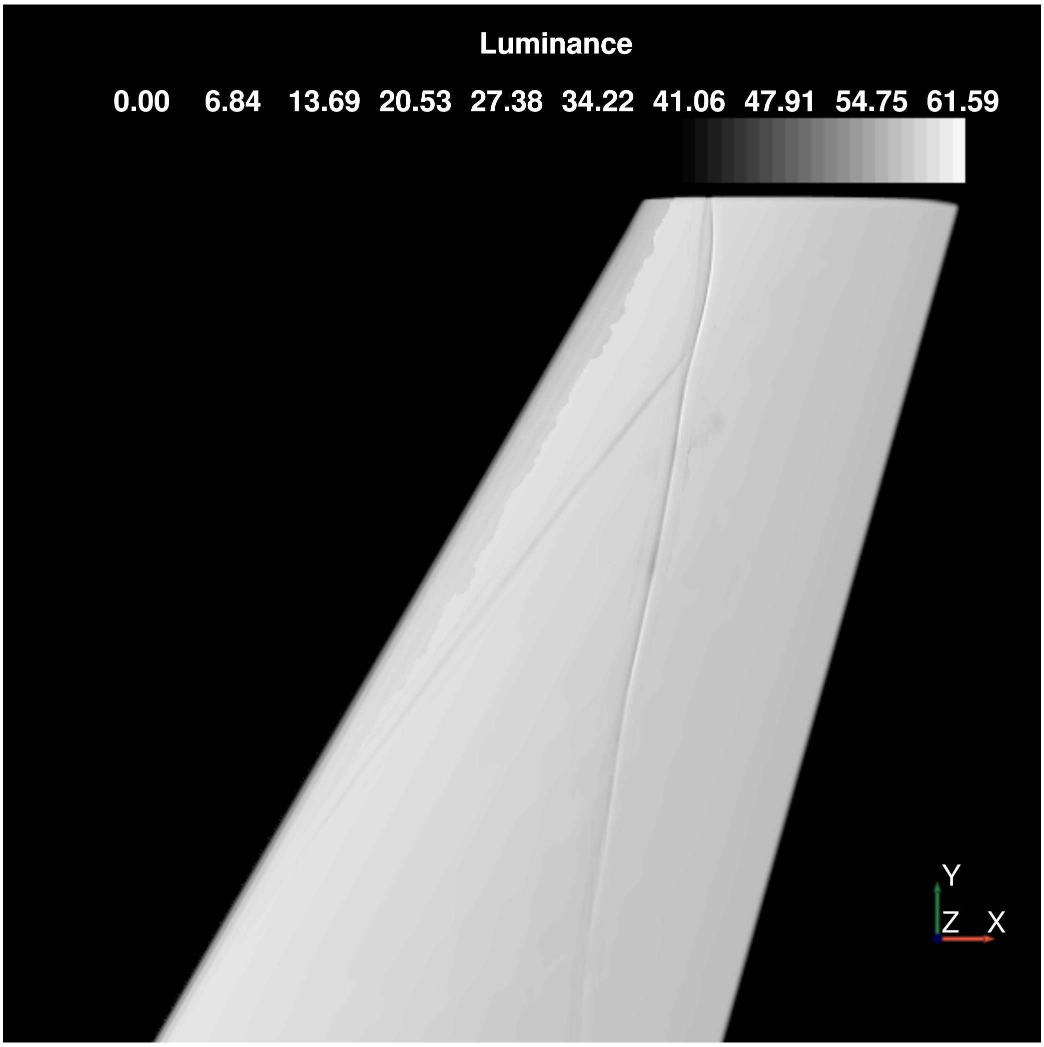


Figure 5.55: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $80°$ and an azimuth angle of $35°$.

Through these images, it is then determined that the clearest shadow and caustics both arise at the sun elevation angle of $90°$, as observed during the flight experiments. The further away that the direction of the light source is from the wing surface average normal, the more the shockshadow images degrade in quality. The visual features become weaker, more sparse or even disappear, blending with the colour of the wing surface. The whole illumination level of the scene decreases, hindering the shockshadow visualisation, and the shape of this shockshadow is suggested to be less correlated to the shape of the shock wave itself, but rather a projection in the direction of illumination.

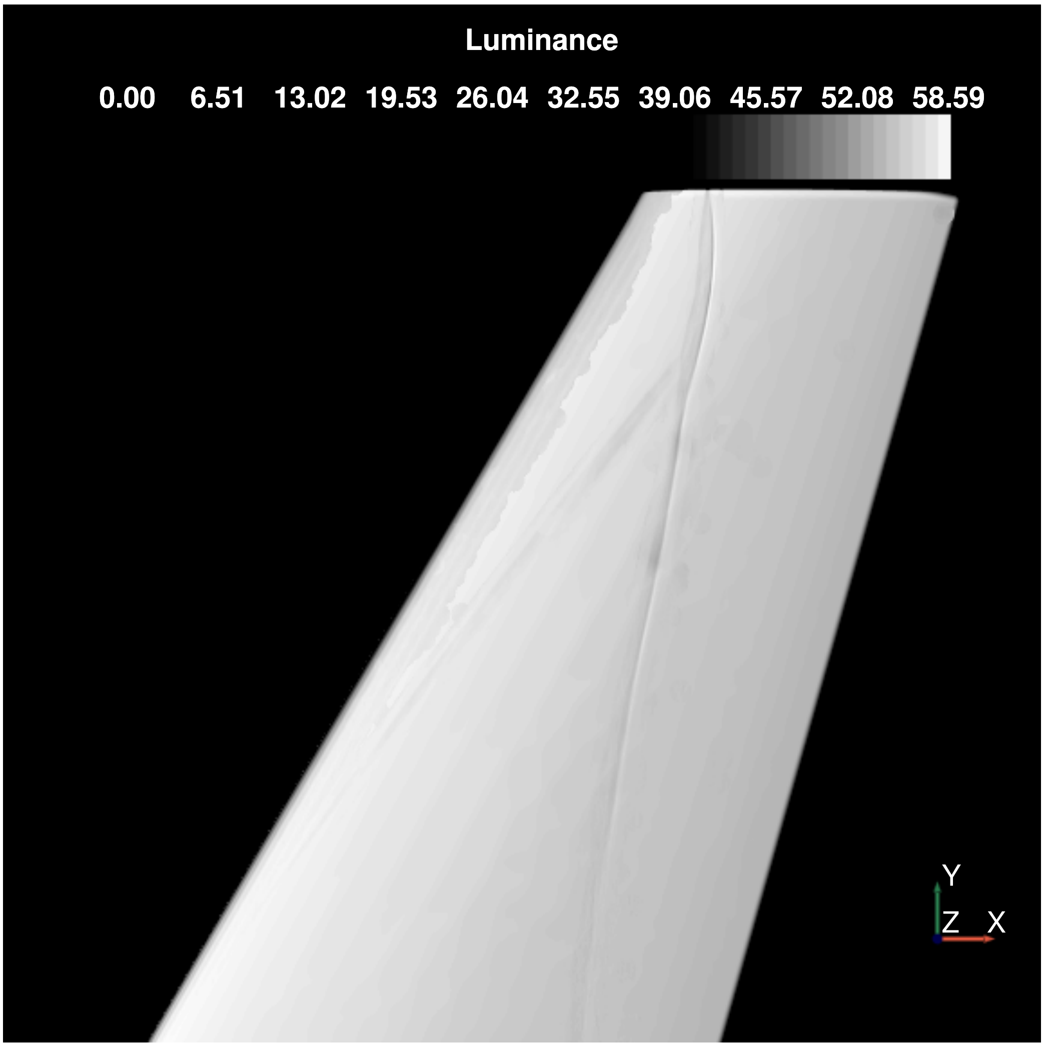


Figure 5.56: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of 70° and an azimuth angle of 35°.

5.2.4 Nonlinear Visualisation of a Shockshadow

All the results presented so far have used direct caustic rendering, i.e. radiance estimation directly from the photon map assuming linear propagation of light after reflection and a perfectly diffuse reflective wing surface. As explained in Section 5.2.2, this approximation removes the visibility dependence on the direction of the reflected light and, instead, only forward nonlinear ray tracing is performed. Thus rendering occurs over the wing surface itself, without taking into account an actual viewing system.

This approximation is more than reasonable for common images rendered in computer graphics. For example, classic models in the field include illumination patterns formed over a table surface where there is either a glass of whisky or a metallic ring, and on the bottom surface of swimming

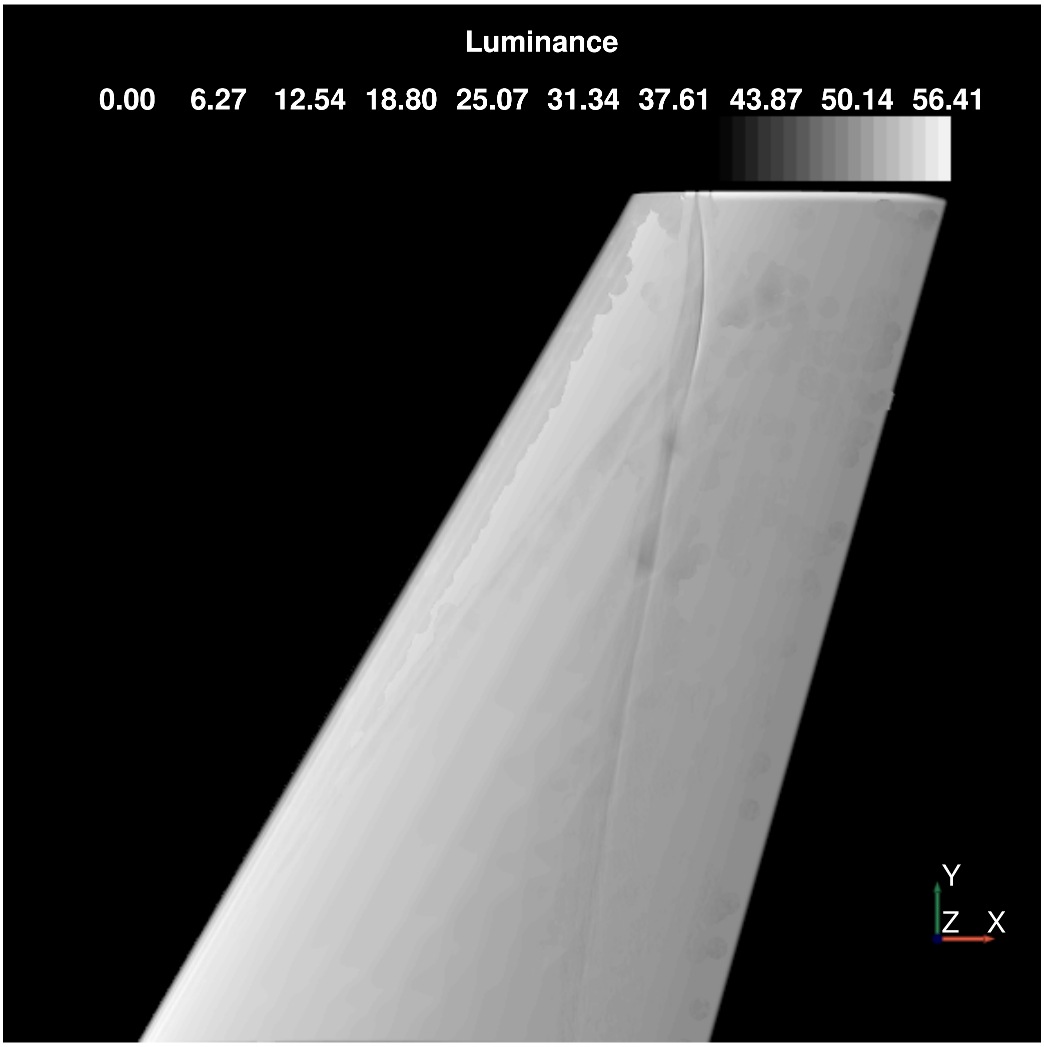


Figure 5.57: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $60°$ and an azimuth angle of $35°$.

pools. In the ring case, a caustic appears inside the ring due to focusing of light via reflection at the highly specular metallic surface and visibility happens in homogeneous air. Regarding the glass, the camera is usually positioned far from it and the caustic that appears near its base, due to multiple refractions through the glass and the whisky, is visible on the table again through homogeneous air outside of these two different optical media. For the swimming pool, images are usually simulated with the camera positioned underwater in order to display the effects, and the optical medium during visualisation, the water, is again homogeneous.

In the visualisation of the shockshadow phenomenon, refraction occurs twice: once when light travels from the sun, crosses the shock wave system and reaches the wing surface; and again when the reflected light crosses the surface of discontinuity again before reaching the eyes or

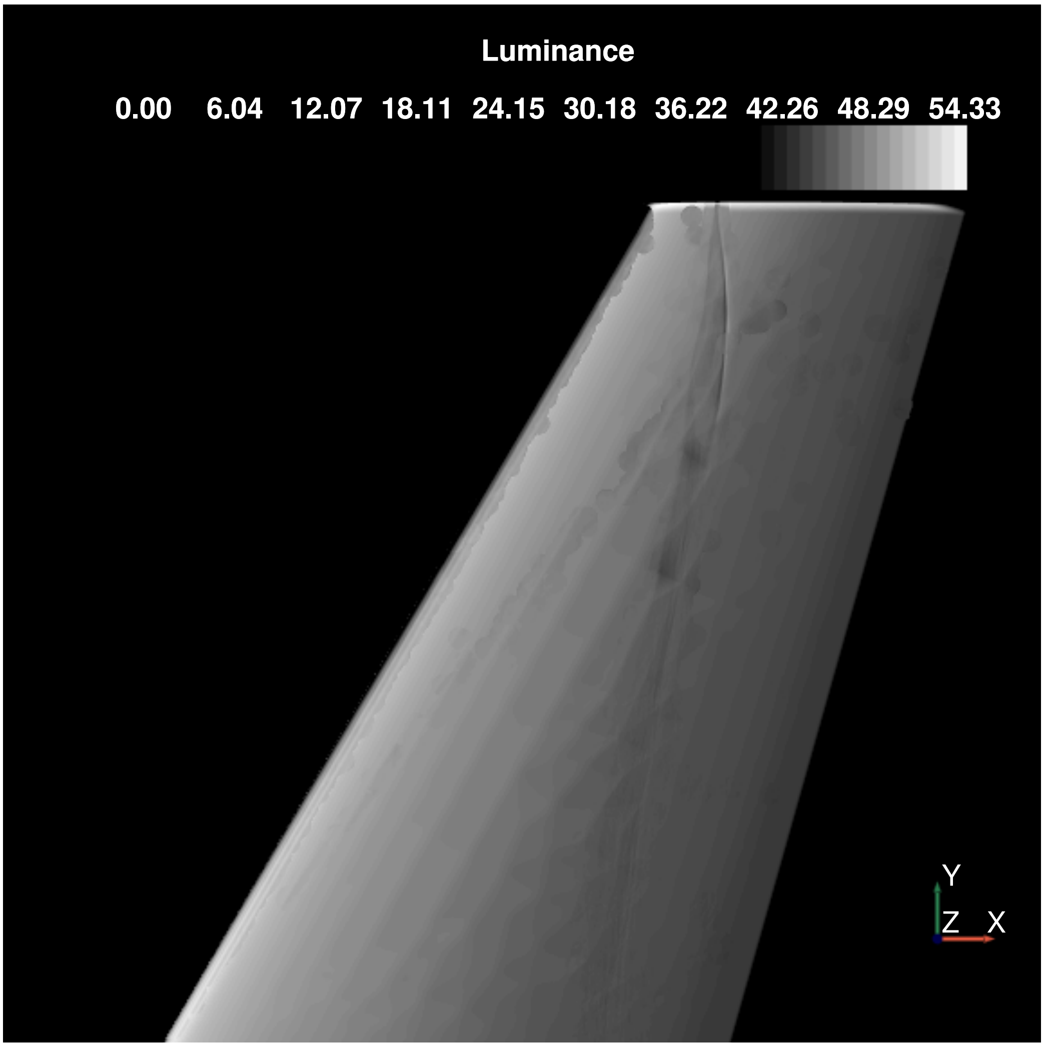


Figure 5.58: Luminance distribution over the wing surface rendered from the illumination of the sun at an elevation angle of $50°$ and an azimuth angle of $35°$.

camera of an observer. It is as if the caustic formed due to the glass of whisky was actually visualised through the glass and liquid, thus resulting in a distorted illumination pattern over the table. In order to account for this effect in the current investigation, the virtual camera model is then appropriately positioned from an out-of-the-window perspective, as depicted in Figure 5.59, and backward *nonlinear* ray tracing via solution of the ray equation is performed. The points that the rays that are emitted from the eye of the viewing system intersect the wing surface are the new query points used in the radiance estimation, thus linking the illumination and visibility passes.

A more accurate simulation of an actual photograph of the shadow and caustic features taken in inhomogeneous air is then generated considering: the previously determined optimal

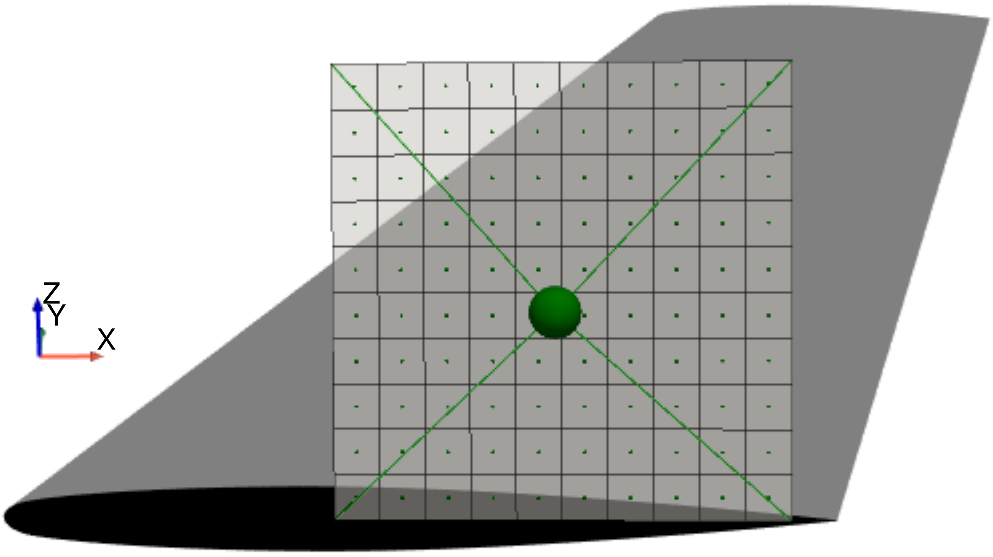


Figure 5.59: Positioning of the virtual camera model to perform a nonlinear visualisation of the shockshadow from an out-of-the-window perspective.

sun orientation of $90°$ elevation and $35°$ azimuth with respect to the wing quarter-chord line; again the Epanechnikov kernel with same radii extrema and bandwidth mapped per searching window; $5M$ photons nonlinearly traced from the source; and an image plane with 1000×1000 resolution. The wing material is still assumed to be lambertian. Figure 5.60 presents this simulated picture, viewed through the shock wave. Figure 5.61, for comparison, shows the picture from the same perspective and under the same illumination conditions, but generated via direct caustic rendering (with the same tuning) considering linear light propagation during visualisation, as in the previous section.

Neither Figure 5.60 nor Figure 5.61 have been processed for contrast enhancement, in order to preserve their original or raw characteristics. At this camera position, the differences between the two images are minimal and rather related to the Python libraries used to plot these two images. While it was easier to employ VTK to render the three-dimensional image over the wing surface for the case of homogeneous air and explore different two-dimensional perspectives, the Pillow library [208] was more appropriate to load the already 2D image rendered on the image plane of the virtual camera. This is the main reason behind the computational instead of physical differing shades of white. The same explanation applies for the apparent clearer shockshadow of the front strand of the shock wave on the image generated considering the influence of the shock wave on the viewing system. Similar contrast could be achieved in inhomogeneous air by simply further processing the image, as in Figure 5.53. Nevertheless,

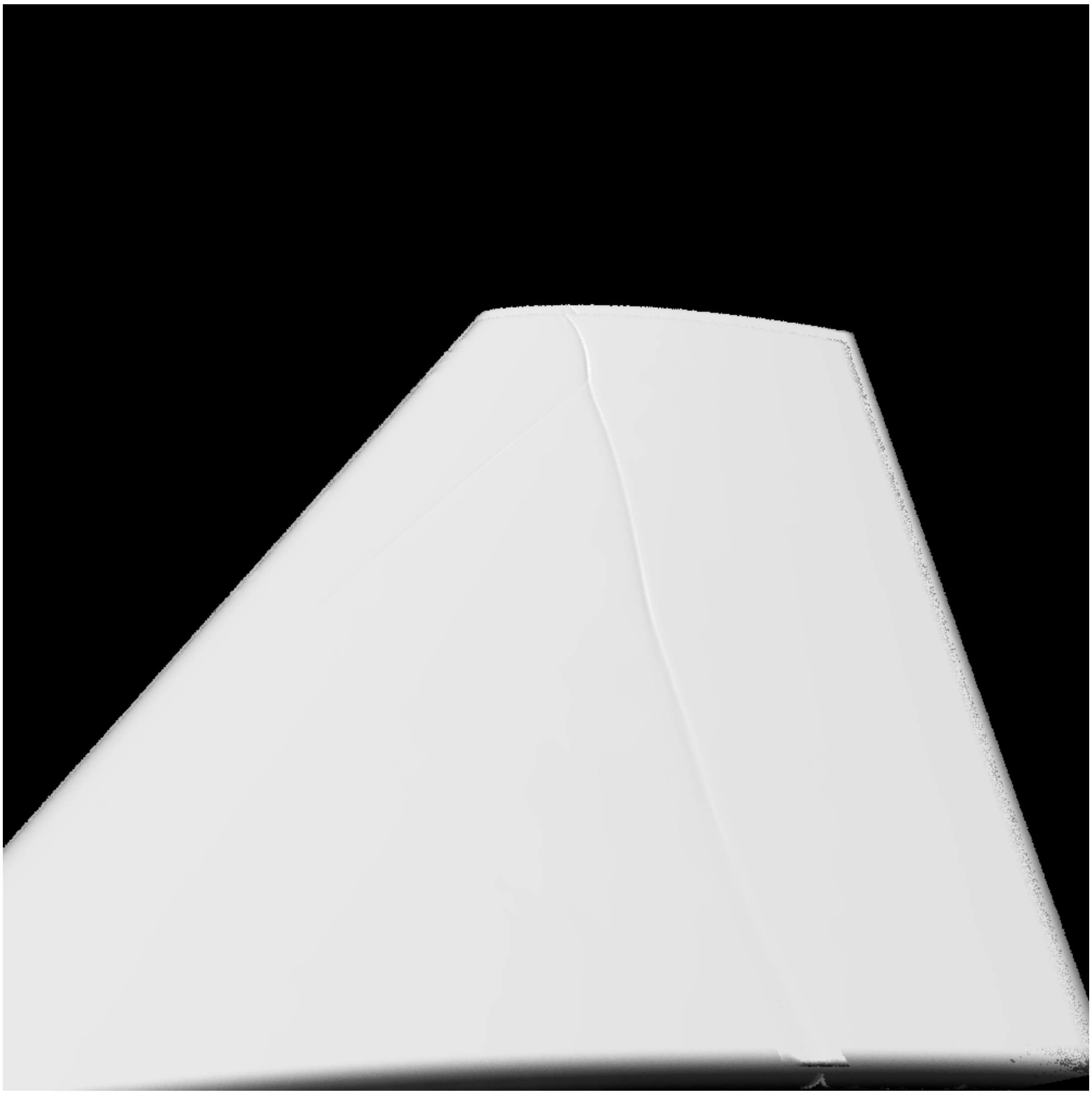


Figure 5.60: Simulation of a photograph of the shockshadow in *inhomogeneous* air from the out-of-the-window view.

the characteristics of the shockshadow are unchanged irrespective of the consideration of nonlinearities during visualisation at this camera angle.

It is expected that, at different camera orientations with respect to the surface of discontinuity, grazing emergence could also be achieved after reflection, during the visibility pass. This observation approximately aligned with parts of the shock wave could then render a discontinuous shockshadow pattern, where the secondary refractions during visualisation alter its real position in some regions, similar to the apparent depth of objects underwater seen from outside

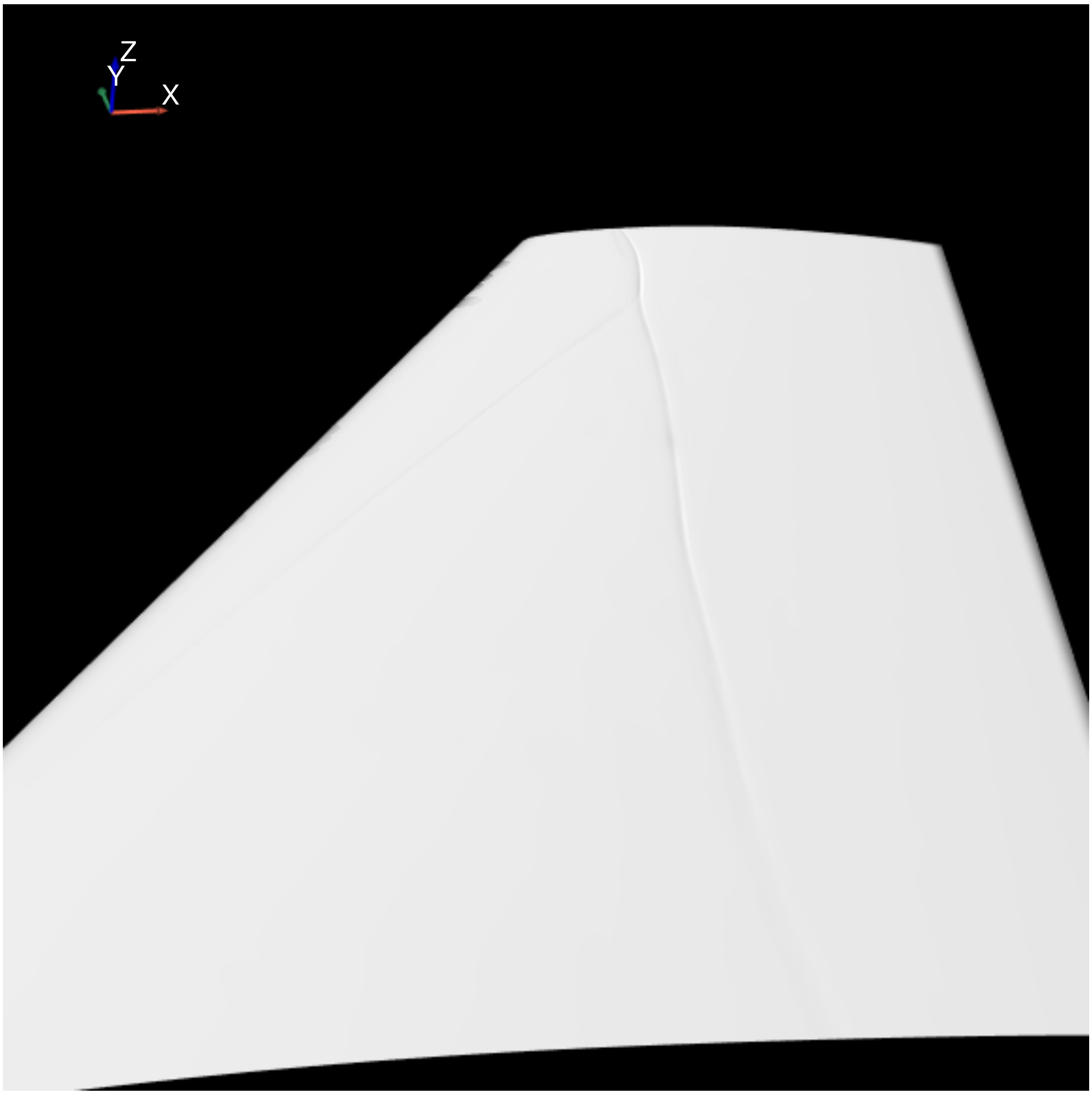


Figure 5.61: Simulation of a photograph of the shockshadow in *homogeneous* air from the out-of-the-window view.

this medium. Other possible event that could arise from a different camera position would be: the imaging of the shock wave as a schlieren feature, through which the background (sky or aircraft fuselage, depending on the point of view) appears distorted due to its presence; or, as in Figure 1.15(b), a shadowgraphy-like visualisation, where other shadows and/or caustics are visible in the plane perpendicular to the wing surface, apart from the shockshadow on the surface itself.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this research, a computational framework to construct an accurate simulation of the behaviour of light within a compressible high-speed flow has been successfully designed. This is the main knowledge contribution of this work. The framework was constructed from the domain of Geometrical Optics and has a fully tested working implementation. Images that exhibit the sought after visual effects of direct natural and artificial shadowgraphy due to light interacting with shock waves have been produced. This virtual experimental environment is the initial step towards establishing a relationship between aerodynamic and light phenomena and subsequently confirming the feasibility of extracting shock wave knowledge from observed shockshadows. The aspects involved in the real image formation and visualisation processes were appropriately modelled, as dictated by computational flow imaging. These included: all of the degrees of freedom of the three-dimensional reality; a higher fidelity aerodynamic flow; the wavelength dependence of the radiant energy emitted from the sun; the nonlinear paths followed by light rays in inhomogeneous optical media; the actual reflection of light from solid surfaces and respective materials; the further scattering of light during additional propagation after reflection; and the inclusion of a recording device to capture the whole scene. The modification of the virtual environment to incorporate other experimental elements, such as lens systems included in indirect shadowgraphy or suitable backgrounds for schlieren imaging, is also feasible with the framework.

To the best knowledge of the author, this is the first published research that reproduces, computationally, all of the elements of the in-flight shock wave shadow formation and visualisation. The majority of previous studies in the field of computational flow imaging pursued only the modelling of the optical setup of traditional flow visualisation techniques. Other purely experimental tests only focused on the physical aspects of the shockshadow, such as the practical conditions that make it visible during flight and how its observation can help the analyses of for examples aircraft aeroelasticity. The shockshadow phenomenon has frequently been suggested as a natural resource that could characterise compressible flows, by making shock waves on typical full-scale commercial aeroplanes detectable. However, an attempt to actually employ shockshadows as an aerodynamic data analysis tool has never been made. One of the barriers to the development of such a tool has been the requirement of a physically realistic simulation framework for the in-flight shockshadow phenomenon. This research addresses this issue by producing a key enabling technology and this new framework will facilitate the future development of an analysis tool.

The development of the new simulation framework considered the intended aerospace application throughout, but was constructed primarily from the perspective of computer graphics. The main goal of recreating realistic images of shockshadows inside a computational environment with a reasonable balance between accuracy and efficiency has been achieved by the new framework. It has been shown that one of the major difficulties in such a framework is the exchange of data between the different components of the simulation pipeline, mainly from fluid flow to light simulation. The initial implementation used C++ and the CGAL library because they are widely used in computer graphics and hence seemed the most appropriate foundation in which to develop the light simulation framework. However, it was found that CGAL had limited capability to implement the processes needed and lacked the flexibility to handle the diversity of CFD meshes and data structures. The translation from C++ with CGAL to the more user-friendly and practical Python programming language with VTK was shown to give greater flexibility. This library is the backbone of the software ParaView, widely used in CFD for preprocessing, postprocessing and, in general, for data visualisation purposes. This means that multiple algorithms used in the CFD field, and also required in the light simulation, are immediately provided as components of VTK.

The verification and validation assessments have provided valuable insights into the developed resource for practical aerodynamic analysis. It was demonstrated in the prism simulation that the nonlinear ray tracing technique via solution of the ray equation can be used to accurately compute the propagation of monochromatic light rays. This shows that the future combination of multiple ranges of the spectrum, traced individually, can account for wavelength dependence, as well as the appearance of chromatic aberrations or colour distortions on the final shockshadow images. This potential for spectrum isolation was utilised in the supersonic cylinder simulation to include a laser light source, which demonstrated the reproduction of a direct shadowgraphy experiment. The computational shadowgraph revealed important aspects of the simulation of geometrical displacements of light. First, it corroborated the unavoidable direct dependence on the underlying CFD solution. Errors of any type are propagated from the flow to the light simulation and, if the former does not accurately represent the shock wave formation, the latter invariably will generate an inaccurate shockshadow. This shadowgraphy setup also demonstrated the necessity of contrast calculation to detect the expected visual effects of dark and bright bands, since the light deflections and displacements resulting from compressible aerodynamic flow are small in magnitude.

A number of important findings were made from the transonic pseudo-three-dimensional wing (stacked 2D NACA 0012 aerofoil solutions) test case:

- A limitation similar to the previous contrast dependence of the shadowgraphs exist for the shockshadow simulation. The shockshadow was not visible in this case if only the wing surface was regarded as a light sensor. Either light had to be allowed to propagate further from the location of reflection, or ray tracing had to be combined with other computer graphics techniques to truly model global illumination. A combination of both techniques would improve the quality of the images.
- Positioning the recording plane at a distance from the region of interest reduced the exposure to which it was subjected, but the photon displacement was magnified the further away the recording plane was positioned. A balance therefore must be established to result in amplified image features without caustic effects becoming too blurred. The wing surface is the screen of the natural direct shadowgraphy and, as in the ground-based experiments, both the light source and the screen have to be appropriately positioned in order to obtain sharp image features. Since the wing itself cannot be moved with respect to the shock wave, in-flight shockshadow observation therefore relies on the characteristics and orientation of the sun and on the process through which images are formed, inside the human eye or inside camera systems.
- The nonlinear characteristic of the light and shock wave interaction phenomenon produces an indirect illumination artifact. This indicated that the photon mapping density-estimation concept was a suitable candidate to complement the sampling-based ray tracing technique. This method solved the same equation that underpins most of the rendering algorithms. However, it was based on a statistical density estimation to compute the illumination distribution. This is a fundamental requirement for a complete light simulation framework and to achieve a representative synthetic shockshadow.
- The creation of a shadowgraph image indicated that neither the illumination nor the visualisation sampling stages were adequate for an accurate image reconstruction. However, the simulated shadowgraph indicated that the tracing of an ever-increasing

number of photons would allow the reduction of the density estimation bandwidth. This distribution would then convert to an accurate image.

- A grazing incidence of light, where rays roughly aligned with the shock wave curvature, produces stronger shockshadows with darker shadows and brighter caustics.

Further findings were made from the transonic Onera M6-wing test case:

- The photon gathering aspect of photon mapping is fundamental to producing a shockshadow with minimal modelling approximations. The lambda patterned shock wave provided an essential test bed to understand how to appropriately employ the statistical density estimator in photon gathering. By rendering the caustic photon map directly, the trade-off between variance and bias became noticeable. It was found that a compromise, universal, bandwidth size could not be established, and the illumination distribution at one of the reference spanwise station was either overfitted and contained too much noise or it was underfitted and the shockshadow was completely suppressed. This was not the only challenge, since when all the spanwise stations were rendered the second trade-off between under and oversmoothing became evident. Different regions of the wing can have contrary requirements of bandwidth size, due to the variations in the shock wave strength. It is clear that it is very difficult to meet these requirements simultaneously and these results are likely to be applicable to most shock wave systems encountered in real aircraft.
- Weaker sections of shock waves produce minimal photon displacement, and therefore these regions require large bandwidths to estimate the radiance on the reflective surface and smooth out the inherent noise of the techniques. On the other hand, regions containing stronger shock waves can be rendered via a small bandwidth to generate sharp features.
- Analysis of the source of noise in the illumination distributions could not be attributed to the image resolution, or to the CFD data, nor to the presence of discontinuities in the inhomogeneous optical medium. It was found to be associated with the computer graphics algorithms used.
- Adaptive bandwidth mapping with suitable image filters resolve these trade-off issues and enable the generation of realistic shockshadows. For this case, determining the bandwidth size based on the strength of the shock wave and also weighting the photon contribution based on their distance from the query point where the illumination was being reconstructed allowed the minimum number of both illumination and visibility samples to be determined.

- The optimal orientation of the sun with respect to the wing quarter-chord line to achieve strong shockshadow visualisation was found to be the grazing incidence of light, confirming flight test data available in the literature.

6.2 Future Work

The studies presented in this research constitute a significant advance in the capability to realistically simulate light propagation, refraction, reflection and image reconstruction in a three-dimensional compressible flow with shock waves. However, the Onera M6-wing test case is not truly representative of a commercial aircraft geometry and hence all the issues associated with the in-flight shockshadow phenomenon have not yet been fully explored. The XRF-1 model is a civil aircraft research configuration developed and shared by Airbus for multi-disciplinary studies [209]. The model contains the wing, fuselage and tail of the aircraft and standard hybrid unstructured meshes are provided for simulating unsteady, viscous and turbulent flow around this configuration. In this research, the flow has been simulated for this case on the standard mesh with adaptation using the DLR TAU-Code software system [210]. However the light propagation process could not be completed, because difficulties were encountered in converting between the TAU and VTK data structures.

The post-processing of the CFD used the open-source research software Cassiopee [211]. These Python modules developed by ONERA can directly manipulate the CFD simulation grid and solution via a CFD General Notation System (CGNS) data tree and perform the conversion of all cells to tetrahedra, using multicore parallel processing. Unfortunately, this process resulted in tetrahedra meshes with holes in them that could not be distinguished from a solid surface. This was linked to the specific standard mesh for the XRF-1 as the conversion process worked on other TAU meshes. If this stage had been completed successfully, then the cells and corresponding air density values stored at their vertices could be selected within specified regions or subdomains of interest for domain isolation. In fact, CGNS is now widely used in the CFD community for exchanging data in a simulation pipeline [211] and this issue was therefore unexpected. Thus it is recommended that this issue is investigated further to try to ascertain the reason for this processing failure for this mesh. Once this has been corrected, the developed simulation framework as shown for the Onera M6 wing should be applied to the XRF1 test case.

It is then recommended that the spectral light source model of the sun is updated to one of the composite extraterrestrial spectra available from measurement [107, 108]. A terrestrial irradiance distribution would then be obtained by taking into account the attenuation of the solar radiation due to a cloudless and clear sky, given its composition [105]. Alternatively, one of the standard relative power distributions representative of light sources that are commonly employed in colorimetry, the so-called CIE Standard Illuminants due to the Commission Internationale de l'Éclairage, could be used to represent sunlight [100]. Although sunlight depends on many astronomical, temporal, geographic, air and atmospheric conditions, and the process of constructing a real light source that represents the $CIE\ D_{65}$ is very complex, it is commonly used as a reference to represent average daylight arriving at the surface of the earth [100]. Other spectral power distributions of outdoor light sources are also available [212].

To complement this wavelength dependence of light scattering an appropriate spectral reflection model should be used. The current implementation only considers the wavelength indirectly in the integration of the spectral irradiance from the sun and in the Gladstone-Dale relation to compute the refractive index of air. Sandford and Robertson [213] presented a valuable study into the reflective properties of airframe surface coatings or paints. The study considered a significant database of bidirectional reflectance distribution function measured from samples of panels from over fifty operational aeroplane models in the infrared (IR) and visible ranges of the spectrum. From these samples, some illustrative paints were fully characterised in terms of their colours, a description of their surfaces, their solar absorption and reflective characteristics. Most samples exhibited reflection that was neither perfect diffuse nor perfect specular. Both the composition of the surface material, aluminium, and the surface finish (paint, condition, roughness and history) were taken into account. Sandford and Robertson [213] then also developed a semi-empirical BRDF model, consistent with this data, that could be used to describe the reflective properties of aircraft paint and could be integrated into the light simulation framework. Effects of the surface roughness, the thickness of the layer of paint and of ageing and weather exposure of the surface are embedded in the model, for example [213].

In the same way that different types of light sources were used to characterise the reflectance and absorption of samples in [213], Fernandez et al. [214] performed similar measurements utilising a laser to determine the BRDFs of a sandblasted, a stock and a white painted sample of aluminium surface. The metal was used as the base material for all the paints due to its wide structural application in fluid flow experimental facilities and materials [214]. These were compared against a reference white perfectly diffuse reflector. From the results presented in polar plots, it was determined that while a treated aluminium surface reflects light more diffusely, the untreated surface behaves more specularly and the surface painted in white is a lambertian reflector for lower incidence angles and causes retro-reflection for higher angles. Similarly to the Robertson-Sandford BRDF model [213], the polar plots in [214] could be used to define the surface model [214], such as the wing surface in the shockshadow simulation.

Since both the light source and the reflective surface would then have spectral properties, the nonlinear ray tracing could also embed this wavelength dependency. As suggested by Zhdanov et al. [215], this could be accomplished through three different ways: by ray tracing independently a certain number of monochromatic scenes (light source, refractive index medium and reflective surface) and combining the monochromatic images a posteriori; ray tracing simultaneously a number of monochromatic light rays that will then generate a polychromatic image; or ray tracing polychromatic light rays that are decomposed when interacting with dispersive surfaces. Thomas [216] proposed a technique similar to the last approach. These approaches to solve the rendering equation in the spectral space are said to provide a more accurate colour rendering, as opposed to distortion, fringing or clipping issues that often occur when working in the RGB space [215, 216], for example. Analogously, to simulate IR effects, including caustics, Boudet et al. [217] used a spectral photon mapping technique. In this adaptation of the traditional algorithm, each photon was assumed to also carry information regarding weights defined for each band of the spectrum. These weights were then used when computing simultaneously all the radiance bands using the rendering equation. Spectral ray tracing and spectral photon mapping could then be coupled in a more advanced implementation that considers the wavelength dependence of the light source and of the surface reflection.

A factor that was explored in the initial implementation of the simulation framework is image contrast computation. This was performed via subtraction of a neutral image, rendered considering a homogeneous optical medium, from the actual disturbed image generated in the compressible flow. In the final implementation only the luminance quantity has been used for rendering. Both the physical and the perceived types of contrast, which quantify luminance and brightness differences between achromatic patches respectively [100], should be further evaluated to potentially extract new information from the shockshadows. The luminance contrasts can be defined by the Michelson, Weber or a simple ratio formula involving the feature to be visualised and a background [100, 218]. Techniques such as histogram equalisation [219] could also be applied to enhance the contrast of the resulting image. Alternatively, the shockshadows currently rendered as radiance values in the sRGB or luminance values in the CIE XYZ colour spaces could be converted to a colour appearance model, such as the CIECAM02 (recommended for digital colour imaging) [100, 220, 221]. These models are responsible for describing additional attributes of achromatic colours, e.g. lightness or the brightness of a patch relative to another that appears to be white. This attribute and others such as chroma, hue and brightness itself, are more closely related to the human visual system and they complement colorimetry in the task of specifying a stimulus. Because they are relative, the environment that the stimulus (shockshadow) is viewed in, including the proximal field (wing surface), the background (sky) and the surround (aircraft interior) are also taken into account in these models [100]. The viewing conditions are then important when dealing with colour appearance phenomena, such as simultaneous contrast and crispening, which affect the lightness perception of an observer [100]. In these two phenomena the proximal field and/or background alter the perception of coloured patches that are identical and that have identical colour differences with respect to another patch in the vicinity, and this would most likely occur in the shockshadow [100].

All of these modifications may degrade the computational efficiency of the framework. The implementation already supports the running of the simulations on a High Performance Computer in parallel, where a single batch of light rays to be traced is divided between multiple processors and run simultaneously. Nevertheless, the algorithms would most likely benefit from further acceleration. Two ways to achieve faster processing times would then be, for example, to either employ message passing interface (MPI) to achieve real parallelism on multiple cores of a computer or to use the graphics processing unit (GPU) instead of the central processing unit (CPU) to perform the main tasks of the rendering engine. These acceleration solutions are both available in Python as the popular mpi4py [222] and CUDA Toolkit [223], respectively. A GPU implementation would certainly facilitate the advancement of the framework towards the generation of shockshadow images at interactive rates [46].

Given the fact that the current implementation includes a virtual camera with all of the degrees of freedom in terms of positioning and orientation, a scene containing multiple cameras should be built. This would then allow the recording of shockshadows from different perspectives. The recorded shapes and positions, and thus distances through triangulation, of the shockshadow with respect to distinct wing features may then allow the extraction of three-dimensional information about the shock wave (photogrammetry), such as depth-perception and an inference of its exact position. In fact, the topic of multi-view stereo is currently trending in computer graphics and computer vision, encompassing algorithms for reconstruction or scanning of three-dimensional models from a large data set of images concerning the required geometrical object. Some available algorithms already take into account transparent and refractive objects [224–226] reconstruction from caustics images [227] or from background-oriented schlieren images [228], and even using the VTK library [229]. These studies strongly suggest that the shock wave reconstruction from multiple respective shockshadow images is possible.

References

[1] G. S. Settles.
Schlieren and shadowgraph techniques: visualizing phenomena in transparent media.
Springer Science & Business Media, 2001.

[2] P. Krehl.
History of Shock Waves.
In *Handbook of shock waves*, pages 1–142. Elsevier, 2001.

[3] E. Mach and J. Sommer.
Ueber die fortpflanzungsgeschwindigkeit von explosionsschallwellen.
1877.

[4] E. Mach and J. Wentzel.
Ein beitrag zur mechanik der explosionen.
Annalen der Physik, 262(12):628–640, 1885.

[5] E. Mach and J. von Weltrubsky.
Über die form der funkenwellen.
Sitzungsber Akad Wiss Wien (II. Abth.), 78:551–560, 1878.

[6] J. Jamin.
Description d'un nouvel appareil de recherches, fondé sur les interférences.
CR Acad. Sci. Paris, 42:482–485, 1856.

[7] L. Zehnder.
A new interferometer.
Ein neuer interferenzrefraktor. Zeitschrift fur Instrumentenkunde, 11:275–285, 1891.

[8] L. Mach.
Ueber einen interferenzrefraktor.
Zeitschrift für Instrumentenkunde, 12(3):89, 1892.

[9] P. Krehl and S. Engemann.
August toepler—the first who visualized shock waves.
Shock Waves, 5(1):1–18, 1995.

[10] A. Toepler.
Beobachtungen nach einer neuen optischen methode. bonn.
1864.

[11] V. Dvořák.
Über eine neue einfache art der schlierenbeobachtung.
Annalen der Physik, 245(3):502–511, 1880.

[12] L. Prandtl.
Zur theorie des verdichtungsstosses.
Z. ges. Turbinenw, 3:341–245, 1906.

[13] E. Hecht.
Optics, 5e.
Pearson Education India, 2017.

[14] M. J. Abzug and E. E. Larrabee.
Airplane stability and control: a history of the technologies that made aviation possible, volume 14.
Cambridge University Press, 2005.

[15] C. L. Johnson.
Development of the Lockheed P-80A jet fighter airplane.
Journal of the Aeronautical Sciences, 14(12):659–679, 1947.

[16] D. F. Fisher, E. A. Haering, G. K. Noffz, and J. I. Aguilar.
Determination of sun angles for observations of shock waves on a transport aircraft.
NASA/TM-1998-206551, 1998.

[17] R. Croman.
Shock wave formation in transonic flight.
https://www.youtube.com/watch?v=HekbC6Pl4_Y, February 2016.
Accessed: 2021-11-19.

[18] T. Austin.
Shockwave on A320 wing.
https://www.youtube.com/watch?v=mu8Ifv-l0Ek, August 2019.
Accessed: 2021-11-19.

[19] bnnnboy.
Shockwave shadowgram on transonic wing.
https://www.youtube.com/watch?v=Fv6fnNJ1-Bo&t=14s, January 2013.
Accessed: 2021-11-19.

[20] M. Hyötyniemi.
Shockwave shadows on Boeing 757-200 wing.
`https://www.youtube.com/watch?v=AHymgT2pWEQ`, June 2011.
Accessed: 2021-11-19.

[21] ntclemens.
Shockwave shadows on Boeing 757-200 wing.
`https://www.youtube.com/watch?v=g4MQqf8wPos`, November 2013.
Accessed: 2021-11-19.

[22] W. Merzkirch.
Flow visualization.
Elsevier, 2012.

[23] H. Kleine.
Flow visualization.
In *Handbook of shock waves*, pages 683–740. Elsevier, 2001.

[24] M. Born and E. Wolf.
Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.
Elsevier, 2013.

[25] W. D. Bachalo.
Methods for compressible flows.
In *Flow Visualization: Techniques and Examples*, pages 227–265. World Scientific, 2012.

[26] O. N. Stavroudis.
The mathematics of geometrical and physical optics: the k-function and its ramifications.
John Wiley & Sons, 2006.

[27] G S Settles.
Schlieren and shadowgraph imaging in the great outdoors.
In *Proceedings of the 2nd Pacific Symposium on Flow Visualization and Image Processing*. Honolulu, Pacific Center of Thermal-Fluids Engineering, Tokyo, 1999.

[28] M. Van Dyke.
An album of fluid motion, volume 176.
Parabolic Press Stanford, 1982.

[29] M. J. Hargather and G. S. Settles.
Natural-background-oriented schlieren imaging.
Experiments in fluids, 48(1):59–68, 2010.

[30] G. S. Settles and M. J. Hargather.
A review of recent developments in schlieren and shadowgraph techniques.
Measurement Science and Technology, 28(4):042001, 2017.

[31] J. T. Heineck, D. Banks, E. T. Schairer, E. A. Haering, and P. Bean.
Background oriented schlieren (bos) of a supersonic aircraft in flight.
AIAA flight testing conference, AIAA-2016-3356, 2016.

[32] J. T. Heineck, D. W. Banks, N. T. Smith, E. T. Schairer, P. S. Bean, and T. Robillos.
Background-oriented schlieren imaging of supersonic aircraft in flight.
AIAA Journal, 59(1):11–21, 2021.

[33] M. Kamlet.
Nasa captures first air-to-air images of supersonic shockwave interaction in flight.
`https://www.nasa.gov/centers/armstrong/features/supersonic-shockwave-interaction.html`, January 2019.
Accessed: 2020-05-05.

[34] G. A. Rathert and G. E. Cooper.
Visual observations of the shock wave in flight.
NACA-RM-A8C25, 1948.

[35] L. Larmore and F. F. Hall.
Optics for the airborne observer.
Optical Engineering, 9(3):090387, 1971.

[36] J. R. Crowder.
Flow visualization techniques applied to full-scale vehicles.
AIAA Atmospheric Flight Mechanics Conference, AIAA-1987-2421, 1987.

[37] W. Lokos, A. Lizotte, N. Lindsley, and R. Stauf.
Shock location dominated transonic flight loads on the active aeroelastic wing.
AIAA Atmospheric Flight Mechanics Conference and Exhibit, AIAA-2005-6315, 2005.

[38] T. M. Tauer, D. L. Kunz, and N. J. Lindsley.
Visualization of nonlinear aerodynamic phenomena during F-16 limit-cycle oscillations.
Journal of Aircraft, 53(3):865–870, 2016.

[39] G. Havener.
Computational flow imaging: Fundamentals and history.
25th AIAA Plasmadynamics and Lasers Conference, AIAA-1994-2615, 1994.

[40] Y. Tamura and K. Fujii.

Visualization for computational fluid dynamics and the comparison with experiments.
Flight Simulation Technologies Conference and Exhibit, AIAA-1990-3031, 1990.

[41] L. A. Yates.
Interferograms, schlieren, and shadowgraphs constructed from real-and ideal-gas, two- and three-dimensional computed flowfields.
NASA CR-190054, 1992.

[42] L. A. Yates.
Images constructed from computed flowfields.
NASA CR-194397, 1993.

[43] R. Samtaney, R. D. Morris, P. Cheeseman, V. Sunelyansky, D. Maluf, and D. Wolf.
Visualization, extraction and quantification of discontinuities in compressible flows.
2000.

[44] D. Brown, T. Cole, B. Peters, J. P. Wilson, and G. Havener.
Cfi-shadowgraph/schlieren photography for aerodynamic applications.
18th AIAA Aerospace Ground Testing Conference, AIAA-1994-2616, 1994.

[45] M. Sun.
Computer modeling of shadowgraph optical setup.
In *27th International Congress on High-Speed Photography and Photonics*, volume 6279, page 62790L. International Society for Optics and Photonics, 2007.

[46] C. Brownlee, V. Pegoraro, S. Shankar, P. McCormick, and C. D. Hansen.
Physically-based interactive flow visualization based on schlieren and interferometry experimental techniques.
IEEE transactions on visualization and computer graphics, 17(11):1574–1586, 2011.

[47] J. M. Delery.
Shock phenomena in high speed aerodynamics: still a source of major concern.
The Aeronautical Journal, 103(1019):19–34, 1999.

[48] G. Jones, L. Gartrell, W. Sewall, and P. Stainback.
Designing transonic wind-tunnel test sections for flow diagnostics and laser velocimeter applications.
In *19th AIAA Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, page 1434, 1987.

[49] M. A. Ramaswamy.
Recent developments in transonic flow—part ii—experimental.
Current Science, pages 109–119, 1987.

[50] D. I. Greenwell.
Transonic industrial wind tunnel testing in the 2020s.
The Aeronautical Journal, 126(1295):125–151, 2022.

[51] T. Davidson, N. Stokes, D. Roberts, and M. Quinn.
Lights, camera, data: optical technique development at ara.
In *Applied Aerodynamics Conference 2018: The Future of Aerodynamics*, 2018.

[52] N. J. Lawson.
The application of laser measurement techniques to aerospace flows.
Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 218(1):33–57, 2004.

[53] J. E. Lamar.
Flow-visualization techniques used at high speed by configuration aerodynamics wind-tunnel-test team.
National Aeronautics and Space Administration, Langley Research Center, 2001.

[54] E. Luthman, N. Cymbalist, D. Lang, G. Candler, and P. Dimotakis.
Simulating schlieren and shadowgraph images from les data.
Experiments in Fluids, 60(8):1–16, 2019.

[55] T. S. Onnink.
Simulated shadowgraphy of transonic wing-bound shock waves.
TU Delft MSc thesis, 2016.

[56] J. H. Ferziger, M. Perić, and R. L. Street.
Computational methods for fluid dynamics, volume 3.
Springer, 2002.

[57] J. D. Anderson Jr.
Fundamentals of aerodynamics.
Tata McGraw-Hill Education, 2010.

[58] F. Moukalled, L. Mangani, and M. Darwish.
The finite volume method in computational fluid dynamics, volume 113.
Springer, 2016.

[59] H. K. Versteeg and W. Malalasekera.
An introduction to computational fluid dynamics: the finite volume method.
Pearson education, 2007.

[60] J. D. Anderson.

Modern compressible flow: with historical perspective.
Tata McGraw-Hill Education, 2003.

[61] D. C. Wilcox.
Turbulence modeling for CFD, volume 2.
DCW industries La Canada, CA, 1998.

[62] R. W. Fox, A. T. McDonald, and J. W. Mitchell.
Fox and McDonald's introduction to fluid mechanics.
John Wiley & Sons, 2020.

[63] C. A. J. Fletcher.
Computational techniques for fluid dynamics 1: fundamental and general Techniques.
Springer Science & Business Media, 1991.

[64] T. Bodnár, G. P. Galdi, and Š Nečasová.
Fluids Under Pressure.
Springer Nature, 2020.

[65] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, and J. A. White.
Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes.
AIAA journal, 48(7):1326–1338, 2010.

[66] B. Diskin and J. L. Thomas.
Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes.
AIAA journal, 49(4):836–854, 2011.

[67] J. D. Anderson.
Computational Fluid Dynamics: The basics with applications.
McGraw-Hill International, 1995.

[68] D. Schwamborn, T. Gerhold, and R. Kessler.
Dlr-tau code - an overview.
In *1st ONERA/DLR Aerospace Symposium*, 1999.

[69] ©OpenCFD Ltd.
OpenFOAM®: about OpenFOAM.
`https://www.openfoam.com/`, 2021.
Accessed: 2021-11-13.

[70] C. J. Greenshields.

OpenFOAM user guide version 9.
The OpenFOAM Foundation, 2021.

[71] OpenFOAMWiki.
Finite volume method (OpenFOAM).
`https://www.openfoamwiki.net`, 2010.
Accessed: 2021-11-13.

[72] SimScale.
CFD: PIMPLE algorithm.
`https://www.simscale.com/forum/t/cfd-pimple-algorithm/81418`, January 2018.
Accessed: 2020-03-18.

[73] T. Holzmann.
Mathematics, numerics, derivations and OpenFOAM®.
Loeben, Germany: Holzmann CFD, 2016.

[74] Kitware.
The visualization toolkit (VTK).
`https://vtk.org/`, 2021.
Accessed: 2021-06-15.

[75] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding.
Two calculation procedures for steady, three-dimensional flows with recirculation.
In *Proceedings of the third international conference on numerical methods in fluid mechanics*, pages 60–68. Springer, 1973.

[76] R. I. Issa.
Solution of the implicitly discretised fluid flow equations by operator-splitting.
Journal of computational physics, 62(1):40–65, 1986.

[77] SU2.
SU2, multiphysics simulation and design software.
`https://su2code.github.io/docs_v7/home/`, 2021.
Accessed: 2021-11-13.

[78] T D Economon, F Palacios, S R Copeland, T W Lukaczyk, and J J Alonso.
SU2: an open-source suite for multiphysics simulation and design.
AIAA Journal, 54(3):828–846, 2016.

[79] A. Loseille.
Mesh generation and adaptation for scientific computing.

PhD thesis, Université Paris-Saclay, 2020.

[80] INRIA.
pyamg.
https://pyamg.saclay.inria.fr/pyamg.html, 2018.
Accessed: 2021-07-05.

[81] A. Loseille.
Unstructured mesh generation and adaptation.
In *Handbook of Numerical Analysis*, volume 18, pages 263–302. Elsevier, 2017.

[82] SU2 Multiphysics Simulation and Design Software 2021.
Governing equations in SU2.
https://su2code.github.io/docs_v7/Theory/#compressible-euler, 2021.
Accessed: 2021-09-01.

[83] SU2 Multiphysics Simulation and Design Software 2021.
Inviscid ONERA M6.
https://su2code.github.io/tutorials/Inviscid_ONERAM6/, 2021.
Accessed: 2021-09-01.

[84] A. Jameson, W. Schmidt, and E. Turkel.
Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes.
14th AIAA Fluid and Plasma Dynamic Conference, AIAA-1981-1259, 1981.

[85] A. Jameson.
The origins and further development of the Jameson-Schmidt-Turkel (JST) scheme.
In *33rd AIAA Applied Aerodynamics Conference, AIAA-2015–2718*, 2015.

[86] R. Swanson and E. Turkel.
A multistage time-stepping scheme for the Navier-Stokes equations.
84, Part 62 of ICASE report, NASA CR-172527, 84, 1985.

[87] D. Mavriplis and A. Jameson.
Multigrid solution of the Navier-Stokes equations on triangular meshes.
AIAA Journal, 28(8):1415–1425, 1990.

[88] A. Jameson and D. Mavriplis.
Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh.
AIAA Journal, 24(6):611–618, 1986.

[89] J. Batina.
Unsteady Euler airfoil solutions using unstructured dynamic meshes.
AIAA Journal, 28(8):1381–1388, 1990.

[90] A. Jameson.
Time dependent calculations using multigrid with applications to unsteady flow past airfoils and wings.
In *10th AIAA Computational Fluid Dynamics Conference, AIAA-1991-1956*, 1991.

[91] A. Arnone, M-S Liou, and L. A. Povinelli.
Multigrid time-accurate integration of Navier-Stokes equations.
In *11th AIAA Computational Fluid Dynamics Conference, AIAA-1993-3361*, 1993.

[92] A. Gaitonde.
A dual time method for the solution of the unsteady Euler equations.
Aeronautical Journal, 98:283–291, 1995.

[93] A. Gaitonde, S. Fiddes, and D. Jones.
A 2D Navier-Stokes method for unsteady compressible flow calculations on moving meshes.
Aeronautical Journal, 102:89–97, 1998.

[94] R. Swanson and R. Radespiel.
Cell centered and cell vertex multigrid schemes for the Navier-Stokes equations.
AIAA Journal, 29(5):697–703, 1991.

[95] S. Chakrabartty, K. Dhanalakshmi, and J. Mathur.
Computation of three-dimensional transonic flow using a cell vertex finite volume method for the Euler equations.
Acta Mechanica, 115:161–177, 1996.

[96] A. Gaitonde.
A dual-time method for two-dimensional unsteady incompressible flow calculations.
International Journal for Numerical Methods in Engineering, 41(6):1153–1166, 1998.

[97] J. Farmer, L. Martinelli, and A. Jameson.
A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces.
In *31st AIAA Aerospace Sciences Meeting and Exhibit, AIAA 1993-0767*, 1993.

[98] H. Gough, A. Gaitonde, and Jones D.
A dual-time central-difference interface-capturing finite volume scheme applied to cavitation modelling.

International Journal for Numerical Methods in Fluids, 66(4):452 – 485, 2011.

[99] ©The CGAL Project.
Cgal the computational geometry algorithms library.
`https://www.cgal.org/`, 2020.
Accessed: 2020-05-29.

[100] E. Reinhard, E. A. Khan, A. O. Akyuz, and G. Johnson.
Color imaging: fundamentals and applications.
CRC Press, 2008.

[101] gamedev.net.
Spectral path tracing.
`https://www.gamedev.net/blog/1455/entry-2255136-path-tracing-is-fun/`, 2012.
Accessed: 2021-26-01.

[102] H. W. Jensen.
Realistic image synthesis using photon mapping.
AK Peters/CRC Press, 2001.

[103] M. Pharr, W. Jakob, and G. Humphreys.
Physically based rendering: From theory to implementation.
Morgan Kaufmann, 2016.

[104] K. Suffern.
Ray Tracing from the Ground up.
CRC Press, 2016.

[105] M. Iqbal.
An introduction to solar radiation.
Elsevier, 2012.

[106] C. A. Gueymard.
The sun's total and spectral irradiance for solar energy applications and solar radiation models.
Solar Energy, 76:423–453, 2004.

[107] M. P. Thekaekara.
Solar energy outside the earth's atmosphere.
Solar energy, 14(2):109–127, 1973.

[108] C. Fröhlich and C. Wehrli.

Spectral distribution of solar irradiance from 25000 nm to 250 nm.
World Radiation Center, Davos, Switzerland, private communication, 1981.

[109] U.S. Department of Energy The National Renewable Energy Laboratory.
Solar spectra.
`https://www.nrel.gov/grid/solar-resource/spectra.html`, 2021.
Accessed: 2021-10-09.

[110] ASTM International.
Standard solar constant and zero air mass solar spectral irradiance tables.
`https://www.astm.org/Standards/E490.htm`, 2021.
Accessed: 2021-10-11.

[111] U.S. Department of Energy The National Renewable Energy Laboratory.
2000 astm standard extraterrestrial spectrum reference e-490-00.
`https://www.nrel.gov/grid/solar-resource/spectra-astm-e490.html`, 2021.
Accessed: 2021-10-09.

[112] U.S. Department of Energy The National Renewable Energy Laboratory.
1985 wehrli standard extraterrestrial solar irradiance spectrum.
`https://www.nrel.gov/grid/solar-resource/spectra-wehrli.html`, 2021.
Accessed: 2021-10-09.

[113] ASTM International.
Standard tables for reference solar spectral irradiances: Direct normal and hemispherical on 37° tilted surface.
`https://www.astm.org/Standards/G173.htm`, 2021.
Accessed: 2021-10-11.

[114] C. Wyman, P.-P. Sloan, and P. Shirley.
Simple analytic approximations to the cie xyz color matching functions.
J. Comput. Graph. Tech, 2(2):11, 2013.

[115] ITURBT ITU.
Parameter values for the HDTV standards for production and international programme exchange.
Recommendation ITU-R BT.709–5, 2002.

[116] J. Vries.
Gamma correction.

https://learnopengl.com/Advanced-Lighting/Gamma-Correction, 2009.
Accessed: 2021-02-25.

[117] D. T. Moore.
Gradient-index optics: a review.
Applied Optics, 19(7):1035–1038, 1980.

[118] A. S. Glassner.
An introduction to ray tracing.
Elsevier, 1989.

[119] C. Cao, Z. Ren, B. Guo, and K. Zhou.
Interactive rendering of non-constant, refractive media using the ray equations of gradient-index optics.
In *Computer Graphics Forum*, volume 29, pages 1375–1382. Wiley Online Library, 2010.

[120] Q. Mo, H. Yeh, and D. Manocha.
Tracing analytic ray curves for light and sound propagation in non-linear media.
IEEE transactions on visualization and computer graphics, 22(11):2493–2506, 2015.

[121] A. Sharma, D. V. Kumar, and A. K. Ghatak.
Tracing rays through graded-index media: a new method.
Applied Optics, 21(6):984–987, 1982.

[122] S. Dorić.
Ray tracing through gradient-index media: recent improvements.
Applied optics, 29(28):4026–4029, 1990.

[123] P. J. Sands.
Classification scheme and nomenclature for refractive-index distributions.
Applied optics, 22(3):430–431, 1983.

[124] B. D. Guenther.
Modern optics.
OUP Oxford, 2015.

[125] T. Sakamoto.
Ray trace algorithms for grin media.
Applied optics, 26(15):2943–2946, 1987.

[126] J. M. Fine.
Interpolants for runge-kutta-nyström methods.
Computing, 39(1):27–42, 1987.

[127] E. Hairer and G. Wanner.
A theory for nyström methods.
Numerische Mathematik, 25(4):383–400, 1975.

[128] E. Fehlberg.
Classical eight-and lower-order runge-kutta-nyström formulas with stepsize control for special second-order differential equations.
NASA TR R-381, 1972.

[129] D. G. Bettis.
A runge-kutta nyström algorithm.
Celestial mechanics, 8(2):229–233, 1973.

[130] J. R. Cash and A. H. Karp.
A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides.
ACM Transactions on Mathematical Software (TOMS), 16(3):201–222, 1990.

[131] J. R. Dormand and P. J. Prince.
New runge-kutta algorithms for numerical simulation in dynamical astronomy.
Celestial Mechanics, 18(3):223–232, 1978.

[132] D. G. Hull.
Fourth-order runge-kutta integration with stepsize control.
AIAA Journal, 15(10):1505–1507, 1977.

[133] E. Hairer, S. P. Nørsett, and G. Wanner.
Solving ordinary differential equations i, nonstiff problems.
Springer-Verlag, Berlin, DOI, 10:978–3, 1993.

[134] P. J. Prince and J. R. Dormand.
High order embedded runge-kutta formulae.
Journal of computational and applied mathematics, 7(1):67–75, 1981.

[135] E. Hairer.
Fortran and matlab codes.
`http://www.unige.ch/~hairer/software.html`, 2021.
Accessed: 2021-11-29.

[136] The SciPy community.
Integration (scipy.integrate): Ordinary differential equations (solve$_{ivp}$).
`https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html`, 2021.

Accessed: 2021-06-14.

[137] T. Barth.
Numerical aspects of computing high reynolds number flows on unstructured meshes.
page 721. AIAA Paper 91-0721, 1991.

[138] H. Nishikawa.
Beyond interface gradient: a general principle for constructing diffusion schemes.
40th AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2010-5093, 2010.

[139] T. Barth.
Recent developments in high order k-exact reconstruction on unstructured meshes.
AIAA Paper 93-0668, 1993.

[140] D. Eberly.
Least squares fitting of data.
Chapel Hill, NC: Magic Software, 2000.

[141] L. Neumann, B. Csébfalvi, A. König, and E. Gröller.
Gradient estimation in volume data using 4d linear regression.
In *Computer Graphics Forum*, volume 19, pages 351–358. Wiley Online Library, 2000.

[142] The PyVista Developers.
Filters.
`https://docs.pyvista.org/core/filters.html?highlight=compute_derivative#pyvista.DataSetFilters.compute_derivative`, 2021.
Accessed: 2021-03-23.

[143] VTK.
vtkgradientfilter class reference.
`https://vtk.org/doc/nightly/html/classvtkGradientFilter.html#details`, 2021.
Accessed: 2021-03-23.

[144] C. Simpson.
Barycentric coordinates.
`https://www.cdsimpson.net/2014/10/barycentric-coordinates.html`, 2014.
Accessed: 2021-03-23.

[145] C. Hollauer.
Modeling of thermal oxidation and stress effects.
Technischen Universität Wien, PhD Thesis, 2007.

[146] ParaView Support.
Numerics of "gradient of unstructured dataset" filter.
https://discourse.paraview.org/, 2018.
Accessed: 2021-03-23.

[147] S. K. Lodha and R. Franke.
Scattered data interpolation: Radial.
Handbook of computer aided geometric design, page 389, 2002.

[148] G. Fasshauer.
Meshfree approximation with matlab - lecture i: Introduction.
Dolomites Research Week on Approximation, 2008.

[149] K. Jasek, M. Pasternak, W. Miluski, J. Bugaj, and M. Grabka.
Application of gaussian radial basis functions for fast spatial imaging of ground penetration radar data obtained on an irregular grid.
Electronics, 10(23), 2021.
ISSN 2079-9292.
doi: 10.3390/electronics10232965.
URL https://www.mdpi.com/2079-9292/10/23/2965.

[150] ALGLIB®.
Fast rbf interpolation/fitting.
https://www.alglib.net/interpolation/fastrbf.php, 2020.
Accessed: 2020-03-12.

[151] VTK.
vtkpointinterpolator class reference.
https://vtk.org/doc/nightly/html/classvtkPointInterpolator.html#details, 2021.
Accessed: 2021-12-01.

[152] VTK.
vtkgaussiankernel class reference.
https://vtk.org/doc/nightly/html/classvtkGaussianKernel.html, 2021.
Accessed: 2021-12-01.

[153] C. Garth and K. I. Joy.
Fast, memory-efficient cell location in unstructured grids for visualization.
IEEE Transactions on Visualization and Computer Graphics, 16(6):1541–1550, 2010.

[154] VTK.
vtkcelltreelocator class reference.
https://vtk.org/doc/nightly/html/classvtkCellTreeLocator.html, 2021.
Accessed: 2021-12-02.

[155] VTK.
vtkcelllocator class reference.
https://vtk.org/doc/nightly/html/classvtkCellLocator.html, 2021.
Accessed: 2021-12-02.

[156] VTK.
vtkstaticcelllocator class reference.
https://vtk.org/doc/nightly/html/classvtkStaticCellLocator.html, 2021.
Accessed: 2021-12-02.

[157] VTK.
vtkmodifiedbsptree class reference.
https://vtk.org/doc/nightly/html/classvtkModifiedBSPTree.html, 2021.
Accessed: 2021-12-02.

[158] VTK.
vtkobbtree class reference.
https://vtk.org/doc/nightly/html/classvtkOBBTree.html, 2021.
Accessed: 2021-12-02.

[159] J. Vries.
Physically based rendering: Theory.
https://learnopengl.com/PBR/Theory, May 2020.
Accessed: 2020-04-20.

[160] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance.
Microfacet models for refraction through rough surfaces.
Rendering techniques, 2007:18th, 2007.

[161] K. E. Torrance and E. M. Sparrow.
Theory for off-specular reflection from roughened surfaces.
Josa, 57(9):1105–1114, 1967.

[162] R. L. Cook and K. E. Torrance.

A reflectance model for computer graphics.
ACM Transactions on Graphics (ToG), 1(1):7–24, 1982.

[163] A. S. Glassner.
Principles of digital image synthesis.
Elsevier, 2014.

[164] J. T. Kajiya.
The rendering equation.
In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[165] T. Whitted.
An improved illumination model for shaded display.
In *ACM Siggraph 2005 Courses*, pages 4–es. 2005.

[166] E. Gröller.
Nonlinear ray tracing: Visualizing strange worlds.
The Visual Computer, 11(5):263–274, 1995.

[167] J. Arvo.
Backward ray tracing.
In *Developments in Ray Tracing, Computer Graphics, Proc. of ACM SIGGRAPH 86 Course Notes*, pages 259–263, 1986.

[168] R. L. Cook, T. Porter, and L. Carpenter.
Distributed ray tracing.
In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, 1984.

[169] E. P. Lafortune and Y. D. Willems.
Bi-directional path tracing.
Proceedings of Compugraphics, pages 145–153, 1995.

[170] E. Veach and L. Guibas.
Bidirectional estimators for light transport.
In *Photorealistic Rendering Techniques*, pages 145–167. Springer, 1995.

[171] B. W. Silverman.
Density estimation for statistics and data analysis.
Routledge, 2018.

[172] L. Schjøth, J. Sporring, and O. Fogh Olsen.

Diffusion based photon mapping.
In *Computer Graphics Forum*, volume 27, pages 2114–2127. Wiley Online Library, 2008.

[173] M. J. Pavicic.
Convenient anti-aliasing filters that minimize “bumpy” sampling.
In *Graphics gems*, pages 144–146. 1990.

[174] L. Schjøth, J. R. Frisvad, K. Erleben, and J. Sporring.
Photon differentials.
In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 179–186, 2007.

[175] R. Herzog, V. Havran, S. Kinuwaki, K. Myszkowski, and H. Seidel.
Global illumination using photon ray splatting.
In *Computer Graphics Forum*, volume 26, pages 503–513. Wiley Online Library, 2007.

[176] B. Spencer and M. W. Jones.
Into the blue: Better caustics through photon relaxation.
In *Computer Graphics Forum*, volume 28, pages 319–328. Wiley Online Library, 2009.

[177] T. Hachisuka, S. Ogaki, and H. W. Jensen.
Progressive photon mapping.
In *ACM SIGGRAPH Asia 2008 papers*, pages 1–8. 2008.

[178] NPARC Alliance.
Overview of cfd verification and validation.
`https://www.grc.nasa.gov/www/wind/valid/tutorial/overview.html`, 2021.
Accessed: 2021-03-29.

[179] NPARC Alliance.
Verification assessment.
`https://www.grc.nasa.gov/www/wind/valid/tutorial/verassess.html`, 2021.
Accessed: 2021-03-29.

[180] NPARC Alliance.
Validation assessment.
`https://www.grc.nasa.gov/www/wind/valid/tutorial/valassess.html`, 2021.
Accessed: 2021-03-29.

[181] J. Panda and G. Adamovsky.
An experimental investigation of laser light scattering by shock waves.
33rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA-1995-0518, 1995.

[182] A. Roshko and H. W. Liepmann.
Elements of Gas-dynamics.
Wiley, 1957.

[183] M. D. Van Dyke.
The supersonic blunt-body problem-review and extension.
Journal of the Aerospace Sciences, 25(8):485–496, 1958.

[184] M. Inouye, H. Lomax, and J. V. Rakich.
A description of numerical methods and computer programs for two-dimensional and axisymmetric supersonic flow over blunt-nosed and flared bodies.
Technical report, 1965.

[185] T. Eymann.
Focus on ability to predict separated flow.
`https://aerospaceamerica.aiaa.org/year-in-review/focus-on-ability-to-predict-separated-flow/`, 2017.
Accessed: 2022-05-01.

[186] C. J. Greenshields.
OpenFOAM user guide.
OpenFOAM Foundation Ltd, version, 3(1):47, 2015.

[187] CFD Direct The Architects of OpenFOAM.
OpenFOAM v6 user guide: 5.2 boundaries.
`https://cfd.direct/openfoam/user-guide/v6-boundaries/`, 2020.
Accessed: 2020-09-28.

[188] OpenFOAMWiki.
Axisymmetric.
`https://openfoamwiki.net/index.php/Main_ContribExamples/AxiSymmetric`, 2016.
Accessed: 2020-09-28.

[189] B. Willis, D. Davis, and W. Hingst.
Flow coefficient behavior for boundary layer bleed holes and slots.
33rd AIAA aerospace sciences meeting and exhibit, AIAA-1995-0031, 1995.

[190] K. D. Seablom, R. H. Soeder, D. E. Stark, J. F. Leone, and M. W. Henry.
Nasa glenn 1-by 1-foot supersonic wind tunnel user manual.
Technical report, 1999.

[191] J. D. Martel and B. Jolly.
Analytical shock standoff and shape prediction with validation for blunt face cylinder.
In *AIAA Atmospheric Flight Mechanics Conference*, page 0523, 2015.

[192] G. E. Solomon.
Transonic flow past cone cylinders.
NACA-TN-3213, 1955.

[193] J. W. Heberle, G. P. Wood, and P. B. Gooderum.
Data on shape and location of detached shock waves on cones and spheres.
NACA-TN-2000, 1949.

[194] M Yu Timokhin, M Tikhonov, IV Mursenkova, and IA Znamenskaya.
Shock-wave thickness influence to the light diffraction on a plane shock wave.
Physics of Fluids, 32(11):116103, 2020.

[195] W. G. Vincenti.
Introduction to physical gas dynamic.
Introduction to physical gas dynamics/Huntington, 1975.

[196] S. G. Jennings.
The mean free path in air.
Journal of Aerosol Science, 19(2):159–166, 1988.

[197] K. E. Moore.
Physical and geometric methods combined to model optics.
https://www.photonics.com/Articles/Physical_and_Geometric_Methods_Combined_to_Model/a13327, 2002.
Accessed: 2022-05-02.

[198] F. Dias Paulino da Costa, A. Gaitonde, and D. Jones.
Virtual experiments of light and shock wave interaction using nonlinear ray tracing and photon mapping.
AIAA Scitech 2021 Forum: Visualization and Analytics I, AIAA-2021-1483, 2021.

[199] A. Lagae and P. Dutré.
Accelerating ray tracing using constrained tetrahedralizations.
In *Computer Graphics Forum*, volume 27, pages 1303–1312. Wiley Online Library, 2008.

[200] H. Si.
Tetgen, a delaunay-based quality tetrahedral mesh generator.
ACM Transactions on Mathematical Software (TOMS), 41(2):1–36, 2015.

[201] *MATLAB version 9.4.0.813654 (R2018a)*.
The Mathworks, Inc., Natick, Massachusetts, 2018.

[202] J. Ahrens, B. Geveci, and C. Law.
Paraview: An end-user tool for large data visualization.
The visualization handbook, 717(8), 2005.

[203] V. Schmitt and F. Charpin.
Pressure distributions on the onera-m6-wing at transonic mach numbers.
Experimental Data Base for Computer Program Assessment, AGARD-AR-138, 1979.

[204] NPARC Alliance Validation Archive.
Onera m6 wing.
`https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing.html`, 2021.
Accessed: 2021-05-11.

[205] A Balan, M. A. Park, W. K. Anderson, D. S. Kamenetskiy, J. A. Krakos, T. Michal, and F. Alauzet.
Verification of anisotropic mesh adaptation for turbulent simulations over onera m6 wing.
AIAA Journal, 58(4):1550–1565, 2020.

[206] SU2 Multiphysics Simulation and Design Software 2021.
Physical definition.
`https://su2code.github.io/docs_v7/Physical-Definition/#free-stream-definition-compressible`, 2021.
Accessed: 2021-09-01.

[207] K. R. Spring, T. J. Fellers, and M. W. Davidson.
Introduction to charge-coupled devices (ccds).
`https://www.microscopyu.com/digital-imaging/introduction-to-charge-coupled-devices-ccds`, December 2021.
Accessed: 2021-12-15.

[208] A. Clark.
Pillow (pil fork) documentation, 2015.
URL `https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf`.

[209] J. Pattinson, T. Wilson, and M. Herring.
High fidelity simulation of the folding wing tip for loads alleviation.
In *International Forum on Aeroelasticity and Structural Dynamics, Saint Petersburg, Russia*, 2015.

[210] D. Schwamborn, T. Gerhold, and R. Heinrich.
The DLR TAU-code: recent applications in research and industry.
European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, 5-8 September, 2006.

[211] ONERA The French Aerospace Lab.
What is cassiopee? a set of python modules for cfd pre- and post-processing.
`http://elsa.onera.fr/Cassiopee/More.html`, 2020.
Accessed: 2020-11-06.

[212] J. Tian, Z. Duan, W. Ren, Z. Han, and Y. Tang.
Simple and effective calculations about spectral power distributions of outdoor light sources for computer vision.
Optics express, 24(7):7266–7286, 2016.

[213] B. P. Sandford and D. C. Robertson.
Infrared reflectance of aircraft paints.
Technical report, PHILLIPS LAB HANSCOM AFB MA, 1994.

[214] R. Fernandez, R. G. Seasholtz, L. G. Oberle, and J. R. Kadambi.
Comparison of the bidirectional reflectance distribution function of various surfaces.
NASA TM-101317, 1988.

[215] D. D. Zhdanov, I. S. Potemin, V. A. Galaktionov, B. K. Barladyan, K. A. Vostryakov, and L. Z. Shapiro.
Spectral ray tracing in problems of photorealistic imagery construction.
Programming and Computer Software, 37(5):236–244, 2011.

[216] S. W. Thomas.
Dispersive refraction in ray tracing.
The Visual Computer, 2(1):3–8, 1986.

[217] A. Boudet, M. Paulin, P. Pitot, and D. Pratmarty.
Low memory spectral photon mapping.
2004.

[218] Color Usage Research Lab: NASA AMES Research Center.
Luminance constrast.

https://colorusage.arc.nasa.gov/luminance_cont.php, December 2021.
Accessed: 2021-12-13.

[219] OpenCV Open Source Computer Vision.
Histogram equalization.
https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html, December 2021.
Accessed: 2021-12-13.

[220] M. R. Luo and C. Li.
CIECAM02 and its recent developments.
In *Advanced Color Image Processing and Analysis*, pages 19–58. Springer, 2013.

[221] Colorspacious developers.
Colorspacious.
https://colorspacious.readthedocs.io/en/latest/overview.html, 2015.
Accessed: 2021-12-13.

[222] L. Dalcin.
MPI for Python.
https://mpi4py.readthedocs.io/en/stable/index.html, November 2021.
Accessed: 2021-12-13.

[223] NVIDIA Corporation.
GPU-accelerated computing with python.
https://developer.nvidia.com/how-to-cuda-python, December 2021.
Accessed: 2021-12-13.

[224] M. Cassidy, J. Mélou, Y. Quéau, F. Lauze, and J-D Durou.
Refractive multi-view stereo.
In *2020 International Conference on 3D Vision (3DV)*, pages 384–393. IEEE, 2020.

[225] K. N. Kutulakos and E. Steger.
A theory of refractive and specular 3D shape by light-path triangulation.
International Journal of Computer Vision, 76(1):13–29, 2008.

[226] J. Lyu, B. Wu, D. Lischinski, D. Cohen-Or, and H. Huang.
Differentiable refraction-tracing for mesh reconstruction of transparent objects.
ACM Transactions on Graphics (TOG), 39(6):1–13, 2020.

[227] M. Kassubeck, F. Burgel, S. Castillo, S. Stiller, and M. Magnor.
Shape from caustics: Reconstruction of 3D-printed glass from simulated caustic images.

In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2877–2886, 2021.

[228] K. O. Winter and M. J. Hargather.
Three-dimensional shock wave reconstruction using multiple high-speed digital cameras and background-oriented schlieren imaging.
Experiments in Fluids, 60(6):1–13, 2019.

[229] M. Hafizah, T. Kok, and E. Spriyanto.
3D ultrasound image reconstruction based on VTK.
In *Proceedings of the 9th WSEAS International Conference on SIGNAL PROCESSING*, pages 102–106, 2010.

Appendix 1 Tetgen to CGAL Constrained Tetrahedralization Converter

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <math.h>
#include "boost\filesystem.hpp"
#include <CGAL\Exact_predicates_inexact_constructions_kernel.h>
#include <CGAL\Linear_algebraCd.h>
#include <CGAL\Triangulation_3.h>
#include <CGAL\Triangulation_cell_base_3.h>
#include <CGAL\Triangulation_vertex_base_3.h>
#include <CGAL\Triangulation_vertex_base_with_info_3.h>
#include <CGAL\Triangulation_data_structure_3.h>
// ----------------------------------------------------------------- //
typedef CGAL::Exact_predicates_inexact_constructions_kernel K;
typedef K::FT                                                doubleType;
typedef CGAL::Linear_algebraCd<doubleType>                   LA;
typedef K::Point_3                                           Point3D;
typedef K::Vector_3                                          Vector3D;
typedef K::Segment_3                                         Segment3D;
typedef CGAL::Triangulation_cell_base_3<K>                   CB3D;
typedef CGAL::Triangulation_vertex_base_3<K>                 VB3D;
typedef CGAL::Triangulation_data_structure_3<VB3D, CB3D>     TDS3D;
typedef CGAL::Triangulation_3<K, TDS3D>                      Tri3D;
typedef Tri3D::Vertex_handle                                 VertexHandle;
typedef Tri3D::Cell_handle                                   CellHandle;
typedef std::map<int, VertexHandle >                         vertexHandleMap;
typedef std::map<int, CellHandle >                           cellHandleMap;
typedef std::map<int, Tri3D::Point >                         intTriPointMap;
typedef std::map<int, std::vector<int> >                     intIntsMap;
// ----------------------------------------------------------------- //

Tri3D tetgen2cgal(const std::string filePath, const std::string fileName){
```

```
// ----------------------------------------------------------------------- //
// ACCESS THE DIRECTORY CONTAINING THE CONSTRAINED TETRAHEDRALIZATION
// ----------------------------------------------------------------------- //
boost::filesystem::current_path(filePath);
// ---------------------------------------------------------------------- //
// ".NODE", ".ELE" AND ".NEIGH" FILE NAMES
// ---------------------------------------------------------------------- //
std::string nodeFile    = fileName + ".1.node";
std::string eleFile     = fileName + ".1.ele";
std::string neighFile   = fileName + ".1.neigh";
// ---------------------------------------------------------------------- //
// INITIALIZE TETRAHEDRALIZATION
// ---------------------------------------------------------------------- //
Tri3D tri;
// ---------------------------------------------------------------------- //
// INITIALIZE STD::MAPS
// ---------------------------------------------------------------------- //
vertexHandleMap vertexMap;
cellHandleMap   cellMap;
intTriPointMap  nodeMap;
intIntsMap      eleMap;
intIntsMap      neighMap;
// ---------------------------------------------------------------------- //
// READ ".NODE" FILE -> LIST OF VERTICES
// IDX VERTEX - X-COORD - Y-COORD - Z-COORD - SCALAR - BOUNDARY MARKER
// ---------------------------------------------------------------------- //
std::string     line;
std::fstream    nodeFileReader(nodeFile.c_str());

if (!nodeFileReader){
        std::cout << "Error␣reading␣.NODE␣file!" << std::endl;
        return 1;
}
else{
        while (std::getline(nodeFileReader, line)){

                int     idxV, bm;
                double  x, y, z, scalar;

                std::stringstream lineStream;
                lineStream << line;

                lineStream >> idxV >> x >> y >> z >> scalar >> bm;

                nodeMap.insert(std::make_pair(idxV, Tri3D::Point(x, y, z))
                    ↪ );
        }
        nodeFileReader.close();
}
// ---------------------------------------------------------------- //
// READ ".ELE" FILE -> LIST OF TETRAHEDRON
// IDX TETRAHEDRON - IDX VERTEX 1 - IDX VERTEX 2 - IDX VERTEX 3 - IDX VERTEX 4
// ---------------------------------------------------------------- //
std::fstream    eleFileReader(eleFile.c_str());

if (!eleFileReader){
        std::cout << "Error reading .ELE file!" << std::endl;
        return 1;
}
else{
        while (std::getline(eleFileReader, line)){
                int     idxC, idxV0, idxV1, idxV2, idxV3;

                std::stringstream lineStream;
                lineStream << line;

                lineStream >> idxC >> idxV0 >> idxV1 >> idxV2 >> idxV3;
                std::vector<int> tmp{ idxV0, idxV1, idxV2, idxV3 };
                // Ele std::map so be used later
                eleMap.insert(std::make_pair(idxC, tmp));

        }
        eleFileReader.close();
}
// ---------------------------------------------------------------- //
// READ ".NEIGH" FILE -> LIST OF NEIGHBOURS
// IDX TETRAHEDRON - IDX NEIGH 1 - IDX NEIGH 2 - IDX NEIGH 3 - IDX NEIGH 4
// ---------------------------------------------------------------- //
std::fstream    neighFileReader(neighFile.c_str());

if (!neighFileReader){
        std::cout << "Error reading .NEIGH file!" << std::endl;
        return 1;
}
else{
        while (std::getline(neighFileReader, line)){
                int     idxC, idxC0, idxC1, idxC2, idxC3;

                std::stringstream lineStream;
                lineStream << line;
```

```
            lineStream >> idxC >> idxC0 >> idxC1 >> idxC2 >> idxC3;
            std::vector<int> tmp{ idxC0, idxC1, idxC2, idxC3 };

            // Neigh std::map to be used later
            neighMap.insert(std::make_pair(idxC, tmp));

        }
        neighFileReader.close();
    }
    // ------------------------------------------------------------------ //
    // CLEAR ANY TETRAHEDRALIZATION PARAMETERS SET DURING CONSTRUCTION
    // ------------------------------------------------------------------ //
    tri.tds().clear();
    // ------------------------------------------------------------------ //
    // SET TETRAHEDRALIZATION DIMENSION
    // ------------------------------------------------------------------ //
    tri.tds().set_dimension(3);
    // ------------------------------------------------------------------ //
    // CREATE FINITE VERTICES
    // ------------------------------------------------------------------ //
    // Vertex handle initialization
    VertexHandle    vH;

    // Iterate over finite vertices
    for (auto iterV = nodeMap.begin(); iterV != nodeMap.end(); iterV++){
        vH = tri.tds().create_vertex();

        // Set its point geometric coordinates
        vH->set_point(iterV->second);

        // Vertex std::map to be used later
        vertexMap.insert(std::make_pair(iterV->first, vH));
    }
    // ------------------------------------------------------------------ //
    // CREATE FINITE CELLS
    // ------------------------------------------------------------------ //
    // Cell handle initialization
    CellHandle  cH;

    // Iterate over finite cells
    for (auto iterC = eleMap.begin(); iterC != eleMap.end(); iterC++){

        cH = tri.tds().create_cell();
        // Tetgen already orients the cells' vertices
        // positively (clockwise) in the ".ele" file
        cH->set_vertices(vertexMap.at(iterC->second[0]),
                         vertexMap.at(iterC->second[1]),
                    vertexMap.at(iterC->second[2]),
                    vertexMap.at(iterC->second[3]));


        // Set incident cell to each one of the vertices
        vertexMap.at(iterC->second[0])->set_cell(cH);
        vertexMap.at(iterC->second[1])->set_cell(cH);
        vertexMap.at(iterC->second[2])->set_cell(cH);
        vertexMap.at(iterC->second[3])->set_cell(cH);

        // Cell std::map to be used later
        cellMap.insert(std::make_pair(iterC->first, cH));
}
// -------------------------------------------------------------------- //
// CREATE INFINITE VERTEX
// -------------------------------------------------------------------- //
VertexHandle infV = tri.tds().create_vertex();
tri.set_infinite_vertex(infV);
// -------------------------------------------------------------------- //
// ASSIGN FINITE NEIGHBOURS TO FINITE AND INFINITE CELLS
// -------------------------------------------------------------------- //
// Iterate over finite cells
for (auto iterC = neighMap.begin(); iterC != neighMap.end(); iterC++){
        // Initialize cell handles
        CellHandle c, c00, c11, c22, c33, cNN, c0, c1, c2, c3;
        // Initialize finite/infinite neighbours identifier and neighbour
            ↪ indexing identifier
        std::vector<int> tmp1{ 1, 1, 1, 1 };
        std::vector<int> tmp2{ 0, 0, 0, 0 };

        // Finite cell
        c = cellMap.at(iterC->first);

        // Convex hull cell
        if ((iterC->second[0] == -1) | (iterC->second[1] == -1) | (iterC->
            ↪ second[2] == -1) | (iterC->second[3] == -1)){
                // ****************************** //
                // tmp1[i] == 1 -> Finite Cell    //
                // ****************************** //
                // tmp1[i] == 0 -> Infinite Cell  //
                // ****************************** //
                auto iter = iterC->second.begin();
                while ((iter = std::find(iter, iterC->second.end(), -1))
                    ↪ != iterC->second.end())
                {
                        tmp1[iter - iterC->second.begin()] = 0;
                        iter++;
```

```
}

// Iterate over neighbours
for (int i = 0; i < 4; i++){
        // Finite neighbours
        if (tmp1[i] == 1){
                // Finite neighbour
                cNN = cellMap.at(iterC->second[i]);

                // ******************************** //
                // tmp2[i] == 1 -> Neighbour index
                    ↪ assigned //
                // ******************************** //
                // Determine which vertices are shared
                    ↪ between the two finite neighbours
                // and assign correspondent neighbour
                    ↪ index
                // Ex.:
                // vertices shared: 0, 1, 2
                // neighbour index: 3
                if ((cNN->has_vertex(c->vertex(1)) == true
                    ↪ ) & (cNN->has_vertex(c->vertex(2))
                    ↪ == true) & (cNN->has_vertex(c->
                    ↪ vertex(3)) == true)){
                        c0 = cNN;
                        tmp2[0] = 1;
                }
                else if ((cNN->has_vertex(c->vertex(0)) ==
                    ↪  true) & (cNN->has_vertex(c->vertex
                    ↪ (2)) == true) & (cNN->has_vertex(c->
                    ↪ vertex(3)) == true)){
                        c1 = cNN;
                        tmp2[1] = 1;
                }
                else if ((cNN->has_vertex(c->vertex(0)) ==
                    ↪  true) & (cNN->has_vertex(c->vertex
                    ↪ (1)) == true) & (cNN->has_vertex(c->
                    ↪ vertex(3)) == true)){
                        c2 = cNN;
                        tmp2[2] = 1;
                }
                else if ((cNN->has_vertex(c->vertex(0)) ==
                    ↪  true) & (cNN->has_vertex(c->vertex
                    ↪ (1)) == true) & (cNN->has_vertex(c->
                    ↪ vertex(2)) == true)){
                        c3 = cNN;
                        tmp2[3] = 1;
                }
        }
}

// Check if finite neighbours were assigned
if (std::accumulate(tmp2.begin(), tmp2.end(), 0) != std::
    ↪ accumulate(tmp1.begin(), tmp1.end(), 0)){
        std::cout << "Issue␣assigning␣finite␣neighbor␣
            ↪ cells" << std::endl;
        return 1;
}

// Create infinite cells = Infinite neighbours of finite
    ↪ convex hull cells
// ************************************************** //
// ATTENTION: IF CGAL THROWS "PB ORIENTATION WITH
// NEIGHBOR" ERROR, SIMPLY CHANGE THE VERTICES
// INDEXES/ORDERING ON CH->SET_VERTICE
// ************************************************** //
if (tmp2[0] == 0){

        cH = tri.tds().create_cell();
        // The neighbour index of the finite cell not yet
        // assigned determines the vertices indexes that
        // comprise the convex hull facet
        // The auxiliary infinite vertex completes
        // the infinite cell

        cH->set_vertex(0, tri.infinite_vertex());
        cH->set_vertex(1, c->vertex(3));
        cH->set_vertex(2, c->vertex(2));
        cH->set_vertex(3, c->vertex(1));

        c0 = cH;
        tmp2[0] = 1;

        tri.infinite_vertex()->set_cell(cH);

        // The index of the finite neighbour of the
        // infinite cell is exactly the index of the
        // infinite vertex
        cH->set_neighbor(cH->index(tri.infinite_vertex()),
            ↪  c);
}
if (tmp2[1] == 0){
        cH = tri.tds().create_cell();
```

```
                        cH->set_vertex(0, c->vertex(2));
                        cH->set_vertex(1, tri.infinite_vertex());
                        cH->set_vertex(2, c->vertex(0));
                        cH->set_vertex(3, c->vertex(3));

                        c1 = cH;
                        tmp2[1] = 1;

                         tri.infinite_vertex()->set_cell(cH);
                        cH->set_neighbor(cH->index(tri.infinite_vertex()),
                            ↪ c);
                }
                if (tmp2[2] == 0){
                        cH = tri.tds().create_cell();

                        cH->set_vertex(0, c->vertex(1));
                        cH->set_vertex(1, c->vertex(0));
                        cH->set_vertex(2, tri.infinite_vertex());
                        cH->set_vertex(3, c->vertex(3));

                        c2 = cH;
                        tmp2[2] = 1;


                         tri.infinite_vertex()->set_cell(cH);
                        cH->set_neighbor(cH->index(tri.infinite_vertex()),
                            ↪ c);
                }
                if (tmp2[3] == 0){
                        cH = tri.tds().create_cell();


                        cH->set_vertex(0, c->vertex(2));
                        cH->set_vertex(1, c->vertex(1));
                        cH->set_vertex(2, c->vertex(0));
                        cH->set_vertex(3, tri.infinite_vertex());

                        c3 = cH;
                        tmp2[3] = 1;

                         tri.infinite_vertex()->set_cell(cH);
                        cH->set_neighbor(cH->index(tri.infinite_vertex()),
                            ↪ c);
                }

                // Set neighbours of finite convex hull cell
                if (std::accumulate(tmp2.begin(), tmp2.end(), 0) == 4){
```

```
                c->set_neighbors(c0, c1, c2, c3);
            }
            else{
                std::cout << "INFINITE␣NEIGHBOR␣MISSING" << std::
                  ↪ endl;
                return 1;
            }
        }
        // Interior cells
        else{
            // Interior cell
            c = cellMap.at(iterC->first);

            // Finite neighbours
            c00 = cellMap.at(iterC->second[0]);
            c11 = cellMap.at(iterC->second[1]);
            c22 = cellMap.at(iterC->second[2]);
            c33 = cellMap.at(iterC->second[3]);

            // Determine which vertices are shared between the two
            // finite neighbours and assign correspondent neighbour
            // index
            // Ex.:
            // vertices shared: 0, 1, 2
            // neighbour index: 3
            if ((c00->has_vertex(c->vertex(1)) == true) & (c00->
              ↪ has_vertex(c->vertex(2)) == true) & (c00->has_vertex
              ↪ (c->vertex(3)) == true)){
                c0 = c00;
                tmp2[0] = 1;
            }
            else if ((c00->has_vertex(c->vertex(0)) == true) & (c00->
              ↪ has_vertex(c->vertex(2)) == true) & (c00->has_vertex
              ↪ (c->vertex(3)) == true)){
                c1 = c00;
                tmp2[1] = 1;
            }
            else if ((c00->has_vertex(c->vertex(0)) == true) & (c00->
              ↪ has_vertex(c->vertex(1)) == true) & (c00->has_vertex
              ↪ (c->vertex(3)) == true)){
                c2 = c00;
                tmp2[2] = 1;
            }
            else if ((c00->has_vertex(c->vertex(0)) == true) & (c00->
              ↪ has_vertex(c->vertex(1)) == true) & (c00->has_vertex
              ↪ (c->vertex(2)) == true)){
                c3 = c00;
                tmp2[3] = 1;
        }
        if ((c11->has_vertex(c->vertex(1)) == true) & (c11->
            ↪ has_vertex(c->vertex(2)) == true) & (c11->has_vertex
            ↪ (c->vertex(3)) == true)){
                c0 = c11;
                tmp2[0] = 1;
        }
        else if ((c11->has_vertex(c->vertex(0)) == true) & (c11->
            ↪ has_vertex(c->vertex(2)) == true) & (c11->has_vertex
            ↪ (c->vertex(3)) == true)){
                c1 = c11;
                tmp2[1] = 1;
        }
        else if ((c11->has_vertex(c->vertex(0)) == true) & (c11->
            ↪ has_vertex(c->vertex(1)) == true) & (c11->has_vertex
            ↪ (c->vertex(3)) == true)){
                c2 = c11;
                tmp2[2] = 1;
        }
        else if ((c11->has_vertex(c->vertex(0)) == true) & (c11->
            ↪ has_vertex(c->vertex(1)) == true) & (c11->has_vertex
            ↪ (c->vertex(2)) == true)){
                c3 = c11;
                tmp2[3] = 1;
        }
        if ((c22->has_vertex(c->vertex(1)) == true) & (c22->
            ↪ has_vertex(c->vertex(2)) == true) & (c22->has_vertex
            ↪ (c->vertex(3)) == true)){
                c0 = c22;
                tmp2[0] = 1;
        }
        else if ((c22->has_vertex(c->vertex(0)) == true) & (c22->
            ↪ has_vertex(c->vertex(2)) == true) & (c22->has_vertex
            ↪ (c->vertex(3)) == true)){
                c1 = c22;
                tmp2[1] = 1;
        }
        else if ((c22->has_vertex(c->vertex(0)) == true) & (c22->
            ↪ has_vertex(c->vertex(1)) == true) & (c22->has_vertex
            ↪ (c->vertex(3)) == true)){
                c2 = c22;
                tmp2[2] = 1;
        }
        else if ((c22->has_vertex(c->vertex(0)) == true) & (c22->
            ↪ has_vertex(c->vertex(1)) == true) & (c22->has_vertex
            ↪ (c->vertex(2)) == true)){
```

```
            c3 = c22;
            tmp2[3] = 1;
        }
        if ((c33->has_vertex(c->vertex(1)) == true) & (c33->
          ↪ has_vertex(c->vertex(2)) == true) & (c33->has_vertex
          ↪ (c->vertex(3)) == true)){
            c0 = c33;
            tmp2[0] = 1;
        }
        else if ((c33->has_vertex(c->vertex(0)) == true) & (c33->
          ↪ has_vertex(c->vertex(2)) == true) & (c33->has_vertex
          ↪ (c->vertex(3)) == true)){
            c1 = c33;
            tmp2[1] = 1;
        }
        else if ((c33->has_vertex(c->vertex(0)) == true) & (c33->
          ↪ has_vertex(c->vertex(1)) == true) & (c33->has_vertex
          ↪ (c->vertex(3)) == true)){
            c2 = c33;
            tmp2[2] = 1;
        }
        else if ((c33->has_vertex(c->vertex(0)) == true) & (c33->
          ↪ has_vertex(c->vertex(1)) == true) & (c33->has_vertex
          ↪ (c->vertex(2)) == true)){
            c3 = c33;
            tmp2[3] = 1;
        }

        // Set neighbours of finite interior cell
        if (std::accumulate(tmp2.begin(), tmp2.end(), 0) == 4){
            c->set_neighbors(c0, c1, c2, c3);
        }
        else{
            std::cout << "␣NEIGHBOR␣MISSING" << std::endl;
            return 1;
        }
    }
}




// ------------------------------------------------------------------ //
// ASSIGN INFINITE NEIGHBOURS TO INFINITE CELLS
// ------------------------------------------------------------------ //
int countNeigh = 0;
```

```
// Iterate over all finite/infinite cells
for (auto iterCell = tri.all_cells_begin(); iterCell != tri.all_cells_end
    ↪ (); iterCell++){

    // Find infinite cells - main infinite cell
    if (tri.is_infinite(iterCell) == true){

        // Get index of infinite vertex/finite neighbour
        int idxInf = iterCell->index(tri.infinite_vertex());

        // Iterate again over all finite/infinite cells
        for (auto iterN = tri.all_cells_begin(); iterN != tri.
            ↪ all_cells_end(); iterN++){

            // Find infinite cells different from main
            // infinite cell - infinite neighbours
            if ((tri.is_infinite(iterN) == true) & (iterN !=
                ↪ iterCell)){
                // Determine which of the finite edges is
                // shared between main infinite cell and
                // infinite neighbours and assign
                // correspondent neighbour
                // Ex.:
                // index of finite neighbour = 0
                // finite edges shared between infinite
                // neighbours= (v(1), v(2)), (v(1),v(3)),
                // (v(2),v(3))
                // indexes of infinite neighbours = 3, 2,
                    ↪ 1
                if ((iterN->has_vertex(iterCell->vertex(((
                    ↪ idxInf + 1) & 3))) == true) & (iterN
                    ↪ ->has_vertex(iterCell->vertex(((
                    ↪ idxInf + 2) & 3))) == true)){
                        iterCell->set_neighbor(((idxInf +
                            ↪ 3) & 3), iterN);
                }
                else if ((iterN->has_vertex(iterCell->
                    ↪ vertex(((idxInf + 1) & 3))) == true)
                    ↪  & (iterN->has_vertex(iterCell->
                    ↪ vertex(((idxInf + 3) & 3))) == true)
                    ↪ ){
                        iterCell->set_neighbor(((idxInf +
                            ↪ 2) & 3), iterN);
                }
                else if ((iterN->has_vertex(iterCell->
                    ↪ vertex(((idxInf + 2) & 3))) == true)
                    ↪  & (iterN->has_vertex(iterCell->
                      ↪ vertex(((idxInf + 3) & 3))) == true)
                      ↪ ){
                        iterCell->set_neighbor(((idxInf +
                          ↪ 1) & 3), iterN);
                    }
                }
            }
        }
    }
    // ------------------------------------------------------------------  //
    // FINAL CHECKS
    // ------------------------------------------------------------------  //
    std::cout << "--------------------------------------------" << std::endl;
    std::cout << "Constrained␣Tetrahedralization:" << std::endl;
    tri.is_valid(true);
    //std::cout << "Number of finite cells: " << tri.number_of_finite_cells()
      ↪ << std::endl;
    //std::cout << "Number of finite facets: " << tri.number_of_finite_facets
      ↪ () << std::endl;
    //std::cout << "Number of finite vertices: " << tri.number_of_vertices()
      ↪ << std::endl;
    std::cout << "--------------------------------------------" << std::endl;
    // ------------------------------------------------------------------ //

    return (tri);
}
```