Geometric Optimization of Solar Concentrating Collectors using Quasi-Monte Carlo Simulation

by

Andrew James Marston

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2010

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

ABSTRACT

This thesis is a study of the geometric design of solar concentrating collectors. In this work, a numerical optimization methodology was developed and applied to various problems in linear solar concentrator design, in order to examine overall optimization success as well as the effect of various strategies for improving computational efficiency.

Optimization is performed with the goal of identifying the concentrator geometry that results in the greatest fraction of incoming solar radiation absorbed at the receiver surface, for a given collector configuration. Surfaces are parametrically represented in two-dimensions, and objective function evaluations are performed using various Monte Carlo ray-tracing techniques. Design optimization is performed using a gradient-based search scheme, with the gradient approximated through finite-difference estimation and updates based on the direction of steepest-descent.

The developed geometric optimization methodology was found to perform with mixed success for the given test problems. In general, in every case a significant improvement in performance was achieved over that of the initial design guess, however, in certain cases, the quality of the identified optimal geometry depended on the quality of the initial guess. It was found that, through the use of randomized quasi-Monte Carlo, instead of traditional Monte Carlo, overall computational time to converge is reduced significantly, with times typically reduced by a factor of four to six for problems assuming perfect optics, and by a factor of about 2.5 for problems assuming realistic optical properties.

It was concluded that the application of numerical optimization to the design of solar concentrating collectors merits additional research, especially given the improvements possible through quasi-Monte Carlo techniques.

ACKNOWLEDGEMENTS

There are a number of people to whom I am thoroughly indebted for their support and guidance over the past couple of years:

To my parents, Donna and Jim, for their love and encouragement, and to Kim and Nathan, for providing a rent-free environment, without which I would likely have starved.

To Professors Kyle Daun and Michael Collins for their guidance and expertise, and for granting me the opportunity to better myself. Also, for allowing me to take far too many vacations.

To my snuggle-bunny Chelsea, for always being there for me, and for giving me a reason to take all those vacations.

To the various members of STEP, Team North, and the Solar Lab who provided a continuous supply of excellent times, both on and off campus.

I would also like to extend thanks to the late Dr. M. Chandrashekar, whose memorial award has allowed me some added comfort in this past year.

Finally, I would like to acknowledge both NSERC and the Government of Ontario whose Discovery Grant and Early Researcher Award formed the funding for this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A, B	heuristics for logarithmic sample size increase
A_a	area of collector opening aperture
A_r	area of receiver aperture
B	Bernstein polynomial
a, b	endpoints for linear section of concentrator
a, b	exponents for KW diminishing sequences
a, b, c	step size for three-point-pattern search (Section 5.4.3 only)
C	points along the Bezier curve
C	concentration ratio
c	coefficient for error trend expression
d	ray direction vector
ê	unit direction vector
F	vector for Newton-Raphson root finding
F	objective function
$F_{1\text{-}2}$	view factor from surface 1 to surface 2
$f(x_j)$	binary outcome of ray originating at position x_j
$g(a)$, $g(b)$, $g(\mathrm{c})$	objective function for three-point-pattern search
h	difference parameter
i	design dimension
K	scalar value for sample size control condition
k	iteration number
M	number of sub-trials in an MC or RQMC estimation
m	polynomial degree of Bezier curve
N	sample size
n	surface normal vector
n	problem dimensionality (number of design parameters)
$\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$	vectors of coordinates for control points (Chapter 3 only)
$\mathbf{P}_k$	descent direction vector
P	cumulative probability distribution function
R	distance from the centre of the sun (m) (Chapter 1 only)

R	uniformly-distributed random number between zero and one
$\mathbf{r}$	ray vector
$\mathbf{r''}$	reflected ray vector
$\mathbf{r_o}$	ray origin
r	radius of the sun (m)
s	integral dimensionality
U	expected improvement of design update
u	parameter for position along concentrator surface
w	parameter for position along ray
$\mathbf{x}$	vector containing ray-trace parameters u and w
x_j	sampled value for MC or RQMC evaluations
y	length of vertical surface for example view factor calculation
~ (tilde)	indicates quantity is an estimate, subject to uncertainty

Greek symbols:

α	surface absorptivity
α^*	minimizing step size (found through quadratic line search)
α_k	step size
β	convergence rate (for error trend)
Δ	initial search step size (for three-point-pattern search)
ΔF	divided-difference term
δ	value for random shift
γ	expansion parameter (for three-point-pattern search)
ε	estimate uncertainty
$\mathbf{\Phi}$	vector of design parameters
$\mathbf{\Phi}^*$	vector of optimal design parameters
Φ	design parameter
σ	standard deviation
θ	solar half-angle, angle of ray incidence/reflection
θ_c	collector acceptance half-angle
ρ	surface reflectivity

Subscripts:

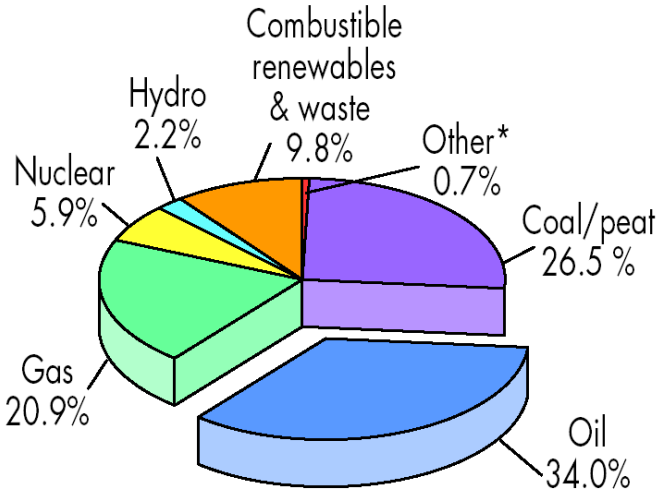
est	estimated
exp	experimental
MC	Monte Carlo
QMC	quasi-Monte Carlo
RQMC	randomized quasi-Monte Carlo
st	stochastic
TS	Taylor Series

Chapter 1

INTRODUCTION

1.1 MOTIVATION

As the worldwide population continues to grow and societies continue to industrialize, the overall demand for energy increases. From 1973 to 2007, the total primary energy supply increased from 6115 to 12 029 million tonnes of oil equivalent (Mtoe), the latter roughly equal to 500 EJ (500×10^{18} J) of energy (IEA 2009). Although a variety of means are currently employed to meet these growing energy demands, the vast majority is met through the use of fossil fuels. As shown in Figure 1.1, coal, oil, and gas make up 81.4% of the total global energy supply, while energy from renewable sources such as geothermal, wind, and solar make up only 0.7%.



* Other includes geothermal, solar, wind, etc.

FIGURE 1.1 - GLOBAL SHARE OF TOTAL PRIMARY ENERGY SUPPLY, 2007 (IEA 2009)

The various consequences of energy production and consumption, at the massive scales undertaken, play a significant role in many important global phenomena. It is generally accepted among the scientific community that there is a link between greenhouse-gas (GHG) emissions and climate change. In its *Fourth*

Assessment Report, the Intergovernmental Panel on Climate Change (IPCC) concluded that "most of the observed increase in global average temperatures since the mid-20th century is *very likely* (i.e. with a probability of over 90%) due to the observed increase in anthropogenic greenhouse-gas concentrations" (2007). As shown in Figure 1.2, carbon dioxide (CO_2) from fossil fuel use accounts for 56.6% of total global GHG emissions. It is hard to deny, then, that energy production from fossil fuels plays a significant role in global climate change. Although it is difficult to accurately predict the effects of climate change and to define tolerable greenhouse-gas levels, there is a consensus among scientists that action should be taken to reduce emissions.

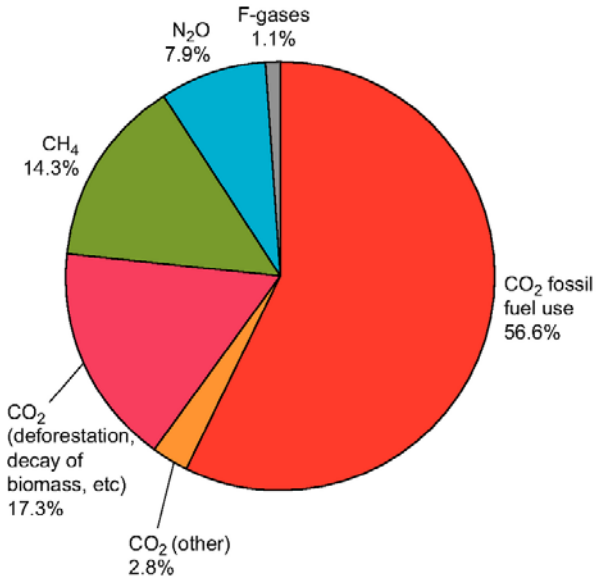


FIGURE 1.2 - GLOBAL ANTHROPOGENIC GREENHOUSE GAS EMISSIONS IN 2004 (IPCC 2007).

In addition to the issues of global warming and climate change, there are other more obvious and immediate problems associated with the use of fossil fuels. Many fossil fuel power plants release various other forms of air pollutants, directly impacting environmental and human health. Coal-fired power plants, in particular, are among the largest contributors to the unhealthy levels of ozone, mercury, and particulate pollution which as many as sixty percent of Americans are exposed to (American Lung Association 2009). Other social and environmental issues are associated with the acts of mining, processing, and transportation of fossil fuels.

Finally, issues of economic and political instability often arise due to reliance on foreign supplies of oil.

There are many strategies that can help reduce the use of fossil fuels, from overall energy conservation to the use of alternative energy sources. Options such as nuclear power or carbon capture and sequestration (CCS) offer GHG-free possibilities for energy production, but not without associated negative environmental implications as well as uncertainties with respect to issues of safety or reliability. Indeed, no alternative currently exists to which no single social, environmental, or moral objection may be made. It can be argued, however, that many of the sustainable sources of energy (wind, solar, hydro, geothermal, wave, biomass, etc.) offer merit as alternatives that have, under the right conditions, limited negative implications. Furthermore, as technology continues to improve, many of these sustainable alternatives are becoming competitive with fossil fuel sources on an economic basis.

This thesis focuses on the improvement of the state of the art in solar concentrating technology, through the development of a solar concentrating collector geometric design optimization methodology.

1.2 SOLAR ENERGY

The powerful presence of the sun is hard to ignore in one's everyday life: indeed, the majority of life on Earth could not exist without its vast output of radiant energy. At any given moment, the Earth's upper atmosphere receives solar radiation amounting to 174×10^{15} W (or 174 PW) of power. As shown in Figure 1.3, about 55% of this reaches the Earth's surface and is either absorbed or reflected by land and oceans. With such a vast amount of solar energy available, humanity could meet its demands by harnessing just a small fraction of this. Indeed, the total annual solar radiation falling on the Earth is more than 7500 times greater than the world's total annual primary energy consumption (WEC 2007). Furthermore, unlike fossil fuels, solar energy will continue to be available for billions of years.

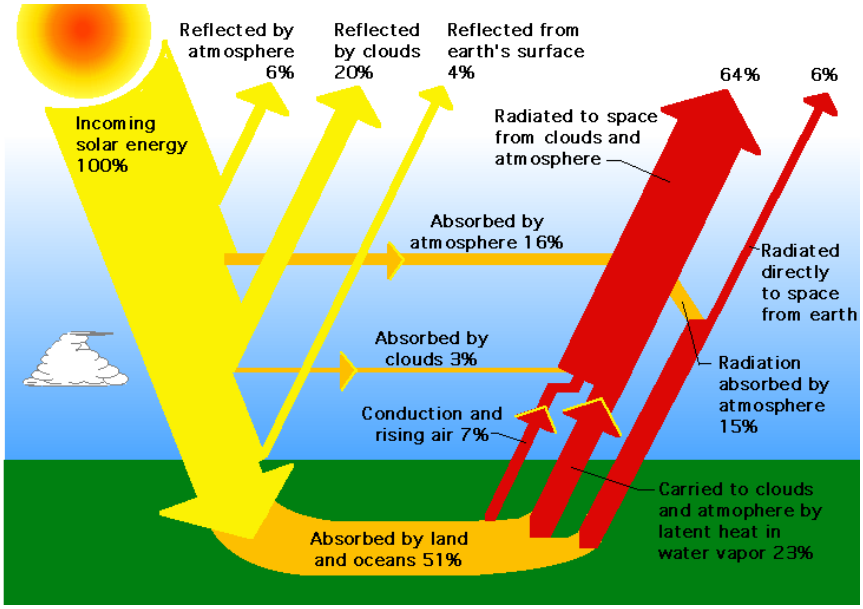


FIGURE 1.3 - EARTH'S ENERGY BUDGET (NASA 2005)

Although there are practical limitations to the amount of solar energy humanity can utilize, it is clear that it is an attractive and sustainable source of energy. Currently, solar energy is captured as either solar photovoltaic or solar thermal energy.

1.2.1 Solar Photovoltaic (PV) Energy

Solar photovoltaic (PV) cells are semiconductor devices that convert part of the incident solar radiation directly into electrical energy. With little requirement for maintenance and the ability to be constructed as stand-alone systems with outputs ranging from microwatts to megawatts, PV cells are useful in a variety of applications, from powering calculators to larger-scale systems for power generation. With such a wide variety of applications and increasing instances of government incentives, the demand for photovoltaics is increasing every year. In 2005, over 1700 MW (peak) worth of PV panels were sold for terrestrial uses, with a growth rate of about 35% per year worldwide (WEC 2007).

There are a variety of factors which constrain the performance of solar PV cells. Each type of semiconductor material used has a maximum wavelength of solar radiation that causes an electron to be freed and a current to be induced in the

connected circuit. Considering this factor alone, silicon cells, for example, have a maximum theoretical performance of 23% (Duffie and Beckman 2006). Newer thin-film technologies are emerging made up of a variety of semiconductor materials, allowing for higher-efficiency multi-layer, or multi-junction, cells to overcome some of these limitations. Because these multi-junction cells show even higher efficiencies when exposed to concentrated sunlight, the relatively new domain of concentrating photovoltaics (CPV) is emerging. In locations with high direct insolation, multi-junction concentrator technology has the potential to reduce the cost of solar electricity by about a factor of two (WEC 2007).

1.2.2 Solar Thermal Energy

Although sunlight may be converted directly into electricity via photovoltaics, it is more easily and efficiently captured as thermal energy. While PV cells can have efficiencies as high as 25-30 %, some solar thermal collectors can capture up to 80 % of incoming solar energy. There are numerous ways in which solar thermal energy can be utilized: simple passive strategies exist in which a mass is exposed to sunlight so that it can store heat during the day and release it at night; several forms of active solar collectors also exist which capture and transport solar thermal energy using a circulated fluid medium. While simple flat plate collectors are useful for things like supplying building space heat or domestic hot water, concentrating collectors are often used to obtain higher temperature heat, for use in industrial processes or large-scale electricity generation (via a heat engine).

1.2.3 Combined Photovoltaic/Thermal (PV/T) Energy

In general, the performance of photovoltaic cells decreases as the operating temperature increases. Depending on the arrangement of the system, it can be worthwhile to implement some form of active cooling in order to keep temperatures down and improve cell efficiency. In some cases, combined photovoltaic/thermal (PV/T) collectors can be built that harness this excess heat while increasing electricity production, improving overall solar energy capture to 40-50 %. This

strategy becomes increasingly important in concentrating photovoltaics, as PV cells may reach temperatures high enough to cause them permanent damage.

1.3 Concentrating Solar Energy

Concentrating solar energy refers to the use of reflecting or refracting optical devices to focus or redirect incoming solar radiation onto a receiver, usually to improve performance and/or economics. Because a major cost of a PV installation are the PV modules, for example, concentrating photovoltaic (CPV) systems take advantage of cheaper optical materials to effectively expose a given area of PV to a greater amount of sunlight, reducing the overall cost-per-kW. Along similar lines, concentrating solar thermal (CST) collectors allow greater levels of radiation to be collected by smaller receivers, enabling higher receiver temperatures and reducing surface area over which heat loss occurs. Systems of CST collectors can be scaled to provide large quantities of heat and designed to achieve a wide range of temperatures, well-suited to various industrial or power generation processes.

Many different concentrating collector arrangements and geometries can be used, typically chosen based on a desired degree of concentration, which is often a result of the required absorber temperature or the characteristics of the PV material. The degree of concentration of a system is most commonly stated as a geometrical concentration ratio (C), defined as

$$C = \frac{A_a}{A_r} \tag{1.1}$$

where A_a and A_r are the areas of the collector aperture and receiver, respectively. This ratio serves as an approximate factor by which the radiative flux is increased by the system, although the true concentration ratio is typically lower, due to non-ideal geometric and optical properties. Often expressed as a "number of suns", the concentration ratio can vary over several orders of magnitude with the numerous possible collector configurations.

1.3.1 LIMITS TO SOLAR CONCENTRATION

Just as with any energy conversion process, the laws of thermodynamics cannot be broken. In the case of collecting energy from the sun, then, the second law of thermodynamics implies that the maximum achievable radiative flux cannot exceed that which is found at the source of the radiation, the surface of the sun (Winston et al. 2005). By taking the sun as a spherically symmetric source of radiation, energy conservation dictates that the radiant flux decrease with $1/R^2$, where R is the distance from the centre of the sun. If we take r to be the radius of the sun, the flux on the earth's surface is smaller than that on the sun's surface by a factor of $(r/R)^2$, where R is the distance from the centre of the sun to the surface of the earth. From Figure 1.4, basic geometry shows that $r/R = \sin\theta$, where θ is the half angle of the sun. The maximum possible concentration ratio on the surface of the earth, then, is given by $C_{max} = 1/\sin^2\theta$, known as the sine law of concentration limit. This limit is achievable in theory through perfect concentration in both transverse dimensions, that is, by using a 3D, or circular, concentrator. Through a similar development, an ideal 2D, or linear, concentrator has a theoretical limit of $C_{max} = 1/\sin\theta$. Based on a solar half-angle of 0.27°, these maximum concentration ratios are about 45 000 and 212 for a 3D and 2D concentrator, respectively (Duffie and Beckman 2006).

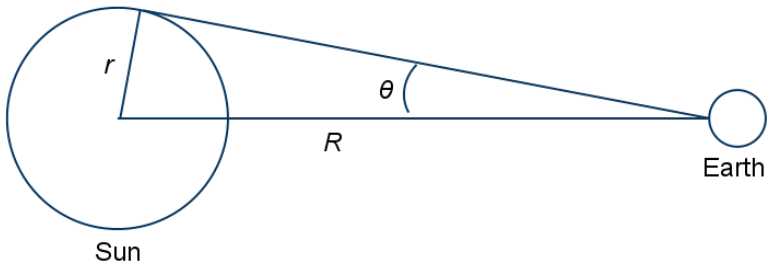


FIGURE 1.4 - GEOMETRY OF THE SUN AND EARTH

1.3.2 TYPES OF CONCENTRATORS

Various concentrator geometries exist to suit a variety of potential receiver requirements. Some of the more familiar types are the imaging concentrators, which form an exact image of the sun on the receiver surface. Parabolic mirrors fall

into this category, and are commonly used in both 2D and 3D geometries to achieve moderate levels of concentration. Such concentrators are designed to be pointed directly at the sun, and so generally require some form of solar tracking. Because, in the collection of solar energy, producing an exact image of the sun is not required, so-called non-imaging concentrators can be designed to suit a more broad range of applications. Fresnel-type concentrators are capable of achieving performance comparable to that of their parabolic counterparts, but with greatly reduced mirror complexity. Other non-imaging concentrators can achieve low-to-medium concentration ratios without the need for solar tracking, making them much more affordable. It also turns out that, by removing the requirement for an exact image, the compound parabolic concentrator (CPC) has been proven theoretically capable of achieve the sine-law limit to concentration. A more detailed discussion of each of these concentrator types is found in Appendix A.

1.4 Concentrator Shape Design

As described, concentrating collectors can be employed to increase the performance and viability of solar energy for a variety of applications. Although much work has already gone into the design and testing of an assortment of possible concentrator geometries, with each new application and change in technology come unique design requirements and increased room for improvement. This is especially true in the relatively new areas of stationary collection and concentrating photovoltaics. With so much potential for novel applications, the development of a fully customizable and straightforward procedure for determining optimal concentrator geometry would be of great benefit.

Traditionally, reflecting-surface devices are designed using a combination of "trial-and-error" and experimental or numerical analytical techniques, resulting in an acceptable though usually sub-optimal design and taking a great deal of time. The relatively new field of non-imaging optics can be applied to the design of theoretically ideal concentrators, but its methods are either limited to a narrow set of problems or require advanced mathematical skills. Recently, much work has gone

into the application of optimization methods to the design of reflecting-surface devices, enabling optimal designs to be reached in a reasonable amount of time and without the need for any complex or limiting techniques. This thesis presents a methodology for the geometric optimization of solar concentrating collectors.

The objectives of this work are:

- To develop a methodology for the geometric optimization of solar concentrating collectors, and demonstrate it on a number of practical problems
- To investigate ways in which performance of the developed optimization algorithm may be improved, in terms of both reliability and computational efficiency

1.5 Outline of Thesis

In the past, a variety of techniques have been used to design the geometry of systems of reflecting surfaces. Chapter 2 provides a review of the literature, in which methods derived from concepts in non-imaging optics, as well as those which employ optimization techniques, are discussed.

One prerequisite to a solar concentrating collector geometric optimization routine is a way of defining collector geometry in terms of a set of *design parameters*. In this thesis, this is accomplished through parametric surface representation using either Bezier splines or points connected by straight lines. Chapter 3 provides a discussion of the necessary theory for such parametric surface representation. The three test problems examined throughout this thesis are also presented. Finally, a brief discussion of the theory necessary for ray-tracing each type of surface is provided.

Another required feature is an automated way of evaluating design performance through an *objective function*, so that designs may be compared to one another and intelligent design updates can be made. In this thesis, the goal is to arrive at the concentrator geometry which results in the greatest fraction of incoming radiation that is absorbed at the receiver surface. This objective is

evaluated using Monte Carlo ray-tracing techniques, which rely on stochastic sampling to simulate radiative heat transfer within a system. In order to reduce the amount of stochastic uncertainty inherent in such evaluations, quasi-Monte Carlo methods are employed, which use quasi-random sequences to reduce estimate variance. Chapter 4 provides a discussion of Monte Carlo and quasi-Monte Carlo ray-tracing techniques, and their associated uncertainties. Both techniques are applied to the three geometries introduced in Chapter 3, and the resulting levels of uncertainty are compared and discussed.

In this thesis, a non-linear optimization technique is applied, in which design updates are performed based on the local objective function gradient. Because objective function evaluations are subject to uncertainty, the Kiefer-Wolfowitz method, a strategy for optimizing stochastic systems, is applied. Chapter 5 presents a discussion of this approach, which uses finite-differences to estimate the gradient and performs steepest descent updates to decrease the objective function. In an attempt to improve algorithm performance and reduce dependence on designer-specified heuristics, modifications to the conventional step-size and sample-size control strategies are presented. Finally, the overall design optimization methodology is developed and then demonstrated on the three test problems.

Although the optimization methodology developed in this thesis proves quite successful for the cases considered, there are a number of aspects which warrant further development. Chapter 6 provides a discussion of these elements.

Chapter 2

LITERATURE REVIEW

Systems of reflecting or refracting surfaces are commonly used to redirect or concentrate thermal radiation in a variety of applications. Although this thesis focuses on the geometric design of solar concentrating collectors in particular, many key characteristics are shared with shape design problems of illuminating devices, radiative furnaces, telescopes, and radiation detectors for applications in physics. In general, any device which consists of surfaces for redirecting radiation from a source to a target can be designed using many of the same tools. This chapter provides a review of the concepts and techniques currently used in the shape design of these devices.

2.1 NON-IMAGING OPTICS TECHNIQUES

The field of non-imaging optics is a relatively new branch of optical physics which began in the 1960's with the introduction of the concentrating parabolic concentrator (CPC), originally developed for the detection of faint radiation in high-energy particle physics experiments (Baranov 1965, Hinterberger and Winston 1966). Until such non-imaging concentrating collectors were introduced, no design had proven theoretically capable of achieving the sine-law limit to concentration. Furthermore, the maximum possible concentration ratio for solar collection without diurnal tracking, based on conventional imaging concentrator technology, was shown to be three (or four, if a second concentrator stage is employed) (Tabor 1958). In 1974, Winston showed that non-imaging concentrators can exceed this, as demonstrated with a trough-CPC of concentration ratio 9.6, capable of collecting on average 8 hours of sunlight per day without diurnal tracking. In addition, a substantial amount of diffuse insolation is captured (a fraction equal to the reciprocal of the concentration ratio), much of which is lost in conventional imaging concentrators. Rabl (1976a)

subsequently published a study comparing important characteristics of conventional parabolic or Fresnel concentrators to the new non-imaging CPC, in which various advantages of the CPC are described. These developments shook the field of optical physics, and much effort was put into the development and application of non-imaging optics techniques.

2.1.1 EDGE-RAY ("STRING") METHOD

One of the simpler non-imaging optics techniques known as the edge-ray, or "string", method has been used extensively in the design of non-imaging optical devices, and is explained thoroughly in Rabl (1994), Ries and Rabl (1994) and Winston et al. (2005). For simple solar collection, for example, the particular distribution of radiation is not important, only whether or not it reaches its goal. The edge-ray method exploits the observation that in order to redirect a beam of radiation onto a region, it is sufficient to simply redirect the boundaries of the beam onto the boundaries of the region. So, by designing a concentrating collector that reflects the boundary or "edge" rays of the source onto the boundary of the target, all other rays from the source will also reach the target.

The geometry of the concentrating parabolic reflector (CPC) serves as a good example to illustrate the basics of the edge-ray principle. As shown in Figure 2.1, each side of the CPC is a parabola rotated and shifted so that any rays incoming at the acceptance half-angle θ_c are focused, after a single reflection, directly onto the opposite edge of the receiver. As a result, any rays entering the collector aperture at an angle of less than θ_c will be reflected onto the receiver somewhere within its edges, and any entering at angles greater than θ_c will be reflected onto the opposite reflecting surface and thereby rejected.

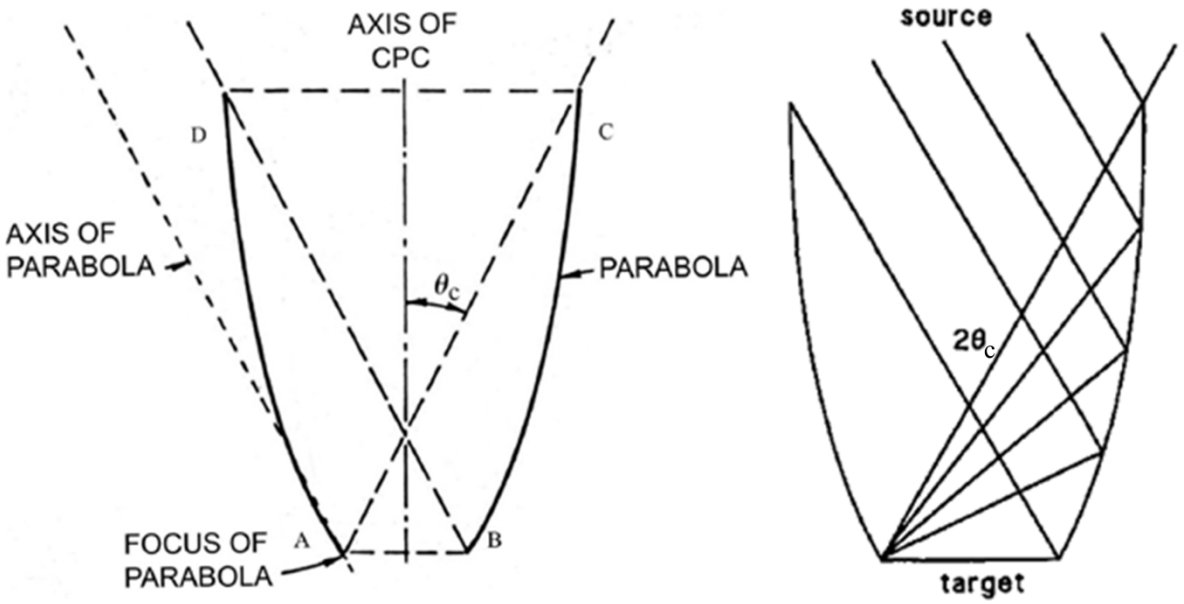


FIGURE 2.1 - THE EDGE-RAY PRINCIPLE AND THE CPC (RIES AND RABL 1994)

The edge-ray method can be applied to the design of symmetric or asymmetric concentrators for an absorber of any non-concave shape. Winston and Hinterberger (1975) apply the principles of the method to the design of ideal linear concentrators for the different absorber geometries including cylindrical tube and vertical fin absorbers. It should be noted that the basic edge-ray method cannot be used to specify reflector geometry in areas located in the shadow of the absorber. This geometry is instead defined by an involute, which is the shape drawn by unwrapping a taut string from around the absorber (Rabl 1976b). The edge-ray method has been used to design several asymmetric CPCs, such as that presented by Mallick et al. (2004) for building-integrated PV concentrators, or those presented by Adsten et al. (2005) for stationary collection in high-latitude locations.

Generally, application of the edge-ray method results in a differential equation for each side of the reflector shape that will maximize collection in two dimensions. As an alternative, Winston et al. (2005) provide an easy-to-follow formulation of the technique called the "string" method, in which a rod and string can be used to physically trace out the required concentrator surface. The edge-ray method has the advantage of being relatively simple, but is limited to a

narrow set of design problems, namely the concentration of a light beam with a given angular spread from one area onto another, under the assumptions of homogeneous irradiance and perfectly specular reflection. Although it is known to yield optimal reflector geometries in these cases, these are often not reasonable assumptions in practice.

2.1.2 TAILORED EDGE-RAY APPROACH

In response to the limited application of the simple edge-ray method to problems of homogenous irradiation, the basic principles have been extended to handle the design of reflecting surfaces to achieve some desired radiation distribution. The tailored edge-ray approach was first used in lighting design to achieve uniform angular distribution of light from a given source. Unlike the basic edge-ray method described above, which uses only a single set of edge-rays to completely determine the reflector profile (resulting in a collector with a single acceptance angle), the tailored edge-ray method solves the reflector geometry for the entire family of edge-rays, which can be defined to achieve a desired irradiation. Winston and Ries (1993) show that by first deriving the required acceptance-angle function based on the radiation distribution at the source and the desired distribution on the target, it is possible to solve for the required reflector shape in closed form. Ries and Winston (1994) apply the method to produce a reflector that provides constant irradiance to ±43° from a cylindrical source of constant brightness. In addition to the design of illumination devices, the methodology also applies to the design of concentrators. Gordon and Ries (1993) and Gordon (1996) apply the method to design a tailored edge-ray collector for secondary concentration in 2D and 3D parabolic or Fresnel primary systems. Despite the improvements the tailored-edge ray approach offers, it still relies on limiting assumptions of perfect optics.

2.1.3 FLOW-LINE METHOD

Another commonly used non-imaging optics technique is the flow-line method, also known as the geometric vector-flux approach, in which the propagation of light is defined and manipulated in "phase space". In analogy with fluid dynamics, phase space consists of both the position and momentum of elements of a fluid, and thus has twice the dimensions of ordinary space. In optics, a light ray's momentum is its direction multiplied by the index of refraction of the medium through which it "flows". The concept of phase space is useful because the physical laws of radiation propagation result in quantities that must remain conserved, namely the geometric vector flux, analogous to the conservation of volume in incompressible fluid flow.

By expressing the propagation of radiation in this way, it is possible to map the flow lines in vector phase space from a given Lambertian (uniformly-emitting) source, and from these deduce possible reflecting geometries that conserve the geometric vector flux. Winston and Welford (1979a and 1979b) describe the overall methodology in detail, and give various examples of its application. Winston et al. (2005) show that by inserting a reflective surface along a set of flow lines from a Lambertian disc, for instance, one arrives at the hyperboloid of revolution as shown in Figure 2.2, which is an ideal 3D concentrator (though it relies on many instances of multiple reflection). Although the flow-line method has definite merit in certain circumstances, it, too, relies on assumptions of perfect optics, and in addition is limited to those with a high level of mathematical knowledge and training.

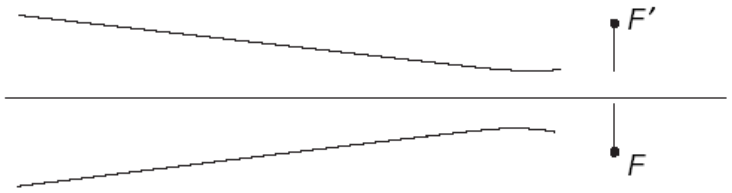


FIGURE 2.2 - HYPERBOLOID OF REVOLUTION (WINSTON ET AL. 2005)

2.2 Design via "Trial and Error" and Parametric Studies

Traditionally, the design of radiative transfer devices consists of a trial-and-error approach, in which a design is proposed based on some previous knowledge or experience, and a prototype is built and tested. If the proposed design does not perform to a satisfactory level, the process is repeated. If enough time and resources are available, a detailed study can be performed on a few select design variables, in an attempt to expose important trends in performance and gain insight into possible design improvements. In general, such approaches are time-consuming, expensive, and only reach an acceptable, and not necessarily optimal, solution. With the advent of computers came various techniques for quickly and accurately calculating heat transfer, allowing potential designs to be evaluated without the need for physical prototyping, and so greatly reducing the requirement for time and resources. Many reflecting surface design improvements have been reached as a result of such studies.

Ryan et al. (1998), for example, use a Monte Carlo radiative heat transfer technique to perform a series of parametric studies on cylindrical solar collector arrays, taking into account directional material properties, as well as both specular and diffuse components of reflection. For a given collector geometry, yearly thermal performance is estimated by using Monte Carlo to calculate the optical performance over a range of possible incidence angles, and inputting the results into a software package for transient energy system simulation. Simulations are performed for collectors with varying array size, tube spacing, and material properties of cover, absorber, and back-plane reflector, examining the effects of one variable at a time. An improved design is suggested from the overall results.

Along the same lines, Haeberle et al. (2007) perform a study on a linear Fresnel collector with a secondary non-imaging concentrator and cylindrical absorber. A commercial ray-tracing software package is used to estimate optical

performance for a range of incidence angles, taking into account material absorptivity and reflectivity, mirror manufacturing and alignment error, and a realistic solar disc profile. As above, a transient simulation tool is used to estimate yearly thermal performance given the resulting optical efficiencies. Based on results for collectors of varying length, an optimal length is deduced.

The above univariate studies neglect any nonlinear interactions between the design variables, and as such are not typically of much use for complex designs. Muschaweck et al. (2000) perform a more complex study in order to optimize reflector geometry for a non-tracking linear CPC solar collector, considering up to two variables at a time. With the objective of investigating how real-world effects may make the optimal reflector geometry differ from the solution predicted by non-imaging optics, designs are compared based on a maximum yearly average utilizable power per absorber area. This is calculated by first determining the concentration ratio for various directions using ray-tracing, and using the results in a simplified time-integration, assuming constant thermal losses. Examining the acceptance angle of the left and right sides of an asymmetrical CPC, the design topography is mapped over the two-dimensional design space, limited to a rectangular domain surrounding the ideal edge-ray solution. The optimal design is then selected from the region(s) with the best performance, identified through inspection. Although this strategy includes nonlinear interaction between variables, the required computation increases greatly with the size of design space and number of variables treated. Furthermore, once the number of variables exceeds two, it is not possible to provide a simple visualization of the design topography, making identification of an optimal region more complicated.

2.3 Design Optimization Methods

Through the application of known numerical optimization techniques, it is possible to treat design problems of greater dimension and complexity than could be reasonably handled using traditional methods. For relatively simple

cases, it may be possible to perform analytical optimization arriving exactly at an optimal design. The majority of problems of interest, however, rely on a combination of computational heat transfer techniques and nonlinear optimization schemes to arrive at an approximate optimum. In order to formulate a problem in reflecting surface shape design as an optimization problem, the first step is to define an objective function whose minimum corresponds to the ideal design performance. The objective function must be dependent upon on a set of design parameters that represent the shape of a design, and may be evaluated using various ray-tracing or numerical radiative heat transfer techniques. Two basic formulations commonly are used in this context: for maximum power collection, the objective can simply be to minimize radiation that enters the system but does not arrive at the intended target; to achieve some desired irradiance distribution, on the other hand, the goal can be to minimize some function which measures the deviation between the resulting and desired distributions. Finally, an optimization technique must be employed to search the design space and arrive at an optimal design configuration.

Generally, optimization algorithms are classified as either: metaheuristic, which rely on some means of random sampling of the objective function over the design space in order to search for the global minimum, or; gradient-based, which progressively reduce the objective function in a deterministic manner based on some local measurement of its curvature.

2.3.1 METAHEURISTIC OPTIMIZATION METHODS

Metaheuristic (also referred to as "global") search schemes enable the intelligent search of a design space (often limited to a distinct region to reduce computation), generally without requiring input of a set of initial design parameters. Because searches occur over the entire design space and without the bias of an initial guess, such schemes have the potential to arrive at novel solutions and are in theory capable of identifying the global optimum, even for problems in which many local optima exist. Generally, metaheuristic algorithms

are best-suited to problems involving discrete design variables, although they can sometimes be adapted to those with continuously variable design parameters. In addition, they are typically slower for problems involving computationally-expensive objective function evaluations, as they rely on a large sampling over the design space. A "total enumeration" search is the simplest example of such a scheme, in which every possible configuration (or some sufficiently fine sampling thereof) is tested, and that with the best performance is chosen. This strategy is obviously computationally inefficient and impractical for anything but the simplest of problems involving a low number of design parameters. Various algorithms have been proposed in an attempt to search a design space in a more intelligent and efficient manner.

Genetic, or evolutionary, algorithms are one such global optimization scheme that has seen much application in reflecting surface design. In a genetic algorithm (GA), design parameters are each assigned to a "gene", and potential designs are generated by random "mating" of "parent" designs. In a way that is meant to mimic natural selection, each design generation is ranked according to "fitness", and when random mating occurs, parents who are deemed more fit have a greater chance of passing on their genes. After many rounds of repeated reproduction and elimination, the routine should converge to a family of one or more optimal designs. In theory, because the routine is initiated with a randomly generated set of parent designs, and because superior performance is favoured without being strictly required, GAs do not get "stuck" in local optima. Furthermore, since there is a chance of random gene "mutation" during mating, there is the possibility of generating novel designs. On the other hand, genetic algorithms generally require a large number of design evaluations, which may prove impractical if each simulation is computationally-expensive to carry out. Moreover, because the algorithm relies on random occurrences, it is not repeatable, and there can be no guarantee that convergence will occur. Despite this, genetic algorithms have been applied to a number of complex design optimization problems with great success. Genetic algorithms are explained in

detail by Goldberg (1989) and Holland (1992), and are first adapted to reflecting surface design problems by Ashdown (1994), in the context of lighting design.

Doyle et al. (1999) apply an evolutionary strategy to the design of a two-dimensional reflector for desired illumination given a point light source. The reflector surface is modeled using a cubic Bezier curve with the x- and y-coordinates of all four control points as design variables, and illumination distributions are calculated using a parametric ray-tracing approach. A custom fitness function is used which takes into account the absolute difference between the desired and resultant lighting distribution (on an element by element basis), as well as the difference between the reflected and desired light power. Generations are mated using a routine known as differential evolution, in which child designs take on randomly selected and mutated genes from parents. If a resulting child design improves in fitness, it replaces its parent in the population. In order to validate the algorithm, it is applied successfully to the design of simple parabolic and elliptical reflectors using their known illumination distributions.

Corcione and Fontana (2003) apply a similar methodology to the design of a lighting configuration for a tennis court and a football field. Design variables are light pole location and height, type, and direction of floodlights on each pole, and are treated discretely. The fitness value takes into account both average illuminance and horizontal uniformity across the field. Optimization is performed a number of times using the total enumeration approach, a genetic algorithm, a Monte Carlo method which considers a random sampling of possible designs, and combined gradient-Monte Carlo and gradient-genetic techniques. A gradient technique is useful because it can treat continuous variables and so does not limit the possible number of designs, although to limit computation it is only applied after many rounds of discrete optimization. It is found that, because of the large design space, problems are best handled by a genetic algorithm, with slight improvements in performance possible by applying a gradient method after-the-fact.

Sarvari (2007) applies a genetic algorithm to the design of two-dimensional radiant enclosures with diffuse-gray walls and containing transparent media, with the goal of producing a desired heat flux distribution over the design surface. Surfaces are represented by cubic B-spline curves, as they are found to strike a balance between complexity of computation and flexibility of curves handled. Radiative heat transfer is solved using the discrete transfer method, in which radiant intensity is solved using the radiative transfer equation for the numerous surface elements and their discretized solid angles. In an attempt to improve convergence performance over regular GAs, a microgenetic algorithm is used which allows for a very small population size (typically ranging from 5 to 8). The proposed method is applied to two surface design problems, each time successfully resulting in a set of designs which satisfy the design goal and problem constraints. It is found that, by first optimizing the coordinates of the B-spline control points, and then post-processing by optimizing the weights used at each, further design improvements could be achieved.

In the context of solar energy collection, genetic algorithms have been applied to problems of optimizing spacing and arrangement of a field of stationary solar thermal collectors (Weinstock and Appelbaum 2007), and optimizing aspects such as payback period or CO_2 emission reduction for a large solar hot water system (Loomans and Visser 2002), though no work has been found dealing with the task of collector shape design.

In addition to genetic algorithms, a number of attempts have been made to apply other global search schemes to problems in reflecting surface design. Patow et al. (2004), for example, apply a custom "brute force" algorithm to a problem in lighting design. Reflecting-surface geometry is represented by a height field (a surface mesh of points with varying heights), and designs are evaluated using a Monte Carlo ray-tracing technique, able to account for realistic irradiance distribution from the light source as well as both specular and diffuse reflection. Original attempts to apply a local descent optimization

method failed due to Monte Carlo estimate noise and the existence of multiple local minima. Instead, a search method is devised which starts with an initial design guess, and relies on repeated rounds of testing sensitivities of performance to each variable, and subsequently generating a new family of designs by iteratively shifting only the most important variables. Although their method is found to result in improved designs, computation is quite slow. Trials with a high fraction of diffuse reflectivity are deemed impossible, as an unreasonable number of rays are required to keep Monte Carlo estimation errors at manageable levels.

2.3.2 Gradient-based Optimization Methods

Gradient-based (also known as "local") optimization methods attempt to hone in on an optimal solution by taking into account first- and sometimes second-order local objective curvature. Such methods are known for their high rates of convergence, although their success typically depends highly on the particular problem curvature and the proximity of the initial guess to the optimum. Local search algorithms identify the nearest significant optimum, and so generally have trouble with problems for which multiple optima exist. A good deal of work has gone into the application of these methods to the solution of reflective surface design problems.

Daun et al. (2003a) apply gradient-based optimization to the design of a diffuse-walled radiant enclosure for uniform irradiation over the target surface. Radiosity distribution is solved using an infinitesimal-area analysis technique (Daun and Hollands 2001), in which a numerical solution is found for the integral equations of radiosity for the given parametrically represented (B-spline) enclosure surfaces. First- and second-order sensitivities are found through solution post-processing, and are used to perform gradient design updates using the steepest descent, Newton, and quasi-Newton methods. Although Newton's method exhibits improved convergence rates, it requires calculation of second-order sensitivities, which can be computationally expensive

to determine. The quasi-Newton method, on the other hand, gradually forms an approximation of the Hessian using objective function gradient detail as it becomes available and thus does not require second-order information. The optimization methodology was successfully applied to a problem of two-dimensional enclosure design, for which Newton's method was found to perform the fastest, with quasi-Newton close behind, followed by steepest descent. These results are attributed in part to the relative ease with which second-order derivates are calculated for the given geometry and analysis technique.

Daun et al. (2003b) again apply a gradient-based optimization technique to the radiant enclosure design problem, this time with both specularly- and diffusely-reflecting surfaces. Radiative flux distribution is solved using Monte Carlo ray-tracing (Siegel and Howell 2002) between surface elements. Because Monte Carlo provides approximations subject to random error, the Kiefer-Wolfowitz method (Kiefer and Wolfowitz 1952) for optimization of stochastic systems is applied. This method performs steepest descent iterations, using divided-differences to estimate the gradient. Diminishing sequences are used to specify step size and central-difference parameter, and sample sizes for Monte Carlo ray-tracing are controlled using a logarithmic relationship based on iteration number and predetermined heuristics (a more detailed discussion of these factors is presented in a later section). The method is successfully applied to the optimization of two two-dimensional radiant enclosure problems, with the identified solutions achieving near-uniform irradiation over the design surface.

Rukolaine (2010) describe a gradient-based method to be used to determine optimal geometry for a similar two-dimensional radiant enclosure problem, for which surfaces are taken to be gray and diffuse. Because the goal is to achieve uniform irradiation over a surface, a least-squares objective functional is defined. Based on this, radiative transfer is solved through inverse analysis of the steady-state radiative transfer equation, and an objective function gradient is formed through shape sensitivity analysis and successive numerical solution of the direct and adjoint problems. The paper focuses on the mathematical

formulation and solution of the objective function gradient, with little recommendation on a particular minimization method.

In this thesis, the geometric design of solar concentrating collectors is examined. As with the majority of previous work dealing with systems of highly-specular surfaces, a Monte Carlo ray-tracing method is employed to solve for radiative heat transfer. Because this method generally leads to relatively computationally expensive evaluations, in order to reduce overall design time, a gradient-based algorithm is used. The optimization methodology presented and applied in this thesis, then, is an extension of that presented by Daun et al. (2003b). Although this thesis focuses on concentrators made up of reflecting surfaces, many of the aspects and techniques introduced also apply to lens-based systems.

Chapter 3

DEFINING COLLECTOR GEOMETRY THROUGH DESIGN PARAMETERS

In order to conveniently facilitate both design updates and objective function evaluations, the optimization routine developed in this thesis relies on parametric representation of both concentrator surfaces and ray trajectories. In the three design optimization cases considered, concentrator surface geometry is defined by a number of points in space whose coordinates are stored in a vector of *design parameters*, **Φ.** These points act either as vertices of a tessellated surface mesh or as control points for B-spline or Bezier curves. In this way, the overall surface geometry can easily be modified by changing the value of the design parameters. Such parametric surface representation is an industry standard technique in computer graphics and CAD. In addition to being easily adaptable, parametric representation also allows for easy objective function evaluation through application of standard ray-tracing techniques. Throughout this thesis, the number of design variables stored in this vector is referred to as the *problem dimensionality*, and is given the symbol *n*.

Throughout this thesis, geometry and analysis are presented as dimensionless, as none of the concepts or methods are dependent on any particular system of units. Although this thesis only presents the theory necessary for solving two-dimensional surfaces, the same basic principles can be applied to three-dimensional problems.

3.1 FACETED SURFACE REPRESENTATION

One of the simplest ways to represent a curve is through tessellation, in which a mesh of planar sections is formed to approximate a given surface. A two-dimensional curve can be represented in this way with a series of points

connected by straight lines. Although such tessellations do not represent a surface with complete accuracy, they often greatly reduce the computational requirement of calculations such as ray-tracing. In solar concentrating collector design, such surface constructions can have further benefits, as resulting designs are typically more easily manufactured than others involving complex curves. Fresnel-type concentrators, for example, could be modeled in this way.

Figure 3.1 shows the geometry of a faceted surface linear concentrating collector that is referred to as "test case two" throughout this thesis. Like the other test geometries to be presented, the two points defining the opening are at a fixed distance of ten units apart, in order to allow an unbiased comparison of design performance. In order to limit problem dimensionality to $n = 3$, symmetry is imposed, so that the remaining three points are defined by the vector of design parameters $\mathbf{\Phi} = [\Phi_1, \Phi_2, \Phi_3]^T$. For design problems based on this case, the objective is to redirect the maximum fraction of incoming radiation onto a cylindrical receiver of radius 0.5, located at the centre of the collector opening. Incoming radiation is taken to be normal to the collector opening, as would be the case for a perfectly-tracking collector.

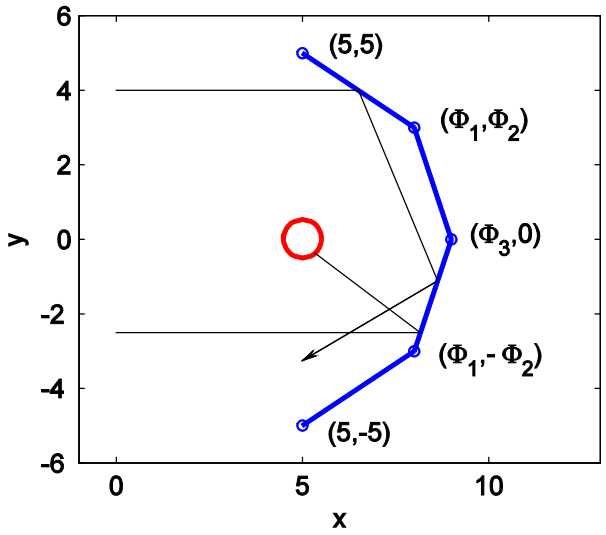


FIGURE 3.1 - TEST CASE 2: FACETED SURFACE LINEAR CONCENTRATOR

3.2 BEZIER CURVE SURFACE REPRESENTATION

More accurate representation of curved surfaces can be achieved through polynomial functions. For computer modeling or simulating, Bezier curves are commonly used, as they are able to model a variety of complex shapes with relatively few specified points. Furthermore, their formulation allows for easy and efficient computation of the coordinates and derivatives of points along the represented curve, which is important for the ray-tracing techniques to be discussed shortly. A great deal of background theory goes into the derivation of Bezier curves, which are a subset of a wider family of parametric representation known as Non-Uniform Rational B-Splines (NURBS); for the purpose of this thesis, only equations necessary for simple two-dimensional curves are presented and applied. For a more comprehensive background, see *The NURBS Book* (Piegl and Tiller 1997).

The coordinates of a Bezier curve, **C**, of polynomial degree m and based on parameter u are defined by

$$\mathbf{C}(u) = \sum_{i=0}^{m} B_{i,m}(u)\mathbf{P}_i, \qquad 0 \le u \le 1 \tag{3.1}$$

where $\mathbf{P}_i$ is a vector of *control points* and $B_{i,m}(u)$ are *basis functions*, defined using the m^{th}-degree Bernstein polynomials given by (Piegl and Tiller 1997)

$$B_{i,m} = \frac{m!}{i!(m-i)!}u^i(1-u)^{m-i} \tag{3.2}$$

As a simple example, a first-order Bezier curve is defined as follows: from Equation (3.2), $B_{0,1} = 1 - u$ and $B_{1,1} = u$; substituting these into Equation (3.1) results in the curve $\mathbf{C}(u) = (1 - u)\,\mathbf{P}_0 + u\,\mathbf{P}_1$. As the parameter u is increased from zero to one, then, the coordinates given by C(u) follow a straight line from $\mathbf{P}_0$ to $\mathbf{P}_1$. Similarly, a second-order Bezier curve is defined by

$$\mathbf{C}(u) = (1-u)^2\mathbf{P}_0 + 2u(1-u)\mathbf{P}_1 + u^2\mathbf{P}_2, \quad 0 \le u \le 1 \tag{3.3}$$

and forms a parabola from $\mathbf{P}_0$ to $\mathbf{P}_2$ with curvature defined by coordinates of $\mathbf{P}_1$. Figure 3.2 shows example (a) second-order (quadratic) and (b) third-order (cubic) Bezier curves.

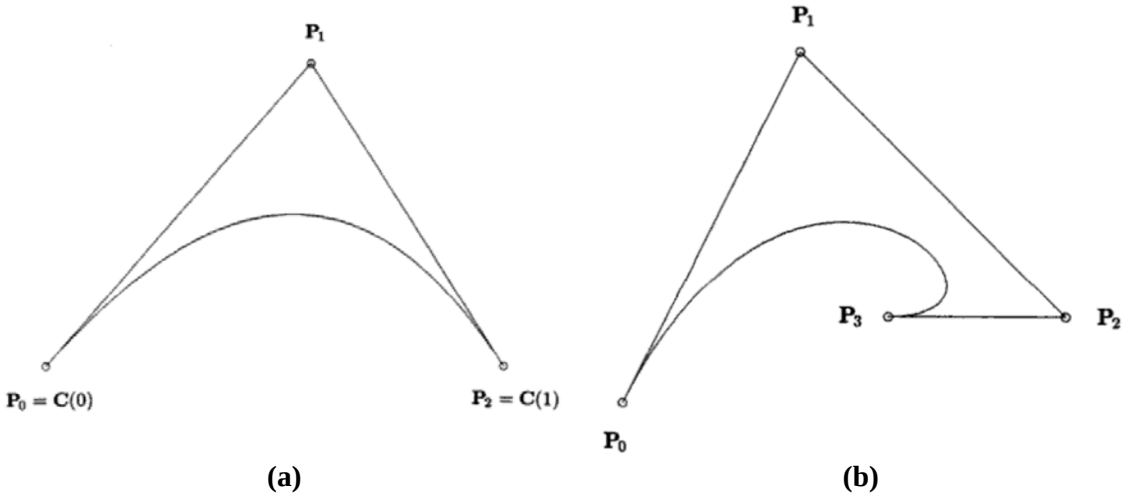


FIGURE 3.2 - (a) SECOND- AND (b) THIRD-ORDER BEZIER CURVES (PIEGL AND TILLER 1997)

Because the control points for a given curve are constants and independent of u, the derivative of the curve can be computed by first calculating the derivatives of the basis functions of Equation (3.2) as

$$\frac{d}{du}B_{i,m}(u) = B'_{i,m}(u) = m(B_{i-1,m-1}(u) - B_{i,m-1}(u)) \tag{3.4}$$

then applying these to the derivative of $\mathbf{C}(u)$ resulting in

$$\frac{d}{du}\mathbf{C}(u) = \mathbf{C}'(u) = \sum_{i=0}^{m-1} B_{i,m-1}(u)\left[m(\mathbf{P}_{i+1} - \mathbf{P}_i)\right] \tag{3.5}$$

This relation for $\mathbf{C}'(u)$ is used in the ray-tracing techniques discussed below.

A convenient property of Bezier curves is that their end points always coincide with the first and last control points, making it easy to fix the collector aperture width. The remaining control points determine the overall curvature, allowing intuitive shape adaptation through modification of their coordinates. Figure 3.3 shows the geometry of "test case one", a linear parabolic concentrating collector modeled by a second-order Bezier curve, with two design

parameters defining coordinates of the second control point, $\mathbf{P}_1$. Note that because both the x- and y-coordinates of this point are variable, this representation does not require symmetry. As with test case two, the receiver is a cylinder of radius 0.5 units and located at the centre of the collector opening, and incoming radiation is taken to be normal to the collector opening.

Figure 3.4 (a) shows the geometry of "test case three", represented by a fourth-order Bezier curve with the three middle control points variable in both the x- and y-dimensions, resulting in a problem with dimensionality $n=6$. In this case, the receiver is a horizontal line two units in length and starting from the rightmost edge of the collector opening. This geometry was inspired by the linear asymmetric CPC proposed by Adsten et al. (2005) for stationary concentration onto a photovoltaic receiver in northern climates, as shown in Figure 3.4 (b). Although in reality such a stationary collector would experience radiation at a variety of angles, in this case it is simplified to be incoming at a fixed angle of thirty degrees from the normal to the opening. In this case, a fourth-order curve is used to allow a variety of unconventional shapes to be considered in the design process, in contrast to the simple curves represented by a second-order spline.

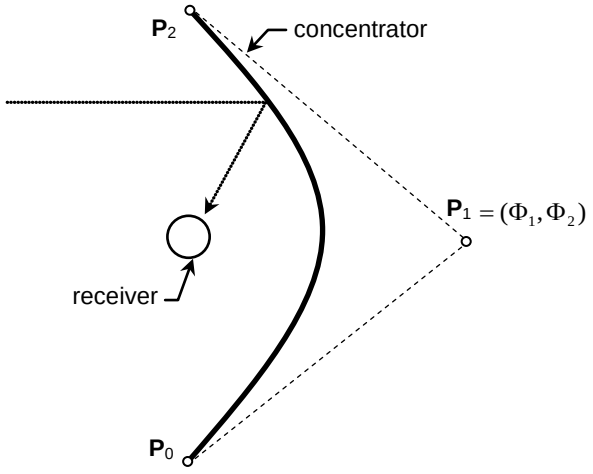


FIGURE 3.3 - TEST CASE ONE, BEZIER REPRESENTATION

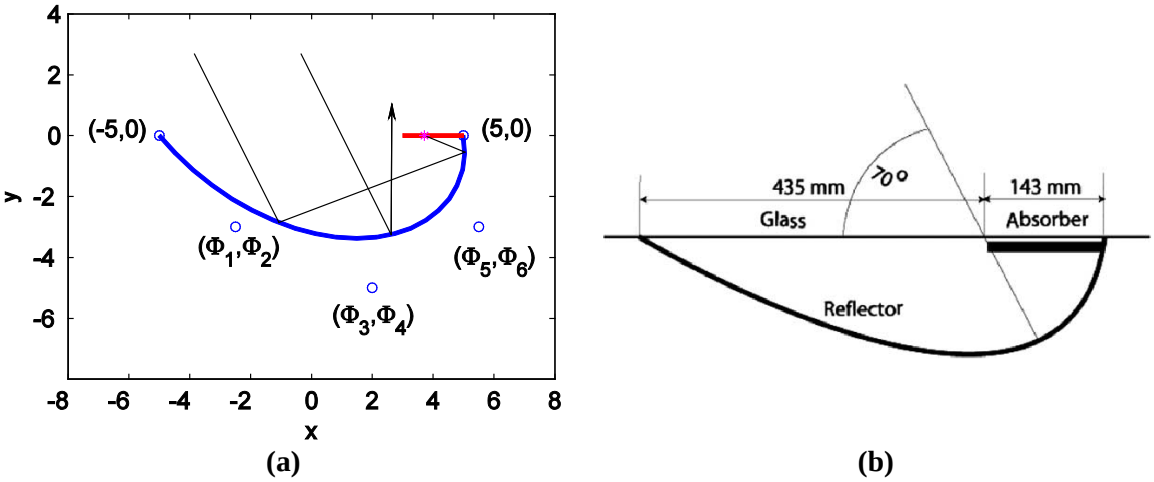


FIGURE 3.4 - TEST CASE THREE, (a) BEZIER REPRESENTATION AND (b) ASYMMETRIC CPC (ADSTEN ET AL. 2005)

3.3 RAY-TRACING

Ray-tracing, the act of following the trajectory of photon bundles through a system of reflecting and/or refracting surfaces, is an important technique for characterizing the performance of optical devices, and is used throughout this thesis to facilitate objective function evaluation. Based on known physical laws and optical properties, it is possible to determine the outcome of a given ray based on its point of origin and direction of movement. As such, rays are conveniently represented in the parametric vector form

$$\mathbf{r}(w) = \mathbf{r}_0 + w\mathbf{d}, \qquad w > 0 \tag{3.6}$$

where $\mathbf{r}$ is the ray vector, $\mathbf{r}_0$ and $\mathbf{d}$ are the point of origin and ray direction, respectively, and w is the ray parameter. As w is increased from zero, the path of the ray from its origin is defined.

There are three basic steps in a ray-tracing procedure: (i) identify the point at which the ray first intersects a surface; (ii) determine the surface normal at this point, and; (iii) calculate the resulting reflected or refracted ray direction. A specular reflection, for example, is computed using the vector algebra shown in Figure 3.5, where $\mathbf{n}$ is the surface normal at the point of intersection, and $\mathbf{r}$ and $\mathbf{r}''$ are the incident and reflected rays, respectively

(Winston et al. 2005). Similar algebra is used for dealing with refraction, based on the well-known Snell's law. These three steps are repeated for each ray until it either is absorbed at a surface or escapes from the system.

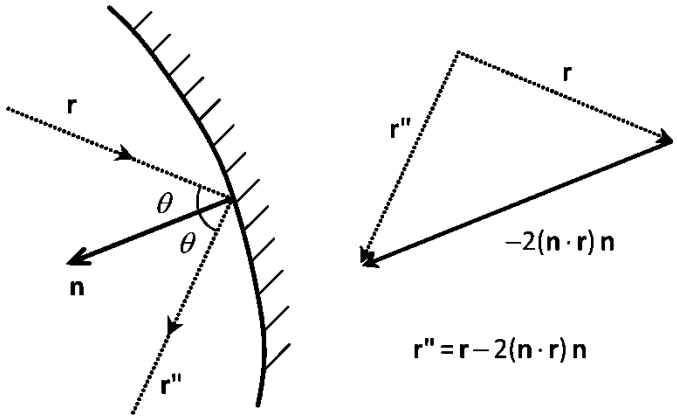


FIGURE 3.5 - RAY VECTOR REFLECTION

Generally, it is the identification of the first point of intersection which proves most complex, especially in systems for which a ray may intersect a surface at multiple locations. The strategy employed depends on the particular method of surface representation used. The main strategies used in this thesis for solving for faceted surface and Bezier curve concentrating collectors are described below. Actual MATLAB functions are provided in Appendix E.

3.3.1 RAY-TRACING FACETED SURFACES

Ray-tracing of surfaces made up of only flat planes or straight lines is relatively fast and simple, as ray-surface intersections and corresponding surface normals can be computed using basic vector algebra. For the 2D systems that are dealt with throughout this thesis, points of intersection between a ray vector and a linear surface section can be found using the following strategy (Press et al. 1992): (*i*) define the linear section $\mathbf{C}(u)$ (with endpoints $\mathbf{a}$ and $\mathbf{b}$) by $\mathbf{C} = \mathbf{a} + s(\mathbf{b} - \mathbf{a})$ and ray vector $\mathbf{r}$ as in Equation (3.6); (*ii*) set the equations for $\mathbf{C}$ and $\mathbf{r}$ equal and solve for parameters u and w, resulting in

$$u = \frac{\mathbf{d} \times (\mathbf{a} - \mathbf{r}_0)}{(\mathbf{b} - \mathbf{a}) \times \mathbf{d}}, \qquad w = \frac{(\mathbf{a} - \mathbf{r}_0) \times (\mathbf{b} - \mathbf{a})}{(\mathbf{b} - \mathbf{a}) \times \mathbf{d}} \tag{3.7}$$

(*iii*) if u is between zero and one, then the ray vector intersects section **C** within its endpoints, at the point given by $\mathbf{C}(u)$ or $\mathbf{r}(w)$. For a collector system made up of multiple surfaces, every possible ray-surface intersection point should be found, and only that with the lowest nonzero w value (corresponding to the first surface hit) should be taken. Intersection points with $w = 0$ correspond to the point of origin of the ray, and so should be neglected. Once the appropriate point of intersection is found, determining the angle of incidence and surface normal is trivial.

3.3.2 Ray-Tracing Bezier Curves

In contrast to the simplicity of faceted surfaces, ray-tracing Bezier curves requires a much greater deal of both computational and programmer effort. Because a Bezier curve is nonlinear, ray-surface intersections cannot be solved through matrix inversion, and must instead be found using a numerical root-finding method. Throughout this thesis, the Newton-Raphson scheme is used with updates based on the current inverse Jacobian, from the basic approach explained in Toth (1985). Points of intersection are found by solving for

$$\mathbf{C}(u) - \mathbf{r}(w) = \mathbf{F}(u, w) = \mathbf{F}(\mathbf{x}) = \mathbf{0} \tag{3.8}$$

where $\mathbf{C}(u)$ is made up of points along the Bezier curve found by applying equations (3.1) and (3.2), $\mathbf{r}(w)$ is the ray vector defined by Equation (3.6), and vector $\mathbf{x}$ contains the two parameters, $\mathbf{x} = (u, w)$. Based on an initial guess at $\mathbf{x}$, Newton-Raphson updates

$$\mathbf{x}_{\text{new}} = \mathbf{x} - \mathrm{J}(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x}) \tag{3.9}$$

are performed until $\mathbf{F}(\mathbf{x})$ is sufficiently close to zero, where $\mathrm{J}(\mathbf{x})$ is the Jacobian given by

$$\mathrm{J}(\mathbf{x}) = \begin{pmatrix} \dfrac{\partial F_x}{\partial u} & \dfrac{\partial F_x}{\partial w} \\ \dfrac{\partial F_y}{\partial u} & \dfrac{\partial F_y}{\partial w} \end{pmatrix} = \begin{pmatrix} \mathbf{C}'_x(u) & \mathbf{r}'_x(w) \\ \mathbf{C}'_y(u) & \mathbf{r}'_y(w) \end{pmatrix} \tag{3.10}$$

$\mathbf{C}'(u)$ is found using Equation (3.5), $\mathbf{r}'(w)$ is simply the direction vector $\mathbf{d}$, and subscripts x and y denote the x- and y- components. As before, the validity of the identified root(s) may be judged based on the values of u and w: if u is between zero and one, the ray hits the curve $\mathbf{C}$ within its endpoints, and; if w is greater than zero, the point of intersection occurs in the forward direction of the ray. If multiple solutions satisfy these criteria, that with the lowest value of w should be taken.

The intersection-finding method described thus far is relatively straightforward, and indeed performs quite well for very simple cases. In many instances, however, multiple points of intersection may exist, and so identification of the correct root depends highly on the initial guess. For a ray that has been previously reflected from the concentrator surface, for example, the initial guess must be sufficiently far from the root found at $w = 0$ to allow identification of any other more appropriate points of intersection. Other difficulties can arise when dealing with more complex surface geometries, as it is possible for a single ray to intersect the concentrator curve multiple times. In these cases, it may make sense to perform the root-finding method a number of times, each with a different initial guess, and in the end take the solution with the lowest nonzero value for the ray parameter, w.

In the strategy employed in this thesis for ray-tracing the fourth-order Bezier spline of test case three, for example, initial guesses are based on a preliminary calculation of intersections between the ray and sections of the Bezier curve's *control polygon*. To illustrate this, take the initial ray shown in Figure 3.6 (a). This ray intersects only the second section of the control polygon (shown as a dashed black line), and so Newton-Raphson need only be performed a single time. Because, in this case, the second section is hit, an appropriate initial guess for u is somewhere between 0.25 and 0.5. Since this particular ray is heading toward the right, the upper limit of $u = 0.5$ is used. The ray direction is taken into account in this way so that the Newton-Raphson scheme will approach the solution from a distance ahead of the ray, and so will be less likely

to gravitate to the root found at the ray origin. An initial ray parameter of $w = 6.91$ is used, corresponding to the intersection with the control polygon. From this initial guess, Newton-Raphson iterations are performed until the vector defined by $\mathbf{C}(u) - \mathbf{r}(w)$ is sufficiently close to zero. In the implementation used, iterations are simply continued until the actual distance between $\mathbf{C}(u)$ and $\mathbf{r}(w)$ (i.e. the vector norm) is less than 10^{-10}. For the given ray, four updates are required to achieve this, resulting in the root defined by $u = 0.2529$ and $w = 6.1219$. Figure 3.7 shows the resulting steps on a contour plot of the distance between the points (although the last two steps are too small to be noticed), and Table 3.1 provides corresponding values of u, w, and the resulting distance at each iteration.

If a specular reflection were to occur at this point, the resulting ray would be slightly more difficult to handle, as it intersects the Bezier curve at two distinct points, as shown in Figure 3.6 (b). Through the same process as above, two roots are found: one at $u = 0.253$, $w = -1.4 \times 10^{-15}$, and another at $u = 0.795$, $w = 6.005$. In this case, the first of these points corresponds to the ray origin and so is thrown out.

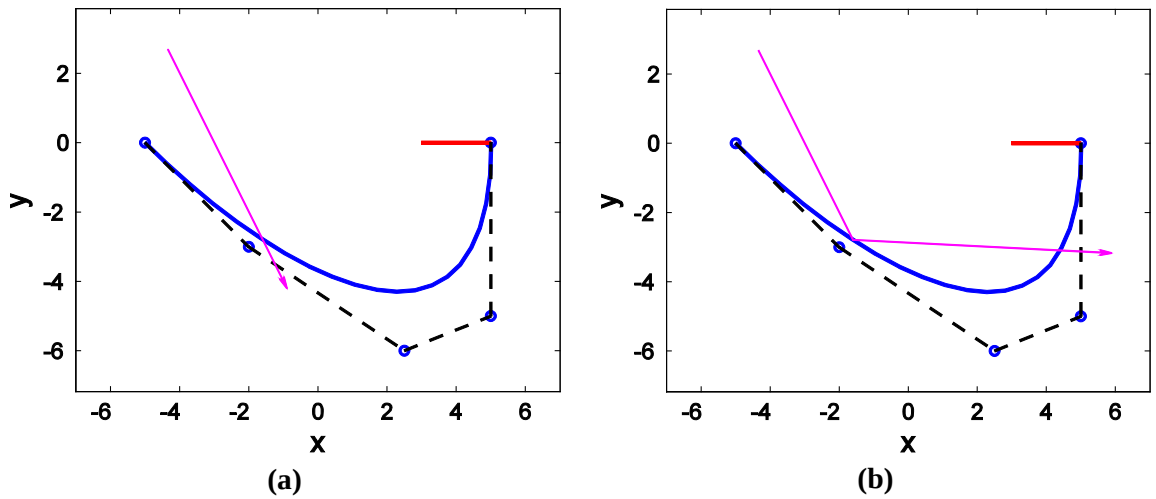


FIGURE 3.6 - EXAMPLE (a) INITIAL AND (b) REFLECTED RAY

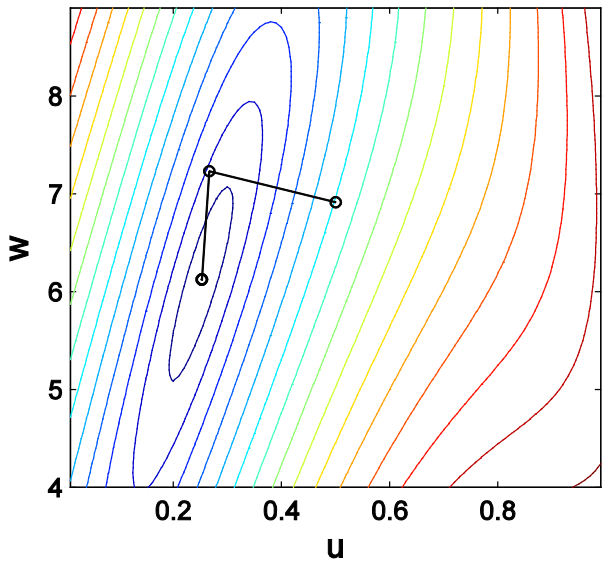


FIGURE 3.7 - NEWTON-RAPHSON STEPS ON CONTOUR PLOT OF DISTANCE

TABLE 3.1 - PARAMETERS DURING INITIAL NEWTON-RAPHSON RAY-TRACE

Iteration	*u*	*w*	distance
1	0.50000	6.91312	3.03173
2	0.26705	7.23074	0.913545
3	0.25299	6.12531	0.00216667
4	0.25289	6.12188	1.01702e-07
5	0.25289	6.12188	1.73422e-15

Chapter 4

EVALUATING THE OBJECTIVE FUNCTION USING MONTE CARLO METHODS FOR RADIATIVE HEAT TRANSFER

In order to employ a nonlinear optimization technique to design a solar concentrating collector, we must define an *objective function*, $F(\mathbf{\Phi})$, which is dependent on the vector of design parameters, $\mathbf{\Phi}$, and representative of design performance. Furthermore, there must be an algorithm capable of evaluating the objective function over the domain of possible designs. Throughout this thesis, the objective function (to be minimized) is taken to be the negative fraction of incoming radiation that is absorbed at the receiver surface, given a particular collector configuration. In this way, the optimal concentrator design will be that which results in the greatest fraction of solar energy collected. As discussed in Chapter 2, because the concentrator system is made up of highly specular surfaces, objective function evaluations are performed using a Monte Carlo ray-tracing technique. (It should be noted that, although overall performance of a solar concentrating collector depends on multiple modes of heat transfer, the scope of the present analysis is limited to radiative heat transfer.)

4.1 MONTE CARLO RAY-TRACING TECHNIQUE

Although the term Monte Carlo encompasses a broad range of numerical integration tools based on random sampling, in the case of concentrating collector analysis there is a convenient physical analogy: "bundles" of photons are ray-traced as they enter the collector from randomly-sampled locations over the aperture and undergo multiple randomized surface-interaction events in until they are either absorbed at a surface or escape from the collector aperture. The overall collector performance is inferred by aggregating results of a large

sample of simulated trajectories. The concept is illustrated using the geometry of test case one, a linear parabolic concentrator with cylindrical absorber, as shown in Figure 4.1.

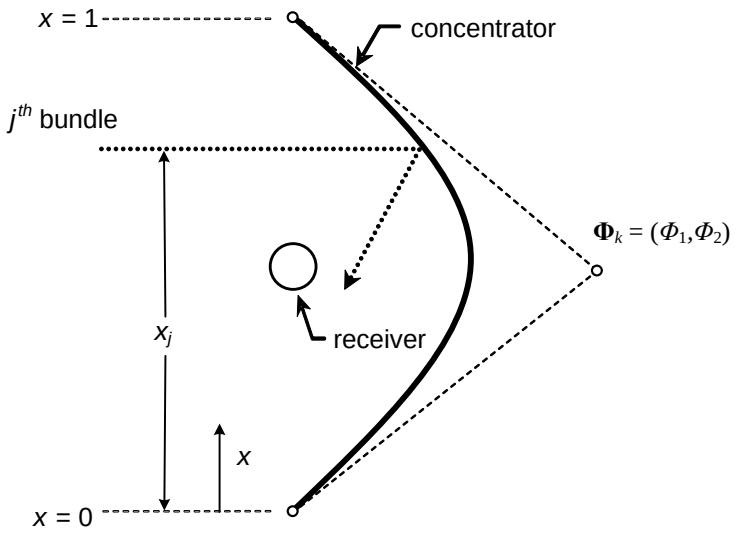


FIGURE 4.1 - MONTE CARLO FOR TEST CASE ONE

It should be noted that, in order to simplify calculation throughout this thesis, geometry is treated as fully two-dimensional, although an extension to include 3D effects is straightforward. In addition, incoming radiation is assumed to be perfectly collimated, so that only the initial ray position must be sampled. Although this is not quite true in reality, as solar radiation is distributed over a half-angle of about 0.27°, it is not necessary to take this into account for demonstration of the methodology. In order to further simplify initial discussion of the Monte Carlo method, perfect optics is assumed (that is, perfectly specular reflection at the concentrator surface and blackbody absorption at the absorber surface), so that the complete path of each ray is determined by its initial position and direction. Under the assumptions of collimated incoming radiation and perfect optics, the Monte Carlo ray-tracing technique is reduced to an estimation of the one-dimensional (single variable) integral

$$F(\mathbf{\Phi}) = \int_{x=0}^{1} f(x)\,dx \approx \widetilde{F}(\mathbf{\Phi}) = \frac{1}{N}\sum_{j=1}^{N} f(x_j), \qquad 0 < x_j < 1 \tag{4.1}$$

where N is the sample size (the number of rays traced) and $f(x_j)$ is the binary outcome for the ray originating at randomly sampled position, x_j. Since the objective is to minimize the negative fraction of collected rays, $f(x_j) = -1$ if the ray is absorbed at the absorber surface and otherwise $f(x_j) = 0$. In this way, then, an ideal collector for which every incoming ray is successfully absorbed at the receiver has a performance given by $F(\mathbf{\Phi}) = -1$. Performing a large sample of such ray-traces provides an unbiased estimate of the objective function, $\widetilde{F}(\mathbf{\Phi})$, in which the tilde indicates that the estimate is subject to some statistical uncertainty, as is discussed below.

For the ideal case presented above, an objective function evaluation through Monte Carlo ray-tracing is straightforward to formulate. Although some additional analysis is required in order to simulate more realistic systems, the method continues to remain relatively intuitive. In addition, unlike some analytical techniques, MCRT can handle arbitrarily complex geometries: a ray-tracing routine could in theory be devised for any system, with the only practical limitation being an increased computational cost with increased complexity. A further advantage of MCRT is that, through simple modifications to this basic formulation, a simulation can be made to accurately account for any number of physical phenomena, such as realistic directional and/or spectral surface properties, or a particular distribution of incoming radiation. This too, however, is associated with an increase in computation.

The sections to follow provide a discussion of concepts necessary for objective function evaluation in this thesis; for further reference refer to Hammersley and Handscomb (1975) and Lemieux (2009) for a discussion of Monte Carlo and quasi-Monte Carlo methods in general, and to Siegel and Howell (2002) for specific discussion of Monte Carlo in its application to modeling radiation heat transfer.

4.1.1 ROLE OF EVENT LIKELIHOOD IN STOCHASTIC SAMPLING

The key to using Monte Carlo to simulate a physical process is ensuring that the randomly sampled values, x_j, have a probability distribution that corresponds to that of the process at hand. In this way, an outcome which is physically more likely to occur will tend to occur more frequently in simulation. In general, the most effective way to achieve this (and that which is used exclusively throughout this thesis) is through *function inversion*. It can be shown that appropriately distributed random samples, x_j, can be generated by transforming a uniformly distributed random number, R_j, by the inverse of the cumulative distribution function (CDF) of the process (Eckhardt 1987). For a univariate process with a CDF of $P(x)$, values of x_j are found by

$$x_j = P^{-1}(R_j) \tag{4.2}$$

where R_j, and thereby x_j, are between zero and one, and P^{-1} is the inverse function of the CDF, P. Multivariable processes are sampled in the same way, requiring one uniformly distributed random number for each independent variable.

Although the above formulation can require some initial analysis and coding on part of the designer, the only increase in computational effort is that due to the required evaluation of Equation (4.2) to sample x_j at every occurrence of the process. For the single variable problem of test case one, each ray is equally likely to originate at any position along the aperture, and so the CDF and its inverse are simply equal to unity. As such, initial ray position x_j is sampled directly from a uniformly distributed random number, R_j. Other physical processes are not quite as convenient. To simulate a diffuse emission (or reflection), for example, the radiative intensity distribution as dictated by Lambert's cosine-law, and shown in Figure 4.2, must be taken into account in order to determine an appropriate ray direction, θ.

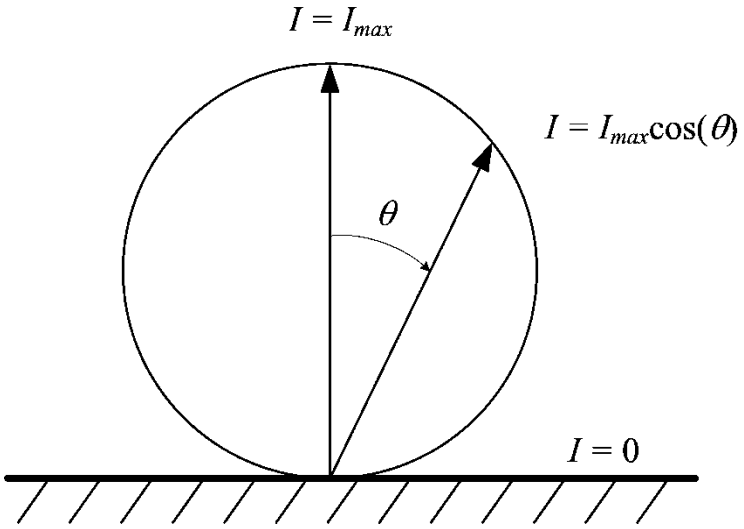


FIGURE 4.2 - LAMBERT'S COSINE LAW

From this, an appropriately distributed random number R_θ can be found through (Siegel and Howell 2002)

$$R_\theta = 2\int_0^\theta \cos\theta^* \sin\theta^* d\theta^* = \sin^2\theta \tag{4.3}$$

where θ^* is a dummy variable of integration. In order to better suit computation, θ is normally left in its sine form as

$$\sin\theta = \sqrt{R_\theta} \tag{4.4}$$

where R_θ is a uniformly distributed random number. From this, the angle of the emitted ray, θ, is determined. For a 3-D treatment, a similar approach is used to determine the azimuth angle; because geometries in this thesis are treated in two dimensions, however, we must simply determine whether the ray is emitted at angle θ clockwise or counter-clockwise from the normal, each of which is equally likely. So, a random number R is sampled, with $R < 0.5$ resulting in one outcome, and $R > 0.5$ resulting in the other.

4.1.2 Simulating Realistic Optical Properties

In order to extend the above perfect optics formulation to cases which treat realistic optical properties, additional stochastic sampling must be implemented.

For each non-ideal ray-surface intersection, a uniformly distributed random number (between zero and unity) must be sampled, with the event outcome dependent upon its relationship to the optical properties of the surface. For a basic treatment of reflection, as is performed in this thesis, the surface reflectivity, ρ, is simply equivalent to the likelihood that a successful reflection will occur. At a ray-surface intersection, then, random number R is sampled, and if $R < \rho$, the ray is successfully reflected. The same logic can be applied to the simulation of absorption, based on surface absorptivity, α.

For the cases throughout this thesis considering "realistic" optics, the following assumptions are made: (*i*) the concentrator surface has a specular reflectivity of 0.92, based on the reflecting surface data used in Haeberle et al. (2005); (*ii*) rays that are not reflected at the concentrator surface are instead absorbed (and henceforth neglected); (*iii*) the receiver surface has a directional absorptivity modeled after the sputtered Ni-NiO_x coating examined in Tesfamichael and Wäckelgård (1999) and presented in Appendix C; (*iv*) rays not absorbed at the receiver are instead reflected diffusely.

4.1.3 Estimating Monte Carlo Uncertainty

Although Monte Carlo ray-tracing is a powerful tool for estimating radiative heat transfer, it has the drawback of introducing stochastic uncertainty into a calculation. Because a Monte Carlo approximation is simply the mean result of a number of individual samples, it contains fluctuations about a mean value. Since in practical applications the exact solution is unknown, this uncertainty, denoted ε_{MC}, must be estimated. This can be done by binning the results into a number of independent sets of trials and measuring the sample standard deviation (Hammersley and Handscomb 1975)

$$\varepsilon_{MC} \approx \frac{\sigma}{\sqrt{M}} \tag{4.5}$$

where M denotes the number of trials, each using N/M rays, and σ is the standard deviation of the set of trials, given by

$$\sigma = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}[\widetilde{F}_i(\mathbf{\Phi}) - \widetilde{F}(\mathbf{\Phi})]^2} \tag{4.6}$$

In this relation, $\widetilde{F}_i(\mathbf{\Phi})$ is the mean result of the i^{th} trial. Because the distribution of resulting means can, by the Central Limit Theorem, be taken as approximately normal, resulting objective function evaluations can be stated, with 68% confidence (Siegel and Howell 2002), as

$$F(\mathbf{\Phi}) = \widetilde{F}\left(\mathbf{\Phi}\right) \pm \varepsilon_{MC} \tag{4.7}$$

For traditional Monte Carlo, in which random values x_j are derived from uniformly distributed pseudorandom numbers, the magnitude of uncertainty ε_{MC} decreases according to (Hammersley and Handscomb 1975)

$$\varepsilon_{MC} \propto N^{-1/2} \tag{4.8}$$

so, in order to halve the uncertainty in a given estimation, four times as many rays must be traced. Herein lies the major drawback to Monte Carlo, as the number of rays traced is nearly directly proportional to the required amount of computation time, since ray-tracing makes up the majority of processing, especially for more complex geometries (Kersch et al. 1994). Throughout this thesis, the negative exponent in an uncertainty trend is referred to as the *convergence rate*. As such, from Equation (4.8), traditional Monte Carlo exhibits a convergence rate of 1/2.

4.2 Reducing Monte Carlo Uncertainty

When sampling from a pseudorandom sequence, there is no guarantee that the integral domain will receive even coverage. This leads to the relatively high degree of uncertainty in traditional Monte Carlo estimations. Indeed, it has been shown that estimate uncertainty has a theoretical upper bound directly

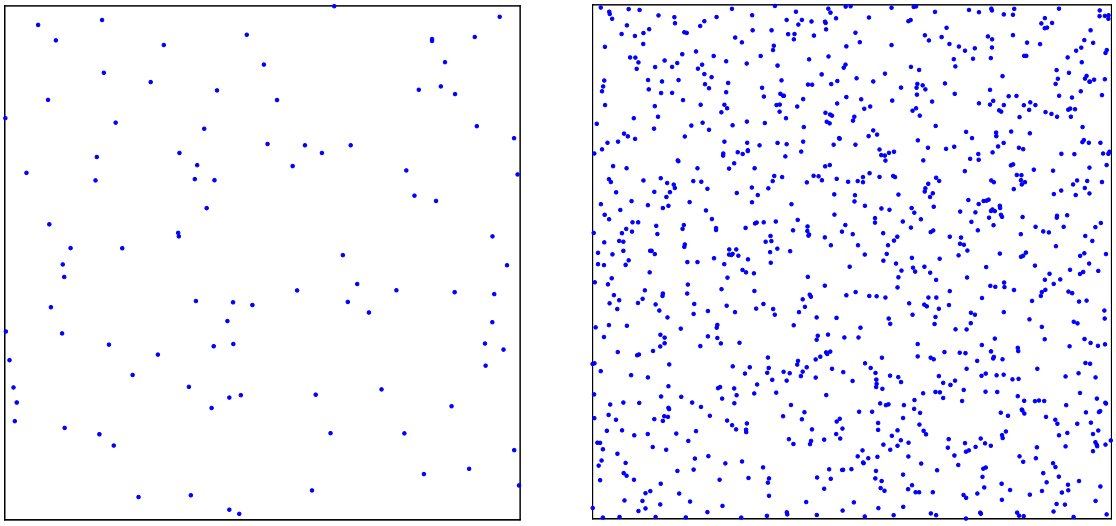
proportional to the *discrepancy* (a statistical measure of non-uniformity) of the sampled points (Hickernell 1998).

An obvious solution for uncertainty reduction, then, is to sample from a set of points that lie on a uniform grid (sampling each point exactly once), and indeed, a single-variable integral approximation of this sort (analogous to performing quadrature using the *rectangle* or *mid-point rule*) results in an uncertainty which decreases in proportion to N^{-1}, and so exhibits a convergence rate of unity (Press et al. 2007). Although such an approach can outperform traditional Monte Carlo for the simplest of integrations, it becomes nearly impossible to include enough points to achieve sufficient resolution for multi-variable integrals (Morokoff and Caflisch 1994). Furthermore, when sampling from a uniform grid, the grid resolution must be defined in advance, leaving no way of sampling *until* some desired level of accuracy is achieved; every single grid point must be sampled in order to maintain uniformity. If a resulting approximation turns out to have too large an uncertainty, then, accuracy can be improved by either repeating the computation from scratch with a finer grid, or enhancing the current estimate by reducing grid-spacing by a factor of two (thereby increasing the number of points by a factor of 2^s, where s is the integral dimensionality). Although this technique is generally impractical, algorithms have been developed for generating number sequences that achieve similar uniformity, without sacrificing flexibility of sample size.

4.2.1 Quasi-Random Sequences

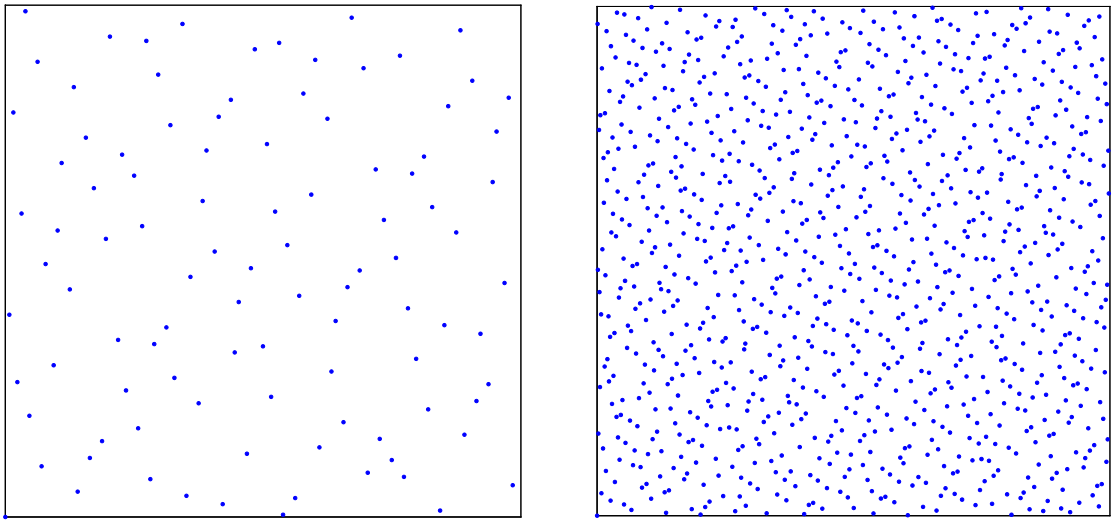
A more practical strategy than that described above is to perform Monte Carlo approximation based on samples from a low-discrepancy, or *quasi-random*, sequence, generated using algorithms specially developed so that any point achieves "maximal avoidance" of all others. In this way, a quasi-random sequence has, for any sequence length, a significantly more uniform distribution when compared to a pseudorandom sequence. Furthermore, as the sequence length increases, the resolution is effectively made finer, allowing

approximations to be concluded once a desired level of accuracy is achieved, and so limiting the amount of excess computational effort. These attributes can be seen by comparing Figure 4.3 and Figure 4.4, which show the first 100 and 1000 points of a two-dimensional pseudorandom and quasi-random (Halton) sequence, respectively. Both sets of plots were created using built-in MATLAB functions, with random number sequences generated the function "rand", and with the Halton sequence generated using the function "haltonset". By estimating the integral using points sampled from a quasi-random, instead of a pseudorandom, sequence the quasi-Monte Carlo (QMC) method is a deterministic scheme, capable of achieving a greater degree of accuracy than traditional Monte Carlo for a given sample size.



(a) **(b)**

FIGURE 4.3 - FIRST (a) 100 AND (b) 1000 POINTS OF A PSEUDORANDOM SEQUENCE



(a) **(b)**

FIGURE 4.4 - FIRST (a) 100 AND (b) 1000 POINTS OF A QUASI-RANDOM (HALTON) SEQUENCE

A number of different algorithms have been developed for the generation of quasi-random sequences, with the Halton, Sobol, and Fauve sequences most commonly applied to QMC approximations. Morokoff and Caflisch (1995) have demonstrated that, between these three, the Halton sequence frequently outperforms in estimations of integrals with dimensionality below about six, with Sobol outperforming in higher dimension. This thesis employs quasi-random sequences in a single dimension, for which Halton and Sobol are identical.

4.2.2 Additional Notes on Quasi-Random Sequences

In the use of quasi-random sequences, there are a few measures which may be taken in order to make the most of their highly uniform distribution. Figure 4.5 shows the order of the first sixteen points in a one-dimensional Halton or Sobol sequence. From this, two important characteristics can be seen. Firstly, uniformity is maintained through a pattern of sweeping over the space, with each new layer made up of twice as many points as the previous, resulting in a spacing that is half of the previous layer. In this way, each point is, as stated previously, "maximally avoiding" all others. Due to this method of sweeping,

however, the most uniform sequences are those with completely finished layers present, occurring at sequences of length defined by the powers of two. Take only the first fourteen points from the below sequence, for example, and suddenly the distribution is slightly skewed to the lower half of the space. This is especially important to take into account for short sequences, but quickly becomes negligible for larger sample sizes. Throughout this thesis, quasi-Monte Carlo is performed using sample sizes of powers of two whenever possible. It should be noted that, for a multi-dimensional problem, this rule depends on the construction of the particular quasi-random sequence used.

Another important characteristic to note is that for a quasi-random sequence the first entry is very close to zero (and identical to zero for a one-dimensional sequence), and there is no other entry present equally close to one. For this reason, a quasi-random sequence will show a slight bias toward the lower side, although as the sequence length increases this bias becomes negligible. In practice, this can be avoided by simply discarding the first few points (the number is irrelevant) from the sequence, as is recommended by Morokoff and Caflisch (1995). Throughout this thesis, whenever a quasi-random sequence is generated for a QMC estimation, the first 64 points are skipped.

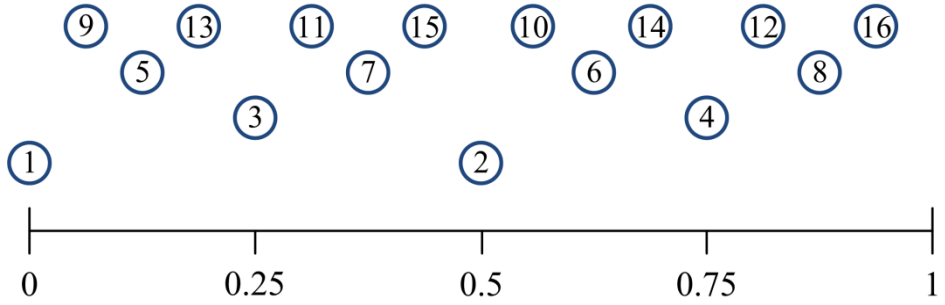


FIGURE 4.5 - FIRST SIXTEEN POINTS OF A QUASI-RANDOM SEQUENCE

4.3 QUASI-MONTE CARLO (QMC) METHOD

By performing Monte Carlo based on sampling from a quasi-random sequence, the so-called quasi-Monte Carlo (QMC) technique can be used to estimate the same problems as Monte Carlo, but can achieve a greater degree of accuracy for

a given sample size. Although, in theory, QMC can exhibit a convergence rate as high as unity, in practice performance depends on factors such as function smoothness and integral dimensionality. As integral dimensionality increases, the nature of the quasi-random sequence used can cause its uniformity to begin to break down, thereby reducing convergence rate. For a general problem, the following error bound has been shown (Morokoff and Caflisch 1995)

$$\left|\varepsilon\right| = cN^{-s/(2s-1)} \tag{4.9}$$

where s is the integral dimensionality and c is a problem-specific constant. Based on this relationship, it can be seen that the worst-case convergence rate quickly decreases with increasing integral dimensionality from unity, with a value of about 0.667, 0.6, and 0.571 for a problem with an s of 2, 3, and 4, respectively. Morokoff and Caflisch (1995) state that this error bound is often pessimistic, with the majority of examined cases vastly outperforming this prediction. At any rate, there is little reason not to apply QMC over traditional MC, since even in cases of high dimensionality, QMC can only perform as badly as MC. Furthermore, the additional computation required for quasi-random sequence generation is negligible, especially as simulation complexity increases.

The merits of QMC are well-documented and have been applied in various areas, though at the time of writing the author is aware of only two examples in radiative heat transfer analysis. In the first, QMC integration was used to accelerate computation of the radiative transfer equation in a number of slabs containing a scattering medium (O'Brien 1992). In the second example, QMC ray-tracing was applied in the estimation of radiative transfer in a semiconductor wafer thermal processing furnace containing specularly reflecting surfaces (Kersch et al. 1994). Although limited application of QMC can be found in the area of radiative heat transfer, its principles have been applied extensively in the simulation of light transport in computer graphics, for which many of the same principles apply. Keller (1996) and Veach (1997) provided thorough discussion of the theory and application of QMC in the solution of

illumination in a three-dimensional scene, and Cieslak et al. (2008) applied randomized-QMC (to be discussed shortly) to speed up the simulation of light distribution in the canopy of a plant.

In this variety of applications, quasi-Monte Carlo was found to consistently outperform traditional Monte Carlo. As will be demonstrated, similarly positive results are achieved for the solar concentrating collector objective function evaluations performed throughout this thesis.

4.3.1 QMC FOR A SIMPLE ONE-DIMENSIONAL INTEGRATION

As an initial demonstration of quasi-Monte Carlo, the simple integral estimation

$$\int_0^1 x^2 dx = \frac{1}{N}\sum_{j=1}^{N} x_j^{\,2} \tag{4.10}$$

is performed using both MC and QMC, with varying sample size N. For the case of traditional MC, x_j is sampled from a uniformly distributed pseudo-random sequence, and for QMC it is sampled from a one-dimensional Halton sequence. Error magnitudes are computed with respect to the known analytical solution of 1/3. As can be seen in Figure 4.6 on a log-log (base 2) plot, QMC integration exhibits a smaller error for all N, with a significantly steeper convergence rate when compared to MC integration.

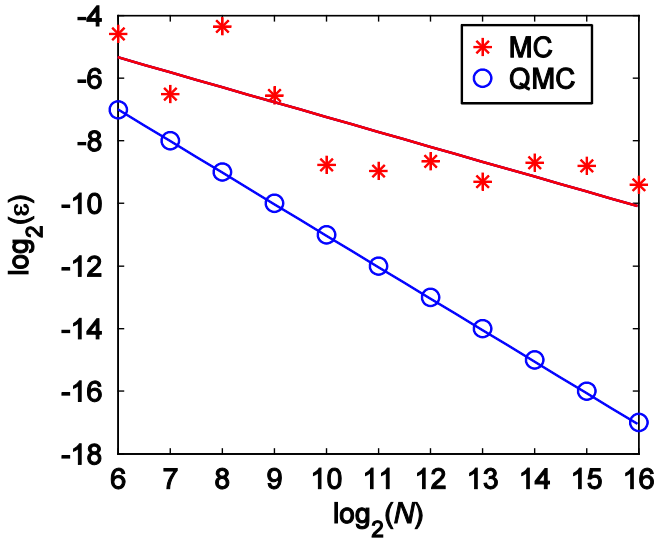


FIGURE 4.6 - ERROR TREND FOR A SIMPLE ONE-DIMENSIONAL INTEGRAL ESTIMATE

Throughout this thesis, trends in estimation uncertainty are presented in this way, in terms of the convergence rate, β, and coefficient, c, found through a linear least-squares fit to the log-log data, so that on a plot of the error trend as in Figure 4.6, the error has a slope of $-\beta$. This leads to the familiar form

$$\varepsilon(N) = c \cdot N^{-\beta} \tag{4.11}$$

The error trends for this simple one-dimensional integral estimation, then, are

$$\begin{aligned} \varepsilon_{MC} &= 0.1753 \cdot N^{-0.4706} \\ \varepsilon_{QMC} &= 0.4974 \cdot N^{-0.9995} \end{aligned} \tag{4.12}$$

As expected for a one-dimensional integral estimation of a smooth function, then, the Monte Carlo convergence rate is close to 1/2, while quasi-Monte Carlo achieves close to the theoretical improved convergence rate of unity. Accordingly, QMC needs two orders of magnitude fewer rays than MC to achieve an error of 10^{-3}, and three orders of magnitude fewer for an error of 10^{-4}, and consequently QMC can achieve estimations of required accuracy with far less computational time, with savings becoming increasingly significant as the required degree of accuracy increases.

4.3.2 Randomized Quasi-Monte Carlo (RQMC)

Because quasi-random sequences are generated using a deterministic algorithm, every instance is identical, and so successive trials are not mutually-independent. As a consequence of this, QMC error cannot be directly estimated from the sample standard deviation, as in Equation (4.5). One way to remedy this is to *randomize* the quasi-random sequence between each successive trial, ideally in a manner that does not compromise uniformity. In this way, so-called randomized quasi-Monte Carlo (RQMC) can achieve the benefits of improved convergence, while providing a group of independent trials for use in error estimation (L'Ecuyer and Lemieux 2002).

This randomization can be achieved in a number of ways; the simplest of which is "random shift modulo one" in which the objective function is estimated by (Cranley and Patterson 1976)

$$\widetilde{F}\left(\mathbf{\Phi}\right)=\frac{1}{N}\sum_{j=1}^{N}f\left(\left(x_j+\delta\right)\bmod 1\right) \tag{4.13}$$

where x_j is the j^{th} term from a quasi-random sequence and δ is a random number uniformly distributed between zero and one. For a one-dimensional estimation, for example, we simply shift the quasi-random sequence by adding the same random number to each point, and then subtract one from any points that are greater than one in order to "wrap them back around" to the beginning. To demonstrate this, random shift modulo one is repeatedly applied to the first 32 points of a one-dimensional Halton sequence, with results shown in Figure 4.7. Although methods of randomization exist which are better-suited to problems of higher-dimensionality, such as those discussed in (Lemieux 2009), this is the method exclusively applied to the problems dealt with in this thesis.

Through this technique, for a given RQMC estimation, only a single quasi-random sequence of length N/M need be generated and stored in memory. It should be noted, however, that for very small sample sizes, such repeated randomization may compromise the uniformity of the quasi-random sequence. As an extreme example, an estimation of sample size $N=20$, when split into twenty trials, is simply based on a single randomly shifted point, and so is essentially equivalent to a traditional Monte Carlo estimation. For estimations of practical sample size, this is not thought to be an issue.

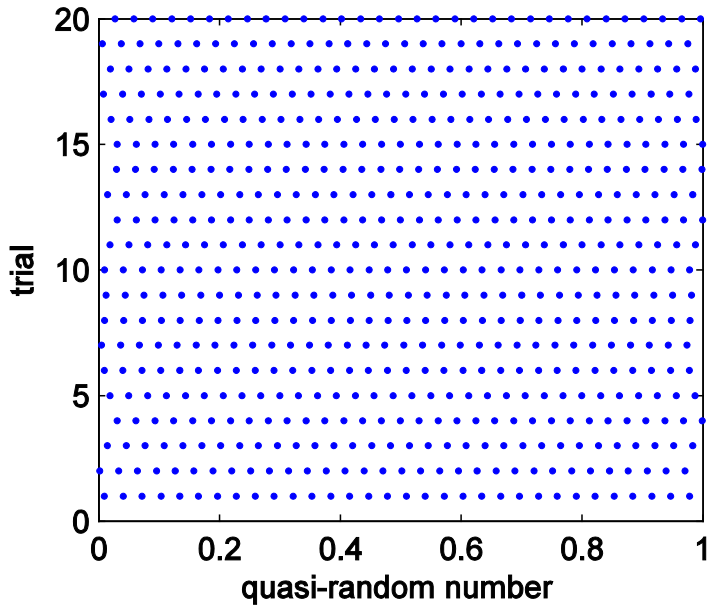


FIGURE 4.7 - RANDOMIZED ONE-DIMENSIONAL HALTON SEQUENCE (FIRST 32 POINTS)

4.4 APPLICATION OF RQMC TO TEST CASES

In order to measure the performance of randomized quasi-Monte Carlo for the estimations performed throughout this thesis, a number of objective function evaluations are performed using both RQMC and traditional MC, for the three test case geometries as described in Chapter 3. In RQMC trials, the initial ray position is sampled from a one-dimensional quasi-random sequence, with any additional sampling (for simulating realistic optical properties, for example) performed using pseudorandom numbers. In order to make resulting uncertainty trends more apparent, each set of results presented is based on the mean outcome of twenty repeated trials. For each example, tests are performed for a number of different sample sizes, chosen to provide an even spread over the log (base 2) scale. As such, trials are performed with $N = \{2^6, 2^7, \ldots, 2^{18}\}$, with each split into sixteen independent groups. Finally, all error magnitudes are measured with respect to an "exact" solution, based on an RQMC trial using 2^{23} rays.

4.4.1 Test Case One

For test case one, repeated objective function evaluations are performed for the designs given by $\Phi = \{[8, -3]^T, [12, 4]^T, [11, -4]^T\}$, all assuming perfect optics. Figure 4.8 shows the geometry and initial RQMC ray-trace for the first two of these designs. For each design, uncertainty trends as in Figure 4.9 are found based on the resulting measured uncertainties. Table 4.1 provides results for c, β, and the resulting sample sizes required to achieve error magnitudes of 10^{-3} and 10^{-4} for all three designs.

The results show the expected uncertainty trends for such a one-dimensional problem, with MC exhibiting a convergence rate of about 1/2 and RQMC achieving its theoretical convergence rate of about one. Although coefficient c is larger for RQMC by a factor of about 7.6, its higher convergence rate quickly outweighs this as the sample size is increased. Accordingly, RQMC needs two orders of magnitude fewer rays than MC to achieve an evaluation with uncertainty 10^{-3}, and three orders of magnitude fewer for an uncertainty of 10^{-4}. As before, then, RQMC can achieve estimations of required accuracy with far less computational effort than MC, with increasingly significant savings as required accuracy increases.

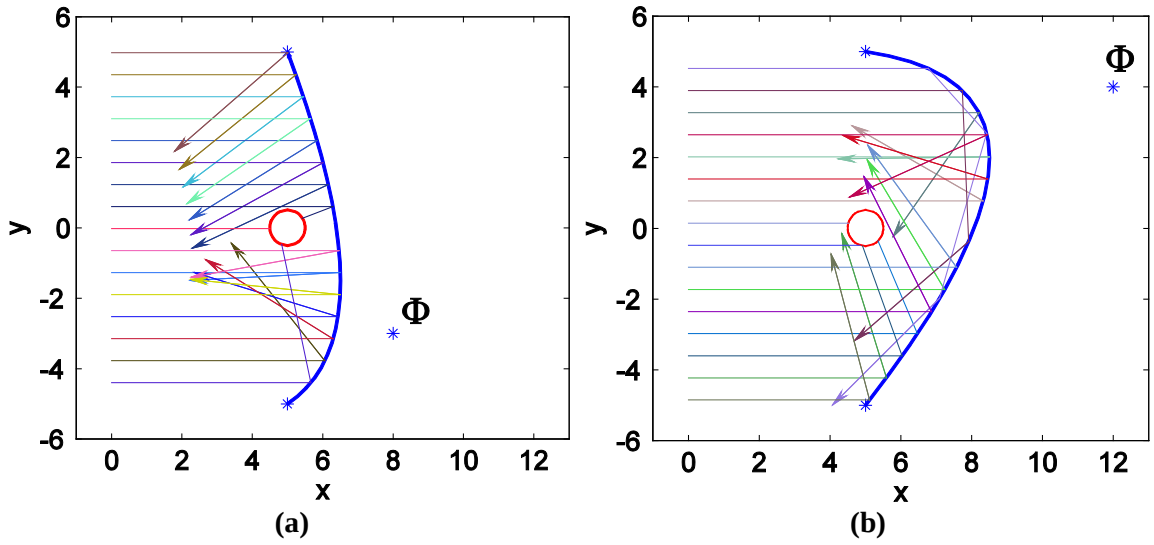


FIGURE 4.8 - GEOMETRY FOR TEST CASE 1, DESIGNS (a) $\Phi=[8,-3]^T$ AND (b) $\Phi=[12,4]^T$

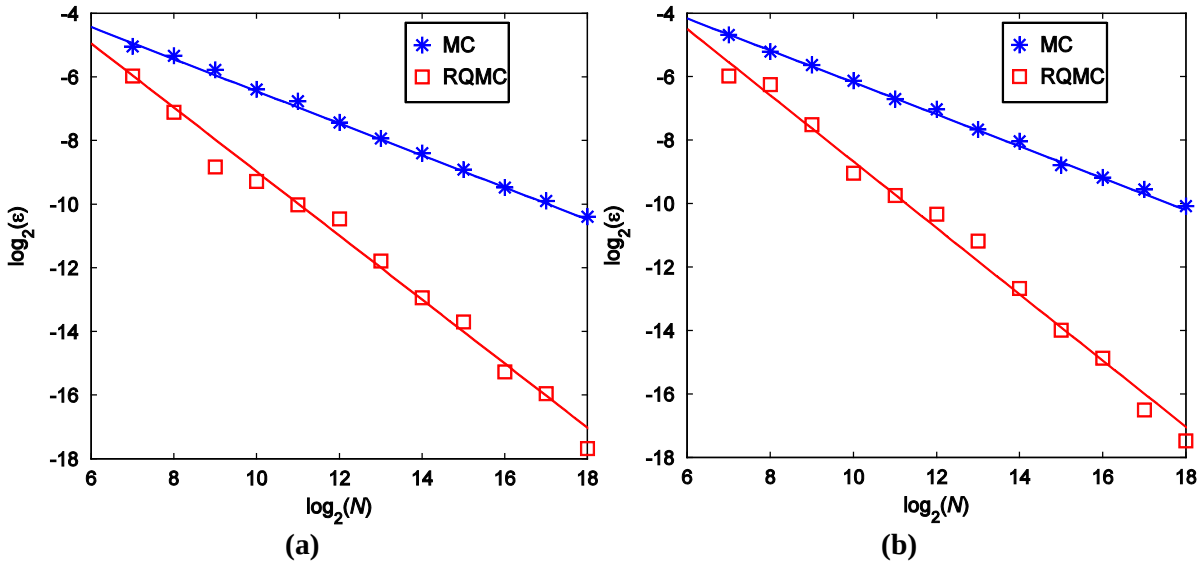


FIGURE 4.9 - UNCERTAINTY TRENDS FOR DESIGNS SHOWN ABOVE

TABLE 4.1 - UNCERTAINTY TRENDS FOR TEST CASE ONE

Φ	method	c	β	$N(\varepsilon = 10^{-3}) \times 10^{-3}$	$N(\varepsilon = 10^{-4}) \times 10^{-4}$
$\begin{bmatrix} 8 \\ -3 \end{bmatrix}$	MC	0.3617	0.4956	145.3	1514
	RQMC	2.1883	1.0148	1.956	1.892
$\begin{bmatrix} 12 \\ 4 \end{bmatrix}$	MC	0.4362	0.4953	213.7	2233
	RQMC	3.4913	1.0503	2.362	2.116
$\begin{bmatrix} 11 \\ -4 \end{bmatrix}$	MC	0.4490	0.5044	181.6	1740
	RQMC	3.9539	1.0339	3.013	2.794

4.4.2 Test Case Two

For test case two, the designs given by $\Phi = \{[8.6, 2.3, 11.2]^T, [6.8, 2.5, 7.8]^T\}$, are evaluated, assuming both perfect and "realistic" optical properties (based on the assumptions outlined previously). Figure 4.10 shows the geometry and initial RQMC ray-trace for the two designs, and Figure 4.11 shows the resulting uncertainty trends for the first of these, based on the assumptions of both perfect and realistic optics. Overall results are provided in Table 4.2.

From these results, it can be seen that RQMC achieves close to the expected degree of improvement over MC when perfect optics are assumed, with

similar orders of magnitude reduction in required sample size as in the previous case. The convergence rate for MC is close 1/2 as is expected, while for RQMC it averages 0.9, slightly lower than unity. For estimations based on realistic optical properties, however, RQMC provides a much smaller improvement over MC, exhibiting a convergence rate of about 0.55 and so achieving less than an order of magnitude reduction in required sample size. Specifically, to provide estimations with an average error of 10^{-3} and 10^{-4}, RQMC requires a factor of about 3.3 and 4.3 fewer rays than MC, respectively. Unlike in previous cases, this uncertainty reduction is due to a combination of lower coefficient (on average 6.5% lower) as well as slightly lower convergence rate (about 1.5% lower).

The significant decrease in convergence rate offered by RQMC over MC when considering realistic surface properties is likely due to the stochastic effects introduced by the additional random sampling required at ray-surface interactions. When perfect optics is assumed, the outcome of a ray is deterministic once the initial position and direction are specified. As such, the highly uniform distribution of initial rays over the collector aperture achieved through the low-discrepancy quasi-random sequence carries throughout the collector. Simulation of realistic surface properties, on the other hand, injects randomness into the low-discrepancy ray-inception points and accordingly increases the variance in the RQMC estimates.

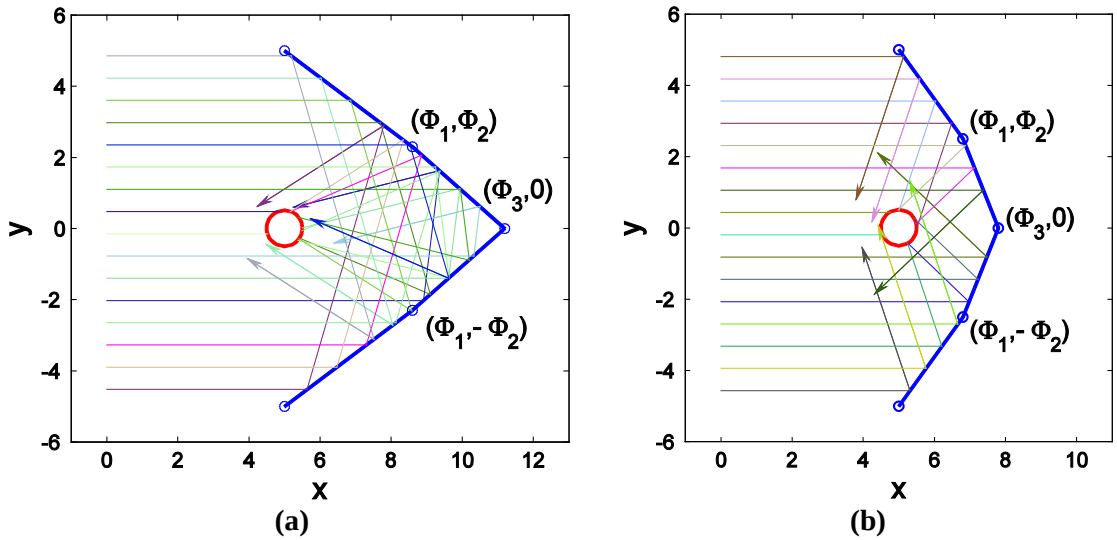


FIGURE 4.10 - GEOMETRY FOR TEST CASE 2, DESIGNS (a) $\Phi=[8.6, 2.3, 11.2]^T$ AND (b) $\Phi=[6.8, 2.5, 7.8]^T$

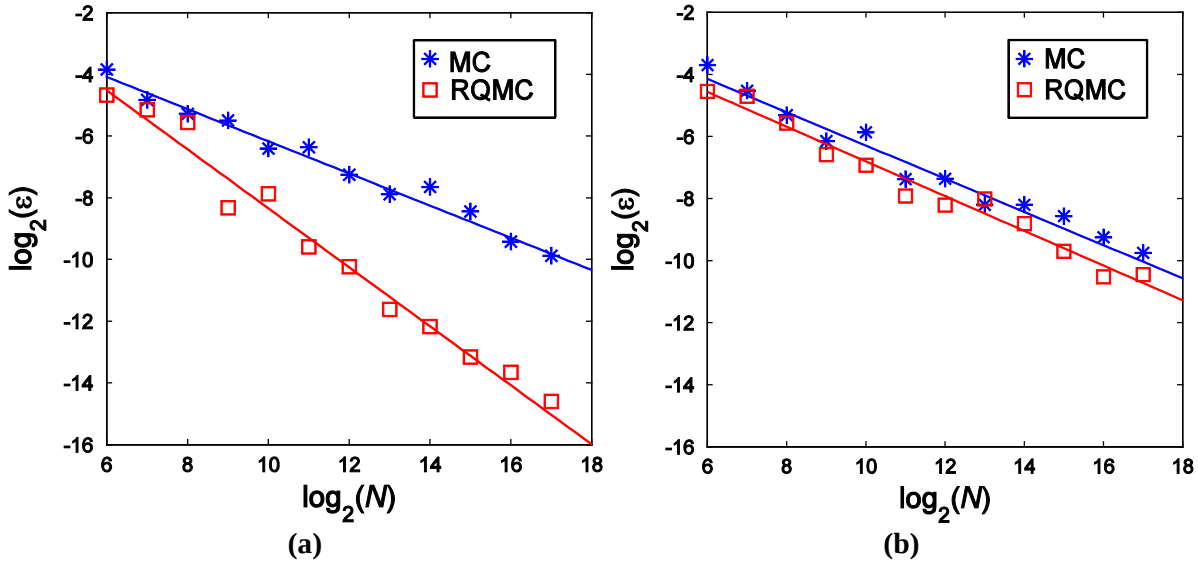


FIGURE 4.11 - UNCERTAINTY TRENDS FOR TEST CASE 2, ASSUMING (a) IDEAL AND (b) REALISTIC OPTICAL PROPERTIES

TABLE 4.2 - UNCERTAINTY TRENDS FOR TEST CASE TWO

Φ	method	c	β	$N(\varepsilon = 10^{-3}) \times 10^{-3}$	$N(\varepsilon = 10^{-4}) \times 10^{-4}$
Perfect Optics					
$\begin{bmatrix} 8.6 \\ 2.3 \\ 11.2 \end{bmatrix}$	MC	0.4659	0.5108	187.2	1698
	RQMC	2.228	0.9455	3.476	3.970
$\begin{bmatrix} 6.8 \\ 2.5 \\ 7.8 \end{bmatrix}$	MC	0.3502	0.4824	187.9	2222
	RQMC	0.7347	0.8669	2.024	2.882
Realistic Optical Properties					
$\begin{bmatrix} 8.6 \\ 2.3 \\ 11.2 \end{bmatrix}$	MC	0.4918	0.5218	144.2	1190
	RQMC	0.4356	0.5623	49.35	296.2
$\begin{bmatrix} 6.8 \\ 2.5 \\ 7.8 \end{bmatrix}$	MC	0.4069	0.5070	140.4	1318
	RQMC	0.2727	0.5326	37.42	282.2

4.4.3 TEST CASE THREE

For test case three, repeated objective function evaluations are performed assuming both ideal and realistic optics, for the designs given by $\mathbf{\Phi} = \{[-2, -3, 2.5, -6, 5, -5]^T, \ [-3.5, -5, 1, -7, 5, -5]^T\}$. Figure 4.12 shows the geometry and the path of the first sixteen RQMC rays for the two designs, and Figure 4.13 shows example uncertainty trends for evaluations assuming perfect and realistic optics. Overall results are provided in Table 4.3.

As for the test case 2, simulations based on perfect optics exhibit the expected uncertainty trends for MC and RQMC, while significantly less improvement is seen for cases assuming realistic optics. Similar to the previous case, in order to provide estimations with an average uncertainty of 10^{-3} and 10^{-4}, RQMC requires a factor of about 4.0 and 4.7 fewer rays than MC, respectively.

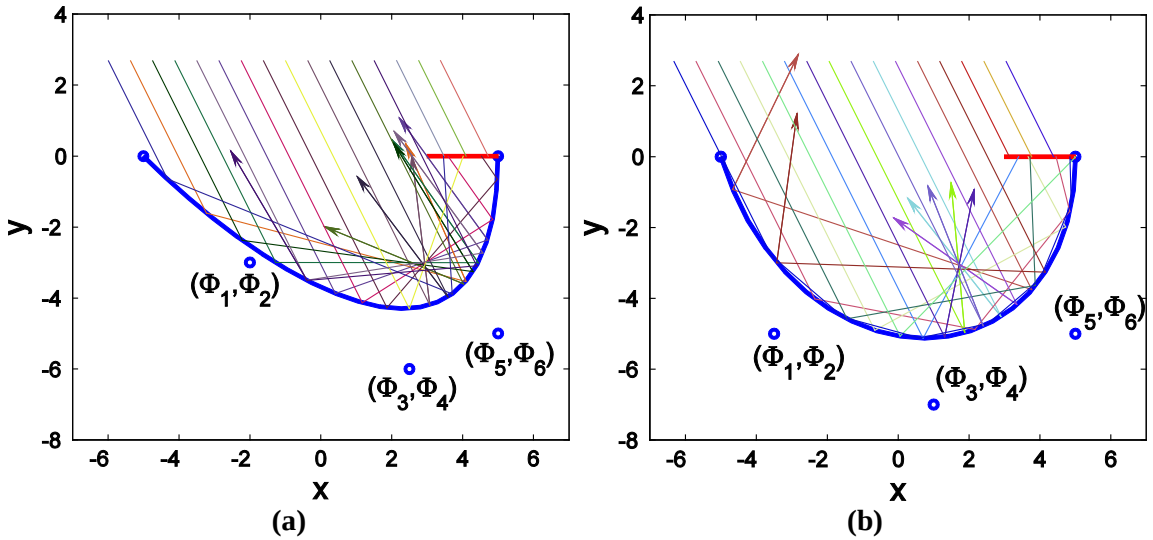


FIGURE 4.12 - GEOMETRY FOR TEST CASE 3, DESIGNS (a) $\Phi=[-2,-3,2.5,-6,5,-5]^T$ AND (b) $\Phi=[-3.5,-5,1,-7,5,-5]^T$

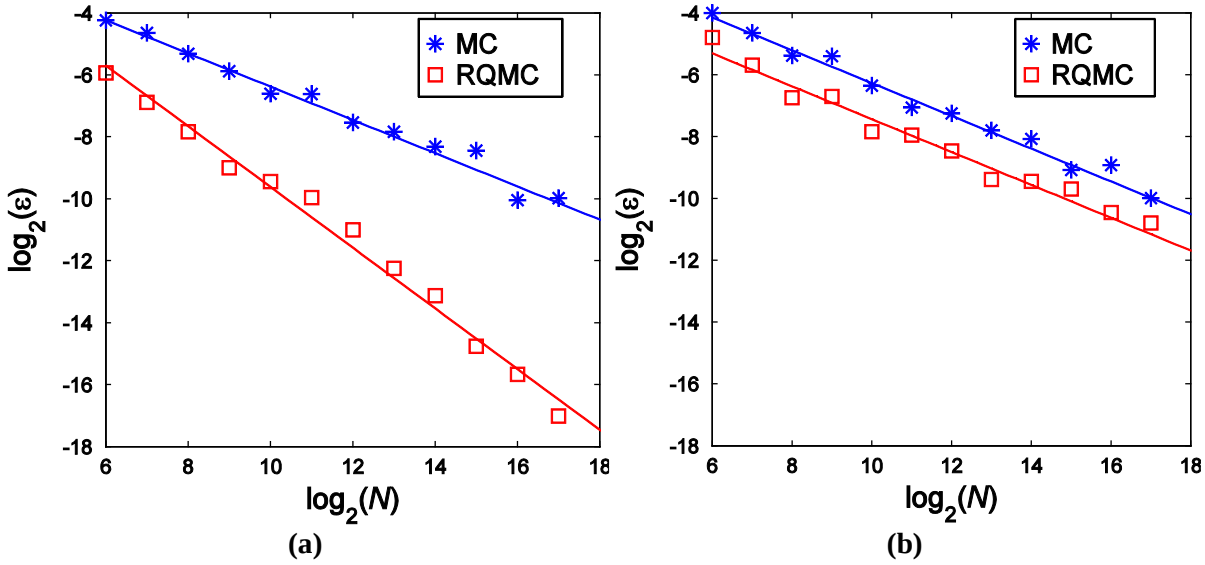


FIGURE 4.13 - UNCERTAINTY TRENDS FOR TEST CASE 3, ASSUMING (a) IDEAL AND (b) REALISTIC OPTICAL PROPERTIES

TABLE 4.3 - UNCERTAINTY TRENDS FOR TEST CASE 3

Φ	method	c	β	$N(\varepsilon = 10^{-3}) \times 10^{-3}$	$N(\varepsilon = 10^{-4}) \times 10^{-4}$
Perfect Optics					
$[-2, -3, 2.5, -6, 5, -5]^T$	MC	0.4892	0.5220	141.9	1169
	RQMC	1.260	0.9909	1.346	1.375
$[-3.5, -5, 1, -7, 5, -5]^T$	MC	0.4734	0.5256	123.0	983.4
	RQMC	1.121	0.9773	1.319	1.392
Realistic Optical Properties					
$[-2, -3, 2.5, -6, 5, -5]^T$	MC	0.2596	0.4607	173.9	2575
	RQMC	0.2186	0.4950	53.22	557.3
$[-3.5, -5, 1, -7, 5, -5]^T$	MC	0.4856	0.5178	154.0	1314
	RQMC	0.2181	0.5189	32.14	271.7

4.5 VALIDATION OF UNCERTAINTY ESTIMATION FOR RQMC

As previously discussed, in practical applications uncertainties cannot be evaluated based on an "exact" solution, and so must be estimated from the resulting sample mean, as in Equation (4.5). This relationship was presented, however, in the context of traditional Monte Carlo evaluation, and not for RQMC. As such, in order to ensure that the optimization methodology to be presented is robust, it is important to verify that these estimations are indeed appropriate. To confirm this, a number of trials are performed for two test problems representative of concentrator designs encountered during optimization. For each problem, one thousand objective function evaluations are performed using both MC and RQMC methods, and resulting uncertainty magnitudes are estimated as in Equation (4.5), as well as measured via comparison to an "exact" solution, taken to be the result of an evaluation using a sufficiently large sample size. For simplicity, the former is referred to as the *estimated uncertainty*, ε_{est}, and the latter as the *experimental uncertainty*, ε_{exp}.

The first problem examined is based on test case one, assuming perfect optics and with design parameters given by $\mathbf{\Phi} = [12, 4]^T$. For this problem, evaluations are performed using a sample size of 2^{12}, with an "exact" solution found using 2^{23} rays. This exact solution is deemed sufficiently accurate, as its corresponding estimated uncertainty is 1.74×10^{-7}, three orders of magnitude less than the lowest mean values of ε_{est} and ε_{exp}. The second problem is based on test case two, assuming realistic optics and with design parameters given by $\mathbf{\Phi} = [8.6, 2.3, 11.2]^T$. Evaluations are performed for this problem using 2^9 rays, with an "exact" solution found using 2^{23} rays. In this case, the estimated uncertainty of the exact solution is 1.04×10^{-4}, two orders of magnitude less than the lowest mean uncertainty.

Table 4.4 provides overall results of this validation study. As can be seen, over the one thousand evaluations, the estimated and experimental uncertainties are quite close, with ε_{est} over-predicting ε_{exp} by an average of 5.4%. Furthermore, when uncertainty predictions for individual evaluations are examined, it is found that on average 68% of experimental uncertainties are within $\pm\varepsilon_{est}$ of the exact solution, and 93% are within $\pm 2\varepsilon_{est}$, almost identical to the statistical predictions for a normal distribution. To help demonstrate this, Figure 4.14 shows a representative frequency distribution of experimental uncertainty, this one for RQMC evaluations of the second problem. In this figure, the indicated standard deviation σ is equal to the overall estimated mean, ε_{est}. Based on these results, it is deemed that objective function uncertainties as estimated using Equation (4.5) are indeed appropriate, and as such can reliably be used to quantify the accuracy of a given evaluation.

TABLE 4.4 - UNCERTAINTY VALIDATION RESULTS

test problem #	method	mean ε_{est}	mean ε_{exp}	fraction within $\pm\varepsilon_{est}$	fraction within $\pm 2\varepsilon_{est}$
1	MC	0.006830	0.005329	0.682	0.940
	RQMC	0.0007495	0.0006113	0.684	0.930
2	MC	0.02159	0.01860	0.660	0.918
	RQMC	0.01322	0.01009	0.691	0.945

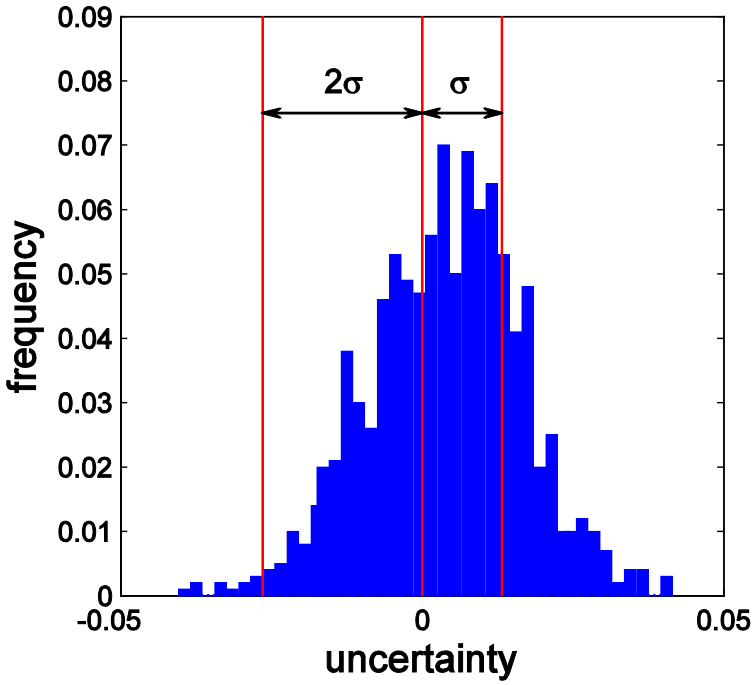


FIGURE 4.14 - FREQUENCY DISTRIBUTION FOR RQMC EVALUATIONS OF TEST CASE TWO

Chapter 5

DESIGN OPTIMIZATION METHODOLOGY

In order to identify the best performing solar concentrating collector geometry for the given conditions, an unconstrained local minimization scheme is applied to perform intelligent updates to the vector of design parameters, $\mathbf{\Phi}$, in an attempt to reduce the value of the objective function, $F(\mathbf{\Phi})$. These design updates continue until some termination criterion is satisfied, at which point the optimal design, $\mathbf{\Phi}^*$, corresponding to the minimum value, $F(\mathbf{\Phi}^*)$, has likely been reached. As previously described, in this thesis, $F(\mathbf{\Phi})$ is defined as the negative fraction of incoming radiation that is absorbed at the receiver surface (as estimated using a Monte Carlo or randomized quasi-Monte Carlo ray-tracing technique), and the design parameters stored in $\mathbf{\Phi}$ correspond to coordinates which define the concentrating surface geometry.

5.1 GRADIENT-BASED STOCHASTIC OPTIMIZATION

Gradient-based optimization techniques are commonly used when the feasible region is convex and the defined objective function is continuously differentiable. Such algorithms progressively reduce the objective function by updating the set of design parameters by

$$\mathbf{\Phi}_{k+1} = \mathbf{\Phi}_k + \alpha_k \cdot \mathbf{P}_k \tag{5.1}$$

where $\mathbf{P}_k$ is a descent direction vector and α_k is a scalar step size, both for the k^{th} design iteration. $\mathbf{P}_k$ is usually a function of the first- and sometimes second-order curvature of $F(\mathbf{\Phi})$. In this work, the steepest-descent direction is used, in which $\mathbf{P}_k$ is defined by the normalized gradient

$$\mathbf{P}_k = -\frac{\nabla F(\mathbf{\Phi}_k)}{\left\| \nabla F(\mathbf{\Phi}_k) \right\|} \tag{5.2}$$

Although search schemes such as Newton's method or the quasi-Newton method can achieve convergence rates far superior to that of steepest-descent in a deterministic setting, in this case neither the objective function nor its gradient can be evaluated deterministically. Instead, the gradient vector is estimated by applying divided differences, one-dimension-at-a-time, as

$$\nabla F(\mathbf{\Phi}_k) \approx \sum_{i=1}^{n} \frac{\Delta \widetilde{F}_i(\mathbf{\Phi}_k)}{2h_k} = \frac{1}{2h_k} \sum_{i=1}^{n} \Delta \widetilde{F}_i(\mathbf{\Phi}_k) \tag{5.3}$$

where $\Delta \widetilde{F}_i(\mathbf{\Phi}_k)$ is the divided-difference term given by

$$\Delta \widetilde{F}_i(\mathbf{\Phi}_k) = \widetilde{F}\left(\mathbf{\Phi}_k + h_k \hat{\mathbf{e}}_i\right) - \widetilde{F}\left(\mathbf{\Phi}_k - h_k \hat{\mathbf{e}}_i\right) \tag{5.4}$$

and h is the *difference parameter*, $\hat{\mathbf{e}}$ is a unit vector, and subscripts k and i denote the current design iteration and dimension, respectively. As before, the tilde is a reminder that estimations contain stochastic uncertainty.

It should be noted that the optimum identified by a gradient-based method is only guaranteed to be the global minimum if the objective function is strictly convex, which is not always the case. Nevertheless, finding a strong local minimum will often suffice for practical design purposes. A discussion of the convexity of the three test cases examined is presented in the following sections.

5.1.1 Uncertainties in Gradient Approximation

Each gradient approximation is subject to two sources of uncertainty. The first of these is due to the compounded effect of the uncertainty present in each objective function estimation. This compounded stochastic uncertainty can be estimated through the method of Kline and McClintock (1953), which states that for a function, R, dependent on n measured variables, $R = R(X_1, X_2, ..., X_n)$, the overall uncertainty can be estimated by

$$\varepsilon_R \approx \sqrt{\left(\frac{\partial R}{\partial X_1}\varepsilon_{X_1}\right)^2 + \left(\frac{\partial R}{\partial X_2}\varepsilon_{X_2}\right)^2 + \ldots + \left(\frac{\partial R}{\partial X_n}\varepsilon_{X_n}\right)^2} \tag{5.5}$$

where ε_R is the compounded uncertainty and ε_{X_i} is the uncertainty of the i^{th} measured variable. From Equation (5.3), then, a divided-difference gradient approximation in one dimension is subject to a stochastic uncertainty, denoted ε_{st}, given by

$$\varepsilon_{\text{st}} = \frac{1}{2h_k}\varepsilon_{\Delta\tilde{F}_i} \tag{5.6}$$

where $\varepsilon_{\Delta\tilde{F}_i}$ is the compounded uncertainty in the divided difference term, estimated by applying Kline-McKlintock to Equation (5.4), as

$$\varepsilon_{\Delta\tilde{F}_i} \approx \sqrt{\varepsilon_{F_1}{}^2 + \varepsilon_{F_2}{}^2} \tag{5.7}$$

Here, ε_{F_1} and ε_{F_2} denote the uncertainty for the forward and backward components of the divided difference term, respectively. Since h_k is relatively small and each evaluation is performed using the same sample size, it is assumed that component uncertainties are equal, ε_F, and so Equation (5.7) is reduced to

$$\varepsilon_{\Delta\tilde{F}_i} \approx \sqrt{2}\varepsilon_F \tag{5.8}$$

and so, finally, ε_{st} is given by

$$\varepsilon_{st} = \frac{1}{\sqrt{2}}\frac{\varepsilon_F}{h_k} \tag{5.9}$$

From this relationship, it can be seen that is ε_{st} is inversely proportional to h_k and directly proportional to ε_F, which from previous discussion we know decreases to zero as the sample size, N, approaches infinity.

The second source of uncertainty is due to the fact that the divided-difference gradient approximation is simply a Taylor-series expansion with truncated higher order terms. Because in practice the exact gradient is

unknown, the magnitude of this Taylor Series uncertainty, denoted ε_{TS}, cannot be known, although for a deterministic function, it decreases to zero as the difference parameter h_k approaches zero. In this case, however, because the objective function is subject to uncertainty, excessively small values of h_k can compromise accuracy of the gradient approximation, unless a sufficiently large sample size is used. This is because a small h_k is associated with a small value for $\Delta\widetilde{F}_i(\mathbf{\Phi}_k)$ in Equation (5.3) (the difference between the forward and backward objective function estimations), thus increasing sensitivity to stochastic uncertainties, ε_F.

As the above discussion implies, the accuracy of a gradient approximation is highly dependent on an appropriate choice of parameters h_k and N, to achieve a balance of the two interrelated sources of uncertainty. To demonstrate this relationship, a series of Monte Carlo simulations are performed to estimate the view factor between two orthogonal, infinite flat surfaces with a common edge, as shown in Figure 5.1 (a). For this one-dimensional problem, the length of surface two, y, is varied by h and the gradient is estimated using the divided-difference scheme. Errors in the gradient approximations are quantified through comparison with results found from the known analytical solution and its corresponding gradient (Siegel and Howell 2002)

$$F_{1-2} = \frac{1}{2}\left(1 + y - \sqrt{1+y^2}\right)$$
$$\nabla F_{1-2} = \frac{\partial F_{1-2}}{\partial y} = \frac{1}{2}\left(1 - \frac{y}{\sqrt{1+y^2}}\right) \tag{5.10}$$

where $F_{1\text{-}2}$ denotes the view factor from surface one to surface two. The resulting magnitudes of ε_{TS}, ε_{st}, and their combined error found by applying the Kline-McClintock formula, $\varepsilon_{combined}$, are shown in Figure 5.1 (b). As can be seen, ε_{TS} and ε_{st} follow the expected trends with h, while $\varepsilon_{combined}$ reaches a minimum at about $h = 0.12$, increasing to either side. This result demonstrates that an inappropriate value for h can result in a large combined uncertainty, which can

lead to an inaccurate gradient approximation. In addition, it should be noted that this relationship is not constant throughout optimization, changing as the current design and sample size are modified. The results shown here are for two surfaces each of unit length, with view factor estimations resulting from Monte Carlo evaluations consisting of 10^6 rays.

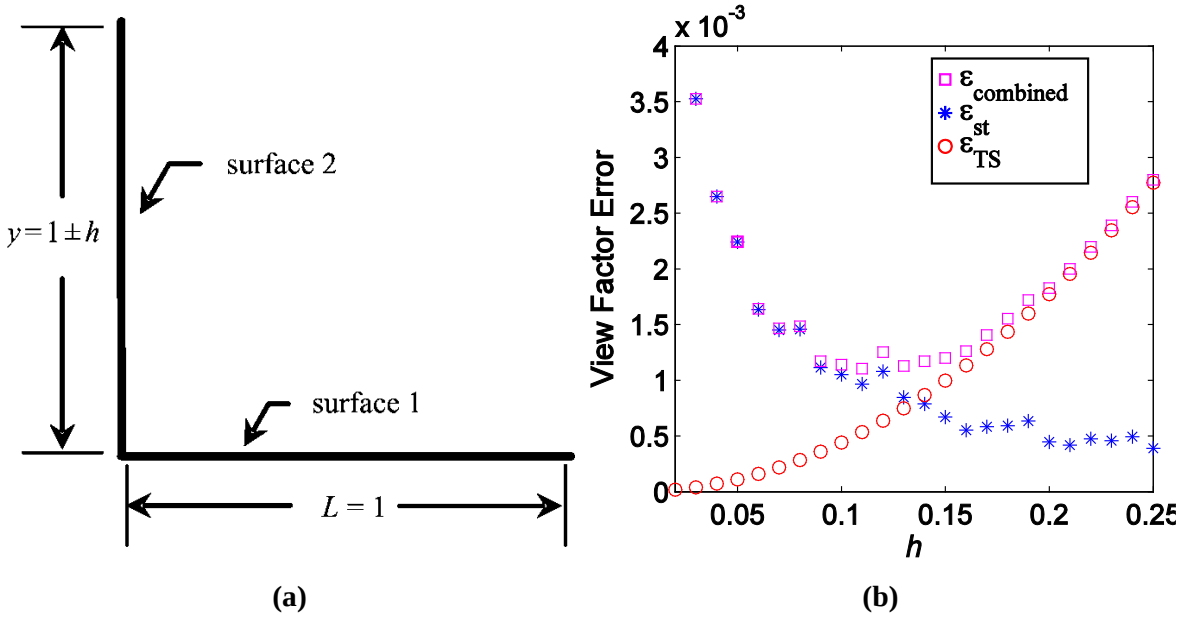


FIGURE 5.1 - VIEW FACTOR EXAMPLE (a) GEOMETRY AND (b) ESTIMATION ERRORS

5.1.2 KIEFER-WOLFOWITZ TECHNIQUE

In order for a gradient-based optimization strategy to successfully locate an optimal design, each arbitrary iteration, k, must involve a step of appropriate size, α_k, in an appropriate direction, $\mathbf{P}_k$. In this case, this implies that α_k is defined in a way that will not prevent convergence, and that the difference parameter, h_k, and sample size, N, are defined such that approximated gradients are of sufficient accuracy. The technique proposed by Kiefer and Wolfowitz (1952) is the standard method for optimization of stochastic systems such as this, for which unbiased estimates are available for the objective function, but not the gradient (Pflug 1996). This method performs steepest-descent updates

based on divided-difference gradient approximations, with step size and difference parameter specified by the power-law sequences

$$\alpha_k = \frac{\alpha_0}{k^a}, \quad h_k = \frac{h_0}{k^b} \tag{5.11}$$

The values of $a = 1$ and $b = 1/3$ are the standard values recommended for the above denominator exponents (Pflug 1996).

In this approach, the designer specifies initial values for step size and difference parameter, α_0 and h_0, usually chosen based on heuristics, and these values are decreased with each successive iteration. Such a diminishing sequence is standard practice for control of step size, as it allows the first few design steps to cover substantial ground, and then slows progress as an optimum is neared to prevent it from being passed by. A diminishing difference parameter, on the other hand, allows initial gradient approximations to be performed based on broader objective function curvature, with estimations increasingly localized as the optimum is approached. Because decreasing the difference parameter in this way also increases the demand for objective function accuracy, often used alongside is an increasing sequence to control sample size, decreasing uncertainty with each successive iteration. Such increasing sequences, referred to as means of "sampling control", typically take the form of the logarithmic series (Daun et al. 2003b, Simha 2003)

$$N = A\log k + B \tag{5.12}$$

where A and B are heuristics chosen based on an initial sample size refinement study.

As an initial demonstration, the Kiefer-Wolfowitz (KW) method as described above is applied to the optimization of test case one, with the objective function estimated using traditional Monte Carlo. Diminishing sequences are used to control both step size and difference parameter, with the recommended power-law exponents and with initial values of $\alpha_0 = 2$ and $h_0 = 1/4$, found to work well from extensive testing. The above logarithmic series is used to control

sample size, with heuristics of $A = 4000$ and $B = 4000$, found to provide sufficient accuracy from trial-and-error. Design updates are allowed to continue until the gradient magnitude (based on the L2-norm) is equal to zero, indicating that an optimal design has been reached. Using these settings, the algorithm converges after four updates, following the design path shown in Figure 5.2, with initial and final designs corresponding to those shown in Figure 5.3 (a) and (b), respectively. It should be noted that, for this particular problem, there is not a single distinct optimum: instead, there is a region of optimal designs that closely resemble a perfectly symmetrical parabola, with foci lying completely within the receiver circle and so achieving an objective function of $F(\boldsymbol{\Phi}^*) = -1$.

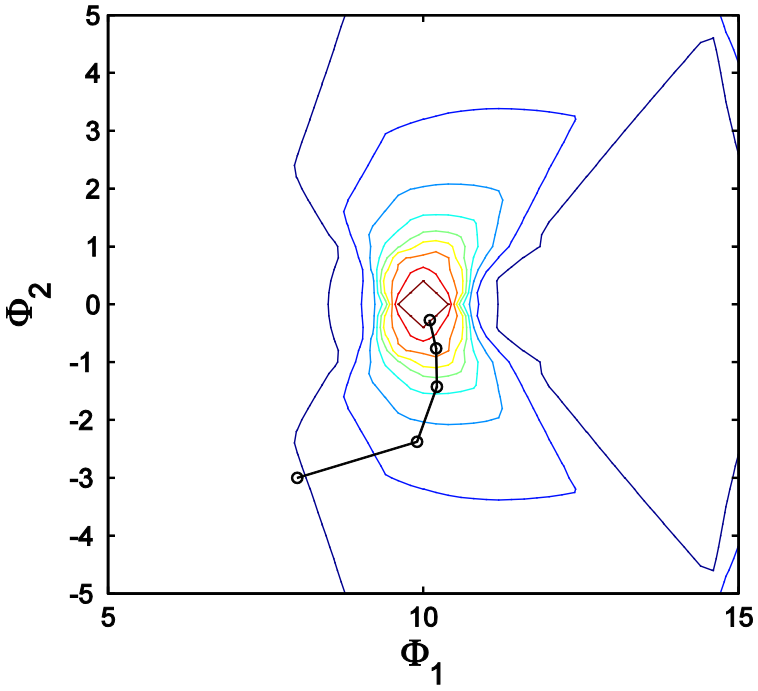


FIGURE 5.2 - KW DESIGN PATH FOR TEST CASE ONE

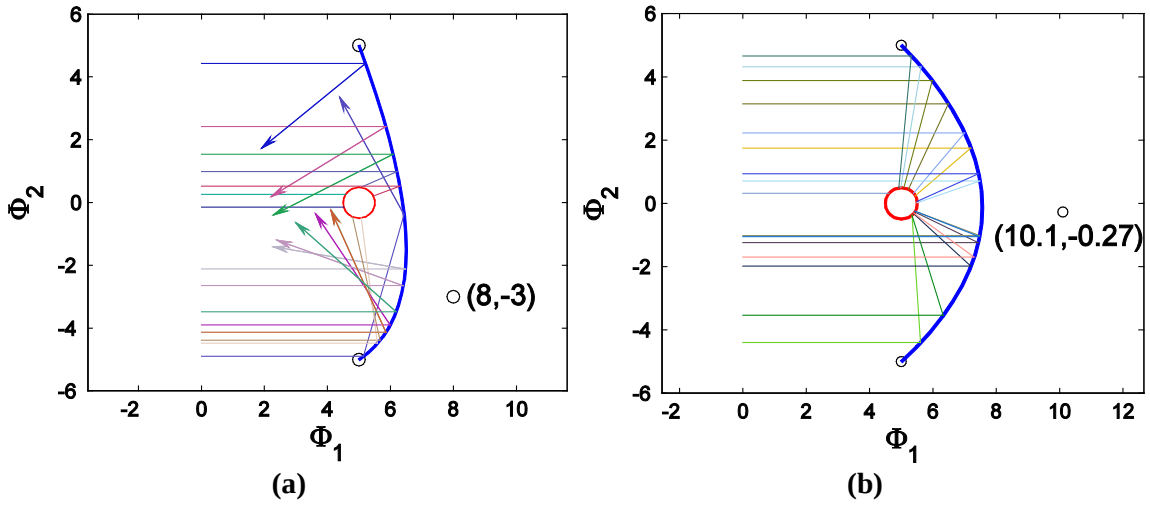


FIGURE 5.3 - KW OPTIMIZATION OF TEST CASE 1: (a) ORIGINAL AND (b) OPTIMIZED DESIGN

Although the KW technique has been successfully applied in the optimization of a number of stochastic systems, in practice its performance is highly dependent upon an appropriate choice of initial parameters and heuristics, which is not always straightforward. In an attempt to develop a method that is both more computationally efficient and less-dependent on such heuristics, alternate means of step size and sample size control are examined and presented below.

5.2 Sample Size Control

The performance of the aforementioned gradient-based stochastic optimization method relies heavily on an appropriate choice of sample size. If insufficient rays are used, objective function estimations and resulting gradient approximations will be inaccurate, leading to an increase in the number of required design updates or even preventing convergence entirely. Using a larger sample size than is necessary, however, causes an unneeded increase in computational demand. Indeed, some compromise must exist for which the sample size is just large enough to ensure design improvement, without resulting in "wasted" computation. The logarithmic sample size sequence described above attempts to achieve this balance, but in reality exact

requirements will depend on the unique objective function behaviour and current difference parameter, and so cannot always be achieved through predetermined heuristics. This section provides development of an alternative approach, in which the sample size is tailored to the local objective function at each iteration, through the use of an accuracy-based control condition.

5.2.1 CONDITION-BASED APPROACH

In order to minimize overall computation, an ideal optimization algorithm would continue to ray-trace only until a desired level of accuracy is reached, in order to be confident that a design update will indeed result in an improvement in performance. The resulting sample size, then, would be just sufficient to ensure that the reduction in the objective function is "statistically-significant" at each iteration. Such an approach can be emulated through careful monitoring and comparison of the magnitude of expected improvement and its associated uncertainty. From a Taylor-series expansion of $F(\mathbf{\Phi}_k)$,

$$F\left(\mathbf{\Phi}_k + \alpha_k \mathbf{P}_k\right) \approx F\left(\mathbf{\Phi}_k\right) + \alpha_k \mathbf{P}_k{}^{\mathrm{T}} \nabla F\left(\mathbf{\Phi}_k\right) \tag{5.13}$$

the expected improvement, denoted U_k, can be estimated as

$$U_k = F\left(\mathbf{\Phi}_k\right) - F\left(\mathbf{\Phi}_k + \alpha_k \mathbf{P}_k\right) \approx \alpha_k \mathbf{P}_k{}^{\mathrm{T}} \nabla F\left(\mathbf{\Phi}_k\right) \tag{5.14}$$

By substituting $\mathbf{P}_k$ as defined in Equation (5.2), this reduces to

$$U_k \approx \alpha_k \frac{\nabla F\left(\mathbf{\Phi}_k\right)^{\mathrm{T}}}{\left\|\nabla F\left(\mathbf{\Phi}_k\right)\right\|} \nabla F\left(\mathbf{\Phi}_k\right) = \alpha_k \left\|\nabla F\left(\mathbf{\Phi}_k\right)\right\| \tag{5.15}$$

By computing the norm of the gradient defined in Equation (5.3), U_k becomes

$$U_k \approx \frac{\alpha_k}{2h_k} \left[\sum_{i=1}^{n} \Delta F_i^{\,2} \right]^{1/2} \tag{5.16}$$

where $\Delta \widetilde{F}_i(\mathbf{\Phi}_k)$ is shortened to ΔF_i for convenience.

This expected improvement is subject to an uncertainty, denoted ε_{Uk}, due to the compounded effect of stochastic uncertainties in the divided-difference term, ΔF_i. By applying the Kline-McClintock formula, ε_{Uk} can be estimated as

$$\varepsilon_{U_k} = \left[\sum_{i=1}^{n} \left(\frac{\partial U_k}{\partial \Delta F_i} \right)^2 \left(\varepsilon_{\Delta F_i} \right)^2 \right]^{1/2} \tag{5.17}$$

where the sensitivity of U_k to ΔF_i can be found by differentiating Equation (5.16), resulting in

$$\frac{\partial U_k}{\partial \Delta F_i} \approx \frac{\alpha_k}{2h_k} \Delta F_i \left(\sum_{i=1}^{n} \Delta F_i^2 \right)^{-1/2} \tag{5.18}$$

and the uncertainty in each ΔF_i term, $\varepsilon_{\Delta F_i}$, is defined as in Equation (5.8). Substituting both of these relations into Equation (5.17) yields

$$\begin{aligned} \varepsilon_{U_k} &= \left[\sum_{i=1}^{n} \left(\frac{\alpha_k}{2h_k} \Delta F_i \left[\sum_{i=1}^{n} \Delta F_i^2 \right]^{-1/2} \right)^2 \left(\sqrt{2} \varepsilon_F \right)^2 \right]^{1/2} \\ &= \frac{\alpha_k}{\sqrt{2} h_k} \varepsilon_F \left[\sum_{i=1}^{n} \Delta F_i^2 \right]^{-1/2} \left[\sum_{i=1}^{n} \Delta F_i^2 \right]^{1/2} \\ &= \frac{\alpha_k}{\sqrt{2} h_k} \varepsilon_F \end{aligned} \tag{5.19}$$

In order to be confident in the statistical significance of the expected improvement, its magnitude should be much greater than that of its associated uncertainty. This can be done by ensuring

$$U_k > K \cdot \varepsilon_{U_k} \tag{5.20}$$

where K is a scalar value chosen to control the degree of confidence. Substituting the relations for U_k and ε_{Uk} from Equations (5.15) and (5.19) above results in the final inequality, henceforth referred to as the "sample size control condition"

$$\frac{\sqrt{2}h_k \left\| \nabla F\left(\mathbf{\Phi}_k\right) \right\|}{\varepsilon_F} > K \tag{5.21}$$

which can be easily validated after each gradient approximation is performed. In application, the most appropriate value for K depends on the particular design problem being solved: a lower value leads to faster, less accurate design updates, which may suffice for a simple problem, but in some cases can prevent convergence. For the problems dealt with throughout this thesis, a value of $K = 5$ is found to result in consistent convergence, without leading to excessively large sample sizes.

The sample size control condition can be used to ensure that each gradient approximation is performed with close to the minimum required sample size, thus minimizing the amount of excess computation, while helping to guarantee that progress will indeed be made. The strategy is as follows: at each iteration the gradient is first approximated using a relatively small sample size, N; if the sample size control condition is not satisfied, the estimation is augmented by tracing another N rays, effectively doubling the sample size to $2N$; this is repeated until the condition is satisfied, at which point the optimization routine proceeds and a design update is performed. In this way, the sample size is continuously adapted to current conditions of objective function curvature and difference parameter as the optimization routine proceeds toward the minimum. Appendix D provides an example of how such an estimation augmentation may be performed.

To illustrate this sample size control strategy, it is applied to the optimization of test case one, with all other parameters identical to the trial performed in Section 5.1.2. Using an initial sample size of two hundred and a value of $K = 5$, the design follows the path shown in Figure 5.4, very similar to that of the previous trial. Although the two sample size control strategies result in similar design updates, the trial using the condition-based control requires significantly fewer rays to be traced than that based on the chosen logarithmic

sequence (67 000 compared to 195 760). In addition, as can be seen in Table 5.1, the trend followed by the sample size based on the condition-based approach is actually contrary to that achieved by the increasing logarithmic sequence. For the first two iterations, sample size is increased multiple times to 6400, while for the remainder only two hundred rays are required. This is because, as can be seen from the objective function contour shown in Figure 5.4, the gradient magnitude is smaller in the region surrounding the initial design, causing the gradient approximation to be more sensitive to stochastic noise. As the optimal design is approached, the gradient slope increases, allowing estimates with greater noise to be acceptable, until finally the optimum is identified and the gradient norm is found to be zero.

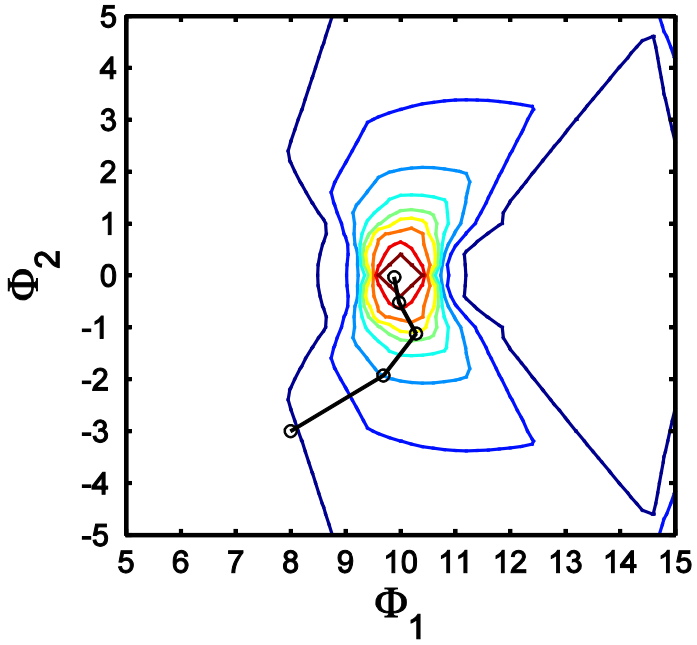


FIGURE 5.4 - DESIGN PATH FOR TRIAL USING CONDITION-BASED SAMPLE SIZE CONTROL

TABLE 5.1 - PARAMETER VALUES FOR ABOVE DESIGN OPTIMIZATION

k	$F(\Phi_k)$	N	U_k	ε_{Uk}	U_k / ε_{Uk}
1	0.1947	6400	0.1711	0.03231	5.2962
2	0.3994	6400	0.1469	0.02296	6.3986
3	0.6800	200	0.4429	0.07486	5.9169
4	0.9800	200	0.3145	0.02072	15.1772
5	1.000	200	0.0	0.0	N/A

It is believed that, through the use of such a condition-based approach to sample size control, overall optimization is made less dependent upon choice of initial heuristics. Instead of having to perform an initial error study in order to gain insight into an appropriate range of sample sizes, optimization can instead be performed without worry using some arbitrarily small initial sample size. At the same time, performance is made less-dependent upon appropriate choice of initial difference parameter, since this approach continuously ensures that the sample size is sufficient given the current difference parameter and objective function characteristics. Throughout the remainder of this thesis, all optimization attempts employ this approach to sample size control.

5.3 STEP SIZE CONTROL

The performance of a gradient-based optimization scheme is highly dependent on performing design updates of appropriate step size. Although a diminishing step size sequence is the standard choice for optimization of stochastic systems, other strategies exist which can achieve more reliable results while being less-dependent upon designer-specified heuristics. This section discusses the merits of strategies based on a diminishing sequence and an approximate line search, and compares results for optimization of test case one.

5.3.1 DIMINISHING SEQUENCE

One way in which convergence to an optimum may be ensured is by defining step size through a predetermined sequence which satisfies the standard conditions (Simha 2003)

$$\alpha_k > 0, \quad \alpha_k \to 0, \quad \sum_{k=1}^{\infty} a_k = \infty \tag{5.22}$$

which require the step size to decrease with each iteration, without ever reaching zero and thereby halting progress. The diminishing sequence used in the Kiefer-Wolfowitz algorithm is an example of such a strategy. Although such

approaches have good theoretical convergence characteristics, in practice convergence tends to be slower than most alternatives (Bertsekas 1999). Furthermore, performance is highly dependent on the chosen initial step size, α_0. Although in theory a decreasing but non-vanishing step size should always allow the algorithm to arrive at an optimum after a sufficient number of iterations, in practice it may take unreasonable computational effort to do so. In cases for which α_0 is too small, for example, the step size can quickly diminish and prevent sufficient progress from being made. Figure 5.5 shows the design path of an optimization attempt with the same settings as for the trial performed in Section 5.1.2, but with a smaller initial step size of one. In this case, although the design eventually reaches the optimum, 21 updates are required, compared only four updates for $\alpha_0 = 2$. Choosing too large an initial step size, on the other hand, can cause the design to repeatedly pass over the optimum before finally converging, resulting in wasted CPU effort.

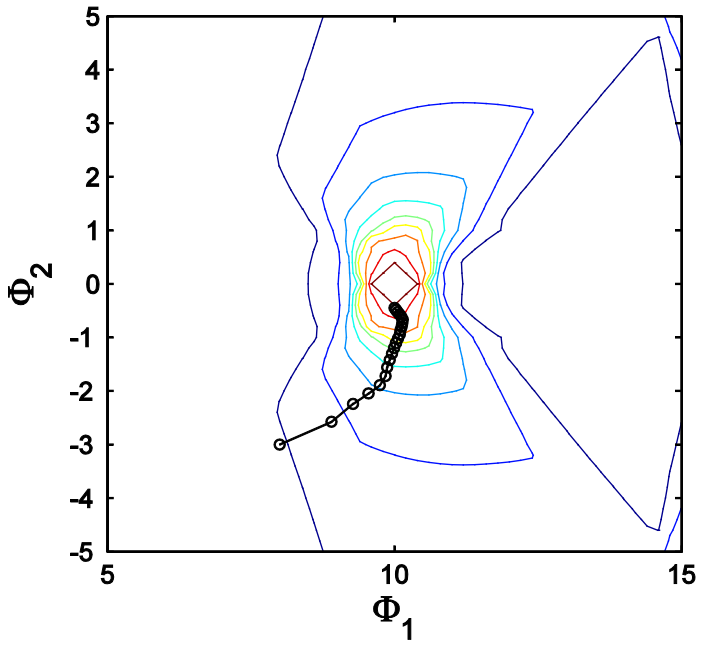


FIGURE 5.5 - DESIGN PATH FOR K-W OPTIMIZATION WITH SMALL α_0

For the simple case presented above, an appropriate choice for initial design and initial step size is obvious. For practical problems with an unknown solution and higher problem dimensionality, however, this is not the case, and it

may be only after substantial trial and error that a suitable initial step size is found.

5.3.2 APPROXIMATE LINE SEARCH

An alternative to defining step size through a predetermined series is to perform a line search along each step direction, finding the step size which minimizes $F(\mathbf{\Phi}_k + \alpha_k \mathbf{P}_k)$, to within some specified tolerance. In this way, the step size continuously adapts to the current local objective function, and so does not require any designer-specified heuristics. Although an exact line search is ideal, it is not possible for a non-deterministic problem such as this; instead, the optimal step size is commonly approximated through a polynomial curve-fit to a number of test points. Because higher-order function information is unavailable in this case, a quadratic line search (QLS) is employed.

The first step of a quadratic line search is to identify a set of points, a, b, and c, that form a three-point pattern around the minimum along the step direction such that

$$a < b < c \quad \text{and} \quad g(a) > g(b) < g(c) \tag{5.23}$$

where $g(a)$ is the value of the objective function after a step of a in the step direction, $\mathbf{P}_k$. Simple search algorithms exist for efficiently identifying a three-point-pattern, such as that presented in Belegundu and Chandrupatla (1999) and outlined in Appendix B. This search works by consecutively evaluating points at distances of Δ, $\gamma\Delta$, $\gamma^2\Delta$, ... from the current design until such a pattern is found. In this strategy, Δ is referred to as the *initial search step*, and γ as the *expansion parameter*. In order to ensure that these evaluations are sufficiently accurate, unless otherwise stated, in this thesis the initial search step, Δ, is set equal to the current difference parameter, h_k, since the sample size has already been increased as necessary for a step of this size.

Once a three-point pattern is found, the minimizing step size α^* is approximated based on the quadratic interpolation (Bertsekas 1999)

$$\alpha^* = \frac{1}{2}\frac{g(a)(c^2-b^2)+g(b)(a^2-c^2)+g(c)(b^2-a^2)}{g(a)(c-b)+g(b)(a-c)+g(c)(b-a)} \tag{5.24}$$

To illustrate this, design optimization is again performed on the problem dealt with in Section 5.1.2, with step size determined through a quadratic line search, and all other settings identical. Three-point-pattern search parameters of $\Delta = 0.5$ and $\gamma = 1.5$ are used. Figure 5.6 (a) shows a plot of the line search performed for the initial design step, along the identified direction of steepest descent. The five blue circles indicate test points examined, with the rightmost three identified as a three-point pattern. The solid blue line shows the resulting interpolated quadratic function, while the dashed black line indicates the actual objective function, as estimated using a fine mesh of Monte Carlo evaluations with a large sample size. As can be seen, the optimal step size is estimated to be about 2.52 units, quite close to the actual optimum of about 3.4. As can be seen from the resulting overall design path as shown in Figure 5.6 (b), in this case only two design updates are required to reach an optimal design.

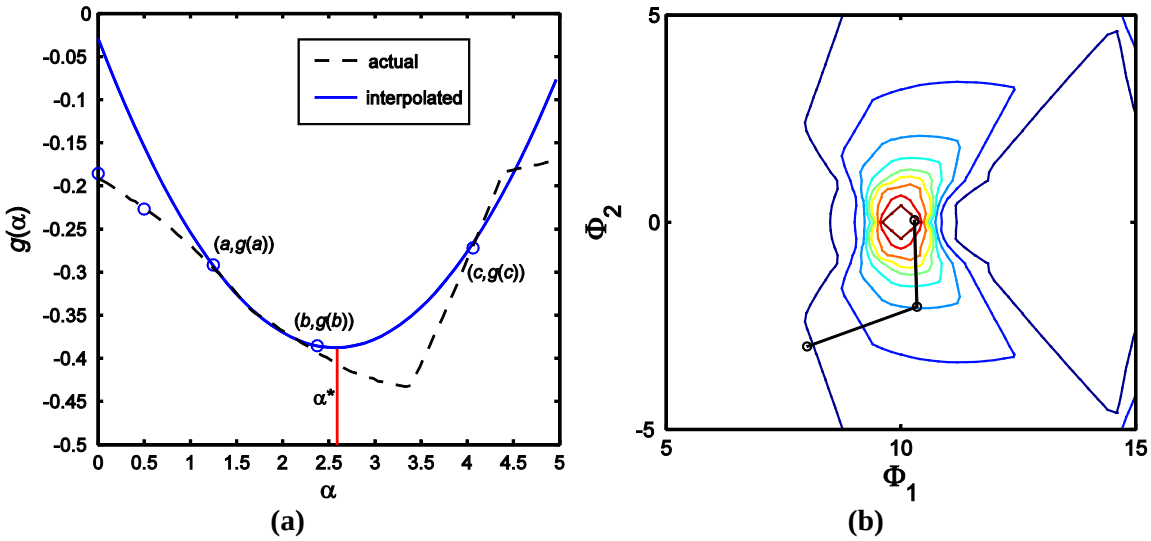


FIGURE 5.6 - QLS OPTIMIZATION (a) INITIAL LINE SEARCH AND (b) DESIGN PATH

5.3.3 Step Size Performance Study

In order to compare the above step size control strategies, a number of optimization attempts using various combinations of initial parameters are

performed for test case one. All simulations are based on Monte Carlo estimations with a sample size controlled using the condition-based sample size control strategy, with an initial sample size $N = 20\,000$ and accuracy scalar $K = 5$, found to result in consistent convergence. Three-point pattern searches are performed using the same sample size as deemed appropriate for gradient approximation, with an initial line search step size of $\Delta = 1/2$. In all trials, the Kiefer-Wolfowitz recommended diminishing sequence for difference parameter is used, with $h_0 = 1/4$. Each CPU time provided is based on the performance of a 1.83 GHz dual core laptop PC (only using one core) with 2.49 GB of RAM. Finally, each optimization attempt is allowed a maximum of 300 seconds to arrive at the optimum, after which point it is taken to be a failure and hence terminated.

Table 5.2 shows the elapsed CPU time and number of design updates for each attempt, based on the average of five repeated trials. As can be seen, trials employing QLS tend to converge in fewer design iterations than those using the diminishing step size approach. Due to the additional computation required for each line search, however, little to no improvement is seen in total computational time to convergence, with comparable results seen for the both strategies. There is no clear choice that consistently achieves superior performance, with results dependent on the initial design and strategy-specific settings. It can be seen, however, that for the settings tested, results for QLS have a significantly lower standard deviation than those for diminishing step size, indicating that the success of QLS is less-dependent on these designer-specified settings. This is further confirmed through observation of the number of "failures" experienced: the algorithm failed to reach the optimum within 300 seconds seven out of the ninety trials using the diminishing sequence approach, but not once when using the quadratic line search method. Although both schemes should, given reasonable parameter choices, converge successfully, the quadratic approach is seen to be more reliable. Based on these results, the

quadratic line search scheme is used exclusively throughout the remainder of this thesis.

TABLE 5.2 - RESULTS OF STEP SIZE STUDY

Φ_0	Quadratic Line Search			Diminishing Step Size		
	γ	time (s)	# of updates	α_0	time (s)	# of updates
$\begin{bmatrix} 8 \\ -3 \end{bmatrix}$	1	36.29	3	1	54.52	12.8
	1.25	34.47	3	1.5	22.69	5
	1.5	34.96	3.8	2	14.44	3.2
	2	37.96	4.2	2.5	12.45	2.6
	2.5	29.94	3.2	3	22.68	5
	3	34.38	4	4	108.80	8.8
average		34.67	3.53		39.26	6.23
standard deviation		2.684	0.5317		37.29	24.01
$\begin{bmatrix} 12 \\ 4 \end{bmatrix}$	1	141.88	3.4	1	242.62	14.4
	1.25	78.80	3.8	1.5	108.22	10.8
	1.5	49.29	2.2	2	210.50	6.6
	2	46.29	3	2.5	68.94	7
	2.5	128.10	4.4	3	122.19	9.8
	3	104.94	3	4	97.00	8.4
average		91.55	3.30		141.58	9.50
standard deviation		40.11	0.7563		68.86	2.887
$\begin{bmatrix} 11 \\ -4 \end{bmatrix}$	1	65.79	4.8	1	179.81	18.8
	1.25	95.63	3.8	1.5	34.63	5.6
	1.5	93.97	2.6	2	41.01	5.6
	2	61.42	3.8	2.5	39.14	5.2
	2.5	76.57	4.4	3	77.99	6.2
	3	44.63	4	4	51.74	5.8
average		73.00	3.90		70.72	7.87
standard deviation		19.78	0.7457		55.67	5.366

5.4 TERMINATION CRITERIA

In order to minimize computational effort, it is important to ensure that an optimization algorithm is terminated once an optimal design has been reached. A number of termination criterion may be used to help identify when this has occurred, some better suited to a particular type of design problem. For the

problems dealt with in this thesis, no single approach is found to work perfectly, so all three of the below criteria are applied in parallel.

5.4.1 Gradient-Based Termination

One method of termination typically used for a gradient-based algorithm is simply to continue until the local objective function gradient has a magnitude of zero, indicating that an optimum has been reached. For most practical problems, an optimal region is infinitely small and cannot be identified exactly, so termination is based on locating a design with a sufficiently small gradient norm, based on a threshold value such as 'terminate if $\|\nabla F(\mathbf{\Phi}_k)\| < 0.001$'. Such a gradient-based approach is ideal in theory, as it causes updates to cease as soon as an optimal design has been reached, preventing additional unnecessary computational effort. For optimization in stochastic systems, however, uncertainty in objective function estimations can often cause very small gradient norms to be overestimated, rendering this method ineffective unless a relatively large sample size is used. Through the use of one of the aforementioned sample size control strategies, such problems may be avoided. The condition-based approach, in particular, has been found to greatly improve the success of such termination, as it causes the sample size to increase greatly in regions of low gradient magnitude. This criterion is used throughout this thesis, with a threshold value chosen to suit the particular design problem.

5.4.2 Significant Improvement of Objective Function

In some cases, although noise may prohibit an algorithm from identifying an optimum based on gradient magnitude, it can be assumed that an optimal design has been found once design updates no longer achieve significant improvement in the objective function. Such a conclusion can be reached through application of a technique employed in a commercial evolutionary algorithm package, and described in Law and Kelton (2000). This technique compares the current performance to that of the last few iterations, based on the

two designer-specified variables: *GNI*, the number of generations (analogous to design iterations) allowed without significant improvement, and; *PI*, the percentage improvement regarded as being significant. Termination occurs at iteration k, based on the current objective function, $F(k)$, if the following condition holds true:

$$\frac{\left|F(k)-F(k-GNI)\right|}{\left|F(k-GNI)\right|} \leq \frac{PI}{100} \tag{5.25}$$

For example, say $GNI=5$, and $PI=5$, and say iterations $k-5$ through $k-1$ have occurred without termination. Termination will occur at generation k if $|F(k)-F(k-5)| \,/\, |F(k-5)| \leq 0.05$, indicating that less than a five percent improvement has been achieve over the last five design generations.

Because, in this thesis, objective function evaluations are subject to stochastic uncertainty, care needs to be taken to ensure that the algorithm is terminated only once improvement has indeed stopped, and not when it has simply appeared to due to random fluctuation. Instead of using a single design to determine performance, then, the average performance of a number of recent designs can be taken. By representing "current" performance with the average of that of the last few iterations, and "previous" performance with the average of the few before that, improvement can be made less sensitive to fluctuations. For the optimization attempts later performed in this thesis, the "current" average is based on the latest five designs, and the "previous" average is based on the ten designs before that. Once enough iterations have been performed to define these averages, termination occurs if performance has not improved by at least 0.5 percent, unless otherwise stated.

5.4.3 ALLOTTED EFFORT

For design problems for which no other termination criteria are found to be successful, a designer will often simply terminate optimization once an allotted amount of computational time or number of design iterations has been exceeded.

At this point, the hope is that enough updates have been performed to result in a satisfactory design. This approach is obviously a last resort, as it offers no indication that a true optimal design has been reached, and may also result in a great deal of wasted computation if an optimal design is found much before termination. Throughout this thesis, each attempt at design optimization is allowed a maximum of thirty minutes. The chosen maximum number of allowed iterations depends on the particular problem.

5.5 Recap of Optimization Algorithm

Before the results of the design optimization procedure are discussed, a brief summary of the final algorithm is presented. Before the algorithm is started, of course, appropriate settings are chosen for the parameters of initial design, $\mathbf{\Phi}_0$, initial difference parameter, h_0, and initial sample size, N. This is discussed in greater detail in the following sections. At this point, optimization can begin, following these steps:

1. A divided-difference gradient approximation is performed based on the initial sample size, N, and the resulting estimation uncertainty is calculated.
    a. If the sample size control condition (with scalar $K=5$) is not satisfied, the sample size is repeatedly doubled until it is.
2. If the current iteration, k, is greater than one, the termination criteria are then checked:
    a. Calculate the gradient norm, $\|\nabla F(\mathbf{\Phi}_k)\|$, from the divided difference results. If it is less than the specified threshold value, go to step 6.
    b. Calculate the percentage improvement, if enough iterations have occurred to define the necessary moving averages. If the improvement is less than the value specified, go to step 6.
    c. Check the total number of design updates and the total elapsed CPU time. If either exceed its specified maximum, go to step 6.

3. A three-point-pattern search is then performed along the identified steepest descent direction, $\mathbf{P}_k$. An expansion parameter, γ, of 1.5 is used, and as previously discussed, the initial search step, Δ, is set equal to the current difference parameter, h_k.
4. A quadratic interpolation is then performed on the identified three-point-pattern, in order to identify an appropriate step size, α_k.
5. A design update is then performed, in which the design takes a step of α_k in the direction $\mathbf{P}_k$. Return to step 1.
6. Design updates have been terminated. The optimal design is taken to be that which has the lowest value of $F(\mathbf{\Phi}_k)$ (i.e. the best performance) of the designs identified.

5.6 Performance of Algorithm for Test Cases

In order to demonstrate the above-developed design optimization methodology, the final algorithm is applied to the three test cases introduced previously. The goal of this study is twofold: to evaluate the performance and reliability of the overall optimization methodology, and; to determine the degree of improvement achieved through application of randomized quasi-Monte Carlo estimation over traditional Monte Carlo. Because each test case introduces a unique element of design complexity, for each a discussion is provided of any required changes to the parameter settings. The results of various optimization trials are then presented, based on objective function estimations performed using both MC and RQMC. For test cases two and three, trials are performed for both ideal and "realistic" optical properties, and the resulting differences are discussed. In order to gain insight into overall performance trends, each set of results presented is based on the average outcome of ten repeated trials. All computations are performed using the MATLAB functions included in Appendix E, on a 1.83 GHz dual core laptop PC (using only one core).

5.6.1 TEST CASE ONE

A series of trials are performed for the design problem of test case one, from three different initial designs: $\mathbf{\Phi}_0 = \{[8, -3]^T, [12, 4]^T, [11, -4]^T\}$. The threshold for termination based on gradient magnitude is set to 0.0001, an initial sample size of $N = 256$ is used, and the initial difference parameter is set to $h_0 = 0.5$. Average results of the trials are provided in Table 5.3. As can be seen, in every case the algorithm successfully converges to the ideal optimal design, similar to that shown in Figure 5.3(b), and corresponding to an objective function value of $F(\mathbf{\Phi}^*) = -1$. In every trial, termination occurs due to a small gradient magnitude, since once the design reaches a point at a distance of about h_k from the edge of the optimal region, the gradient norm is found to be exactly zero.

For this relatively simple design problem, the optimization algorithm proves highly successful, consistently reaching the expected optimal solution. This is because, as can be seen from the objective function contours as shown in Figure 5.7, the objective function associated with this problem has a single optimal region and is convex in the surrounding design space. As such, the gradient-based approach used easily converges on the optimum, although time to convergence depends on the initial design used and their proximity to the optimal solution. Trials starting from the second initial design, in particular, take over twice as long as other trials, due to local curvature of the objective function gradient. As illustrated by Figure 5.7, the second initial design starts on one side of a slight "valley" in the objective function contour, causing the first few steepest-descent updates to "zig-zag" toward the optimum. In addition to resulting in more design updates, this design path spends the first few iterations in a region with low gradient magnitude, resulting in larger required sample sizes (as discussed previously).

From the results in Table 5.3, it can be seen that there is no appreciable difference in optimal performance or average number of design updates for MC- and RQMC-based trials. This is evidence that both approaches result in a

similar design path, which makes sense since the sample-size control should ensure about the same level of accuracy throughout optimization. As expected, however, RQMC-based trials converge significantly faster than those using traditional MC, with an average improvement of a factor of about 4.1, while requiring about 4.9 times fewer rays. The difference between these two factors can be explained by the 20% increase in required CPU time per ray for RQMC over MC, likely due to the generation and subsequent randomization (and storage in memory) of the required quasi-random sequence. Although this suggests that slightly more time is required for handling the quasi-random sequences than for the comparable generation of pseudo-random numbers, the increase is negligible when compared to the overall savings due to reduced required number of rays.

Although there is a significant reduction in time to convergence is achieved through the use of RQMC over MC, it is appreciably less than the orders of magnitude reduction in sample size based on the error trend analysis performed in Chapter 4. One reason for this is that, in general, the sample sizes required during optimization are generally smaller than those used to predict the order-of-magnitude levels of improvement (improvement becomes more significant as sample sizes increase). Furthermore, since the condition-based approach used only makes attempts at discrete sample sizes (in this case 256, 512, 1024, etc…) in reality it does not use the true minimum allowable sample size.

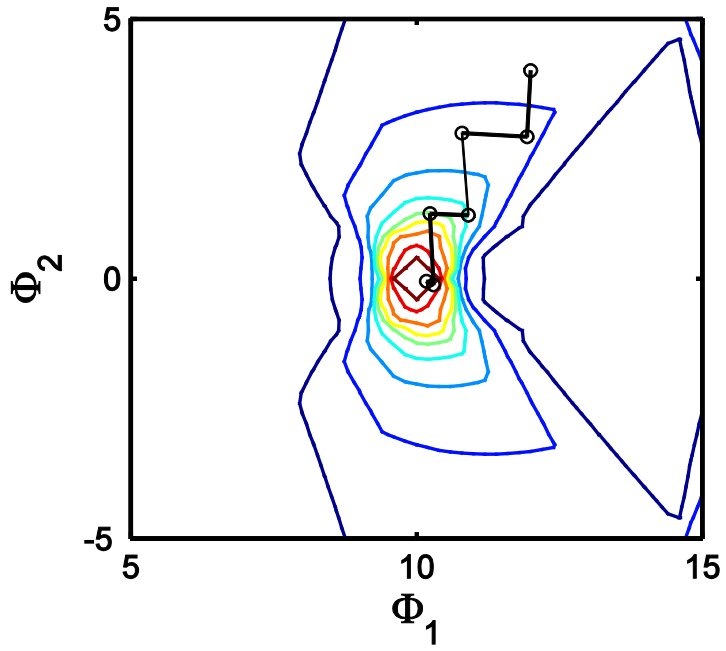


FIGURE 5.7 - DESIGN PATH FROM INITIAL DESIGN $\Phi_0 = [12, 4]^T$

TABLE 5.3 - OPTIMIZATION RESULTS FOR TEST CASE 1

Φ_0	method	average $F(\Phi^*)$	elapsed CPU time (s)	# design updates	# rays traced $\times 10^{-4}$	time per ray (s) $\times 10^{4}$
$\begin{bmatrix} 8 \\ -3 \end{bmatrix}$	MC	-1	8.4453	4.6	6.9171	1.2209
	RQMC	-1	4.3734	4.0	2.7878	1.5688
$\begin{bmatrix} 12 \\ 4 \end{bmatrix}$	MC	-1	50.233	4.9	37.775	1.3298
	RQMC	-1	9.4703	5.5	6.0698	1.5602
$\begin{bmatrix} 11 \\ -4 \end{bmatrix}$	MC	-1	23.059	4.8	17.692	1.3020
	RQMC	-1	6.0078	4.4	3.8890	1.5450
average	MC	-1	27.24577	4.8	20.795	1.2842
	RQMC	-1	6.617167	4.6	4.2489	1.5580

5.6.2 Test Case Two

Based on a number of optimization attempts for the design problem of test case two, certain modifications to the previously used settings are found beneficial. First of all, the design space for this problem is found to much less regular than for the previous case, containing a number of local optima (this is discussed in

greater detail below). In order to prevent rapid convergence on some of the less significant optima, then, the initial difference parameter is increased to $h_0 = 1$, causing the algorithm to begin the search with a more broad view of the objective function. Secondly, because, unlike in the previous case, this problem does not have a finite region of "perfect" performance, the gradient magnitude never reaches a value of exactly zero. In order to improve termination, then, the threshold for gradient magnitude is increased to 0.01, found to be a reasonable value from initial trials.

With these modifications, design optimization is performed a number of times from the initial designs given by $\mathbf{\Phi}_0 = \{[8, 3, 6]^T, [7, 3, 8]^T, [7, 2, 10]^T\}$, corresponding to the geometries in Figure 5.8 (a), (c), and (e), respectively. Of the trials performed, about half terminated based on a low gradient magnitude, while the rest terminated once significant improvement was no longer being made. Overall average results of these trials are provided in Table 5.4.

For this problem, it is found that the identified optimal design depends highly on the initial design used. Figure 5.8 (b), (d), and (f) show three possible optimal designs which result from initial design(s) as indicated by the arrows, when perfect optics are assumed. As can be seen, trials starting from both the first and second initial designs converge to a geometry that approximates an ideal parabolic concentrator, for which a performance of about $F(\mathbf{\Phi}^*) = -0.505$ is achieved. Although both instances consistently result in this approximate design, those starting from the second initial design tend to achieve slightly better performance. This is likely due to the close proximity of the second initial design to this optimum, allowing the majority of design effort to be dedicated to final "fine tuning". Trials starting from the third initial design, which as can be seen is quite different from the other two, tend to converge to one of two possible optimal designs: the first, which achieves a performance nearly identical to that of the expected optimum; and the second, which has a much poorer performance of about $F(\mathbf{\Phi}^*) = -0.334$. Although this second design exhibits inferior performance, it is identified as optimal because the three control points are

located in the "shadow" of the receiver, causing design updates to have little effect on overall performance and so resulting in a small gradient magnitude. Due to occasional convergence to this second local minima, the attempts starting from this design achieve an inferior average performance of about $F(\mathbf{\Phi}^*) = -0.4471$.

When accounting for "realistic" optical properties, there is no noticeable difference in the geometries of the optimal designs identified. As for the case of ideal optics, attempts starting from the first and second design converge to the expected parabola-like geometry, while those from the third initial design converge to one of the two nearby optima. As for the ideal case, average performance is significantly worse for attempts starting from the third initial design, as expected, while those starting from the second slightly outperform the first. Due to the effects of non-perfect reflection and absorption, the best design in this case exhibits a performance of about $F(\mathbf{\Phi}^*) = -0.44$, about 10% worse than that based on ideal optics. Such a reduction is reasonable, given a concentrating surface reflectivity of 0.92 and the receiver surface directional absorptivity as shown in Appendix C, averaging about 0.84.

As evidenced by the multiple optimal designs identified, the design space of this problem is not convex. As expected for a gradient-based scheme, then, the optimization algorithm tends to converge on the first significant optimum encountered, usually that which is closest to the initial design. Identifying the "correct" optimum, then, depends on its proximity to the chosen initial guess. In the case of this particular design problem, it is easy to start with such an initial guess, though for a more complex problem this may not be the case.

As for the previous design problem, there is no obvious difference in optimal performance or required number of design updates between trials based on MC and RQMC, indicating that the overall design path is not strongly influenced by the choice of evaluation method. The total CPU time and required number of rays, however, are significantly reduced through the use of RQMC when compared to traditional MC. For trials assuming perfect optics, CPU time

and total number of rays are decreased by factors of about 10.5 and 6.35, respectively. The discrepancy between these two factors is due to a significantly larger required CPU time per ray for MC than for RQMC, a discrepancy for which, at this point, no explanation is offered. For trials based on realistic optical properties, savings due to RQMC are less significant, with CPU time and total number of rays decreased by factors of about 2.55 and 2.67, respectively. Such a difference between degrees of improvement for ideal and realistic cases makes sense, since, as found in Chapter 4, RQMC offers a much more substantial improvement in convergence rate for cases assuming perfect optics.

TABLE 5.4 - OPTIMIZATION RESULTS FOR TEST CASE 2

Φ_0	method	average $F(\Phi^*)$	elapsed CPU time (s)	# design updates	# rays traced $\times 10^{-5}$	time per ray (s) $\times 10^{5}$
Perfect Optics						
$\begin{bmatrix}8\\3\\6\end{bmatrix}$	MC	-0.4973	131.58	21.7	8.5053	15.471
	RQMC	-0.4936	17.582	24.4	1.9448	9.0408
$\begin{bmatrix}7\\3\\8\end{bmatrix}$	MC	-0.5024	267.30	20	17.828	14.993
	RQMC	-0.5002	15.942	18.5	1.7871	8.9205
$\begin{bmatrix}7\\2\\10\end{bmatrix}$	MC	-0.4379	86.189	11.4	4.8233	17.869
	RQMC	-0.4562	12.720	16.4	1.1745	10.830
average	MC	-0.4792	161.69	17.7	10.386	16.111
	RQMC	-0.4833	15.415	19.8	1.6355	9.5971
Realistic Optics						
$\begin{bmatrix}8\\3\\6\end{bmatrix}$	MC	-0.4012	66.322	15.2	7.5274	8.8107
	RQMC	-0.4309	38.938	21.6	4.2246	9.2176
$\begin{bmatrix}7\\3\\8\end{bmatrix}$	MC	-0.4381	94.780	21.5	11.007	8.6105
	RQMC	-0.4367	36.847	19	4.0748	9.0427
$\begin{bmatrix}7\\2\\10\end{bmatrix}$	MC	-0.3888	60.367	14.9	6.3091	9.5682
	RQMC	-0.3396	11.211	10.9	1.0184	11.009
average	MC	-0.4094	73.823	17.2	8.2812	8.9965
	RQMC	-0.4024	28.999	17.2	3.1060	9.7564

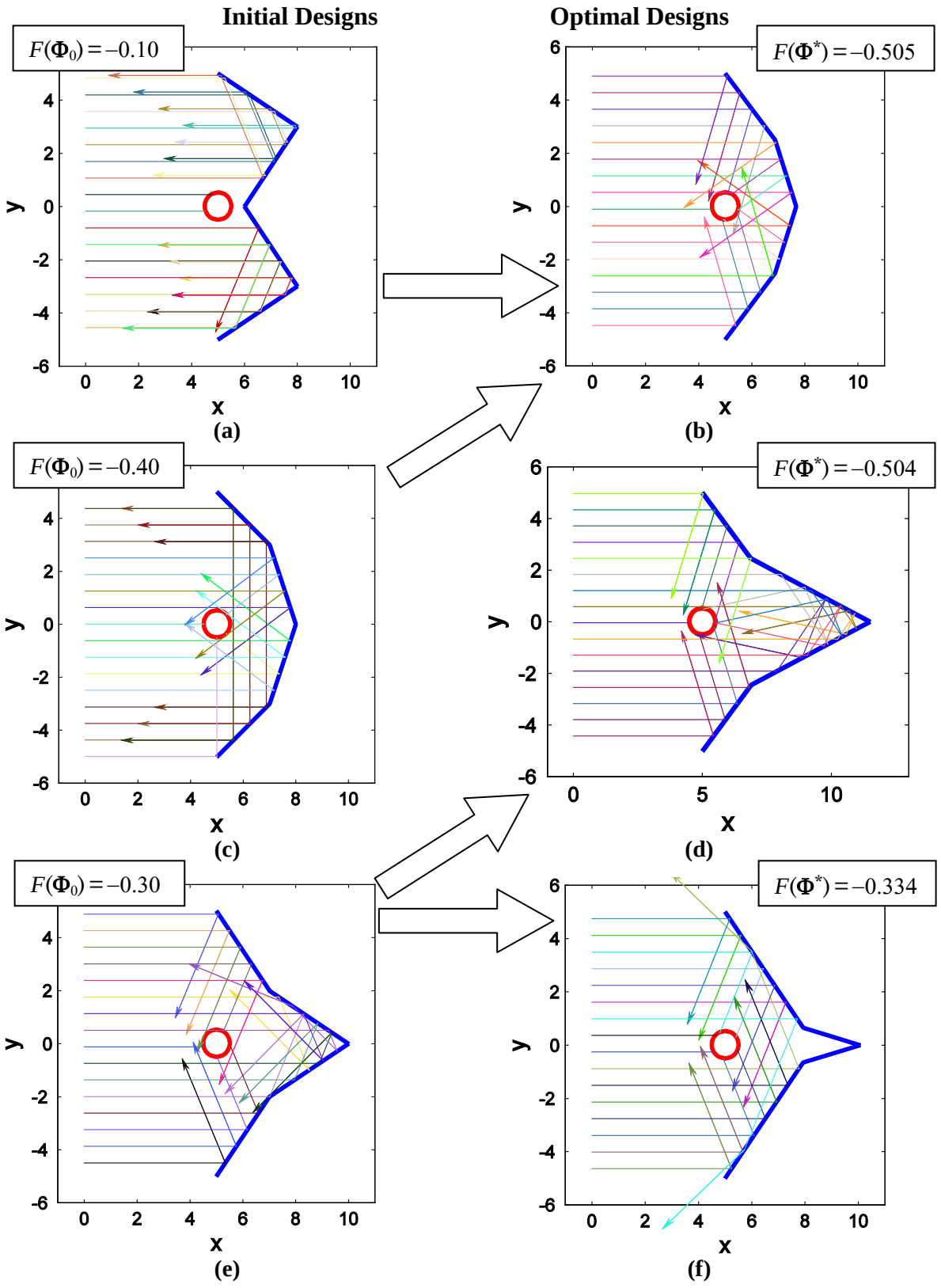


FIGURE 5.8 - TEST CASE 2: (a), (c), (e) INITIAL AND; (b), (d), (f) RESULTING OPTIMAL DESIGNS (ARROWS INDICATE TYPICAL RESULT)

5.6.3 TEST CASE THREE

For the design problem of test case three, a series of optimization attempts are performed from an initial design of $\mathbf{\Phi}_0 = [-5, -5, 0, -7, 5, -5]^T$, corresponding to the semi-circular geometry as shown in Figure 5.9 (a). Upon initial testing, it is found that the design performance progresses quite slowly, requiring a relatively large number of iterations in order arrive at an optimal performance. In order to prevent premature termination, then, the percentage improvement required to be considered significant is decreased from 0.5% to 0.1%. All other settings as used for test case two are found to work well, namely $K = 5$, initial sample size $N = 256$, $h_0 = 1$, and a gradient magnitude termination threshold of 0.01. Average results of the performed trials are provided in Table 5.5.

All of the resulting optimization attempts follow a design path very similar to that as described by Figure 5.9 (a) through (d), gradually approaching a design similar to the asymmetrical CPC of Adsten et al. (2005). As can be seen from Figure 5.9 (e) and (f), significant design improvement is achieved in the first ten or so iterations, over which a number of larger design steps of over two or three units are performed. After this point, however, a large number of very small design updates occur, resulting in a very gradual improvement in performance. Some of these small steps are taken in the negative descent direction, an occasional consequence of applying a quadratic line fit to the identified three-point pattern when the optimal step size is very close to zero. After many such small updates, termination eventually occurs due to a lack of significant improvement.

The optimal performance achieved is found to be highly dependent on the number of updates that are performed before termination, in the ideal case ranging around $F(\mathbf{\Phi}^*) = -0.75$ for trials that last 15 iterations, and up to $F(\mathbf{\Phi}^*) = -0.88$ for those lasting into the thirties and forties. On average, about 32 design updates are performed before such termination, resulting in a performance of about $F(\mathbf{\Phi}^*) = -0.7885$ for cases assuming perfect optics and

$F(\boldsymbol{\Phi}^*) = -0.6870$ for those based on realistic optical properties. There is no noticeable difference in geometry between optimal designs for either case.

As with the previous test cases, the design path and resulting optimal performance are not affected by the particular method of estimation used. Also as before, results show that optimization based on RQMC requires significantly less computation than for those based on traditional MC, with a greater degree of improvement seen for cases assuming perfect optics than for those based on realistic optical properties. For trials assuming perfect optics, CPU time and total number of rays are decreased by factors of about 4.4 and 4.6, respectively, and for trials based on realistic optics, quantities are reduced by factors of about 2.1 and 2.2.

TABLE 5.5 - OPTIMIZATION RESULTS FOR TEST CASE 3

$\boldsymbol{\Phi}_0$	method	average $F(\boldsymbol{\Phi}^*)$	elapsed CPU time (s)	# design updates	# rays traced $\times 10^{-5}$	time per ray (s) $\times 10^{4}$
Perfect Optics						
$[-4,-5,0,-7,4,-5]^T$	MC	-0.7939	694.11	27.3	8.4245	8.2393
	RQMC	-0.7830	157.66	34.9	1.8437	8.5513
Realistic Optical Properties						
$[-4,-5,0,-7,4,-5]^T$	MC	-0.6873	1408.3	33.6	18.236	7.7224
	RQMC	-0.6867	662.92	32.6	8.1293	8.1547

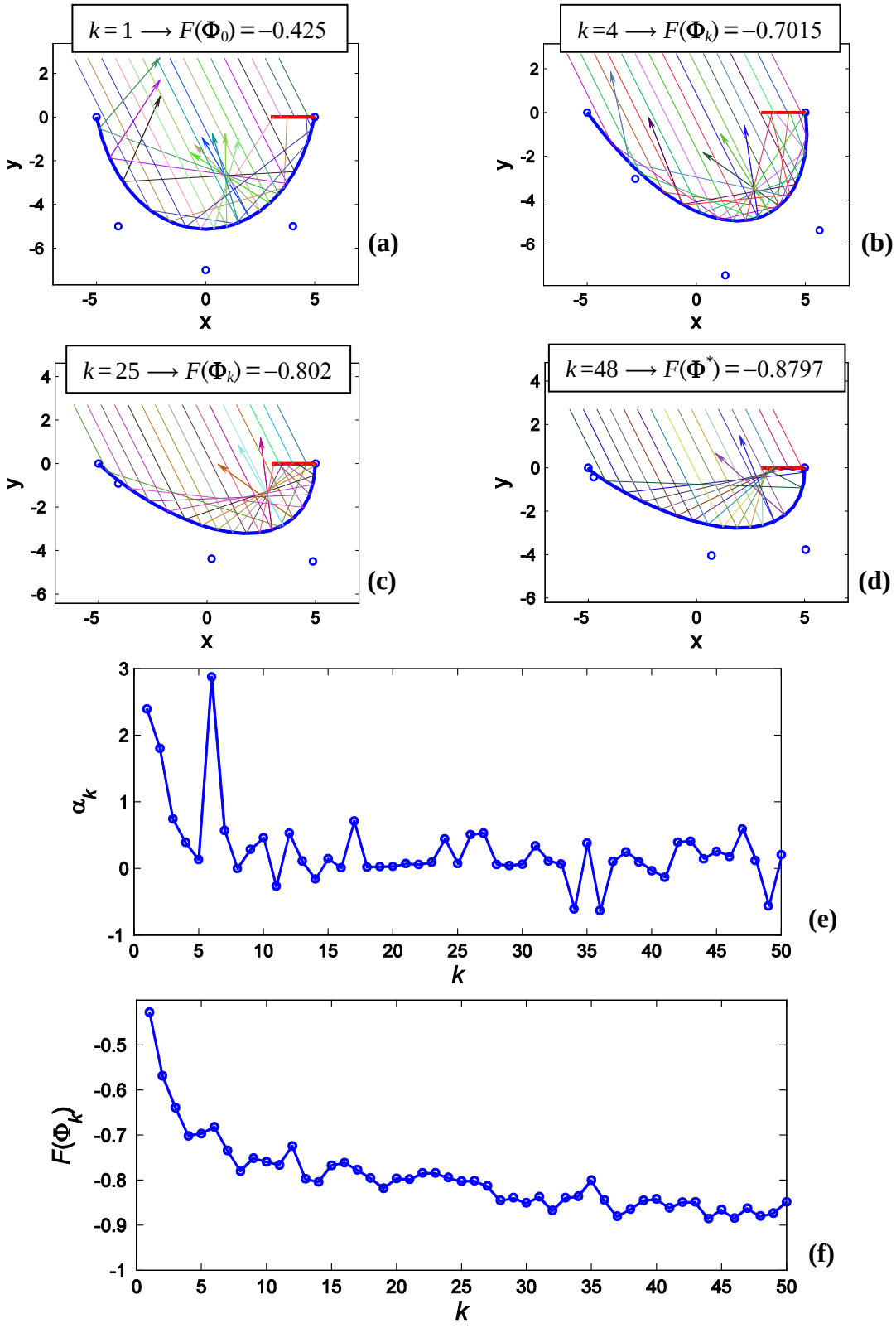


FIGURE 5.9 - EXAMPLE OPTIMIZATION PROGRESS FOR TEST CASE 3

Based only on the above set of optimization trials, the overall convexity of the problem cannot be known. Due to the large number of very small design steps that occur before eventual termination, it is suspected that some sort of "valley" is present in the objective function contour, which greatly impedes design progress. In an attempt to determine whether or not other optimal designs are present in the design space, a number of additional trials are performed from various different initial designs. In most cases, a design similar to the above is identified. In one attempt, however, starting from initial design $\mathbf{\Phi}_0 = [-7, -5, 0, -5, 7, -5]^T$, a "true" optimal design is identified, with an associated performance of $F(\mathbf{\Phi}^*) = -1$. Figure 5.10 shows the initial and optimal design for this case. Because this design was not identified by trials from the previous initial design, it is likely that this design problem has multiple regions of local optima.

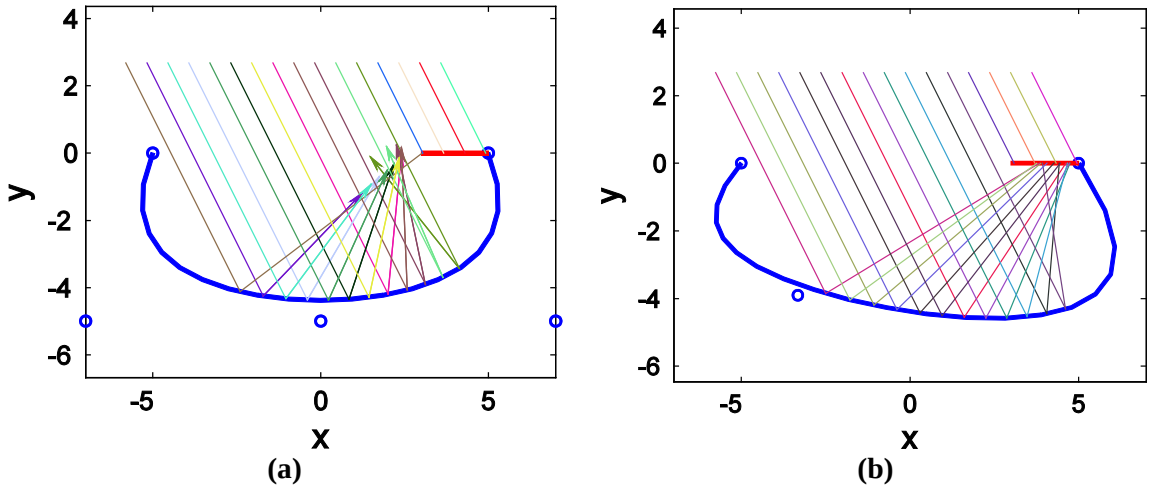


FIGURE 5.10 - ADDITIONAL (a) INITIAL AND (b) OPTIMAL DESIGN FOR TEST CASE 3

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

The developed solar concentrating collector geometric optimization methodology, when applied to the three introduced test problems, performs with mixed results. In each case, an optimal design is successfully identified, though as additional difficulties are introduced, performance becomes less reliable. For the simple problem of test case one, for which the objective function is strictly convex, the algorithm performs very well, consistently reaching the known ideal solution. For the problem of test case two, for which multiple optimal regions are present, performance is found to be dependent upon the quality of the initial guess, although the majority of optimal designs perform quite well. For the problem of test case three, the majority of trials were unable to identify a "true" optimal design, although they did consistently result in a design with a performance far superior to that of its respective initial design. In each of the cases, such a dependence of optimization performance on initial design highlights the importance of performing optimization from a variety of initial designs, either manually or through some kind of "re-start" algorithm. For the problems examined, accounting for realistic optical properties versus perfect optics does not lead to a noticeable change in the geometry of identified optimal designs, though it does, of course, reduce overall performance.

The use of randomized quasi-Monte Carlo evaluation can offer substantial reduction in the sample size required to achieve a given degree of estimation uncertainty, when compared to traditional Monte Carlo. For objective function evaluations of cases which assume perfect optics, RQMC exhibits convergence rates of near unity, significantly higher than the rate of 1/2 shown by MC. This leads to typical reductions of two or three orders of magnitude in required

sample size, with the degree of improvement increasing as uncertainty decreases. For evaluations which simulate realistic optical properties, RQMC exhibits much less significant, albeit non-trivial, improvement, leading to typical sample size reductions of a factor of three or four. Due to these reductions in required sample size, overall computational effort of optimization is significantly reduced through the use of RQMC over MC. For design problems assuming perfect optics, computational time and required number of rays are typically decreased by factors in the range of four to six, while for problems based on realistic optical properties factors of around 2.5 are more likely.

6.2 RECOMMENDATIONS

In light of the conclusions gained through this research, there are a number of aspects which warrant future work.

Because, for the more complex of the problems examined, the objective function is generally not found to be well-behaved and strictly convex, it may be worthwhile investigating the application of optimization techniques more sophisticated than the steepest-descent scheme used. While it was previously deemed impractical to apply higher-order gradient-based schemes to stochastic systems such as this, uncertainty reduction through the use of RQMC estimation may change this. Alternatively, since RQMC can greatly reduce the sample sizes required for objective function estimation, the application of global optimization schemes may be more attractive, which are thought to perform better in such situations.

In order to further demonstrate the utility of such a geometric optimization methodology, its application to problems of increased complexity would prove valuable. Identification of an optimal non-tracking concentrating collector, for example, could be performed through the development of a cumulative distribution function to account for realistic seasonal or yearly meteorological data of solar intensity with respect to azimuth and zenith angle. In addition, the objective function could be modified to take into account the

uniformity of radiation over the receiver surface as well as the overall fraction of collected radiation, in order to achieve optimal performance for application in concentrating photovoltaics. In CPV, uniformity is important, as variation in radiative intensity leads to performance degradation (Duffie and Beckman 2006).

Because, when performing quasi-Monte Carlo, stochastic sampling is still used to determine the outcome of non-ideal surface interactions, the degree of reduction in required sample size is less significant for problems based on realistic optical properties. As such, it would be worthwhile to perform additional research into ways to utilize quasi-random sequences throughout Monte Carlo ray-tracing. From recent discussion with Professor Lemieux (2010), it has been found that this is indeed possible for some systems. The benefit to this particular application remains to be determined, however, since in cases involving highly-specular surfaces most reflection events are deterministic. Problems involving a high-degree of diffuse reflection or a scattering medium would likely benefit greatly from this technique, however, due to large amount of random sampling needed to predict resulting ray directions.

REFERENCES

Adsten, M., Helgesson, A., Karlsson, B., 2005, "Evaluation of CPC-collector designs for stand-alone, roof- or wall installation," Solar Energy **79**, pp. 638-647.

American Lung Association, *State of the Air*, 2009. Accessed on March 29, 2010 at <http://www.lungusa.org/assets/documents/publications/state-of-the-air/state-of-the-air-report-2009.pdf>.

Ashdown, I., 1994, "Non-Imaging Optics Design Using Genetic Algorithms," Journal of the Illuminating Engineering Society, **23**, pp. 12-21.

Baranov, V.K., 1965, "Properties of parabolic focons," Opt. Mekh. Prom., **6**, pp. 1-6.

Belegundu, A. D., Chandrupatla, T. R., 1999, *Optimization Concepts and Applications in Engineering, 2nd Ed.*, Pearson Education, Singapore, pp. 28-29.

Bertsekas, D. P., 1999, *Nonlinear Programming, 2nd Ed.*, Athena Scientific, Belmont MA, pp. 723-728.

Cieslak, M., Lemieux, C., Hanan, J., Prusinkiewicz, P., 2008, "Quasi-Monte Carlo simulation of the light environment of plants," Functional Plant Biology **35** (10), pp. 837-849.

Corcione, M., Fontana, L., 2003, "Optimal design of outdoor lighting systems by genetic algorithms," Lighting Res. Technol. **35** (3), pp. 261-280.

Cranley, R., Patterson, T. N. L., 1976, "Randomization of number theoretic methods for multiple integration," SIAM J. Numer. Anal. **13**, pp. 904–914.

Daun, K. J., Hollands, K. G. T., 2001, "Infinitesimal-area radiative analysis using parametric surfaces, through NURBS," J. Heat Transfer **123**, pp. 249-256.

Daun, K. J., Howell, J. R., Morton, D. P., 2003a, "Geometric optimization of radiative enclosures through nonlinear programming," Numerical Heat Transfer, Part B **43,** pp. 203-219.

Daun, K. J., Howell, J. R., Morton, D. P., 2003b, "Geometric optimization of radiant enclosures containing specular surfaces," J. Heat Transfer **125**, pp. 845-851.

Desertec-UK, 2009, "CSP Pictures," Accessed April 12, 2010 at <http://www.trec-uk.org.uk/resources/pictures/stills1.html>.

Doyle, S., Corcoran, D., Connell, J., 1999, "Automated mirror design using an evolution strategy," Opt Eng. **38** (2), pp. 323-333.

Duffie, J. A., Beckman, W. A., 2006, *Solar Engineering of Thermal Processes, third ed.*, John Wiley & Sons, Inc., Hoboken, New Jersey.

Eckhardt, R., 1987, "Stan Ulam, John von Neumann and the Monte Carlo method," Los Alamos Science, Special Issue, pp. 131–143.

Evergreen Energy Solar, "CPC 6 Lite Evacuated Tube Collectors," (Product Webpage), accessed on April 8, 2010 at <http://www.evergreenenergy.ie/cpc6.htm>.

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Gordon, J. M., 1996, "Simple string construction method for tailored edge-ray concentrators in maximum-flux solar energy collectors," Solar Energy **56** (3), pp. 279-284.

Haeberle, A., Berger, M., Luginsland, F., Zahler, C., Baitsch, M., Henning, H.-M., Rommel, M., 2007, "Linear Concentrating Fresnel Collector for Process Heat Applications," Proc. Local Renewables, Freiburg, Germany, June 13-15.

Hammersley, J. M., Handscomb, D. C., 1975. *Monte Carlo Methods*, Fletcher & Son Ltd, Norwich, Great Britain, accessed 24 April, 2010 at <http://www.cs.fsu.edu/~mascagni/Hammersley-Handscomb.pdf>.

Hickernell, F. J., 1998, "A generalized discrepancy and quadrature error bound," Mathematics of Computation **67**, pp. 299–322.

Hinterberger, H., and Winston, R., 1966, "Efficient light coupler for threshold Cerenkov counters," Rev. Sci. Instrum. **37**, pp. 1094–1095.

Holland, J. H., 1992, *Adaptation in Natural and Artificial Systems*, 2nd Ed., MIT Press, Cambridge, MA.

Intergovernmental Panel on Climate Change (IPCC), *Fourth Assessment Report: Climate Change 2007*, 2008. Accessed March 25, 2010 at <http://www.ipcc.ch/publications_and_data/publications_and_data_reports.htm#1>.

International Energy Agency (IEA), 2009, *Key World Energy Statistics 2009*, accessed on March 25, 2010 at <http://www.iea.org/publications/free_new_Desc.asp?PUBS_ID=1199>.

Keller, A., 1996, "Quasi-Monte Carlo methods in computer graphics: the global illumination problem," Lectures in Applied Mathematics **32**, pp. 455-469.

Kersch, A., Morokoff, W., Schuster, A., 1994, "Radiative Heat Transfer with Quasi-Monte Carlo Methods," Transp. Theory Stat. Phys. **23**, pp. 1001-1021.

Kiefer, J., Wolfowitz, J. R., 1952, "Stochastic Estimation of the Maximum of a Regression Function," Ann. Math. Stat., **23**, pp. 462-466.

Kline, S. J., and McClintock, F. A., 1953, "Describing Uncertainties in Single-Sample Experiments", Mechanical Engineering: Journal of the American Society of Mechanical Engineers, **75**, pp. 3-12.

L'Ecuyer, P., and Lemieux, C., 2002, "Recent advances in randomized quasi-Monte Carlo methods," *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, ed. M. Dror, P. L'Ecuyer and F. Szidarovszki, Boston: Kluwer Academic Publishers, pp. 419–474.

Law, A. M., and Kelton, W. D., 2000, *Simulation Modeling and Analysis, 3rd Ed.*, Chapter 12.6.2, McGraw-Hill.

Lemieux, C., 2009, *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer, New York, NY.

Lemieux, C., 2010, Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo, *Personal Communication*, September 21.

Loomans, M., Visser, H., 2002, "Application of the genetic algorithm for optimization of large solar hot water systems," Solar Energy **72,** pp. 427-439.

Lovegrove, K., Luzzi, A., 2001, "Solar Thermal Power Systems", *Encyclopedia of Physical Science and Technology, 3rd Edition*, *Vol. 15*.

Mallick, T. K., Eames, P. C., Hyde, T. J., Norton, B., 2004, "The design and experimental characterization of an asymmetric compound parabolic photovoltaic concentrator for building façade integration in the UK," Solar Energy **77**, pp. 319-327.

Morokoff, W. J. and Caflisch, R. E., 1994, "Quasi-Random sequences and their discrepancies," SIAM J. Sci. Comput., **15**, pp. 1251-1279.

Morokoff, W. J., Caflisch, R. E., 1995, "Quasi-Monte Carlo Integration," J. of Comp. Phys., **122**, pp. 218-230.

Muschaweck, J., Spirkl, W., Timinger, A., Benz, N., Dörfler, M., Gut, M., Kose, E., 2000. "Optimized Reflectors for Non-tracking Solar Collectors with Tubular Absorbers," Solar Energy **68**, pp. 151-159.

North American Space Association (NASA), 2005, "The Earth Radiation Budget Experiment (ERBE)," accessed on March 29, 2010 at <http://asd-www.larc.nasa.gov/erbe/ASDerbe.html>.

O'Brien, D. M., 1992, "Accelerated Quasi Monte Carlo Integration of the Radiative Heat Transfer Equation," Journal of Quantitative Spectroscopy and Radiative Transfer **48** (1), pp. 41-59.

Patow G., Pueyo, X., Vinacua, A., 2004, "Reflector design from radiance distributions," Int. J. Shape Modeling **10** (2), pp. 211-235.

Pflug, G. C., 1996, *Optimization of Stochastic Models: The Interface between Simulation and Optimization*, Kluwer Academic Publishers, Boston MA.

Piegl, L., and Tiller, W., 1997, *The NURBS Book*, 2nd ed., Springer-Verlag, Berlin.

Press, W. H., et al., 2007, "Quasi-Random Sequences," *Numerical Recipes*, 3rd Ed., New York, NY: Cambridge University Press.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992, *Numerical Recipes in C, The Art of Scientific Computing, Second Edition*, Cambridge University Press.

Rabl, A., 1976a, "Comparison of solar concentrators," Solar Energy **18**, pp. 93-111.

Rabl, A., 1976b, "Solar concentrators with maximal concentration for cylindrical absorbers," Applied Optics **15** (7), pp. 1871-1873.

Rabl, A., 1994, "Edge-ray method for analysis of radiation transfer among specular reflectors," Applied optics **33** (7), pp. 1248-1259.

Ries, H., Rabl, A., 1994, "Edge-ray principle of nonimaging optics," J. Opt. Soc. Am. A **11** (10), pp. 2627-2632.

Rukolaine, S. A., 2010, "The shape gradient of the least-squares objective functional in optimal shape design problems of radiative heat transfer," Journal of Quantitative Spectroscopy and Radiative Transfer **111**, pp. 2390-2404.

Ryan, J. P., Miller, J. A., Burns, P. J., 1998, "Development and use of a Monte Carlo code to conduct a parametric study of cylindrical solar collector arrays," Proceedings of the 1998 International Solar Energy Conference, Albuquerque, NM, June 14-17.

Sarvari, S. M. H., 2007, "Optimal geometry design of radiative enclosures using the genetic algorithm," Numerical Heat Transfer, Part A **52**, pp. 127-143.

Siegel, R., Howell, J. R., 2002, *Thermal Radiation Heat Transfer*, 4th Ed., Taylor & Francis, New York, NY.

Simha, R. 2003, "An Algorithm for Gradient-Free Simulation Optimization using Sampling Control," Int J Modeling Simulation **23**, pp. 197-204.

T. Tesfamichael, and E. Wäckelgård, 1999, "Angular Solar Absorptance of Absorbers used in Solar Thermal Collectors", Applied Optics **38**, pp. 4189-4197.

Tabor, H., 1958, "Stationary mirror systems for solar concentrators," Solar Energy **2**, pp. 27-33.

Toth, D. L., 1985, "On Ray Tracing Parametric Surfaces," SIGGRAPH '85 Conference Proceedings 19, pp. 171-179.

Veach, E., 1997, "Robust Monte Carlo Methods for Light Transport Simulation," PhD thesis, Stanford University.

Weinstock, D., Appelbaum, J., 2007, "Optimization of economic solar field design of stationary thermal collectors," Solar Energy **129**, pp. 363-370.

Winston, R., 1974, "Principles of solar concentrators of a novel design," Solar Energy **16**, pp. 89-95.

Winston, R., Hinterberger, H., 1975, "Principles of cylindrical concentrators for solar energy," Solar Energy **17**, pp. 255-258.

Winston, R., Miñano, J. C., Benítez, P., 2005, *Nonimaging Optics*, Elsevier Academic Press, Burlington, MA, USA.

Winston, R., Ries, H., 1993, "Nonimaging reflectors as functional of the desired irradiance," J. Opt. Soc. Am. A **10** (9), pp. 1902-1908.

Winston, R., Welford, W. T., 1979a, "Geometrical vector flux and some new nonimaging concentrators," J. Opt. Soc. Am. **69** (4), pp. 532-536.

Winston, R., Welford, W. T., 1979b, "Ideal flux concentrators as shapes that do not disturb the geometrical vector flux field: A new derivation of the compound parabolic concentrator," J. Opt. Soc. Am. **69** (4), pp. 536-539.

World Energy Council (WEC), 2007, "Survey of Energy Resources 2007," accessed on March 29, 2010 at <http://www.worldenergy.org/publications/survey_of_energy_resources_2007/default.asp>.

APPENDIX A

TYPES OF CONCENTRATING COLLECTORS

A.1 PARABOLIC CONCENTRATORS

A 3D (or 2D) concentrator of perfect parabolic geometry is capable of focusing collimated incoming radiation onto to a single point (or line). Because incident solar beam radiation actually forms a cone of angular width 0.53°, however, a parabolic concentrator can at best focus the entire solar image onto a receiver of finite area. For this reason, ideal 3D and 2D parabolic concentrators with planar absorbers, for example, can achieve maximum concentration ratios of only $1/(4\sin^2\theta)$ and $1/(2\sin\theta)$, respectively; factors of four and two less than the previously discussed theoretical maximums (Winston et al. 2005).

Parabolic trough (2D) concentrators have been built with typical concentration ratios of 8 – 80, able to achieve temperatures of 260 – 400 °C (Lovegrove and Luzzi 2001). Large commercial scale solar thermal power plants utilize parabolic troughs with evacuated tube absorbers, such as the one shown in Figure A.1, and collect heat for conventional steam turbine electricity generation. Located in California's Mojave Desert, Solar Energy Generating Systems (SEGS) parabolic trough plant is the largest solar power generating facility in the world with a total electrical generating capacity of 354 MW (Lovegrove and Luzzi 2001). Such systems are well-suited to produce heat for industrial processes or to supplement coal or natural gas in conventional power plants. Parabolic trough concentrators have also been built for concentrating photovoltaics, generally focusing on a rectangular array of cells and requiring some form of active cooling, often lending themselves well to combined PV/T collection. Such linear concentrators require single-axis tracking, and can be oriented either along an east-west axis, adjusting their solar altitude as needed

throughout the day, or along a north-south axis, tracking the sun across the sky daily.

Parabolic dish (3D) concentrators have been built with typical concentration ratios of 800 – 8000, able to reach temperatures of 500 –1200 °C (Lovegrove and Luzzi 2001). CPV systems using these concentrators have been built, such as the one shown in Figure A.2, usually making use of high-performance multijunction solar cells to take advantage of the high solar flux. At these high concentration ratios, high-capacity heat sinks are required to prevent thermal destruction and maintain performance. Parabolic dish solar thermal power plants have also been built, generating electricity via a heat engine, often a Stirling engine, located at the focal point, or by transporting collected heat to a larger central heat engine, often a Rankine engine. In general, parabolic dish concentrators require accurate two-axis solar tracking.

FIGURE A.1 - PARABOLIC TROUGH COLLECTOR (DESERTEC-UK 2009)

FIGURE A.2 - PARABOLIC DISH CPV COLLECTOR (DESERTEC-UK 2009)

A.2 FRESNEL-TYPE CONCENTRATORS

One serious drawback of parabolic concentrators is that it is generally quite difficult and thus expensive to manufacture large curved reflecting surfaces and their required supporting structure. One way to help overcome these difficulties is to split the reflector into a series of smaller sections that are arranged to approximate a parabola. Such Fresnel-type collectors have been demonstrated in both 2D and 3D, and can generally achieve performance that approaches that of their parabolic counterparts, but for a fraction of the cost. In these collectors, each reflecting surface must have its own solar tracking mechanism in order to maintain concentration onto a fixed central receiver.

Linear Fresnel collectors, such as that shown in Figure A.3, use a series of long and flat parallel rectangular strips to concentrate sunlight onto a linear receiver. Simple flat reflecting surfaces further reduce cost of manufacture. Furthermore, each reflecting surface can be mounted close to the ground and beside its neighbors, which results in lower wind loads and thus lighter and cheaper supporting structure. Another advantage of linear Fresnel systems is that because each mirror is adjusted independently, there is more control over

radiative flux distribution, which aids greatly in high concentration PV applications.

A common form of 3D Fresnel concentrators, large-scale central receiver, or "power tower", plants, such as that shown in Figure A.4, are capable of collecting thermal energy in the megawatt range, generally with concentration ratios of 600 – 1000 and at temperatures of 500 – 800 °C (Lovegrove and Luzzi 2001). Such systems use a large field of fully-tracking flat or slightly-curved mirrors known as heliostats to concentrate beam radiation onto a receiver at the top of a central tower. Concentrated radiation is either used to directly heat a flow of steam or gas, or to heat a molten salt or metal to a higher temperature to allow some degree of energy storage, both with the end goal of electricity generation via a conventional heat engine.

FIGURE A.3 - LINEAR FRESNEL COLLECTOR (DESERTEC-UK 2009)

FIGURE A.4 - CALIFORNIA'S SOLAR TWO (DESERTEC-UK 2009)

A.3 NON-IMAGING CONCENTRATORS

As previously discussed, parabolic concentrators can create a perfect image of the sun on the receiver, but are not capable of achieving the maximum theoretical concentration limit. For the purpose of solar energy collection, the goal is to simply redirect incoming radiation onto a given receiver, and a perfect image is thus not necessary. In fact, by removing this requirement, concentrators can be designed which can actually attain, or closely approach, the sine law limit to concentration. So-called non-imaging concentrators are designed with the goal of redirecting to the receiver all solar radiation entering the aperture within some range of incidence angle. Custom designs can thus be created to meet specific needs, such as a wide acceptance angle to reduce the requirement for tracking, or a particular geometry to achieve a desired radiative flux distribution over the absorber surface. In general, collectors with smaller acceptance angles have a greater concentration ratio, but require more frequent and precise solar tracking.

Two- or three-dimensional compound parabolic concentrators (CPCs) are a common form of non-imaging reflecting surface, with cross-sectional geometry

made up of two parabolas with axes of symmetry tilted toward one another. Linear CPCs are often used in residential solar hot water collectors, for example, reflecting both incoming beam and diffuse radiation within a wide acceptance angle onto a cylindrical receiver, as shown in Figure . Numerous other truncated or asymmetrical non-imaging concentrator geometries exist, some of which are modifications of the traditional CPC while others are entirely unique. Details on how these geometries can be designed are the focus of this thesis.

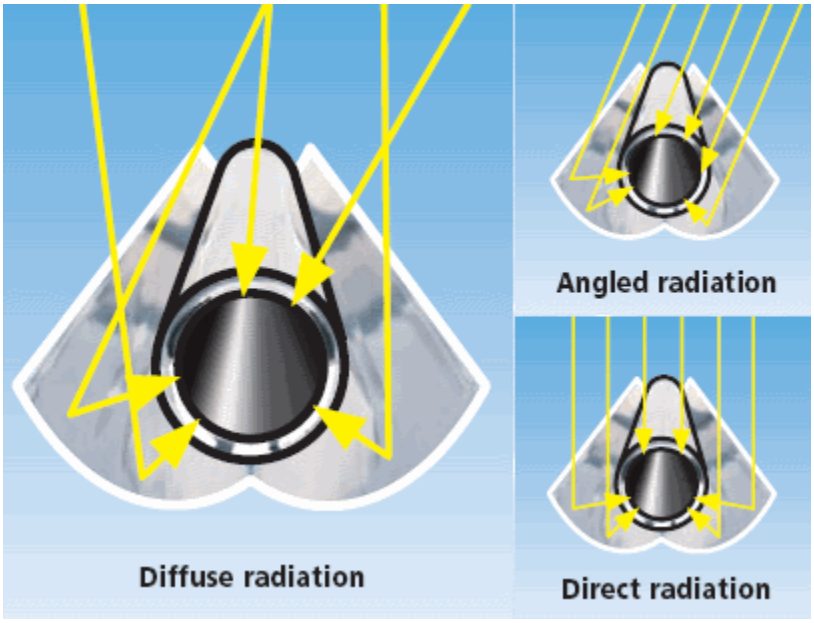


FIGURE A.5 - CPC WITH CYLINDRICAL ABSORBER (EVERGREEN ENERGY SOLAR 2010)

Appendix B

THREE-POINT PATTERN SEARCH ALGORITHM

The algorithm employed in this thesis for identifying a three-point pattern is as provided in pages 28-29 of Belegundu and Chandrupatla (1999), and is as follows: from a choice of initial step size, x1, initial objective function, g1, initial search step, Δ, and expansion parameter, $\gamma >= 1$, perform the following:

1. Set x2 = x1 + Δ
2. Evaluate g1 and g2
3. If g2 <= g1, go to step 5
4. Interchange g1 and g2, x1 and x2, and set Δ = - Δ
5. Set $\Delta = \gamma\Delta$, x3 = x2 + Δ, and evaluate f3 at x3
6. If g3 > g2, go to step 8
7. Rename g2 as g1, g3 as g2, x2 as x1, x3 as x2; then go to step 5
8. Points 1, 2, and 3 satisfy g1 >= g2 < g3 (three-point pattern)

Throughout this thesis, each search is begun at the current design, corresponding to x1 = 0.

Appendix C

DIRECTIONAL ABSORPTIVITY OF RECEIVER SURFACE

For the test cases performed in this thesis assuming "realistic" optical properties, the receiver surface is given the directional absorptivity as measured by Tesfamichael and Wäckelgård (1999) for a sputtered Ni-NiO_x coating (shown in Figure C.1). This data is based on a normalized absorptance of $\alpha_n = 0.96$. In order to apply the above curve in the ray-tracing code, a curve-fit is performed using the online tool found at www.zunzun.com, based on the eight experimental data points sampled from the curve, and resulting in the relation

$$\frac{\alpha}{\alpha_n} = a(c\theta_i + d)^b + e$$

$$a = -1.9184 \times 10^4$$

$$b = -6.6314$$

$$c = -4.1624$$

$$d = 1.0960 \times 10^1$$

$$e = 1.0032$$

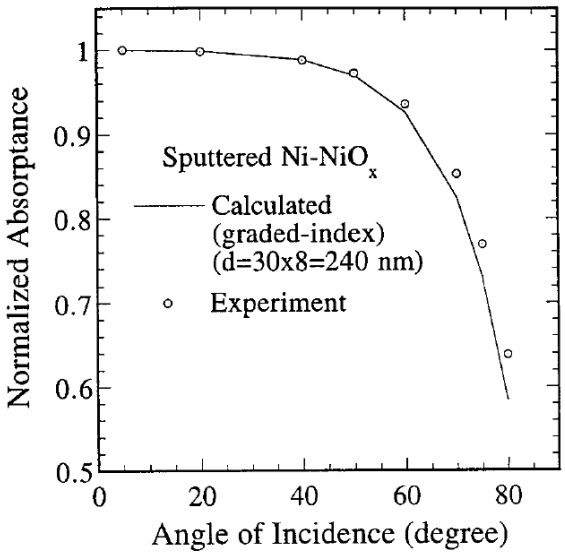


FIGURE C.1 - NORMALIZED ANGULAR ABSORPTANCE OF SPUTTERED NI-NIO_X COATING (TESFAMICHAEL AND WÄCKELGÅRD 1999)

Appendix D

Refining Estimations by Increasing Sample Size

When the sample size control condition calls for the total sample size to be increased from N to $2N$, instead of throwing out the initial result and performing the estimation again from scratch, it is better use of computational effort to update the estimation by combining the result of the initial N rays with that of an additional N rays. In doing this, however, care needs to be taken to ensure that the validity of objective function evaluations and corresponding uncertainty estimations is not compromised.

The strategy used throughout this thesis relies on the fact that the mean of a set of numbers can be calculated as the mean of two sub-means, as long as both sub-means are based on the sets of equal size. The mean value, μ, of a set of numbers, $\{x_1, x_2, \ldots, x_{n-1}, x_n\}$, for example, can be calculated as the mean of the sub-means μ_1 and μ_2, as follows

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \mu_1 = \frac{1}{n/2}\sum_{i=1}^{n/2} x_i, \quad \mu_2 = \frac{1}{n/2}\sum_{i=(n/2)+1}^{n} x_i$$

$$\mu = \frac{1}{n}\left(\sum_{i=1}^{n/2} x_i + \sum_{i=(n/2)+1}^{n} x_i\right) = \frac{1}{n}\left(\frac{n}{2}\mu_1 + \frac{n}{2}\mu_2\right)$$

$$\mu = \frac{\mu_1 + \mu_2}{2}$$

Note that this is valid only for sets that can be divided into two equal-sized sub-sets (i.e. the set contain an even number of entries).

Because an objective function evaluation through Monte Carlo (or quasi-Monte Carlo) as performed throughout this thesis is simply the mean result of the outcome $f(x_i)$ based on N rays, refining a previous result by combining with another the outcome of another N rays is straightforward. A refined estimation

$F_{combined}$, based on a sample size of $2N$, then, is simply the mean of the two estimations based on a sample size of N, F_1 and F_2 as follows

$$F_1 = \frac{\sum_{j=1}^{N} f(x_j)}{N}, \qquad F_2 = \frac{\sum_{j=N+1}^{2N} f(x_j)}{N}$$

$$F_{combined} = \frac{\sum_{j=1}^{2N} f(x_j)}{2N} = \frac{\sum_{j=1}^{N} f(x_j)}{N} + \frac{\sum_{j=N+1}^{2N} f(x_j)}{N} = \frac{F_1 + F_2}{2}$$

An updated uncertainty estimation is similarly achieved. Recall that Monte Carlo uncertainty is estimated as follows, based on the standard deviation of M independent sub-trials of length N/M

$$\varepsilon \approx \frac{\sigma}{\sqrt{M}}$$

where the standard deviation is calculated based on the mean of each sub-trial, F_i, and the overall mean, F_{total}

$$\sigma = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}[F_i - F_{total}]^2}$$

In order to update the estimated uncertainty to reflect the new combined Monte Carlo (or quasi-Monte Carlo) estimation, then, the standard deviation can simply be re-calculated using values for F_i and F_{total} which have been updated to account for the new results, using the above "mean of the sub-mean" strategy. Note that, by using this strategy, the sample size is effectively doubled, while maintaining the same number of M independent sub-trials.

Appendix E

SAMPLE MATLAB CODE

Included here are samples of the MATLAB code created for this thesis work. The first sample included is for evaluations of test case one, using randomized quasi-Monte Carlo. The code for traditional Monte Carlo estimation is identical, except for where an initial ray position is sampled from the shifted quasi-random sequence 'ps', it is instead sampled using the built-in pseudorandom number generator function 'rand'. The second sample included is for performing design optimization, using the final methodology as presented in the thesis.

In the interest of saving paper, the remaining functions are not included, and have been made available online at
https://sites.google.com/site/andrewmarstonresearch/home/masters-1

E.1 OBJECTIVE FUNCTION EVALUATION FOR TEST CASE ONE

```
function [aveAbsorbed, absorbed] = testcase1rQMC(DP, rayTotal, trialTotal)
% This function performs RQMC for test case 1, based on the input current
% state of the design parameter vector DP, and the required number of
% rays. Perfect optics are assumed.
% This function returns the estimated value for F, as well as a vector of
% size 'trialTotal' by one, containing the means of the sub-trials.

% determine if Mandrew wants visual accompaniment
if rayTotal < 41
    visual = 1;
    figure
else visual = 0;
end

%---------------- DEFINE PROBLEM GEOMETRY -------------------------------
% define control points
P0 = [5;5];
P1 = [DP(1);DP(2)];
P2 = [5;-5];
% define receiver properties (circle centred at (f(1),f(2)))
f = [5;0];
radius = 0.5;

if visual == 1  % to visualize control points and Bezier spline
    plot(P0(1),P0(2),'bo')  % plot control points
    hold on
```

```
    plot(P1(1),P1(2),'bo')
    plot(P2(1),P2(2),'bo')
    u = 0:0.05:1;  % create Bezier spline, with points defined by matrix C
    stop = length(u);
    C = zeros(stop,2);
    for i = 1:stop
        C(i,1) = P0(1).*(1-u(i)).^2 + P1(1).*2.*u(i).*(1-u(i)) ↲
+ P2(1).*u(i).^2;
        C(i,2) = P0(2).*(1-u(i)).^2 + P1(2).*2.*u(i).*(1-u(i)) ↲
+ P2(2).*u(i).^2;
    end
    plot(C(:,1),C(:,2),'LineWidth',2)
    y = f(2)-radius:0.05:f(2)+radius;       % for the receiver circle
    x1 = sqrt(radius^2-(y-f(2)).^2)+f(1);
    x2 = -sqrt(radius^2-(y-f(2)).^2)+f(1);
    plot(x1,y,'Color','r','LineWidth',1.5)
    plot(x2,y,'Color','r','LineWidth',1.5)
    axis([-1,11,-6,6])  % adjust axis limits
    axis equal
end

% ---------- RAY TRACING BEGINS ----------------------------------------
n = zeros(2,1);
absorbed = zeros(trialTotal,1); % counts the number of rays absorbed by
the receiver

p = haltonset(1,'skip',64);     % generate quasi-random sequence
p = p(1:(rayTotal/trialTotal));
for trialNum = 1:trialTotal
    % perform random "shift" for each trial
    ps = p + rand;
    ps = mod(ps,1);

    for rayNum = 1:rayTotal/trialTotal
        done = 0;       % if ray has been absorbed, done = 1
        reflected = 0;  % if ray is been reflected previously, reflected>0

        if visual == 1
            LineColor = [rand,rand,rand];  % choose random ray colour
        end

        while done == 0;    % continue to trace ray

            if reflected == 0;      % define a new ray
                Ro = [0;-5+(10*ps(rayNum))];    % ray origin
                dir = [1;0];        % ray direction
            end

            % ------------ determine ray/receiver interaction ------------
            % between ray and receiver (line and circle):
            A = dir(1)^2 + dir(2)^2;    % calculate A, B, and C from ↲
quadratic formula
            B = 2*Ro(1)*dir(1) - 2*f(1)*dir(1) + 2*Ro(2)*dir(2) - ↲
2*f(2)*dir(2);
```

```
            C = Ro(1)^2 - 2*Ro(1)*f(1) + f(1)^2 + Ro(2)^2 - 2*Ro(2)*f(2) ↲
+ f(2)^2 - radius^2;
            determinant = B^2-4*A*C;
            if determinant > 0
                absorbed(trialNum) = absorbed(trialNum) + 1;
                done = 1;
                if visual == 1
                    w(1) = (-B + sqrt(determinant)) / (2*A);
                    w(2) = (-B - sqrt(determinant)) / (2*A);
                    w = min(w);
                    intersection = Ro + w.*dir;
                    line([Ro(1),intersection(1)],[Ro(2),intersection(2)],↲
'Color',LineColor)
                end
                continue
            end

            if reflected > 0 && abs(Ro(2)+((5-Ro(1))/dir(1))*dir(2)) <= 5
                if abs(Ro(2))==5 && dir(1)>0
                    continue
                end
                done = 1;    % ray escapes from aperture
                if visual == 1
                    rayEnd = Ro + 4*dir;
                    arrow(Ro,rayEnd,'EdgeColor',LineColor,'FaceColor',↲
LineColor,'Length',5,'TipAngle',28,'BaseAngle',12)
                end
                continue
            end

            % so the ray must intersect the concentrator ... where?
            % establish an initial guess
            if reflected == 0   % choose reasonable guess
                u = (5-Ro(2))/10;
                w = 5;
            else  % choose guess to prevent convergence to new ray origin
                if dir(2)>0   % i.e. if ray is traveling upward
                    u = 0;      % start search at top of curve
                    w = 7;
                else
                    u = 1;
                    w = 7;
                end
            end

            % find intersection using Newton-Raphson:
            i = 0;  % counter for number of iterations
            J(:,2) = dir;   % Jacobian
            C = P0.*(1-u)^2 + P1.*2*u*(1-u) + P2.*u^2;
            B = Ro + w.*dir;    % ray position
            distance = sum((C-B).^2).^0.5;

            while distance > 1e-4       % Newton-Raphson iterations
                J(:,1) = 2*(1-u)*P0 - (2*(1-u)-2*u)*P1 - 2*u*P2;
                F = B - C;
                update = -J\F;
```

```
                u = u + update(1);
                w = w + update(2);
                C = P0.*(1-u)^2 + P1.*2*u*(1-u) + P2.*u^2;
                B = Ro + w.*dir;
                distance = sum((C-B).^2).^0.5;
                i = i + 1;
                if i > 100
                    disp('possible error in Newtons method ... i > 100')
                    continue
                end
            end
            intersection = B;   % B = Ro + w*dir (point of intersection)

            if visual == 1
                  line([Ro(1),intersection(1)],[Ro(2),intersection(2)], ↲
'Color',LineColor)
            end

            reflected = 1;      % all reflections are specular
            n(1) = -2*(1-u)*P0(2) + (2-4*u)*P1(2) + 2*u*P2(2);
            n(2) = 2*(1-u)*P0(1) - (2-4*u)*P1(1) - 2*u*P2(1);
            n = n/norm(n);      % normalize
            dir = (intersection - Ro) - 2*(n'*(intersection - Ro))*n;
            % define new ray direction
            dir = dir/norm(dir);
            Ro = intersection;  % new ray originates at intersection point
        end
    end
end
absorbed = absorbed./(rayTotal/trialTotal);
aveAbsorbed = sum(absorbed)/trialTotal;
```

E.2 DESIGN OPTIMIZATION

```
function
[Fmax,TimeElapsed,objEval,designIterations,rayCounter,termination, ↲
kmax,normAtOptimal,normal] = DesignOptimizerQLS(MC, DPinit, hInit, ↲
FileName)
% This function performs the developed stochastic optimization algorithm,
% for the case in the objective function evaluation function specified.
% A divided differencing approximation is used to estimate the gradient.
% Steepest-descent method is used to determine direction of each design
% update, and the Kiefer-Wolfowitz recommended power law series are used
% for difference parameter control, while a quadratic line search is
% performed to determine step size.
% A condition for sample size increase is implemented for error anagement,
% based on uncertainties estimated using the Kline-McClintock method.
% MC or RQMC simulation is used to estimate the objective function,
% based on the function indicated.
% Termination is based on a combination of gradient-normal, significant
% performance improvement, and alloted CPU time, according to the chosen
% parameters.
% INPUTS: MC : a string of text corresponding to the name of the Monte
% Carlo ray-trace routine;
% DP : vector of initial design parameters ;
```

```
% rayTotal : scalar for initial  number of bundles to use in the Monte
% Carlo ray-tracing for FEval ;
% hInit : initial difference parameter ;
% FileName : (string)the name of the file to save the workspace to ;
% RETURNS:
% termination == 1 : terminated based on gradient normal
% termiantion == 2 : terminated based on lack of significant improvement
% termination == 3 : terminated due to exceeding alloted design iterations
% termination == 4 : terminated due to exceeding alloted CPU time

disp('-------------------------------------------------')

% ---------------- INITIALIZE VARIABLES & DEFINE COUNTERS ----------------
Time = cputime; % start time for elapsed CPU
objEval = 0;    % objective function evaluation counter
rayCounter = 0; % counter for total number of rays traced
k = 1;          % design iteration counter
trialTotal = 20;    % number of trials for each Monte Carlo estimation
rayTotal = zeros(2,1);
rayNum = 256;   % initial sample size
rayTotal(1) = rayNum;
n = length(DPinit);     % dimension of problem (# of design parameters)

DP = zeros(2,n);
DP(1,:) = DPinit;
F = zeros(2,1);     % objective function value at iteration (exchange ↲
factor of current design)
G = zeros(2,n);     % gradient vectors
normal = ones(2,1); % gradient normal
P = zeros(2,n);     % direction of steepest ascent

F1 = zeros(2,n);    % objective function in forward direction
F2 = zeros(2,n);    % objective function in backward direction
F1new = zeros(2,n);
F2new = zeros(2,n);
absorbed1 = zeros(trialTotal,n);
absorbed1new = zeros(trialTotal,n);
Umc1 = zeros(2,n);  % Monte Carlo uncertainty in the forward direction
timeQLS = zeros(2,1);    % amount of time spent in Quadratic line search

% initialize series for step size and difference parameter
gamma = 1.5;        % expansion parameter >= 1, delta scales with h
g = zeros(3,1);     % objective function during bracketing search
s = zeros(3,1);     % step size (delta) during bracketing search
alpha = zeros(2,1);
h = zeros(2,1);
h(1) = hInit;       % initial difference parameter
b = 1/3;            % power law coefficient (as reccommended by K-W)

K = 5;              % scalar multiplier for sample size increase condition
rayNumMax = 1e06;    % maximum allowed sample size

% ----------  DEFINE SETTINGS FOR TERMINATION CRITERIA  -------------
termination = 0;
if strncmp(MC,'testcase1',9)
    NormalTerminate = 0.0001;   % gradient normal at which to terminate
elseif strncmp(MC,'testcase2',9)
```

```
      NormalTerminate = 0.01;   % gradient normal at which to terminate
else % if strncmp(MC,'testcase3',9)
      NormalTerminate = 0.01;   % gradient normal at which to terminate
end
maxTime = 180000;     % alloted CPU time

if strncmp(MC,'testcase3',9)
    new = 5;    % # iterations considered in current moving average
    old = 15;   % # iterations considered in previous moving average
    PI = 0.001;  % fraction improvement considered significant
    MG = 50;    % maximum number overall iterations
else    % for test cases 1 or 2
    new = 5;    % # iterations considered in current moving average
    old = 15;   % # iterations considered in previous moving average
    PI = 0.005;  % fraction improvement considered significant
    MG = 50;    % maximum number overall iterations
end
improvement = ones(old,1);  % for storing the "fraction improvement"

% Set up strings for function evaluations (1 = forwards, 2 = backwards).
FEval = [MC,'(DP(k,:),rayTotal(k),trialTotal)'];
FEval1 = [MC,'(DP(k,:)+h(k)*unit(i,n),rayTotal(k),trialTotal)'];
FEval2 = [MC,'(DP(k,:)-h(k)*unit(i,n),rayTotal(k),trialTotal)'];

% ------------------- INITIAL GRADIENT APPROXIMATION --------------------
% perform initial estimation
for i = 1:n     % estimate the gradient using central differencing
    [F1(k,i), absorbed1(:,i)] = eval(FEval1);
    Umc1(k,i) = std(absorbed1(:,i))/sqrt(trialTotal);
    [F2(k,i)] = eval(FEval2);
    objEval = objEval + 2;
    rayCounter = rayCounter + 2*rayTotal(k);
    G(k,i) = (F1(k,i)-F2(k,i)) / (2*h(k));
end
normal(k) = norm(G(k,:));

disp(['Umc is ',num2str(mean(Umc1(k,:)))])

while sqrt(2)*h(k)*normal(k) < K * mean(Umc1(k,:)) && rayTotal(k) < ↵
rayNumMax % increase bundle size and repeat until satisfied
    for i = 1:n
        [F1new(k,i), absorbed1new(:,i)] = eval(FEval1);
        [F2new(k,i)] = eval(FEval2);
        rayCounter = rayCounter + 2*rayTotal(k);
        absorbed1(:,i) = mean([absorbed1(:,i),absorbed1new(:,i)],2);
        % updated aveAbsorbed is mean of old and new
        Umc1(k,i) = std(absorbed1(:,i))/sqrt(trialTotal);
        F1(k,i) = mean([F1(k,i),F1new(k,i)]);
        F2(k,i) = mean([F2(k,i),F2new(k,i)]);
        G(k,i) = (F1(k,i)-F2(k,i)) / (2*h(k));
    end
    normal(k) = norm(G(k,:));
    rayTotal(k) = rayTotal(k)*2;
    disp(['Bundle size doubled to ',num2str(rayTotal(k))])
    disp(['Umc is ',num2str(mean(Umc1(k,:)))])
    if normal(k) < NormalTerminate     % should terminate
        termination = 1;
```

```
        break
    elseif (cputime - Time) > maxTime
        termination = 4;
        break
    end
end
P(k,:) = G(k,:)./normal(k); % define direction of steepest descent
[F(k)] = eval(FEval);   % evaluate objective function for current design
disp(['F = ',num2str(F(k))])

% ------------------------ DISPLAY PROGRESS --------------------------
disp('Original Design (k=1):')
disp(['DP = ',num2str(DP(k,:))])
disp(['P = ',num2str(P(k,:))])
disp(['h = ',num2str(h(k))])
disp(['rayTotal = ',num2str(rayTotal(k))])
disp(['normal = ',num2str(normal(k))])
disp(['Umc = ',num2str(mean(Umc1(k,:)))])


% ---------------------- PERFORM DESIGN UPDATES -------------------------
while termination == 0 && F(k) < 1

    % -------------------- QUADRATIC LINE SEARCH ------------------------
    %   A three-point pattern search is performed based on the algorithm
    %   provided in Belegundu & Chandrupatla, pp 28-29
    time2 = cputime;
    delta = h(k); % scale delta with difference parameter
    s(1) = 0;       % start search at current DP value
    g(1) = -F(k);   % we've already evaluated at the first point
    s(2) = s(1) + delta;
    [g(2)] = eval([MC,'(DP(k,:)+s(2)*P(k,:),rayTotal(k),trialTotal)']);
    g(2) = -g(2);
    done = 0;
    objEval = objEval + 2;  % increment counter for objective function ↲
evaluations
    rayCounter = rayCounter + 2*rayTotal(k);
    if g(2) > g(1)  % we must be passing by the minimum ??
        temp = g(2);
        g(2) = g(1);
        g(1) = temp;
        temp = s(2);
        s(2) = s(1);
        s(1) = temp;
        delta = -delta;
    end
    while done == 0     % find an initial "three-point pattern"
        delta = gamma * delta;
        s(3) = s(2) + delta;
        [g(3)] =
eval([MC,'(DP(k,:)+s(3)*P(k,:),rayTotal(k),trialTotal)']);
        g(3) = -g(3);
        objEval = objEval + 1;
        rayCounter = rayCounter + rayTotal(k);
        if g(3) > g(2)     % three-point pattern found
            done = 1;
            continue
```

```
        end
        g(1) = g(2);
        s(1) = s(2);
        g(2) = g(3);
        s(2) = s(3);
        if (cputime - time2) > 3600
            save(FileName)   % save current workspace
            error('stuck in three-point pattern search')
        end
    end
    alpha(k) = 0.5 * ( g(1)*(s(3)^2-s(2)^2) + g(2)*(s(1)^2-s(3)^2) + ↲
g(3)*(s(2)^2-s(1)^2) ) / ( g(1)*(s(3)-s(2)) + g(2)*(s(1)-s(3)) + ↲
g(3)*(s(2)-s(1)) );
    timeQLS(k) = cputime-time2;     % time spent in line search

    disp(['alpha = ',num2str(alpha(k))])    % display more details

    % ----------- update parameters for next design step -----------------
    rayTotal(k+1) = rayNum;     % start at initial ray size at each ↲
iteration
    DP(k+1,:) = DP(k,:) + alpha(k).*P(k,:);     % update design parameters
    h(k+1) = h(1)/((k+1)^b);        % update difference parameter
    k = k + 1;      % increment iteration counter

    disp('----------------------------------------')
    disp(['iteration number ',num2str(k),':'])
    disp(['DP = ',num2str(DP(k,:))])
    disp(['h = ',num2str(h(k))])

    % ------------- determine search direction ---------------
    % perform initial gradient estimation
    for i = 1:n     % estimate the gradient using central differencing
        [F1(k,i), absorbed1(:,i)] = eval(FEval1);
        Umc1(k,i) = std(absorbed1(:,i))/sqrt(trialTotal);
        [F2(k,i)] = eval(FEval2);
        objEval = objEval + 2;
        rayCounter = rayCounter + 2*rayTotal(k);
        G(k,i) = (F1(k,i)-F2(k,i)) / (2*h(k));
    end
    normal(k) = norm(G(k,:));

    disp(['Umc = ',num2str(mean(Umc1(k,:)))])
    while sqrt(2)*h(k)*normal(k) < K * mean(Umc1(k,:))  && rayTotal(k) < ↲
rayNumMax % increase bundle size and repeat until satisfied
        for i = 1:n     % estimate the gradient using central differencing
            [F1new(k,i), absorbed1new(:,i)] = eval(FEval1);
            [F2new(k,i)] = eval(FEval2);
            rayCounter = rayCounter + 2*rayTotal(k);
            absorbed1(:,i) = mean([absorbed1(:,i),absorbed1new(:,i)],2);
           % updated elements of 'absorbed' are mean of old and new
            Umc1(k,i) = std(absorbed1(:,i))/sqrt(trialTotal);
            F1(k,i) = mean([F1(k,i),F1new(k,i)]);   % udpated F values ↲
are mean of old and new
            F2(k,i) = mean([F2(k,i),F2new(k,i)]);
            G(k,i) = (F1(k,i)-F2(k,i)) / (2*h(k));
        end
```

```
        normal(k) = norm(G(k,:));
        rayTotal(k) = rayTotal(k)*2;
        disp(['Bundle size doubled to ',num2str(rayTotal(k))])
        disp(['Umc = ',num2str(mean(Umc1(k,:)))])
        disp(['normal = ',num2str(normal(k))])

        if normal(k) < NormalTerminate     % should terminate
            termination = 1;
            break
        elseif (cputime - Time) > maxTime
            termination = 4;
            break
        end
    end
    P(k,:) = G(k,:)./normal(k); % define direction of steepest descent
    [F(k)] = eval(FEval);   % evaluate objective function for current ↲
design
    disp(['F = ',num2str(F(k))])

    % --------------- DISPLAY RESULTS OF CURRENT ITERATION ---------------
    disp(['P = ',num2str(P(k,:))])
    disp(['rayTotal = ',num2str(rayTotal(k))])
    disp(['normal = ',num2str(normal(k))])
    disp(['Umc = ',num2str(mean(Umc1(k,:)))])

    % check termination criteria:
    if normal(k) < NormalTerminate     % should terminate
        termination = 1;
    elseif k > old
        current = mean(F(k-new+1:k));
        previous = mean(F(k-old+1:k-new));
        improvement(k) = (current - previous) / previous;
        if improvement(k) < PI
            termination = 2;
        end
    end
    if k > MG
        termination = 3;
    elseif (cputime - Time) > maxTime
        termination = 4;
    end
end

disp(['Termination occurred at k = ',num2str(k), ', based on condition ' ↲
,num2str(termination)])

designIterations = k - 1;   % number of design updates is k-1
TimeElapsed = cputime - Time;   % note: below operations not included in
total elapsed time

% ------------------- PLOT OPTIMIZATION PROGRESS ----------------------
H = figure;
for j = 1:designIterations
    subplot(4,1,1)
    hold on
    if j == 1
```

```
        xlabel('iteration')
        ylabel('F')
    end
    plot(j,F(j),'b*')
end
for j = 1:designIterations
    subplot(4,1,2)
    hold on
    if j == 1
        xlabel('iteration')
        ylabel('\alpha')
    end
    plot(j,alpha(j),'b*')
end
for j = 1:designIterations
    subplot(4,1,3)
    hold on
    if j == 1
        xlabel('iteration')
        ylabel('log(N_b)')
    end
    plot(j,log10(rayTotal(j)),'b*')
end
for j = 1:designIterations
    subplot(4,1,4)
    hold on
    if j == 1
        xlabel('iteration')
        ylabel('||\nabla(F)||')
    end
    plot(j,normal(j),'b*')
end
saveas(H,[FileName,' - progress'],'fig')
close all

display('------- beginning post-processing -------')


% test last few designs with more accuracy to determine the best
if k < new
    numberToConsider = k;
else
    numberToConsider = new;
end

for i = 0:numberToConsider-1
    if rayTotal(k-i) < 50000
        F(k-i) = eval([MC,'(DP(k-i,:),50000,20)']);
    end
end
Fmax = max(F(k-numberToConsider+1:k));
kmax = find(ismember(F,Fmax));

if length(kmax) > 1    % if the optimal design has occurred twice,
    kmax = kmax(length(kmax));    % take the last occurrence
end
if kmax ~= MG      % normal hasn't been computed for the latest k
```

```
    normAtOptimal = normal(kmax);
else
    normAtOptimal = 100;
end
DPoptimal = DP(kmax,:);

eval([MC,'(DPoptimal,16,1)']);     % give visualization of final design
H1 = gcf;
if strncmp(MC,'testcase1',9)
    H2 = openfig('testcase1 - contour.fig');
    hold on
    MarkerSize = 4;
    plot(DP(1,1),DP(1,2),'ko','MarkerSize',MarkerSize)
    for i = 1:k-1
        line([DP(i,1),DP(i+1,1)],[DP(i,2),DP(i+1,2)],'Color','k')
        plot(DP(i+1,1),DP(i+1,2),'ko','MarkerSize',MarkerSize)
    end
    saveas(H2,[FileName,' - design path'],'fig')
elseif strncmp(MC,'testcase2',9)
    axis([-1,11,-6,6])
elseif strncmp(MC,'testcase3',9)
    axis([-7,7,-8,4])
end
xlabel('x','FontSize',13)
ylabel('y','FontSize',13)
title(['F = ',num2str(Fmax),', termination = ',num2str(termination)])
saveas(H1,[FileName,' - optimal design'],'fig')

close all
save(FileName)   % save current workspace
```