Computational Methods for the Shape from Shading Problem

by
Paul M. Carter

Technical Report 93-26
July 1993

Department of Computer Science
University of British Columbia
Rm 333 - 6356 Agricultural Road
Vancouver, B.C.
CANADA V6T 1Z2

Telephone: (604) 822-3061
Fax: (604) 822-5485

Abstract

The aim of this thesis is to explore computational methods for the shape from shading problem as formulated through the image irradiance equation. We seek to develop robust, efficient methods and test our algorithms on synthetic images ranging from simple smooth surfaces to complex digital terrain model data. We consider three different approaches.

The first approach revisits the method of characteristic strips with a view to using more stable integration schemes than had been used in earlier works. Stable schemes, coupled with projections onto the image irradiance equation are used. Although the effects of noise are thereby reduced, the solution is still deemed unsatisfactory even for very simple surfaces.

The second approach considers Horn's variational technique as a basis for producing a fast solver. We devise a discretization scheme coupled with a special continuation-multigrid method for this formulation. We also allow for multiple image data and explicit knowledge of the location of discontinuities in surface height and orientation. Given multiple image data, we obtain excellent results even in the presence of discontinuities.

The third approach examines a class of solution techniques based on fluid flow which are new to the shape from shading literature. This formulation is ill-posed in general, so we propose a regularization of the problem. We observe that the algorithm is prone to producing spurious results. Analysis shows that these are due to the non-random accumulation of errors in the computed solution.

Of the three approaches considered, the variational method yields the most promising results. Efficient, good quality reconstructions are obtained, especially when data from more than one image are available.

Table of Contents

List of Tables

List of Figures

Acknowledgement

I would like to acknowledge the inspiration, guidance and enthusiasm of my research supervisor Dr. Uri Ascher. I thank him for having encouraged me to complete this thesis. I would also like to thank the members of my supervisory committee: Drs. Jim Little, Jim Varah, Bob Woodham and Matt Yedlin. Their helpful comments during the research and writing of this thesis are greatly appreciated.

Chapter 1

Introduction

1.1 Overview

The field of computer vision, while having its roots in a biological function which most of us consider effortless, is an area of research containing a wealth of problems which have proved to be anything but effortless to solve. It has become clear that the building of a vision system requires the construction of many component parts. Shape from shading is just one of those components.

Shape from shading consists of a class of problems which aim to recover the depth map of the surface giving rise to a shaded image or images. Under certain lighting conditions and with assumptions on the reflectance properties of the surface, it can be shown that mathematically the problem reduces to one of solving a nonlinear, first order, partial differential equation known as the *image irradiance equation*.

1.2 Historical background

The image irradiance equation has been solved using both the method of characteristic strips, see Horn [21], and by variational techniques, see Horn & Brooks [23]. Woodham [55] developed a technique called photometric stereo which uses data from multiple images to determine surface slopes which are then integrated to produce the surface height.

The method of characteristic strips has proved to be difficult to implement in the presence of image noise as accumulation of errors leads to a tendency for characteristics

to cross. It has also been criticized for not having its roots in a corresponding biological function. This method now appears to have been abandoned in favour of iterative solutions arising from variational formulations of the problem.

The variational formulation arises not out of a desire to solve the image irradiance equation exactly but rather to find the surface which best satisfies this equation (thus allowing for noise) as well as perhaps some other constraint or constraints. The imposition of additional constraints turns out to be not so much an option as a necessity if a reasonable solution is to be attained. Examples of such constraints are penalty terms for non-smoothness and non-integrability of the computed surface slopes.

The differential equations arising from the variational formulation, known as the Euler equations, have been solved using iterative techniques most of which appear to suffer from poor convergence rates. It has been suggested that the multigrid method, which has proved to be a fast, efficient solver for elliptic partial differential equations, may be used to improve the convergence rate, see Terzopoulos [50]. However, as we shall see, the application of this method to the shape from shading problem is one which requires due care and attention.

Oliensis & Dupuis [39] and Lions, Rouy & Tourin [34] [51] [46] have recently developed new techniques based on ideas in Control Theory and Dynamical Systems. These techniques use information at singular points (which we define later in the chapter) to determine the solution at other points in the domain. Hence, singular points, rather than boundary information at the edge of the image, are seen as the determining factor in resolving ambiguities in solutions.

Bruckstein [11] and Kimmel & Bruckstein [28] present algorithms for computing the height of an object directly from a shaded image via level sets. Their work is inspired by wavefront propagation techniques.

Leclerc & Bobick [30] present a variational technique for computing surface height

directly. An iterative minimization technique based on the conjugate gradient method is employed. They also demonstrate the possibility of using stereo processing of images to provide boundary and initial data for their shape from shading algorithm.

Lee & Kuo [32][33] present a finite element solution method which computes surface height directly on a triangular mesh. They use linear basis functions and linearize the reflectance map about the previous iterate. A smoothing term is added to regularize the problem which can gradually be reduced to zero in the case when two images are input. They show that the data from two images serve to greatly improve the accuracy of their algorithm.

While many authors assume that the light source direction is known a priori, others (Zheng & Chellappa [59], Brooks & Horn [7]) have developed methods for estimating the light source direction from the image data. Gibbins, Brooks & Chojnacki [17] present a comparative performance analysis of existing algorithms for estimating light source direction. They indicate that methods which aim to determine light source direction as a preprocessing step in a shape from shading algorithm tend not to be particularly robust. It was suggested that methods which couple together the estimation of light source direction and computation of surface height, such as Brooks & Horn [7], tend to be more reliable.

There has been comparatively little work in the area of existence and uniqueness of shape from shading problems. Bruss [12] and Brooks, Chojnacki & Kozera [8][9] have presented existence and non-uniqueness results for the shape from shading problem when a single image with overhead illumination is given as input. Oliensis [37][38] has also given uniqueness results for the single image problem although his work is generalized to the case of oblique illumination. In the latter case he deduces that the shape from shading problem is only partially well constrained. His results are based on the interpretation of the shape from shading problem as a "flow" of characteristics over the surface, see

also Saxberg [47]. Horn, Szeliski & Yuille [26] demonstrate the existence of images which cannot arise from the shading of a smooth surface with uniform reflectance properties. They refer to such images as impossible shaded images. Onn & Bruckstein [40] and Kozera [29] show that two image photometric stereo generally gives rise to a unique solution of the shape from shading problem.

1.3 Shape from Shading

Ramachandran [45] has demonstrated that shape from shading is without doubt a component of the human vision system. The presence of boundaries (whether illusory or real) was found to be important for depth perception. It was also shown that the same shading pattern with different boundary information gives the impression of quite different surfaces lit from different directions indicating the importance of boundary information for the human vision system. However, depth cues from shading were easily over-ridden by stereographic cues and other higher-level visual processes such as object recognition. Marr [36] has commented that extracting shape from shading is something that the human visual system does not do particularly well but he went on to say that "Nevertheless, we do make some use of shading, so there is definitely something there to be understood."

As a process which explicitly reconstructs the depth-map of a surface, shape from shading has received some criticism. Blake & Zisserman [3] have commented that "direct applications for depth maps, as descriptions of visible surfaces, are at best limited and at worst, perhaps, non-existent." They argued that in terms of object recognition, the depth-map has no place at all. Other uses of vision, such as collision-avoidance, may well require the use of a depth-map but they argue that bounding the object by a polygon is a more compact way of describing the position of the object to be avoided. However, the depth map can be useful for tasks involving object manipulation. There are also

situations when the human visual system reconstructs a surface solely from shading cues, for example, when looking at shaded topographic maps or satellite images of the earth, moon or planet. If the extraction of shape from shading is a function of the human visual system (and it undoubtedly is), we believe that it also has a place in computer vision systems even though its role in the overall scheme of things may not be that great.

The earliest and most general solution to the shape from shading problem was presented as part of Horn's 1970 Ph.D. thesis. The relevant part of this thesis is reprinted in Horn [20]. The solution was general in the sense that the simplifying assumptions of distant viewer and distant light source, which figure in later works, were not made. Also, the application of the solution technique to imaging systems such as the electron microscope and surfaces such as that found on the moon, demonstrated further generality. The governing equations were solved using the method of characteristics, which yields a set of five, coupled, ordinary differential equations.

Later authors, including Horn, found the assumptions of distant viewer and distant light source to be quite reasonable in many imaging situations and with the ensuing mathematical simplification their adoption has proved to be popular. It was Horn & Sjoberg [25] who demonstrated how a reflectance map, giving scene radiance as a function of surface slope, could be derived under these assumptions with a knowledge of the Bidirectional Reflectance Distribution Function (BRDF) and the location of light sources. The BRDF is defined by the National Bureau of Standards to be the ratio of the scene radiance reflected towards the viewer to the scene irradiance coming from the light source for a given source and viewer position and has units of steradian^{-1}.

We denote the reflectance map by $R(p, q)$ where (p, q) is the surface gradient at a point (x, y) that is, $p = z_x$ and $q = z_y$ where $z(x, y)$ denotes surface height and a subscript denotes partial differentiation with respect to the indicated variable. Further, we denote the image brightness at this point by $E(x, y)$. Then under the assumption that image

irradiance equals scene radiance we have:

$$E(x, y) = R(p, q). \tag{1.1}$$

This equation is known as the *image irradiance equation.* In general, $R(p, q)$ is a non-linear function of p and q and so equation (1.1) can be described as a non-linear, first order, partial differential equation in z, the surface height.

Horn later reworked his solution to make use of the reflectance map (see Horn [21]). However, the successful solution of the governing equations by means of characteristic strips depends on a noise-free image which under normal circumstances cannot be guaranteed.

Woodham [54] presented the first attempt at an iterative means of determining shape from shading. Constraints on local topography were used to iteratively restrict the gradient space contours on which the slope (p, q) of a particular point must lie. The algorithm was highly dependent on the ability to hypothesize an ordering of image points with respect to viewing angle and direction of steepest descent along the surface. As implemented, the algorithm required an initial guess which served as a framework to locally order closely spaced points. Iterative refinement then aimed to modify this surface with its local constraints into a surface which represented a "simple distortion" of the initial guess.

Woodham [56] later showed how surface orientation can be determined when a priori knowledge of the object is at hand. Cases dealt with include convex surfaces, developable surfaces (for example, right circular cones and cylinders) and generalized cones (for which, in cylindrical coordinates (r, θ, z), $r = h(z)f(\theta)$ where h and f are continuous functions of the indicated independent variable). It was shown, for example, that points on a developable surface whose axis is parallel to the image plane, have slopes which lie on a one parameter curve in gradient space. Hence surface orientation may be determined by

systematically scanning along image lines and determining the single parameter, t, for which $R(p(t), q(t)) = E(x, y)$.

Also at this time, Woodham [55] presented a scheme for determining surface orientation from multiple images. The method is referred to as *photometric stereo.* Woodham showed that for a special case of a Minnaert surface (which is commonly used to model the Moon's surface) for which we have the linear reflectance map:

$$R(p, q) = \rho \frac{(1 + p_0 p + q_0 q)}{\sqrt{1 + p_0^2 + q_0^2}} \tag{1.2}$$

only two images (obtained from different light sources) are required to uniquely determine p and q provided ρ, the surface albedo, and (p_0, q_0), which specifies the light source direction in each image, are known.

It was also shown that for a Lambertian surface for which we have the nonlinear reflectance map:

$$R(p, q) = \max \left\{ 0,\ \rho \frac{(1 + p_0 p + q_0 q)}{\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2}} \right\} \tag{1.3}$$

only three images are required to uniquely determine ρ, p and q at each point provided the light sources used to obtain each image are not coplanar and that there are no non-illuminated regions in any of the three images. Note that R achieves its maximal value of 1 when $p = p_0$ and $q = q_0$. At such points $R_p = R_q = 0$. We refer to these points as *singular points.* Note that (1.3) can be interpreted as:

$$R = \max \{0,\ \rho \cos(\theta)\} \tag{1.4}$$

where θ is the angle between the light source direction, $(-p_0, -q_0, 1)$, and the surface normal, $(-p, -q, 1)$. Here we assume that the viewer is located at $+\infty$ on the z-axis.

Next we consider the many schemes that have arisen from variational techniques. Such techniques have proved popular given the apparent limitations of the method of

characteristics mentioned earlier. The idea behind the variational technique is to determine the functions $p(x, y)$ and $q(x, y)$ which minimize the functional:

$$F(p, q) = \int\int_{\Omega} (E - R(p, q))^2 d\Omega \tag{1.5}$$

where the domain of integration, Ω, is the image under consideration. Hence we set out not to solve the image irradiance equation exactly but to determine the functions p and q which minimize the error in this equation. In other words, we determine the surface slopes which in some sense *best* solve the image irradiance equation thus allowing for the presence of noise.

Unfortunately, the problem is ill-posed in the sense that it does not always have a unique solution, see Poggio & Torre [44], even under the imposition of boundary conditions. Various authors have presented different means of dealing with the problem of non-uniqueness. An excellent overview is to be found in Horn & Brooks [24]. Of particular note in this work is the realization that the search for a unique solution must ensure that the orientation map produced should be integrable. In other words, it is of little use to provide the surface gradient which minimizes equation (1.5) if that gradient cannot be integrated to yield the depth-map. In this vein, Horn and Brooks considered the addition of a term which adds a penalty for non-integrability of the surface slopes, i.e. they considered the minimization of

$$F(p, q) = \int\int_{\Omega} \left\{ (E - R(p, q))^2 + \mu (p_y - q_x)^2 \right\} d\Omega. \tag{1.6}$$

Notice that here they do not strictly enforce integrability at each point; schemes which attempted to do this by means of a Lagrange Multiplier were found not to be convergent. It was concluded that exact imposition of integrability at each point was too restrictive.

A necessary condition for the functions p and q to be extremals of the functional $F(p, q)$ is that they satisfy the corresponding Euler equations, see Courant & Hilbert

[15]. For the functional given in equation (1.6) it was shown that the corresponding Euler equations are

$$\left.\begin{aligned} \mu(p_{yy} - q_{xy}) + (E - R)R_p &= 0 \\ \mu(q_{xx} - p_{xy}) + (E - R)R_q &= 0 \end{aligned}\right\} \tag{1.7}$$

These equations were solved in [24] using a simple Jacobi iteration. They reported very accurate results for tests conducted on synthetic images but also said that "like most shape from shading methods, it typically takes many iterations to converge." It was suggested that slow convergence could be remedied by the use of the multigrid method but they did not pursue this. Sweldens & Roose [49] however, present a parallel multigrid algorithm for this formulation.

Having obtained the orientation map, the corresponding depth-map is also obtained using variational techniques. The problem now is to determine the function $z(x, y)$ which minimizes the functional

$$\int\int_{\Omega} ((z_x - p)^2 + (z_y - q)^2)\, d\Omega. \tag{1.8}$$

The corresponding Euler equation was shown to be:

$$\nabla^2 z = p_x + q_y \tag{1.9}$$

subject to the natural boundary conditions (see Courant & Hilbert [15])

$$(z_x, z_y) \cdot \mathbf{n} = (p, q) \cdot \mathbf{n} \tag{1.10}$$

where $\mathbf{n}$ is the outward unit normal to the boundary curve. Hence, surface height is obtained to within an arbitrary constant of integration.

The availability of boundary value information appears to be a questionable issue. The gradient space formulation assumes some knowledge of p and q on a closed curve forming the boundary of the image. In many cases it is not unreasonable to assume that

a priori knowledge of the scene being imaged will give us such information. However, there is a class of problems for which gradient space formulation is not the most practical. In cases where the boundary of the object is an occluding boundary, one of p and q will become infinite and cannot therefore be used as boundary information.

Previously, Ikeuchi & Horn [27], presented a scheme where surface orientation is parametrized using stereographic coordinates, (f, g). Such co-ordinates remain finite on occluding boundaries so that the values of f and g on $\partial\Omega$ may be used as boundary conditions for the problem. However, there is a drawback in that the integrability condition cannot easily be expressed in stereographic coordinates. In order to obtain a unique solution, Ikeuchi and Horn considered the addition of a smoothing term and hence set out to minimize the functional

$$\int\int_{\Omega} \lambda\,(f_x^2 + f_y^2 + g_x^2 + g_y^2) + (E - R(f,g))^2\,d\Omega. \tag{1.11}$$

The corresponding Euler equations are

$$\left.\begin{aligned} \lambda\nabla^2 f + (E - R)R_f &= 0 \\ \lambda\nabla^2 g + (E - R)R_g &= 0 \end{aligned}\right\} \tag{1.12}$$

Again, these equations were solved using a simple Jacobi iteration. This scheme had good convergence properties but accuracy depended largely on the parameter λ. In cases where the image was synthetically produced and λ was small, very accurate results were reported, however, when there was noise in the image and λ was large, large errors in the solution were observed due to excessive smoothing of the surface. Terzopoulos [50] reported a successful application of the multigrid method to this scheme, the convergence rate being improved significantly. Lee [31] gives an algorithm based on this variational formulation which is provably convergent provided λ lies within a certain given interval.

Horn & Brooks [24] also presented a scheme which parametrized surface slope in terms of the unit surface normal. Occluding boundary information could again be employed

and a penalty term for non-integrability was also incorporated easily. Later this work was extended by Malik & Maydan [35] who set out to reconstruct the depth map of an object which consists of piecewise smooth surfaces. Their scheme included the labelling of edges in the images. However, the integrability penalty term was dropped and replaced with a constraint on surface smoothness. It was suggested that the scheme of Frankot & Chellappa [16] could be used to project the non-integrable surface slopes onto the nearest integrable surface slopes after each iteration. Vega & Yang [53] also parametrize surface slope in terms of the unit surface normal. Their algorithm, however, rather than being based on a variational formulation, is heuristic in nature. They do not incorporate any penalty for non-integrability of the computed surface slopes.

In his latest paper, Horn [22] returns to the gradient space parametrization of surface slope and presents a new "Height and Gradient from Shading" scheme. This scheme couples together the process of determining an orientation map and determining the depth map which had hitherto been dealt with sequentially. Again using variational techniques, Horn sets out to find functions z, p and q which minimize the functional

$$\int\int_{\Omega} \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu[(z_x - p)^2 + (z_y - q)^2] + (E - R(p,q))^2 d\Omega. \qquad (1.13)$$

The first of these terms represents a penalty for departure from smoothness. In cases where there is little or no noise in the image it was expected that λ may be taken small or even reduced to zero. The second term is the integrability penalty term. The third term requires no explanation. The Euler equations were shown to be

$$\left.\begin{aligned} \lambda\nabla^2 p + \mu(z_x - p) + (E - R)R_p &= 0 \\ \lambda\nabla^2 q + \mu(z_y - q) + (E - R)R_q &= 0 \\ \nabla^2 z - (p_x + q_y) &= 0 \end{aligned}\right\} \qquad (1.14)$$

Horn solves these equations on staggered grids – the grid for z being offset from the one for p and q by half a pixel. A discretization is devised such that the discrete equations

closely follow the underlying continuous equations: a discretization for $\nabla^2 z$ is derived by convolving the discrete operators for z_x and z_y with themselves, the latter being second order central differences. This procedure ensures complete consistency in the last equation of system (1.14).

The Euler equations were again solved using a simple Jacobi iteration; however, an important addition to the scheme was the use of linearization for the reflectance map so that $R(p^{n+1}, q^{n+1})$ is replaced by

$$R(p^n, q^n) + R_p(p^n, q^n)(p^{n+1} - p^n) + R_q(p^n, q^n)(q^{n+1} - q^n) \tag{1.15}$$

where (p^n, q^n) is the previous iterate and (p^{n+1}, q^{n+1}) is the iterate to be computed. Horn reported that this linearization process added stability to his scheme.

It was demonstrated that the scheme was extremely accurate for synthetic images where λ could be reduced to zero as the iteration proceeded. However, the convergence rate was very slow, especially when λ became small. Again it was suggested that the multigrid method could be used to speed up the convergence rate but this was not carried out.

1.4 Objectives and Contributions of the Thesis

The aim of this thesis is to propose and to investigate fast, efficient, robust numerical solutions to the shape from shading problem in computer vision as formulated through the image irradiance equation.

In Chapter 2, we examine stable numerical schemes for the method of characteristic strips which was the first solution method for the shape from shading problem. We show that even with stable integration schemes and projection methods, the scheme is still prone to produce significant errors in the computed solution when there is noise in the image data, as was observed previously for less stable ODE integration schemes.

In Chapter 3, we extend the work of Horn [22] in providing a fast, efficient multigrid solver for his variational formulation (1.14) of the shape from shading problem. Using a discretization on a nonstaggered grid, we propose a special multigrid algorithm where continuation in the regularization parameter λ is incorporated into a nested multigrid scheme. The resulting algorithm provides speedups of more than an order of magnitude over the algorithm in [22]. We also extend the formulation to include availability of data from more than one image and knowledge of the location of discontinuities in surface height and orientation. We show that when the same information is offered to our algorithm as to a photometric stereo algorithm, we can also achieve excellent results.

In Chapter 4, we examine a class of solution techniques which have so far not been considered in the shape from shading literature, namely upwinding schemes for flow formulations. This section is inspired by solution techniques for the incompressible Navier-Stokes equations which govern fluid flow. We establish the connection between the latter equations and a certain formulation of the shape from shading problem. At this point our work ties in with the work of Saxberg [47] and Oliensis [37] [38] who describe the shape from shading problem in terms of a "flow" of characteristics on the surface. We show that these solution techniques perform well in the absence of singular points, that is, points where the gradient of the reflectance map, in (p, q) space, vanishes: $R_p = R_q = 0$. However, when singular points are present we show that the algorithm is prone to producing spurious results. We relate this to the ill-posed nature of this particular formulation when there are singular points in the image. We further examine the tendency of the algorithm to produce spurious results by turning to a one dimensional version of the problem which lends itself to easier analysis. We also examine fundamental differences between the "flow" of shape from shading problems and fluid flow governed by the incompressible Navier-Stokes equations. Particular difficulties associated only with the flow of shape from shading problems are identified.

Finally, in Chapter 5 we state the conclusions arising from our investigations.

Chapter 2

The Method of Characteristic Strips

2.1 Introduction

Historically, the method of characteristic strips was among the first to be used as a solution technique for the general shape from shading problem, see Horn [20]. In light of difficulties arising when using this technique, the method began to receive less attention in favour of variational techniques. However, Saxberg [47] and Oliensis [37][38] have more recently re-examined the method of characteristics from a more theoretical viewpoint. Oliensis has shown that characteristics can be used to determine uniqueness in certain shape from shading problems. The existence of singular points, coupled with the flow of characteristic strips across the surface, uniquely specifies the solution in cases of overhead illumination when a smooth object is wholly contained in the field of view. Oliensis therefore concludes that in the case of overhead illumination, the shape from shading problem is generally well-posed and therefore does not necessarily require regularization. Oliensis also shows that in the case of oblique illumination, the shape from shading problem is partially well-constrained in that there will be some region in the image which gives rise to a unique solution.

Given this resurgence of interest in characteristics, we begin our investigation with a look at the method of characteristic strips.

2.2 Methods

2.2.1 The method of characteristic strips

The solution of first order, non-linear, partial differential equations by the method of characteristic strips is well documented, see Carrier & Pearson [13]. For the problem at hand we consider the image irradiance equation

$$E(x,y) = R(z_x, z_y) \tag{2.1}$$

as a first order, generally non-linear, partial differential equation in $z(x,y)$, the surface height. We begin by writing

$$E(x,y) = R(p,q) \tag{2.2}$$

where $p = z_x$ and $q = z_y$ are the surface slopes. Differentiating (2.2) with respect to x and y we get:

$$E_x = R_p p_x + R_q q_x \tag{2.3}$$

$$E_y = R_p p_y + R_q q_y \tag{2.4}$$

Now consider propagating the solution along a curve C known as a characteristic strip. Let

$$\vec{r}(\xi) = x(\xi)\vec{\imath} + y(\xi)\vec{\jmath} \tag{2.5}$$

be a parametrization of C for some parameter ξ. Then we may write

$$\dot{z}(\xi) = p\dot{x}(\xi) + q\dot{y}(\xi) \tag{2.6}$$

where a dot denotes differentiation with respect to ξ. Now, by choosing

$$\dot{x}(\xi) = R_p \tag{2.7}$$

$$\dot{y}(\xi) = R_q \tag{2.8}$$

we specify the direction of C and substituting into (2.3) and (2.4) we get

$$\dot{p}(\xi) = p_x\dot{x}(\xi) + q_x\dot{y}(\xi) = E_x \tag{2.9}$$

$$\dot{q}(\xi) = p_y\dot{x}(\xi) + q_y\dot{y}(\xi) = E_y \tag{2.10}$$

Equations (2.6) to (2.10) provide us with a means of propagating the solution $z(x, y)$ and its gradient along the curve C. Since the gradient as well as the solution is propagated along C, this curve is known as a characteristic strip as opposed to a characteristic curve.

We start the integration by specifying the solution (x, y, z, p, q) along an initial strip C_0. In order to propagate characteristics from this strip, the tangent vector to C_0 at each point must not be parallel to the vector $(R_p\vec{\imath} + R_q\vec{\jmath})$ at that point. The solution of the 5 non-linear, ordinary differential equations (2.6) to (2.10) with suitable initial conditions as just described, constitutes the method of characteristic strips.

Several problems with this method have been reported in the literature [20]. Many of these problems appear to be independent of the chosen numerical integration scheme used to solve the system. Horn reports that replacing a simple Euler method by a Runge-Kutta method does not yield significant improvement in accuracy. These methods appear to be particularly susceptible to noise: characteristics when propagated from some initial curve tended to cross yielding multiple solutions in certain regions.

Since the path of characteristic strips is dependent on the particular surface under consideration, the density of characteristics will change as they propagate out from the initial curve. Thus, there may be regions where no information is computed while in other regions, the characteristics are densely packed. Horn proposed a method whereby characteristics were computed in parallel. When two characterisitics became too close, one was terminated and when the gap between neighbouring characteristics became too large, a new curve was initiated by interpolating the solution from neighbouring curves.

In an attempt to reduce the effect of noise, Horn "sharpened" his solution for p and q by imposing the conditions

$$\dot{z} - p\dot{x} - q\dot{y} = 0 \tag{2.11}$$

$$E(x, y) - R(p.q) = 0 \tag{2.12}$$

along "rings" or curves consisting of points equidistant from the initial curve. The term "ring" is used since Horn typically took circles around singular points as initial curves. Local sphericity of the surface was assumed in order to specify initial data on this curve. Equations (2.11) and (2.12) were not imposed exactly, rather one Newton iteration was performed to "correct" the computed solution at each point on the "ring".

2.2.2 Integration Schemes

We begin by introducing some terminology which will be used later in the Chapter. Consider the first order initial value problem

$$\dot{y} = f(y); \quad y(0) = \alpha \tag{2.13}$$

where a dot denotes differentiation with respect to the independent variable, x, and f is a generally non-linear function of y. Consider computing discrete approximations to the solution of (2.13) on a regular grid defined by the grid points $x_i = ih$, $i = 0, 1, 2, \cdots$ where h is the step-size. The forward Euler method discretizes (2.13) by:

$$\frac{y_{i+1} - y_i}{h} = f(y_i), \quad i = 0, 1, 2, \cdots \tag{2.14}$$

with $y_0 = \alpha$ so that

$$y_{i+1} = y_i + hf(y_i). \tag{2.15}$$

Notice that (2.15) gives an explicit formula for the discrete solution at each successive grid point. Such a scheme is referred to as *explicit*.

The backward Euler method discretizes (2.13) by:

$$\frac{y_{i+1} - y_i}{h} = f(y_{i+1}), \quad i = 0, 1, 2, \cdots \tag{2.16}$$

so that

$$y_{i+1} = y_i + hf(y_{i+1}). \tag{2.17}$$

In the general case when f is a non-linear function of y, (2.17) defines y_{i+1} implicitly at each grid point. Such schemes are referred to as *implicit*. In such cases, we solve for y_{i+1} approximately using a non-linear solver such as Newton's method.

Another implicit scheme which has the advantage of being second order accurate is the midpoint scheme:

$$\frac{y_{i+1} - y_i}{h} = f((y_{i+1} + y_i)/2) \tag{2.18}$$

so that

$$y_{i+1} = y_i + hf((y_{i+1} + y_i)/2). \tag{2.19}$$

In general, explicit schemes have the disadvantage that a stability restriction must be placed on the magnitude of the step-size, h, in order that a reasonable solution be obtained. In some cases this restriction can be quite severe leading to an expensive computation. However, backward Euler and the midpoint scheme, which are implicit schemes, are stable for all values of h and are therefore referred to as unconditionally stable. For further details see Hairer, Norsett & Wanner [19] or Ortega [41].

While Horn [20] reports no improvement in accuracy when replacing a simple forward Euler scheme with an explicit fourth order Runge-Kutta scheme (see [19] [41]), there is also a question of stability which must be considered especially in the presence of noise.

Since backward Euler and the midpoint scheme are unconditionally stable we have decided to attempt a solution of equations (2.6) to (2.10) using these methods. In either case, the resulting non-linear system is solved approximately using Newton's method.

This requires that the Jacobian of the discrete system be computed. This Jacobian is given by:

$$\begin{pmatrix} -1/\Delta\xi & 0 & E_{xx} & E_{xy} & 0 \\ 0 & -1/\Delta\xi & E_{yx} & E_{yy} & 0 \\ R_{pp} & R_{pq} & -1/\Delta\xi & 0 & R_p + pR_{pp} + qR_{qp} \\ R_{qp} & R_{qq} & 0 & -1/\Delta\xi & R_q + pR_{pq} + qR_{qq} \\ 0 & 0 & 0 & 0 & -1/\Delta\xi \end{pmatrix}$$

Here $\Delta\xi$ is the step size taken along the characteristic curve and all other terms are evaluated at the previous iterate.

Notice that the second derivatives of the image brightness function $E(x, y)$ appearing in this Jacobian yield a system which will be highly susceptible to noise. This is a drawback in using an implicit scheme such as backward Euler. If an explicit scheme were used we would have to contend only with first derivatives of $E(x, y)$.

In order to kick-off the integration scheme, we require an initial strip along which x, y, p, q and z are known. Rather than assuming local sphericity of the surface in the neighbourhood of singular points we assume that z, p and q are known along the boundary of the image. The assumption of local sphericity requires an estimate of the radius of the sphere which Horn suggests can be obtained by estimating the size of the object under consideration. This procedure will clearly lead to errors in the data even before any integration is carried out. For simple surfaces which when illuminated yield only one singular point in the corresponding image, such an initial strip proves to be adequate. Second order estimates of the derivatives of the image brightness are used at all points of the domain *including* the boundary. It is important to resist the temptation of using only first order estimates of these derivatives at the boundary since any errors instilled in the solution at the beginning of each strip will propagate along the entire strip adversely affecting the solution.

We therefore traverse the border of the image and shoot characteristic strips out at regular intervals. If (R_p, R_q) is such that the characteristic strip immediately leaves the domain, we simply integrate in the reverse direction. In the language of fluid mechanics, if the boundary happens to be an outflow boundary, we integrate in the upstream direction. This procedure is quite reasonable in cases where most characterisitics are expected to terminate at a singular point, i.e. when few or no characteristics intersect the boundary of the domain more than once.

We choose the step-size in ξ to be

$$\frac{1}{2\sqrt{R_p^2 + R_q^2}N} \tag{2.20}$$

where N is the number of pixels across the image. Notice that $\sqrt{R_p^2 + R_q^2}$ represents the speed of the curve and so our step-size is chosen to be inversely proportional to this speed. With this particular choice, we expect the solution to be computed approximately every half pixel. A given characteristic is terminated when $(R_p^2 + R_q^2)$ becomes too small. Notice that this condition signals that we are close to a singular point or inside a non-illuminated region. This is equivalent to a stagnation point or no-flow region in fluid mechanics. A tolerance of 1×10^{-10} is used for the Newton iteration. If convergence to within this tolerance is not obtained within 200 iterations, the characteristic strip is terminated.

Although all characteristics are started at image grid points, successive solution points do not necessarily coincide with grid points. In such cases, bi-cubic interpolation is used to determine the image brightness and its first and second derivatives at these points.

2.3 Results: Simple Integration Schemes

Unless otherwise noted, all results reported in this chapter are for the midpoint rule. We choose the midpoint rule over simple backward Euler since the former is expected to yield

more accurate results. Our schemes are tested by computing and reporting errors in the computed surface height and the reconstructed image brightness. The error in surface height is a root mean square (r.m.s.) error given by:

$$z_{err} = \left(\frac{\sum_{i=1}^{N} (z_i - ze(x_i, y_i))^2}{N} \right)^{1/2} \tag{2.21}$$

where z_i, $(1 \leq i \leq N)$, are the computed surface heights at points (x_i, y_i) on the characteristic strips and $ze(x_i, y_i)$ is the corresponding exact solution. The image brightness error is also an r.m.s. error given by:

$$ib_{err} = \left(\frac{\sum_{i=1}^{N} (R(p_i, q_i) - \tilde{E}_i)^2}{N} \right)^{1/2} \tag{2.22}$$

where (p_i, q_i), $(1 \leq i \leq N)$, are the computed surface gradients at points (x_i, y_i) on the characteristic strips and $\tilde{E}_i$ is the interpolation of the image data from regular grid points to the point (x_i, y_i). We use bi-cubic interpolation to achieve this.

We consistently use a Lambertian reflectance map in all tests so that

$$R(p, q) = \max \left\{ 0, \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2}} \right\} \tag{2.23}$$

The coordinate system used to display the images is consistent throughout the Chapter and is given in Figure 2.1.

2.3.1 Images with no noise

Consider the simple surface $z = x^2 + y^2$ on $[-0.5, 0.5]^2$. An image of this surface will have at most one singular point. If the singular point lies in the field of view, characteristics will propagate from the boundary towards this point.

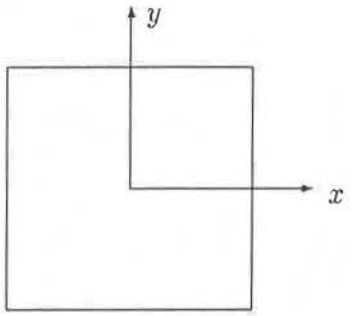


Figure 2.1: Coordinate system for displayed images

First consider the case of overhead illumination which results in a singular point at the origin of the corresponding image. Since characteristic strips follow paths of steepest descent on the surface, (see Oliensis [37], [38]) the tracks of these strips, when projected onto the image plane, form straight lines from the boundary to the origin. The algorithm discussed in the previous section was applied to a 129×129 image. The tracks of the characteristics in the image plane are shown in Figure 2.2. The error in computed surface height was 1.489×10^{-5} with an image brightness error of 1.235×10^{-3}. The small error in the image brightness demonstrates that the image irradiance equation $E(x, y) = R(p, q)$, which underlies the system of equations we are considering, has been solved well.

The computed characteristics indeed appear to follow straight lines from the boundary to the singular point. With a maximum height of 0.5, the error in the computed surface height is around 0.2%. However, notice that the surface height is not computed uniformly throughout the domain despite the initial even distribution of characteristic strips. This problem is inherent in the solution technique.

We now illuminate the paraboloid from the direction $(-0.5, -0.5, 1)$. This yields one singular point in the resulting image. The computed tracks are shown in Figure 2.3. The

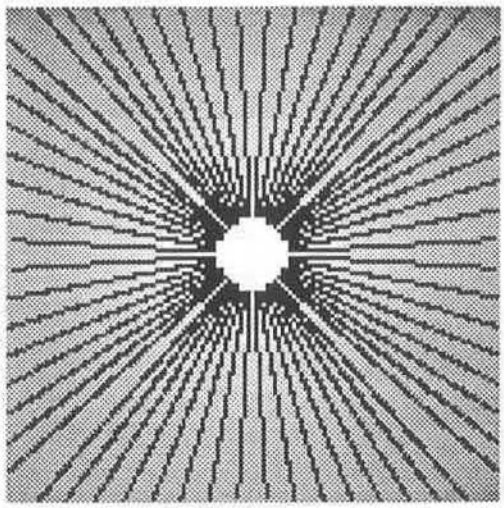


Figure 2.2: Paraboloid, overhead illumination.

error in the computed surface height was still excellent at 1.717×10^{-5} while the image brightness error was 1.752×10^{-3}. Again, our algorithm has performed quite adequately.

We next consider a more challenging example. Consider the surface $z = -\frac{1}{2\pi}\cos(2\pi r)$, $r = \sqrt{x^2 + y^2}$ on $[-0.5, 0.5]^2$ which somewhat resembles an inverted Mexican hat. We start by illuminating this surface from overhead. The resulting image has a singular point at the origin as well as a ring of singular points at $r = 0.5$. The computed tracks are shown in Figure 2.4.

As expected, all the characteristics have terminated near the ring of singular points at $r = 0.5$ leaving a very large portion of the surface untouched. In order to compute the solution in the region $r < 0.5$, we would have to make some assumption about the shape of the surface either just inside the ring $r = 0.5$ or on a closed curve surrounding the singular point at the origin. This, of course, would be in addition to the information we have already used on the boundary of the image. A major drawback in using the method of characteristics is therefore revealed – there will be cases where knowledge of the solution around *one* closed curve in the image is not sufficient to compute the solution

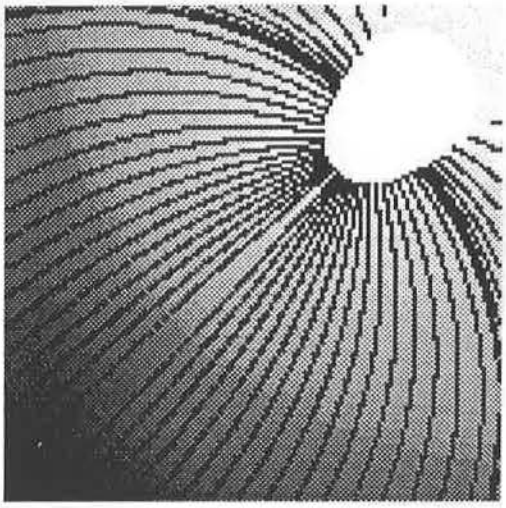


Figure 2.3: Paraboloid, light source direction $(-0.5, -0.5, 1)$.

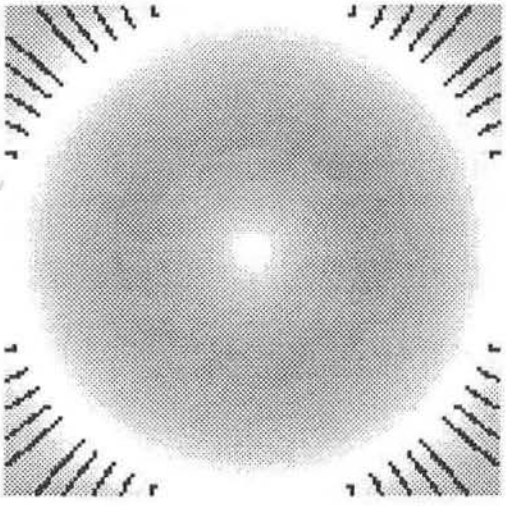


Figure 2.4: Mexican hat, overhead illumination.

Radius	z-Error	IB-Error
0.15	6.185×10^{-2}	4.510×10^{-2}
0.20	1.256×10^{-2}	1.673×10^{-2}
0.25	2.971×10^{-2}	4.164×10^{-2}
0.50	5.240×10^{-2}	7.405×10^{-2}

Table 2.1: Effect of radius of sphere on computed solution

in all regions of the domain.

If we assume knowledge of the exact surface height and gradient along a closed curve surrounding the singular point at the origin our algorithm then computes the solution over the region $r < 0.5$ with an error in surface height of 2.162×10^{-4} and an image brightness error of 2.585×10^{-3}. Given that the range in surface heights is approximately 0.3, we can be quite satisfied with such a small error. However, such initial data may not always be available. We next consider how the assumption of local sphericity near the singular point at the origin affects the computed solution. In order to compute the error in surface height we assume that the height of the sphere that we fit to the surface is coincident with the exact height of the surface at the origin.

We must then estimate an appropriate value for the radius of the sphere. For the surface under consideration, there are few features in the image that give any clue as to an appropriate choice for this variable. The ring of singularities at r = 0.5 would indicate that the radius should have a maximum value of 0.5. We show in Table 2.1 the results of our algorithm with various choices for the radius of the sphere. The columns labelled z-Error and IB-Error are the errors in the computed surface height and image brightness, respectively.

As can be seen, the results are at best nearly an order of magnitude worse than the case when the exact data is given. Cross-sections through the computed surfaces as well

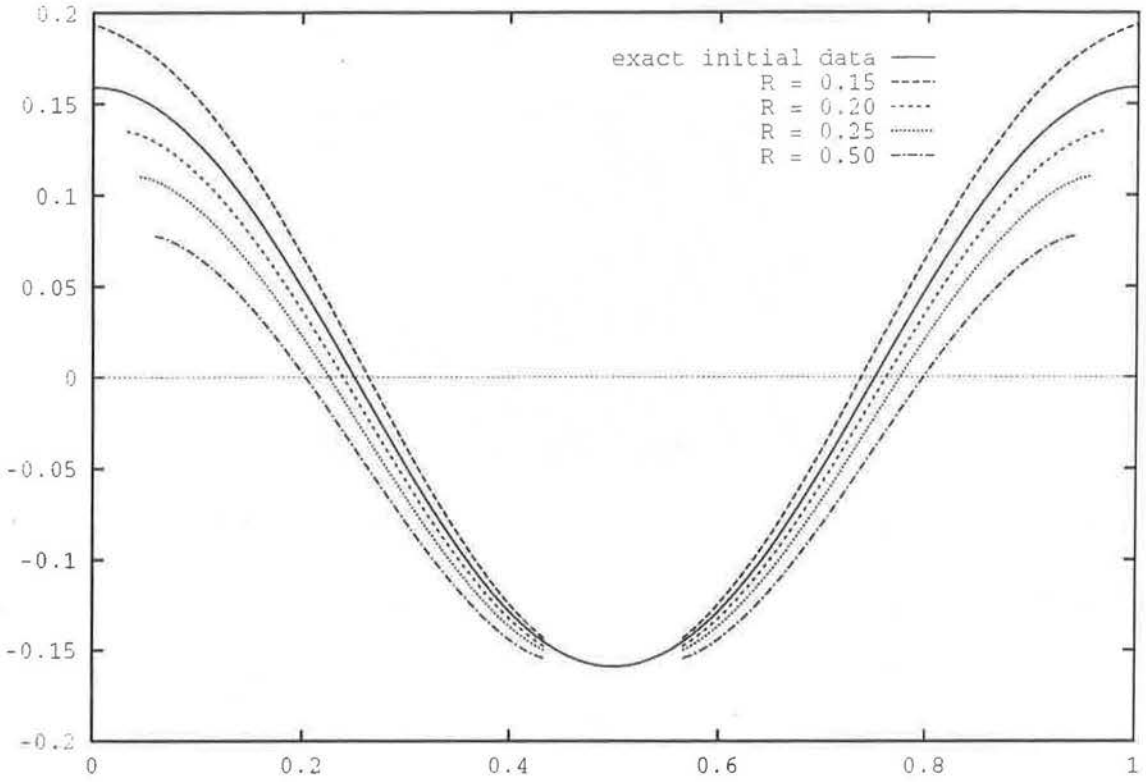


Figure 2.5: Cross-sections through computed solutions for different estimates of radius of sphere

as the exact solution are shown in Figure 2.5. The computed solutions are qualitatively reasonable in that they maintain the correct shape. However, given the input image, it is unlikely that a value of 0.2 (for which we obtained the best results out of those tested) would have been chosen and we conclude that quantitatively, these results may not be satisfactory.

We now illuminate this surface from the direction $(-1, 0, 1)$. The resulting image has only one singular point which is located at $(0.25, 0)$. The path of characteristics across the image is now rather complicated. As pointed out by Oliensis [38], if we now align the z-axis with the light source (rather than the viewer) the characteristics will

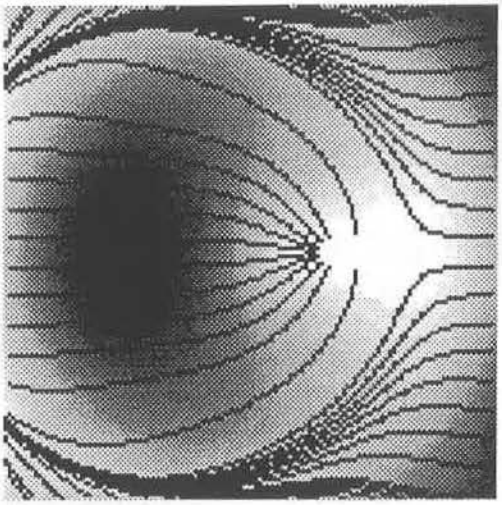


Figure 2.6: Mexican hat, paths of characteristics computed from exact surface slopes.

follow paths of steepest descent in this new co-ordinate system. The point on the surface corresponding to the singular point is neither a local maximum nor minimum but rather a saddle point. Therefore some characteristics will flow towards this point while others emanate from it.

In order to get an idea of the paths that characteristics are expected to follow we solve the system

$$\begin{aligned} \dot{x} &= R_p(p_e, q_e) \\ \dot{y} &= R_q(p_e, q_e) \end{aligned} \tag{2.24}$$

where p_e and q_e represent the exact surface slopes. We solved this system in a manner similar to that described for the system in Section 2.2.2, except that forward Euler was used to integrate the equations. The results are shown in Figure 2.6.

Notice that characteristics shot from the top and bottom of the image do not propagate far into the domain and could be omitted without any great loss. The result of propagating characteristics from only the left and right sides of the image is shown in Figure 2.7. This result is much more easily interpreted.

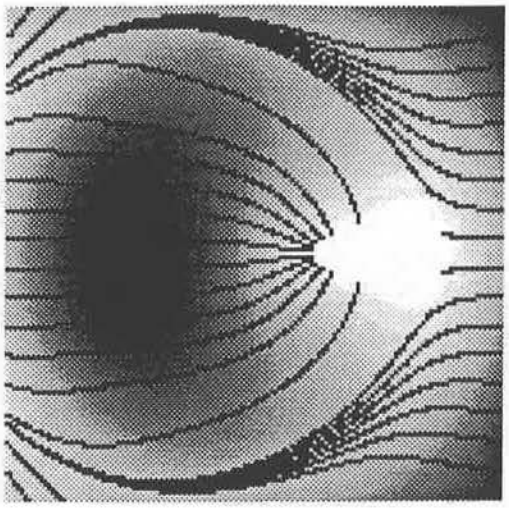


Figure 2.7: Mexican hat, paths of reduced number of characteristics computed using exact surface slopes.

In the remaining tests conducted on this surface, we will continue to propagate characteristics from the left and right sides of the image only. The result obtained from our algorithm is shown in Figure 2.8. The error in the computed surface height was 2.506×10^{-4} while the image brightness error was 3.187×10^{-3}.

For the most part the characteristics have followed the expected path. However, in the top left and bottom left corners, the characteristics have strayed slightly off course and have even crossed in some places resulting in a non-unique solution. This suggests that the underlying image irradiance equation has not been solved as well. This phenomenon is known as "drift". Although on the continuous level, the system we are solving has a solution which is guaranteed to be a solution of the image irradiance equation, on the discrete level this is not the case due to the accumulation of errors caused by discretization. Later in this chapter we will examine a technique for dealing with drift.

At this point we also present the result of using backward Euler as opposed to the midpoint rule. Note that backward Euler is only a first order scheme. The error in surface height was much worse at 1.297×10^{-2} while the image brightness error was similar at

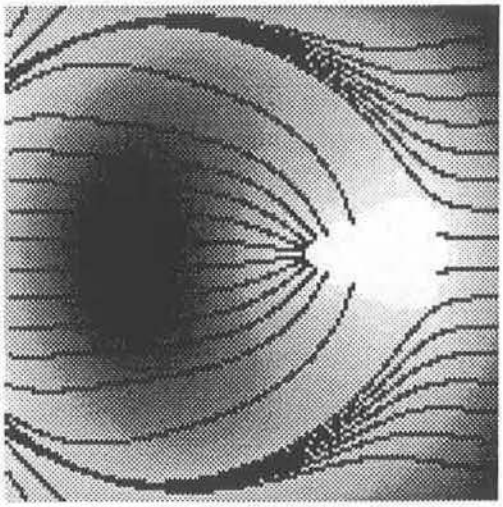


Figure 2.8: Mexican hat, computed paths of characteristics.

7.177×10^{-3}. The tracks of the computed characteristics are shown in Figure 2.9. Clearly, this first order scheme yields results which are not satisfactory in comparison to those of the midpoint scheme. This deterioration in the computed solution when using backward Euler was observed for all tests presented in this chapter. However, from now on we will only report the results of using the midpoint scheme.

2.3.2 Effects of noise in the image

We now consider the effects of noise in the image data. Since errors accumulate during integration along the characteristic strips, this method tends to be very susceptible to noise. We begin by considering the effects of digitization. Images are frequently stored in the format of one byte per pixel so that image brightness is represented as an integer between 0 and 255. However, in all the images presented so far, image brightness has been computed and stored as a double precision variable. In order to model the effects of digitization we perform the following operations on the image brightness during the image creation routine:

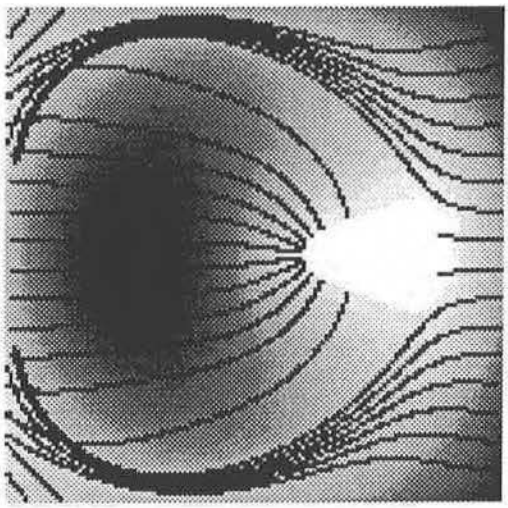


Figure 2.9: Mexican hat, computed using backward Euler.

```
brightness = R(p_exact(x,y), q_exact(x,y));
    temp   = (unsigned char) (brightness * 255 + 0.5);
    E[i][j] = (double) temp / 255;
```

Here we assume that the reflectance map, $R(\cdot,\cdot)$, returns a value between 0 and 1.

With this addition to the code we now perform some of the tests carried out in the previous section. First, consider the surface $z = x^2 + y^2$ illuminated from overhead. The tracks are shown in Figure 2.10. The error in the computed surface height was 1.537×10^{-1} while the image brightness error was 2.590×10^{-3}. The results clearly demonstrate the effects of this rather basic form of noise in the image. Characteristic strips have crossed at various points in the image yielding non-unique results.

The errors in the surface height and image brightness have increased by two orders of magnitude over the case with no noise. This very large increase in the error is due to the fact that some of the characteristics, rather than terminating at the singular point, skirt around it and end up wandering out of the domain. Some of these characteristics are shown in Figure 2.11. If these strips are omitted from the computation, the resulting

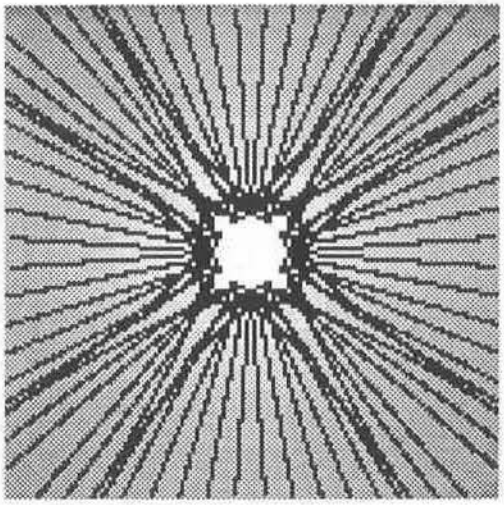


Figure 2.10: Paraboloid with digitization error, overhead illumination.

error in surface height is much lower at 1.013×10^{-3}. This figure is not that much larger than the case when there is no noise in the image.

This result indicates that the error in the computation could be controlled to some extent by shooting characteristics out in parallel and terminating certain characteristics when they become too close together, as suggested by Horn [20]. There is a danger in this procedure, since when two characteristics cross, it is not known which of them is in error and it is therefore possible that a "good" characteristic may be terminated in favour of a "bad" one. If both are in error, then one will survive to go on polluting the solution.

We also test the "Mexican hat" surface illuminated from $(-1, 0, 1)$. The effects on this surface are not quite as serious as for the previous example although some deterioration is clearly noticeable. The tracks are shown in Figure 2.12. The errors in the surface height and image brightness were 1.614×10^{-2} and 3.776×10^{-3}, respectively.

In this case, there were no characteristics that wandered seriously off course and so the very large deterioration in the solution that occured in the previous test was not

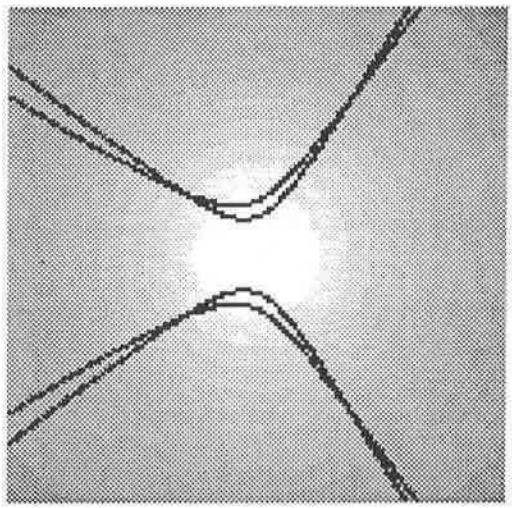


Figure 2.11: Paraboloid with digitization error, overhead illumination. Tracks of some of the characteristics which wander out of domain.

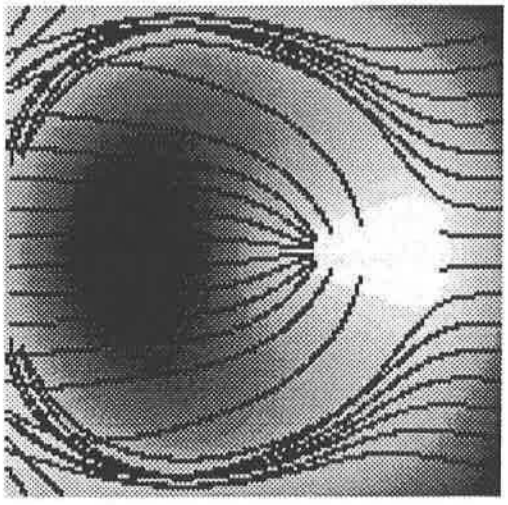


Figure 2.12: Mexican hat with digitization error, light source direction $(-1, 0, 1)$.

observed. However, the effects of noise are clearly noticeable.

2.4 The Method of Projected Invariants

In Section 2.2.1 we saw how the image irradiance equation

$$g(v) = E(x,y) - R(p,q) = 0, \qquad v = (x,y,z,p,q) \tag{2.25}$$

is equivalent to a system of 5 non-linear o.d.e.s which we will now write as

$$\dot{v} = f(v). \tag{2.26}$$

This non-linear system was solved using Newton's method which necessitated the computation of the Jacobian of $f(v)$ which in turn involved the second derivatives of the image brightness function. For this reason, any noise in the image is greatly magnified in this Jacobian which contributes to the errors observed in Section 2.3.2.

However, system (2.26) has an invariant, namely equation (2.25). On the continuous level, any solution of system (2.26) is guaranteed to be a solution of equation (2.25). For the discrete system this is not necessarily so. As mentioned earlier, this phenomenon is known as "drift". We therefore consider projecting the discrete solution on to the manifold defined by (2.25). We achieve this by adding λ times the gradient of (2.25) to the right hand side of system (2.26) and augmenting this system with equation (2.25). Here, λ is an additional variable, somewhat like a Lagrange multipler, which is determined so that equation (2.25) is satisfied by the resulting solution. We thus end up with the following system:

$$\left.\begin{aligned} \dot{v} &= f(v) + \lambda \nabla g(v) \\ 0 &= g(v) \end{aligned}\right\} \tag{2.27}$$

We now expect that errors in the image irradiance equation will be reduced. Notice, however, that this will not necessarily lead to an improvement in the estimate of the

surface height. This is because z does not figure directly in the image irradiance equation. While we may end up computing x, y, p and q such that the image irradiance equation is well satisfied, this will not necessarily be achieved in a manner in which p and q are integrable. Any deviation from integrability will result in errors in the surface height. There is, of course, also discretization error involved in computing z from p and q.

Notice that this procedure is akin to Horn's method for "sharpening" his estimates of p and q. However, Horn's method consisted of a post-computation, correction technique while the method of projected invariants provides a more integrated approach.

2.5 Results: The Method of Projected Invariants

2.5.1 Images with no noise

Again consider the paraboloid $z = x^2 + y^2$ as introduced earlier. In the case of overhead illumination, the errors in surface height and image brightness were 2.224×10^{-5} and 1.157×10^{-14}, respectively. We measure the image brightness error at the midpoint since this is the point where the image irradiance equation (the invariant) is imposed. While there is only a very slight improvement in the computed surface height there is a very obvious improvement in the image brightness error which indicates that the method of projected invariants is indeed ensuring that the solution stays on the manifold defined by the image irradiance equation. Similar results are observed when the light source direction is $(-0.5, -0.5, 1)$. In this case the error in surface height was 8.987×10^{-6} while the image brightness error was 1.742×10^{-13}. In both cases the tracks of characteristics in the image plane are not noticeably different from those depicted in Figures 2.2 and 2.3.

For the "Mexican hat" we again set the illumination direction at $(-1, 0, 1)$. The computed surface height in this case was noticeably improved with an error of 9.871×10^{-5} while the image brightness error was also very small at 1.719×10^{-12}. In this case there

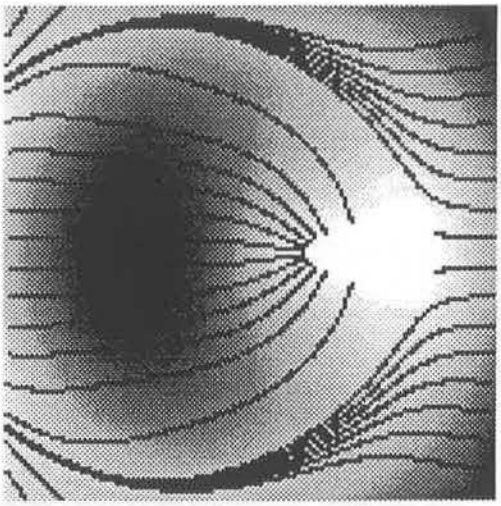


Figure 2.13: Mexican hat, light source direction $(-1, 0, 1)$.

is a noticeable difference in tracks followed by the characteristics. These are shown in Figure 2.13. Notice that characteristics now appear not to cross as compared to Figure 2.8.

2.5.2 Images with noise

We now consider the effects of digitization error as in Section 2.3.2. For the paraboloid illuminated from overhead, the errors in the computed surface height and image brightness were both increased. Convergence difficulties were experienced for some characteristics. Characteristics were terminated if the Newton iteration did not converge within a tolerance of 1×10^{-10} after 200 iterations. Portions of those characteristics for which convergence was obtained are shown in Figure 2.14. For these characteristics, the error in the computed surface height was 2.910×10^{-3} while the image brightness error was 1.010×10^{-12}. It is clear that digitization error has still had a noticeable effect on the solution. Characteristics still have a tendency to cross although the situation is not as bad as in Section 2.3.2 where some characteristics, rather than terminating at the singular

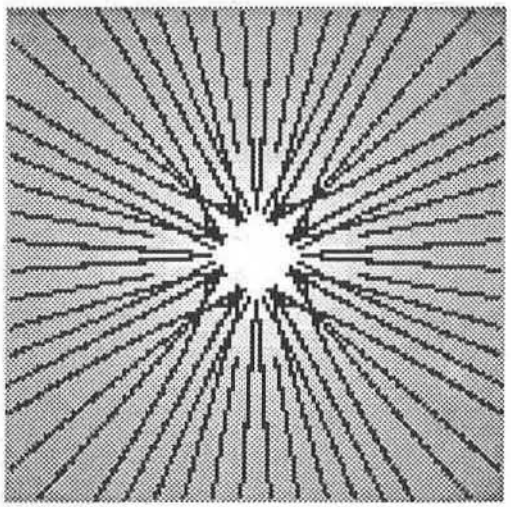


Figure 2.14: Paraboloid with digitization error, overhead illumination.

point, wandered out of the domain causing large errors in the solution.

For the "Mexican hat" illuminated from $(-1, 0, 1)$ we again observe an increase in the error in computed surface height and image brightness error (as compared to the previous section where there was no noise in the image data) which were measured at 5.214×10^{-3} and 1.982×10^{-12}. The result is shown in Figure 2.15. A slight difference in the tracks of some characteristics can be observed as compared to Figure 2.13 but on the whole the method of projected invariants appears to have reduced the effects of digitization error.

2.6 Discussion

Although the effects of digitization error have been reduced by the method of projected invariants, it is clear that there are still many problems. The test cases we have considered here are very simple and as will be seen shortly, can be handled very easily by variational techniques. The development of an automated system that will work on images with an arbitrary number of singular points would involve considerable effort which the results

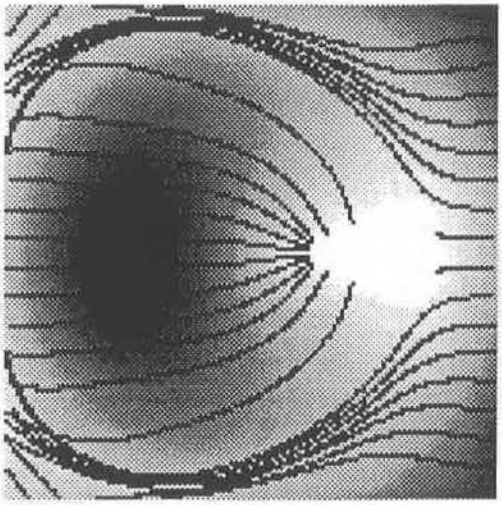


Figure 2.15: Mexican hat with digitization error, light source direction $(-1, 0, 1)$.

of this chapter do not justify. We feel that our results demonstrate that the method of characteristic strips is not appropriate for the problem in hand, even for very simple images.

Chapter 3

Variational Techniques

3.1 Methods

We begin this chapter with a brief overview of the calculus of variations, introducing terminology that is required in order to understand later sections. We also give an introduction to multigrid methods which will be used as solvers for the discrete problem arising from the variational calculus.

3.1.1 Variational Calculus

The calculus of variations concerns itself with determining an extremum of a functional, f, of one or more dependent variables. Consider the problem of determining $z(x, y)$ such that

$$f = \int\int F(x, y, z, z_x, z_y) dA \tag{3.1}$$

is minimized on some domain Ω. Here a subscript denotes partial differentiation with respect to the indicated variable. A necessary condition for $\tilde{z}$ to be an extremum of (3.1) is that the so called first variation $f(x, y, \tilde{z} + \epsilon\eta)$ of f about $\tilde{z}$ be zero, that is

$$\left.\frac{\partial f}{\partial \epsilon}\right|_{\epsilon=0} = 0 \tag{3.2}$$

where $\eta = \eta(x, y) \in H_0^2(\Omega)$ is an essentially arbitrary, smooth function satisfying only the property that it is zero on $\partial\Omega$ and ϵ is arbitrarily small. It is shown in Courant & Hilbert [15] that this condition amounts to requiring that $\tilde{z}$ be a solution of the partial

differential equation:

$$F_z - \frac{\partial}{\partial x} F_{z_x} - \frac{\partial}{\partial y} F_{z_y} = 0 \tag{3.3}$$

under one of the following boundary conditions

(a) $z = z_0(x, y)$ on $\partial\Omega$ where $z_0(x, y)$ is specified.

(b) $F_{z_x} \frac{dy}{ds} - F_{z_y} \frac{dx}{ds} = 0$ on $\partial\Omega$ where s is arclength measured along $\partial\Omega$ from some fixed point.

Here, F_{z_x} denotes the partial derivative of F with respect to $\partial z/\partial x$. Equation (3.3) is known as an Euler equation. Boundary conditions of type (a) are referred to as *essential* and those of type (b) are called *natural.* Boundary conditions which specify the solution on the boundary, as in type (a), are often referred to as *Dirichlet* while those which specify the normal derivative of the solution across the boundary are referred to as *Neumann.*

The above ideas can be generalized to cases where the integrand in equation (3.1) involves a larger number of both dependent and independent variables. For more details see Courant & Hilbert [15].

The existence of natural boundary conditions for this type of problem is of importance in situations where known boundary data is not available. For the problem considered above, it could well be that no condition on $z(x, y)$ is known on $\partial\Omega$ in which case the natural boundary conditions supplied by the variational calculus can be employed.

A simple illustration of these techniques can be found in the problem of determining the height of a surface $z(x, y)$, $(x, y) \in \Omega$ from known surface slopes $p(x, y)$ and $q(x, y)$. We may consider an acceptable solution to this problem as being that function $\tilde{z}(x, y)$ which minimizes the functional

$$f(x, y, z) = \int\int [(z_x - p)^2 + (z_y - q)^2] dx\, dy. \tag{3.4}$$

A necessary (and in this case, sufficient) condition that $\tilde{z}(x,y)$ be a global minimum of $f(x,y,z)$ is that it satisfies the partial differential equation

$$\nabla^2 z = p_x + q_y \tag{3.5}$$

under either of the following boundary conditions

(a) essential : z is specified on $\partial\Omega$

(b) natural : $\partial z/\partial n = \mathbf{n}\cdot(p,q)$ where $\mathbf{n}$ is the outward unit normal to $\partial\Omega$.

Notice that (3.5) with (a) has a unique solution whereas with (b) $\tilde{z}$ is determined only up to an additive constant. This observation is important from the computational viewpoint since this degree of freedom in the solution must be handled by the chosen numerical method. Such considerations turn out to be important in many shape from shading schemes.

3.1.2 Multigrid Methods

The multigrid method has proved to be a fast, efficient solver for elliptic partial differential equations. An excellent introduction may be found in Briggs [6] while more in-depth discussions may be found in Brandt [4] and Hackbusch [18]. Here we give a brief overview introducing terminology that is necessary for understanding later implementations of multigrid methods to the problem in hand.

Consider the solution of

$$\mathcal{N}u = 0 \text{ on } \Omega \tag{3.6}$$

where $\mathcal{N}$ is a non-linear, elliptic differential operator, and

$$u = 0 \text{ on } \partial\Omega. \tag{3.7}$$

Define a grid Ω_h on which a discrete approximation to u is to be computed. Then, u_h, the exact solution of the discrete problem, satisfies:

$$\mathcal{N}_h u_h = 0 \text{ on } \Omega_h \tag{3.8}$$

and

$$u_h = 0 \text{ on } \partial\Omega_h \tag{3.9}$$

where $\mathcal{N}_h$ is a discretization of the differential operator $\mathcal{N}$. The discrete problem (3.8) may be regarded as a large, sparse system. The solution of such a system by direct methods tends to be very inefficient due to fill-in during the decomposition stage. Iterative methods such as Gauss-Seidel tend to be more efficient in terms of storage requirements but suffer from poor asymptotic convergence and therefore tend to be very slow. The multigrid method, however, is a fast, efficient solver for such a system.

The multigrid method may be viewed as a defect-correction method. Given an approximation U_h to u_h, we define a defect, d_h, by:

$$d_h = 0 - \mathcal{N}_h U_h \tag{3.10}$$

and a correction, v_h, satisfying:

$$\mathcal{N}_h(U_h + v_h) = d_h + \mathcal{N}_h U_h \tag{3.11}$$

or

$$\mathcal{N}_h w_h = d_h + \mathcal{N}_h U_h \tag{3.12}$$

where $w_h = U_h + v_h$ and

$$v_h = 0 \text{ on } \partial\Omega. \tag{3.13}$$

Having computed d_h and solved *exactly* for w_h, our corrected solution becomes:

$$u_h := w_h \quad (= U_h + v_h). \tag{3.14}$$

However, the solution of (3.12) is computationally no easier than the solution of the original problem. At this point we propose to construct a hierarchy of successively coarser grids on which the defect on the previous finer grid may be represented. Here, for simplicity, we confine ourselves to only two grids, a fine grid Ω_h and a coarse one Ω_H where, for example, $H = 2h$ and $\Omega_H \subset \Omega_h$. With a coarser grid defined, we propose to solve (3.12) on this coarser grid rather than on the fine one as this will be computationally cheaper. However, the coarse grid cannot compensate for the high frequency components of the error in the solution which can only be seen on the fine grid. Hence, the coarse grid correction is complemented by relaxation iterations on the fine grid which serve to smooth high frequency components of the error. Simple iterative methods such as Gauss-Seidel while having poor asymptotic convergence rates are very efficient smoothers and may be used in the relaxation component in the algorithm.

The basic two-level method consists of the following four components:

1. **Relaxation** : generally consists of ν sweeps of a simple iterative method which serves to update the solution and smooth out high frequency components of the error in the solution. It is generally applied before and after the coarse grid correction.

2. **Restriction** : computes the defect on the fine grid and transfers it to the coarse grid using an appropriate restriction operator, I_h^H.

3. **Solve** : an exact solver for the problem on the coarse grid.

4. **Prolongation** : transfers the correction, v_H, from the coarse grid to the fine grid using an appropriate interpolation operator, I_H^h.

Multigrid algorithms consisting of more than two grids are based on a recursive application of the above two-level method, see Brandt [4] and Hackbusch [18].

In some classical multigrid methods, the iteration commences on the finest grid. If the two-grid method described above is called once recursively on each coarser level, the multigrid cycle is referred to as a $V(\nu_1, \nu_2)$ cycle. Here, ν_1 and ν_2 are the number of relaxation iterations used before and after each coarse grid correction, respectively. If the two-grid method is called twice recursively from each coarser level, the multigrid cycle is referred to as a $W(\nu_1, \nu_2)$ cycle.

One further variant, known as full multigrid or FMG, starts the iteration on the coarsest grid. Having solved the problem exactly on this grid, the solution is interpolated to the next finer grid using a high-order interpolation (typically bi-cubic if the number of grid points is sufficiently high). On each successively finer level, a number of multigrid V or W cycles is performed before once again interpolating to the next finer level and so on.

3.2 Application to Horn's Height and Gradient from Shading Scheme

Recall that in [22], we seek to determine functions z, p and q such that the functional

$$\int\int_{\Omega} \{\lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu((z_x - p)^2 + (z_y - q)^2) + (E(x,y) - R(p,q))^2\} d\Omega \quad (3.15)$$

is minimized. The corresponding Euler equations are

$$\left.\begin{aligned} \lambda\nabla^2 p + \mu(z_x - p) + (E - R)R_p &= 0 \\ \lambda\nabla^2 q + \mu(z_y - q) + (E - R)R_q &= 0 \\ \nabla^2 z - p_x - q_y &= 0 \end{aligned}\right\} \quad (3.16)$$

with natural boundary conditions given by:

$$\frac{\partial p}{\partial n} = \frac{\partial q}{\partial n} = 0 \quad (3.17)$$

$$\frac{\partial z}{\partial n} = \mathbf{n} \cdot (p, q) \quad (3.18)$$

Notice that the non-elliptic integrability terms are of a lower order than the elliptic smoothing terms allowing us to take λ small without losing ellipticity. However, for $\lambda = 0$ this system is non-elliptic.

Horn discretized these equations on staggered grids, the grid for z being offset from the one for p and q by a half cell. His discretization is such that given a red-black checkerboard colouring of the grid for z, the computation on the red grid is entirely independent of that on the black grid, assuming natural boundary conditions as specified in [22]. There are therefore two degrees of freedom in the discrete system whereas the underlying continuous system has only one. The solution of Horn's scheme therefore requires that z is known at one point on each of the red and black grids.

The approach we have chosen is to discretize the functional (3.15) directly rather than the Euler equations (3.16). Our approach is a finite volume method which in fact leads quite naturally to a standard second order discretization of the Euler equations on *non-staggered* grids. Let Ω^h be a discrete approximation of the domain and partition this domain using square cells Ω^h_{ij}, $(1 \leq i, j \leq N)$ each having side h and vertices as shown in Figure 3.1.

The functional

$$\int\int_{\Omega_{ij}} \{\lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu[(z_x - p)^2 + (z_y - q)^2] + (E - R(p,q))^2\} dx dy \qquad (3.19)$$

is approximated by

$$\begin{aligned}
& \frac{\lambda}{2h^2}[(p_{i,j-1} - p_{i-1,j-1})^2 + (p_{i,j} - p_{i-1,j})^2 + (p_{i-1,j} - p_{i-1,j-1})^2 + (p_{i,j} - p_{i,j-1})^2 \\
+ \; & (q_{i,j-1} - q_{i-1,j-1})^2 + (q_{i,j} - q_{i-1,j})^2 + (q_{i-1,j} - q_{i-1,j-1})^2 + (q_{i,j} - q_{i,j-1})^2] \\
+ \; & \frac{\mu}{2}[(h^{-1}(z_{i,j-1} - z_{i-1,j-1}) - (p_{i,j-1} + p_{i-1,j-1})/2)^2 + (h^{-1}(z_{i,j} - z_{i-1,j}) \\
- \; & (p_{i,j} + p_{i-1,j})/2)^2 + (h^{-1}(z_{i-1,j} - z_{i-1,j-1}) - (q_{i-1,j} + q_{i-1,j-1})/2)^2 \\
+ \; & (h^{-1}(z_{i,j} - z_{i,j-1}) - (q_{i,j} + q_{i,j-1})/2)^2]
\end{aligned}$$

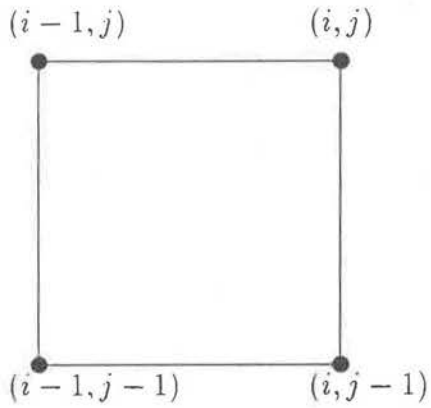


Figure 3.1: Discrete cell.

$$+ \quad \frac{1}{4}[(E_{i-1,j-1} - R(p_{i-1,j-1}, q_{i-1,j-1}))^2 + (E_{i,j-1} - R(p_{i,j-1}, q_{i,j-1}))^2$$
$$+ \quad (E_{i-1,j} - R(p_{i-1,j}, q_{i-1,j}))^2 + (E_{i,j} - R(p_{i,j}, q_{i,j}))^2] =: F_{ij} \tag{3.20}$$

We then consider minimizing

$$F := \sum_{i,j} F_{ij} \tag{3.21}$$

A necessary condition for a local minimum of (3.21) is that the partial derivative of F with respect to each of the unknowns $z_{i,j}$, $p_{i,j}$, $q_{i,j}$ $(0 \leq i,j \leq N)$ be zero.

As mentioned earlier, this leads in a natural way to a second order discretization of (3.16) on non-staggered grids. In cases where Dirichlet boundary data is available, such information can be incorporated into the above scheme by fixing the unknowns on the boundary. When boundary data is not available in any form (as is often the case for the problem under consideration), the above process provides natural boundary conditions which are equivalent to (3.17) and (3.18).

At this point, we note that the resulting discrete system is not h-elliptic, see Brandt & Dinar [5], in the limit as $\lambda \to 0$: h-ellipticity is a measure of the ellipticity of the discrete system. However, as we will see later, it is possible to obtain convergence for very small

values of λ although a noticeable slow–down in the convergence rate of the iteration is observed as λ is decreased. Horn's scheme on staggered grids also loses h-ellipticity as $\lambda \to 0$ but he also reports a succesful implementation of a pointwise relaxation scheme even for $\lambda = 0$, see [22].

Also worthy of note is that a very minor modification to our discrete scheme, namely, replacing

$$\left(\frac{z_{i+1,j} - z_{i-1,j}}{2h} - \frac{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}}{4}\right) \tag{3.22}$$

by

$$\left(\frac{z_{i+1,j} - z_{i-1,j}}{2h} - p_{i,j}\right) \tag{3.23}$$

(and similarly for the discrete estimate of $(z_y - q)$) results in a scheme which *is* h-elliptic in the limit as $\lambda \to 0$. However, experiment has shown that in spite of this, the modified scheme does not perform any better. It therefore appears that the measure of discrete ellipticity is not particularly relevant to these schemes.

Horn employed a continuation algorithm in λ as part of his iterative scheme. The initial value of λ was taken to be 1.0 and this was slowly reduced, even to zero in some cases. Here we employ an FMG continuation algorithm in λ as demonstrated in Figure 3.2. On the coarsest level we take a value of λ which is large enough to ensure a good smoothing rate. A hierarchy of finer, square grids is set up with the usual coarse to fine grid ratio of 2. As we proceed to each new finer grid, λ is reduced by a factor of 4 so that $\bar{\lambda} = \lambda/h_f^2$ is constant where h_f is the step-size on the currently finest grid. For each level we employ one FAS W(2,2) cycle with λ fixed (see [18], [4]). The nonlinear term $(E - R(p,q))^2$ is handled by replacing $R(p,q)$ with its Taylor expansion to first order about the previous iterate (pointwise). This is equivalent to performing one pointwise Gauss-Newton iteration. The usual full-weighting is used for the restriction operator with its adjoint for the prolongation operator. Bi-cubic interpolation is used to transfer

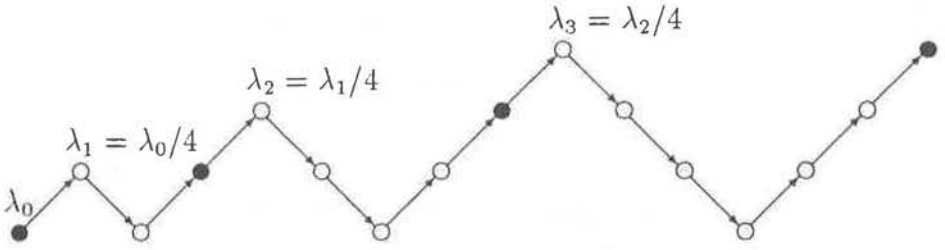


Figure 3.2: FMG continuation algorithm.

the solution to a new finest level in the FMG algorithm. Once the finest level is reached, 5 W(2,2) cycles are performed to end the process.

Notice that within each $W(2,2)$ cycle we keep λ fixed. Thus, for values of $\lambda \sim 1$ for which the discrete scheme is safely elliptic, we expect to achieve good convergence. However, such values of λ cause unwanted smoothing of the solution. It is therefore necessary to consider much smaller values of λ, especially for a noise-free image which is illuminated everywhere (so that information is available at all points in the image). The FMG continuation algorithm in λ then turns out to be essential in order to obtain convergence on the finest level in a reasonable number of iterations (and in some cases, to obtain convergence at all).

The parameter μ is not varied at all during the iteration. This is in contrast to Horn who reduces μ as the algorithm proceeds (although μ cannot sensibly be taken to zero). For a small value of λ we must be sure not to take μ too large (in which case we come close to imposing integrability) nor too small (since the discrete operator becomes singular in the limit as $\lambda, \mu \to 0$). Recall that schemes which have attempted to impose integrability exactly have been shown not to work well [24].

3.3 Results: Single image as input

We now present results of tests applied to various synthetic images. Throughout the Chapter we report errors in the computed surface height and gradient when the exact solutions are known. These errors are root mean square errors and are computed at each grid point in the computational domain unless otherwise noted. We also report residual errors. This is a measure of how well the discrete equations have been solved. Consider the equation

$$\mathcal{N}^h u^h = f^h \tag{3.24}$$

where $\mathcal{N}^h$ is a discrete differential operator, u^h denotes the discrete solution and f^h is a known function at each grid point. We define the root mean square residual error to be

$$r = \left(\frac{\sum_{j=1}^{n} \sum_{i=1}^{n} (\mathcal{N}_{ij}^h u^h - f_{ij}^h)^2}{n^2} \right)^{1/2} \tag{3.25}$$

In the case of systems of equations, we compute the r.m.s. error for each equation and then report the maximum of these errors.

We consistently use a Lambertian reflectance map so that

$$R(p,q) = \max \left\{ 0, \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2}} \right\} \tag{3.26}$$

3.3.1 Effect of regularization parameters on performance of algorithm

In this section we attempt to determine the effects of the regularization parameters, λ and μ, on both the computed solution and the algorithm's convergence rate.

For this purpose it is important to choose a fairly simple image as the test case so that defects in performance or in the computed solution can be blamed only on the algorithm

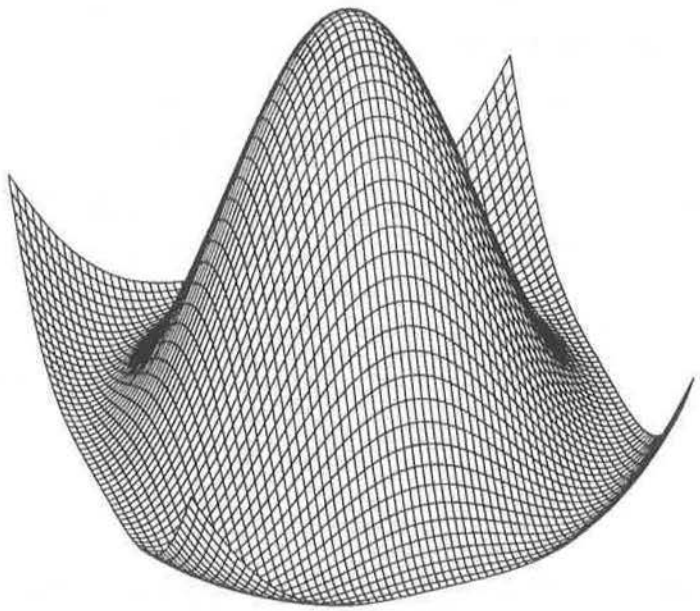


Figure 3.3: Exact Solution: "Mexican Hat".

and not on the model or the complexity of the surface we are attempting to recover. To this end we use the surface

$$z = \frac{\cos(2\pi r)}{2\pi}, \quad r = \sqrt{x^2 + y^2} \tag{3.27}$$

on the domain $[-0.5, 0.5]^2$ as in Chapter 2. We illuminate this surface using a single light source from the direction $(0, -1, 1)$ so that the resulting image has only one singular point (at $(0, -0.25)$) and is illuminated everywhere except at the single point $(0, 0.25)$. The exact surface, resembling a Mexican hat, is shown in Figure 3.3 and the input image is shown in Figure 3.4.

The co–ordinate systems used to display results are consistent throughout the chapter and are demonstrated in Figure 3.5. It should be noted that these co–ordinate systems differ from those used in the computation.

We begin by examining the effect of the integrability parameter μ. Since λ multiplies

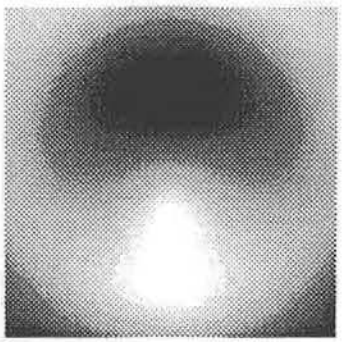


Figure 3.4: Input Image.

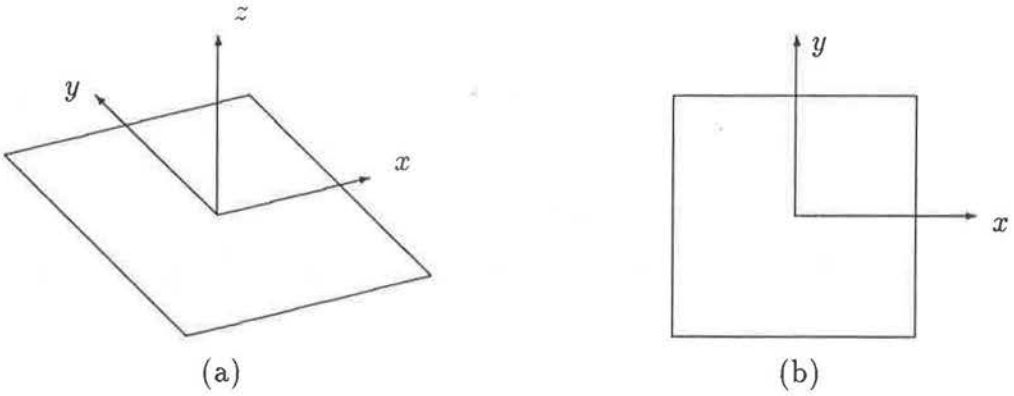


Figure 3.5: Co–ordinate systems : (a) Surfaces, (b) Images.

the elliptic terms and μ the non-elliptic terms in the system, we expect that the convergence rate will depend on the ratio λ/μ. Note, however, that this is true only to the point where the non-linear terms do not interfere with the convergence rate. For a fixed value of λ we therefore expect the convergence rate to increase as μ decreases. However, as we decrease μ we also express less preference for integrability and so some deterioration in the solution may be expected. Our tests have been conducted on a 129×129 image with different boundary conditions for the variables p, q and z. We keep $\bar{\lambda}$ fixed at 0.4. Recall that $\bar{\lambda} = \lambda/h_f^2$ where h_f is the step-size on the currently finest grid. The results are shown in Table 3.1. The column labelled "B.C.'s" gives the boundary conditions used for the gradient (p, q) and the height z, respectively (Dir = Dirichlet, Nat = Natural). The column labelled ρ is the average convergence rate observed on the finest level. The convergence rate at each iteration is the factor by which the residual error is reduced. The column labelled "Residual" is the final residual error after 5 W(2,2) cycles on the finest grid (we report only the largest of the three residuals for the system). The columns labelled "IB Error", "(p, q) Error" and "z Error" are the image brightness error and errors in computed gradient and surface height, respectively.

As can be seen, when Dirichlet boundary conditions are used for the gradient, our results bear out the theory. For large values of μ the convergence rate is poorer, however, the computed solution is more accurate. As μ is decreased, the convergence rate improves but at the expense of accuracy in the computed solution. However, when natural boundary conditions are used for the gradient, the performance of the algorithm is seriously affected. In many cases, convergence is not observed after 5W(2,2) multigrid cycles on the finest level.

We next examine the effect of the regularization parameter, λ, that controls smoothing. We now expect that as this parameter is reduced, the convergence rate will slow down with an increase in the accuracy of the computed solution. In light of the results

B.C.'s	μ	ρ	Residual	IB Error	(p,q) Error	z Error
Dir, Dir	1.0	2.49	7.850×10^{-4}	2.791×10^{-3}	9.355×10^{-3}	2.116×10^{-3}
	0.1	13.42	5.401×10^{-7}	2.677×10^{-3}	2.601×10^{-2}	5.726×10^{-3}
	0.01	32.00	3.135×10^{-8}	2.963×10^{-3}	1.236×10^{-1}	2.730×10^{-2}
Dir, Nat	1.0	1.67	2.735×10^{-3}	4.163×10^{-3}	4.858×10^{-2}	2.384×10^{-2}
	0.1	5.85	2.224×10^{-5}	4.119×10^{-3}	1.110×10^{-1}	5.491×10^{-2}
	0.01	9.96	2.027×10^{-6}	3.686×10^{-3}	2.308×10^{-1}	1.161×10^{-1}
Nat, Dir	1.0	2.51	9.901×10^{-4}	4.136×10^{-3}	2.164×10^{-2}	2.671×10^{-3}
	0.1	9.27	3.064×10^{-6}	3.394×10^{-3}	4.324×10^{-2}	5.985×10^{-3}
	0.01	-	-	-	-	-
Nat, Nat	1.0	1.53	4.912×10^{-3}	6.880×10^{-3}	1.595×10^{-1}	6.178×10^{-2}
	0.1	-	-	-	-	-
	0.01	-	-	-	-	-

Table 3.1: Effect of μ on performance of algorithm.

shown in Table 3.1 we choose to keep μ fixed at 0.1 so as to achieve a balance between accuracy and good convergence.

The results may be found in Table 3.2. When Dirichlet boundary conditions are used for both the gradient and height, our results are as expected. The same can be said for the second set of results for which we have Dirichlet conditions on the gradient and natural boundary conditions for the height. A noticeable improvement in the accuracy of the computed solution is observed as $\bar{\lambda}$ is decreased but this is at the expense of a slow down in the convergence rate.

Rather spurious results are again observed when natural boundary conditions are used for the gradient. In the first set of results (where we have Dirichlet conditions for the surface height) we do not achieve convergence when $\bar{\lambda} = 4.0$. This is contrary to the behaviour predicted by the theory which suggests that the convergence rate should improve with increasing values of $\bar{\lambda}$. When $\bar{\lambda}$ is reduced to 0.4 and 0.04 we in fact observe the expected convergence behaviour.

B.C.'s	$\bar{\lambda}$	ρ	Residual	IB Error	(p,q) Error	z Error
Dir. Dir	4.0	33.23	3.117×10^{-8}	1.900×10^{-2}	1.513×10^{-1}	3.632×10^{-2}
	0.4	13.42	5.401×10^{-7}	2.677×10^{-3}	2.601×10^{-2}	5.726×10^{-3}
	0.04	3.04	2.562×10^{-4}	2.919×10^{-4}	3.303×10^{-3}	6.615×10^{-4}
Dir. Nat	4.0	9.94	1.551×10^{-6}	1.865×10^{-2}	2.610×10^{-1}	1.370×10^{-1}
	0.4	5.85	2.224×10^{-5}	4.119×10^{-3}	1.110×10^{-1}	5.491×10^{-2}
	0.04	1.83	2.929×10^{-3}	6.338×10^{-4}	2.960×10^{-2}	1.412×10^{-2}
Nat, Dir	4.0	-	-	-	-	-
	0.4	9.27	3.064×10^{-6}	3.394×10^{-3}	4.324×10^{-2}	5.985×10^{-3}
	0.04	2.66	5.166×10^{-4}	5.030×10^{-4}	1.042×10^{-2}	1.052×10^{-3}
Nat, Nat	4.0	-	-	-	-	-
	0.4	-	-	-	-	-
	0.04	1.68	7.248×10^{-3}	1.202×10^{-3}	1.155×10^{-1}	2.738×10^{-2}

Table 3.2: Effect of $\bar{\lambda}$ on performance of algorithm.

In the last set of results where natural boundary conditions are employed for all unknowns, we again fail to observe convergence for larger values of $\bar{\lambda}$. A plausible explanation is that the curious results observed when natural boundary conditions are used for the gradient result from the fact that when $\bar{\lambda}/\mu$ is large, the natural boundary conditions

$$\frac{\partial p}{\partial n} = \frac{\partial q}{\partial n} = 0 \tag{3.28}$$

conflict with the image irradiance equation $E(x,y) = R(p,q)$ causing poor convergence.

We present the computed surface for the most accurate test (i.e. Dirichlet conditions on gradient and height, $\bar{\lambda} = 0.04$ and $\mu = 0.1$) in Figure 3.6. It is also of interest to use the computed gradient to produce an image. We illuminate the surface from both the original light source direction and from a direction orthogonal (in gradient space) to the original light source direction. These images can be found in Figure 3.7. As can be seen the solution is in excellent agreement with the original surface. There is also excellent agreement between Figure 3.7a and the original input image (Figure 3.4) demonstrating

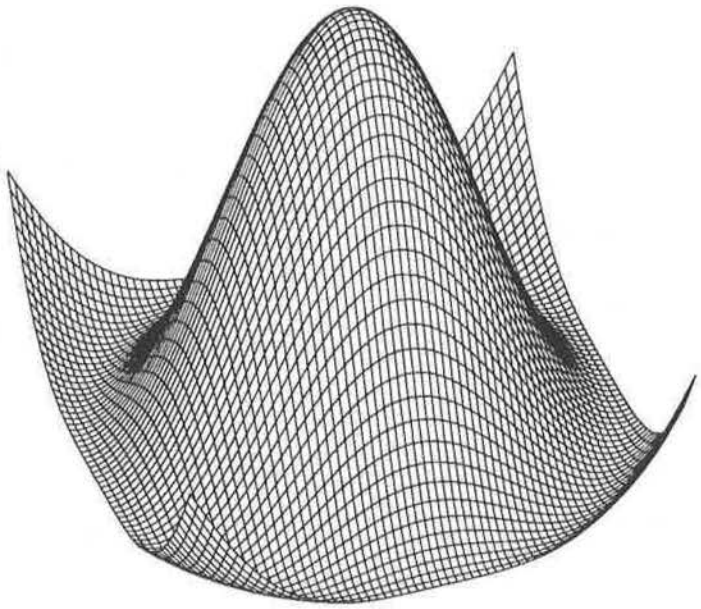
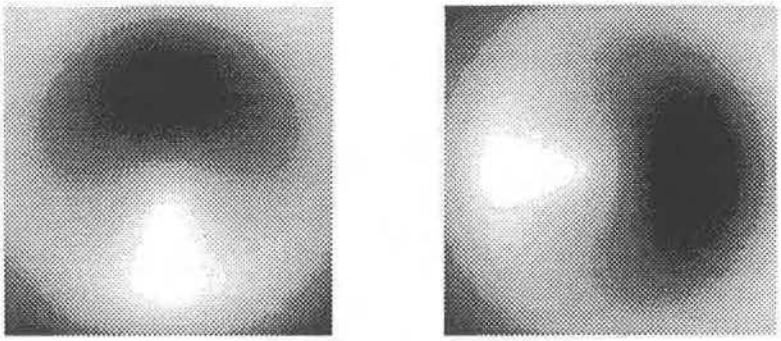


Figure 3.6: Computed Surface: Mexican Hat, $\lambda = 0.01$, $\mu = 0.1$.

that the image irradiance equation has been solved well. The importance of examining the computed surface illuminated from an independent light source direction, as in Figure 3.7b, will be seen later.

3.3.2 Synthetic Hemisphere

We next test our algorithm on an image of a synthetic hemisphere lying on a flat plane. This test is significant in that the entire range of possible gradients (i.e. $(-\infty, \infty)^2$) is covered on a hemisphere, providing a challenge to the algorithm. Also of note is the presence of a discontinuity in surface orientation at the boundary of the hemisphere. We assume that the flat side of the hemisphere is in contact with the plane which forms the background so that there is no discontinuity in surface height. One further difficulty is that we will illuminate the sphere obliquely so that there is a region in the image which is not illuminated and thus contains limited information as to surface orientation in that region. In fact, in such a region we can only deduce that the gradient (p, q) satisfies the



(a) (b)

Figure 3.7: Images of computed surface: using original light source direction (left), using orthogonal light source direction (right).

relation

$$1 + p_0 p + q_0 q < 0 \tag{3.29}$$

where (p_0, q_0) specifies the light source direction. Hence, this test contains several difficulties not encountered in tests conducted previously.

The hemisphere is of radius $1/3$ centred in the region $[-0.5, 0.5]^2$. The resulting image is of size 129x129 and is formed by illuminating the hemisphere from the direction $(0, 1, 1)$. The exact solution and input image are given in Figures 3.8 and 3.9 respectively.

In light of the results of the previous sections, we take $\mu = 0.1$ and $\bar{\lambda} = 0.4$ in the hope of obtaining a good convergence rate while still obtaining reasonable accuracy. The observed convergence rate was 10.5 resulting in a final residual error of 1.414×10^{-5}. The error in surface height was 8.505×10^{-2}. Given that the maximum height in the exact surface is $0.3\dot{3}$, this represents an error of approximately 25%. The computed surface is found in Figure 3.10. Images have been produced using the computed gradient illuminated from both the same direction as the input image and from a direction orthogonal to the input image. These images are found in Figure 3.11.

It is clear from Figure 3.11(a) that the image irradiance equation has been solved

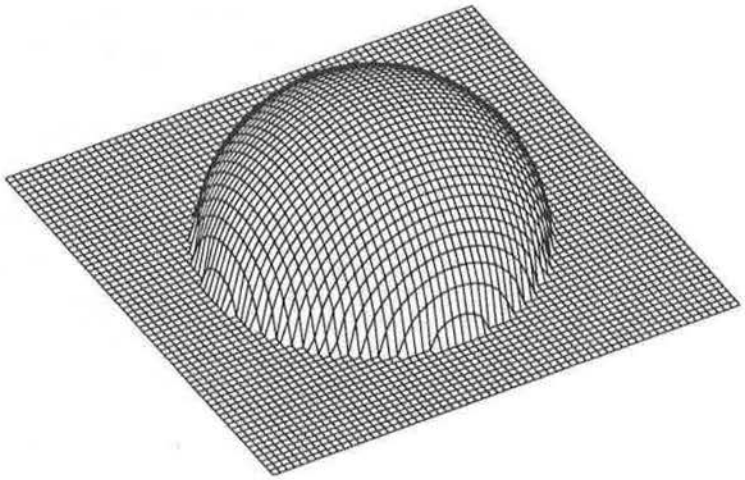


Figure 3.8: Exact solution: Hemisphere.

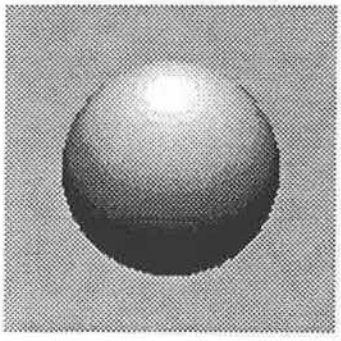


Figure 3.9: Input image.

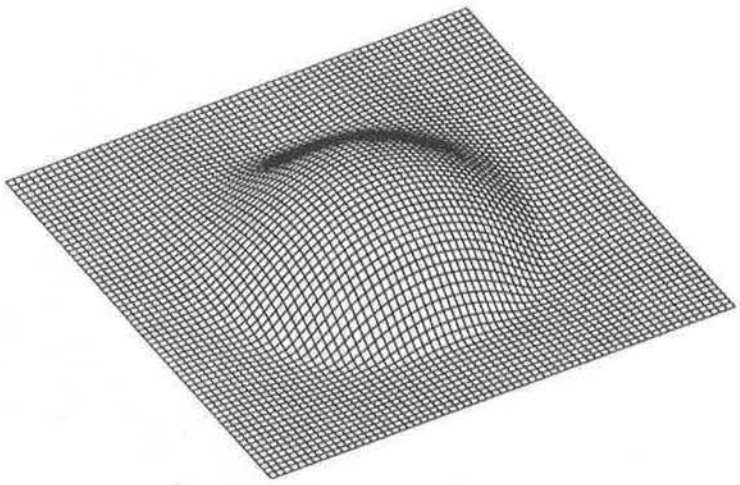
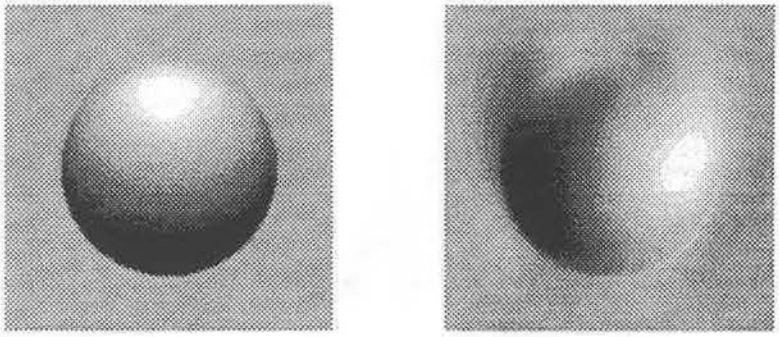


Figure 3.10: Computed surface: Hemisphere from single image.



(a) (b)

Figure 3.11: Images of computed surface: using original light source direction (left), using orthogonal light source direction (right).

reasonably well, there being little difference between the reconstructed image and the input image (except for some smoothing at the boundary of the hemisphere). However, Figure 3.11(b) and Figure 3.10 show that the computed surface in fact contains serious flaws. The fact that the image irradiance equation can be solved so satisfactorily while serious errors in the computed surface are evident is indicative of the ill-posed nature of the problem. Recall also the limited success of projection onto the image irradiance equation reported in the previous chapter. However, we must not forget the problems inherent in this particular test, i.e. the presence of a discontinuity in surface orientation and regions in the image where there is little information due to non–illumination.

3.3.3 Synthetic Vase

We now consider a more interesting image which contains all the problems of the previous test but has the added difficulty that the curve in the image representing the boundary of the object is a little more irregular. We have created a synthetic vase by taking a 6th order polynomial and rotating it to produce a surface of revolution. The exact surface and a 129x129 image of that surface illuminated from the direction $(0, 1, 1)$ are found in Figures 3.12 and 3.13, respectively.

We test our algorithm with parameters $\bar{\lambda} = 0.4$, $\mu = 0.1$. The observed convergence rate of 7.05 was not quite as good as for the synthetic hemisphere but sufficient to reduce the final residual to 7.137×10^{-5}. The error in the computed surface height was 7.106×10^{-2} which when compared to the maximum height of 0.2855 yields an error of approximately 25%, equivalent to that observed for the synthetic hemisphere. Again, our expectations for good results cannot be too high due to the presence of a discontinuity (of fairly irregular shape) and regions where the surface is not illuminated. The computed surface is shown in Figure 3.14 and reconstructed images using the computed gradient illuminated from the same direction as the input image and a direction orthogonal to it

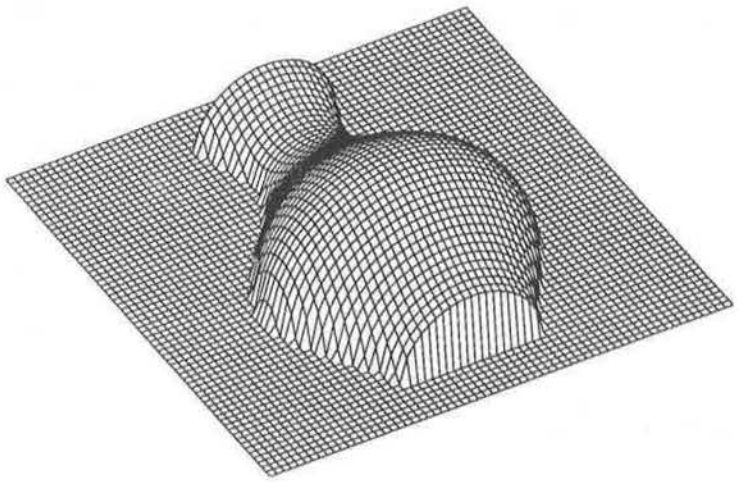


Figure 3.12: Exact solution: Vase.

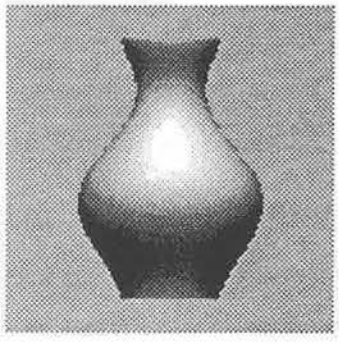


Figure 3.13: Input image.

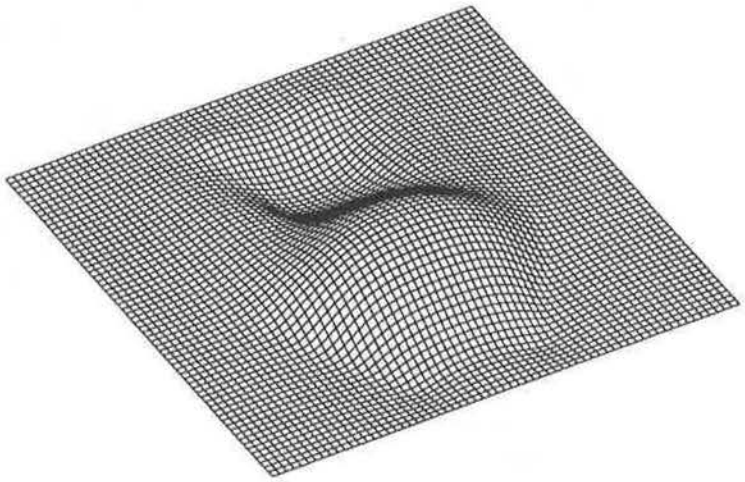
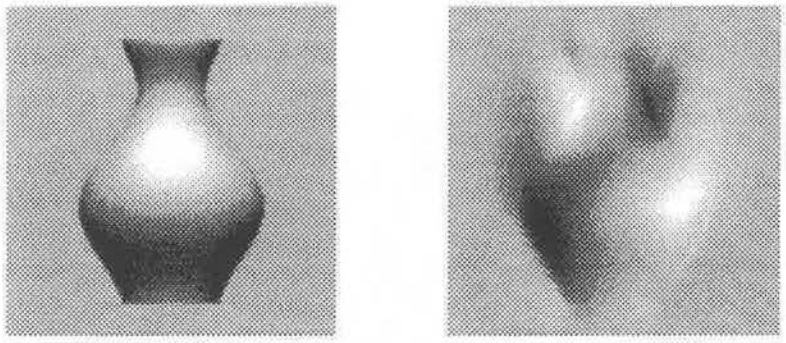


Figure 3.14: Computed surface: Vase from single image.

are found in Figure 3.15. Similar comments to those presented in the previous section are in order; the image irradiance equation has been solved quite well as evidenced by Figure 3.15a, however, there are fairly serious errors in the computed surface.

3.4 Comparison with Horn's scheme

We now compare our algorithm against that proposed by Horn [22]. Since Horn suggests that several thousand iterations must be performed even for a 65 x 65 image, we now restrict ourselves to this smaller sized image for expediency. In keeping with our earlier tests we hold μ fixed at 0.1, even though in the one example presented by Horn [22], μ is reduced somewhat from its initial value. Since tests that we conducted earlier indicate that reducing μ leads to less accurate results, we suggest that holding μ fixed at 0.1 will not adversely affect Horn's algorithm. Recall that Horn uses Dirichlet boundary conditions on (p, q) and natural conditions on z in most of his tests. We do the same here for both Horn's scheme and our multigrid scheme.



(a) (b)

Figure 3.15: Images of computed surface: using original light source direction (left), using orthogonal light source direction (right).

We begin with the "Mexican hat" surface $z = \cos(2\pi r)$, $r = \sqrt{x^2 + y^2}$ (Figure 3.3) as considered earlier. For Horn's scheme we started with $\lambda = 1$ and this was slowly reduced to 1×10^{-5}. The total number of iterations performed was 13,312. We followed a routine for reducing λ that closely followed the one presented by Horn. The final residual was 1.301×10^{-3} while the error in the computed surface height was 3.276×10^{-2}.

We then performed the test on our multigrid scheme. We took $\lambda = 0.01$ on the coarsest grid so that on the finest level we had $\lambda \simeq 1 \times 10^{-5}$ as for the test on Horn's scheme. After 5 W(2,2) cycles on the finest grid, the residual error was 2.277×10^{-3} and the error in the computed surface height was 3.014×10^{-2}. These figures are quite comparable to those produced by Horn's algorithm.

The major difference between these two computations is the time taken to perform them. The 13,312 iterations taken by Horn's scheme took in excess of 4.5 hours on a 33MHz 80486DX machine. However, the 5 W(2,2) cycles performed by our multigrid scheme took approximately 5 minutes! On a finer grid our algorithm is expected to yield even larger savings as compared to [22] (cf. [18]).

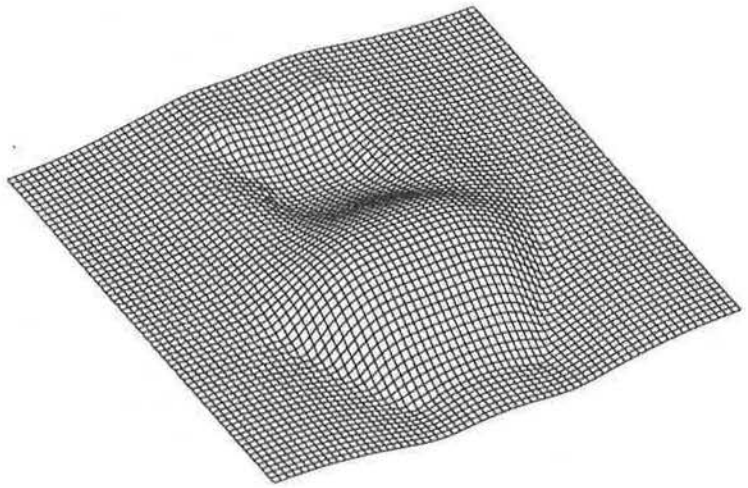


Figure 3.16: Computed Surface: Vase, Horn's scheme.

We also compared the performance of the two algorithms on an image of the synthetic vase with similar results. The same values of λ and μ were used as for the test considered above. Again after 13,312 iterations, the residual error for Horn's scheme was 4.907×10^{-5} and the error in surface height was 1.458×10^{-1}. It was necessary to perform 10 W(2,2) cycles on the finest grid of our multigrid algorithm (which took approximately 8 minutes to compute) in order to reduce the residual error to 1.035×10^{-4} for which the error in surface height was 1.457×10^{-1}. The computed surfaces are shown in Figures 3.16 and 3.17. The results are again quite comparable, the only major difference being computation time.

Close examination of Figures 3.16 and 3.17 shows that the surface computed by Horn's scheme is not as smooth as that computed by our multigrid scheme. Shape from shading schemes are typically subject to a certain degree of shearing in the solution orthogonal to the light source direction. As the computation on Horn's red and black grids are independent, as discussed earlier, we believe that this shearing will occur to differing

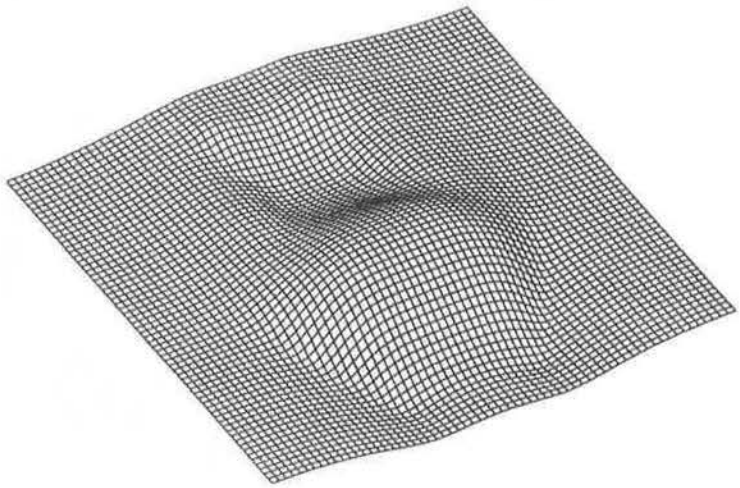


Figure 3.17: Computed Surface: Vase, Multigrid scheme.

degrees on the red and black grids causing the roughness that appears in Figure 3.16.

Also worthy of note at this point is the choice of boundary conditions. Since Horn uses staggered grids, the imposition of Dirichlet boundary conditions on all variables requires that the gradient be known along the boundary of the image but that the surface height be known along a curve offset from the boundary of the image by half a pixel. Such information would not likely be readily available. However, since our multigrid algorithm uses non-staggered grids, knowledge of the gradient along the boundary of the image leads to knowledge of the surface height along that curve since we can simply integrate the tangential component of the gradient along the boundary in order to retrieve the height. This is important since we have shown earlier that more accurate results are obtained when Dirichlet conditions are used for all variables. It can therefore be argued that given the same information (i.e. the gradient along the boundary of the image) our algorithm can potentially retrieve the solution more accurately.

3.5 Multiple Images

In light of the difficulties experienced in obtaining a good solution from a single image, especially in the presence of discontinuities, and when using natural (or Neumann) boundary conditions, we now consider the case when two (or more) images are available. It is assumed that these images are taken from the same viewpoint so that only the light source direction is changed. We modify Horn's variational formulation accordingly so as to avail ourselves of this additional information. We therefore consider minimizing

$$\int\int \{\lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu[(p - z_x)^2 + (q - z_y)^2] + \sum_{i=1}^{l}(E_i - R_i(p,q))^2\}\, d\Omega \quad (3.30)$$

where l is the number of images available. Note that it is not necessary to consider more than three images since it has been shown (Woodham, [55]) that for a Lambertian reflectance map, this number is sufficient to determine the solution uniquely provided the three illumination directions are not colinear and that there are no non-illuminated regions in any of the three images. In fact Onn & Bruckstein [40] have shown that, except for certain degenerate cases, integrability disambiguates the solution arising from only two images.

We begin by repeating some of the tests found in Tables 3.1 and 3.2. We repeat only those tests for which difficulties were experienced, that is, those for which natural boundary conditions are used for the gradient. We now take two images of size 129×129 as input for the algorithm illuminated from the directions $(1, 0, 1)$ and $(0, 1, 1)$. The results are found in Tables 3.3 and 3.4.

As can easily be seen, the convergence rate in these cases is greatly improved over the cases where only one image is given as input. This is likely due to the fact that the non-linear terms, which drive the iteration when λ is small, converge more quickly given the added information contained in the second image. However, for large values of $\bar{\lambda}$, or

B.C.'s	μ	ρ	Residual	IB Error	(p,q) Error	z Error
Nat. Dir	1.0	4.259	1.447×10^{-4}	1.575×10^{-3}	4.313×10^{-3}	8.282×10^{-4}
	0.1	15.40	4.383×10^{-7}	1.737×10^{-3}	1.055×10^{-2}	1.837×10^{-3}
	0.01	1.16	1.290×10^{-2}	2.763×10^{-3}	4.062×10^{-2}	6.046×10^{-3}
Nat. Nat	1.0	3.76	9.256×10^{-5}	2.371×10^{-3}	1.973×10^{-2}	4.499×10^{-3}
	0.1	7.98	4.986×10^{-6}	2.569×10^{-3}	2.645×10^{-2}	8.486×10^{-3}
	0.01	7.88	5.516×10^{-6}	3.246×10^{-3}	6.191×10^{-2}	2.191×10^{-2}

Table 3.3: As for Table 3.1 but with two images input

B.C.'s	$\bar{\lambda}$	ρ	Residual	IB Error	(p,q) Error	z Error
Nat. Dir	4.0	14.62	6.096×10^{-7}	1.470×10^{-2}	7.199×10^{-2}	8.62×10^{-3}
	0.4	15.40	4.383×10^{-7}	1.737×10^{-3}	1.055×10^{-2}	1.837×10^{-3}
	0.04	6.93	8.406×10^{-6}	1.644×10^{-4}	1.133×10^{-3}	2.202×10^{-4}
Nat. Nat	4.0	9.28	2.682×10^{-6}	1.703×10^{-2}	1.023×10^{-1}	4.537×10^{-2}
	0.4	7.98	4.986×10^{-6}	2.569×10^{-3}	2.645×10^{-2}	8.486×10^{-3}
	0.04	5.49	8.052×10^{-6}	3.326×10^{-4}	1.343×10^{-2}	1.151×10^{-3}

Table 3.4: As for Table 3.2 but with two images input

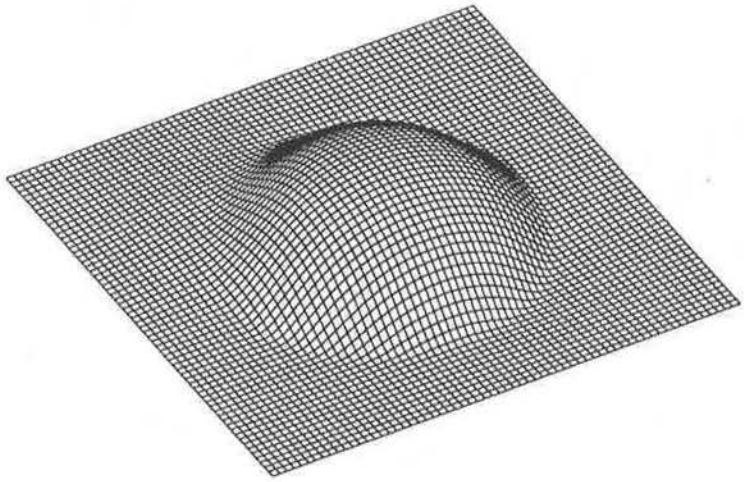


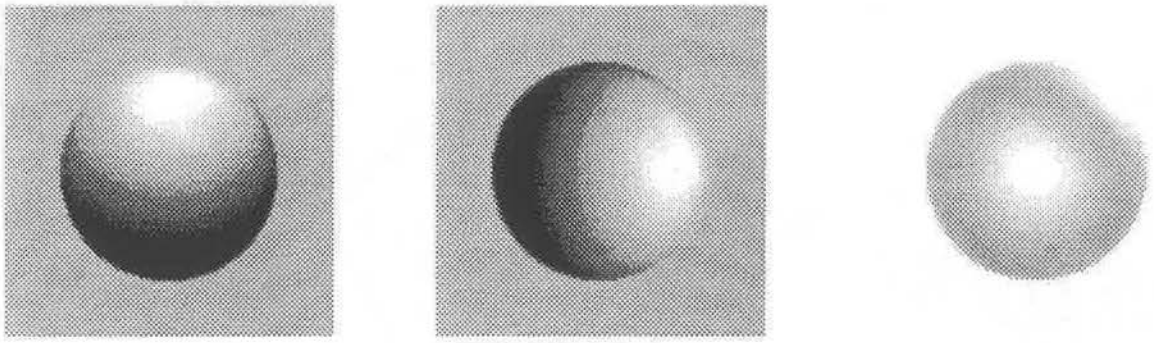
Figure 3.18: Computed Surface: Hemisphere from two images.

small values of μ, convergence is poor. Again this is likely due to the conflict between the natural boundary conditions and the image irradiance equation.

3.5.1 Synthetic Hemisphere and Vase

We now reproduce the results of Sections 3.3.2 and 3.3.3 that were carried out on the synthetic hemisphere and synthetic vase except that we allow two images as input. We illuminate the hemisphere from the directions $(0, 1, 1)$ and $(1, 0, 1)$, all other parameters are as in Section 3.3.2. The computed surface is found in Figure 3.18 while images generated from the computed gradient are found in Figure 3.19. Since we now have two images as input, we use an overhead light source as the independent light source direction.

The convergence rate was found to be good at 12.51, however, there was not a significant improvement in the error in surface height which was 7.980×10^{-2}. As can be seen from the computed surface and reconstructed images, the ridges which were present in



(a) (b) (c)

Figure 3.19: Images of computed surface: using original light source directions (left & centre), using overhead light source direction (right).

earlier results when only one image was used as input are no longer present. This leads to a result which is visually more pleasing.

Similar results were found for the synthetic vase. The light source directions used for the input images were $(0.5, 1, 1)$ and $(-0.5, 1, 1)$. The convergence rate was 12.81 and the error in the computed surface height was 6.250×10^{-2}. The computed surface and reconstructed images are found in Figures 3.20 and 3.21, respectively.

3.6 Discontinuities

It is clear from the previous section that even with information from two images, any discontinuities present in the surface are severely smoothed in the computed solution. However, given the presence of the smoothing terms for non-zero λ and indeed the manner in which z is computed from p and q (i.e. through the equation $\nabla^2 z = p_x + q_y$) we must expect a degree of smoothing across discontinuities. We therefore consider supplying information about the location of the latter. We assume knowledge of such information from preprocessing of the images by an edge detection scheme, see Marr [36] and Torre & Poggio [52]. In cases considered here where we have an object against a background,

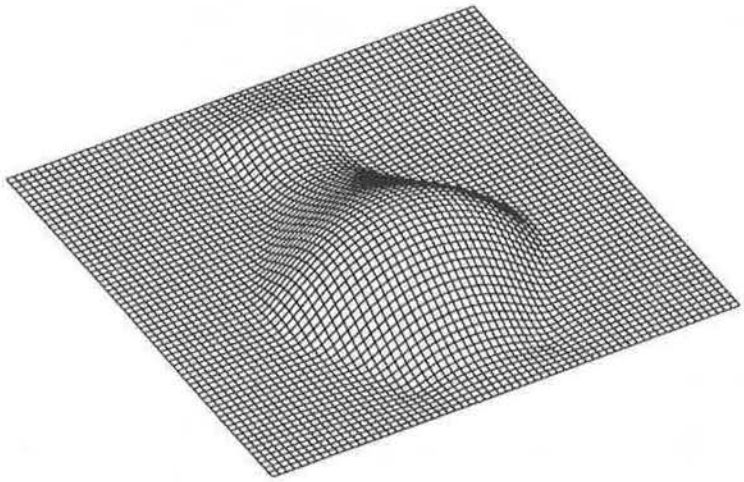


Figure 3.20: Computed Surface: Vase from two images.

(a) (b) (c)

Figure 3.21: Images of computed surface: using original light source directions (left & centre), using overhead light source direction (right).

Figure 3.22: Sample silhouette file: Vase.

knowledge of discontinuities is equivalent to knowledge of where the object is located (assuming that the object itself is smooth). This information is provided to the algorithm by way of a silhouette file, an example of which is shown in Figure 3.22.

We can then detect the presence of a discontinuity between neighbouring grid points by noting a change in the silhouette file between those points. In this manner, a continuity map can be built providing information as to the presence of discontinuities between a given grid point and each of its 8 neighbours. Such information can be conveniently and inexpensively stored using only one byte per grid point: each of the 8 bits is assigned to a neighbouring grid point and is set to 1 or 0 to indicate the presence or absence of a discontinuity. For the modified FMG continuation algorithm described earlier, such a continuity map is built for each level.

The discrete approximation to (3.19) is then modified in such a way that derivatives across discontinuites are inhibited. For example, in the situation shown in Figure 3.23 the discrete approximation to (3.19) becomes:

$$
\begin{aligned}
& \frac{\lambda}{2h^2}[(p_{i,j-1} - p_{i-1,j-1})^2 + (p_{i,j} - p_{i,j-1})^2 + (q_{i,j-1} - q_{i-1,j-1})^2 + (q_{i,j} - q_{i,j-1})^2] \\
+ \; & \frac{\mu}{2}[(h^{-1}(z_{i,j-1} - z_{i-1,j-1}) - (p_{i,j-1} + p_{i-1,j-1})/2)^2 \\
+ \; & (h^{-1}(z_{i,j} - z_{i,j-1}) - (q_{i,j} + q_{i,j-1})/2)^2]
\end{aligned}
$$

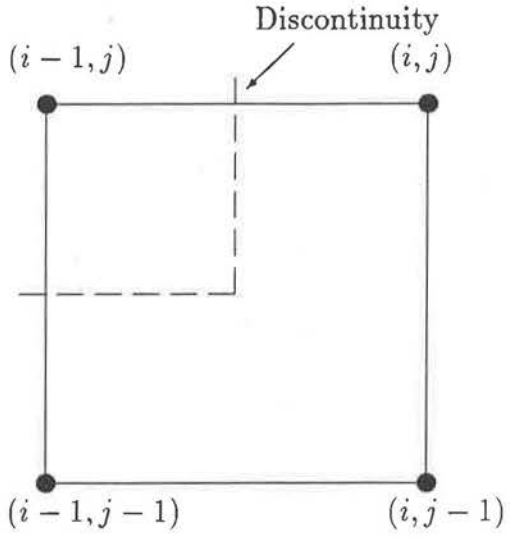


Figure 3.23: Discrete cell with discontinuites.

$$+ \quad \frac{1}{4}[(E_{i-1,j-1} - R(p_{i-1,j-1}, q_{i-1,j-1}))^2 + (E_{i,j-1} - R(p_{i,j-1}, q_{i,j-1}))^2$$
$$+ \quad (E_{i-1,j} - R(p_{i-1,j}, q_{i-1,j}))^2 + (E_{i,j} - R(p_{i,j}, q_{i,j}))^2] \tag{3.31}$$

Notice that this is entirely consistent with the natural boundary conditions

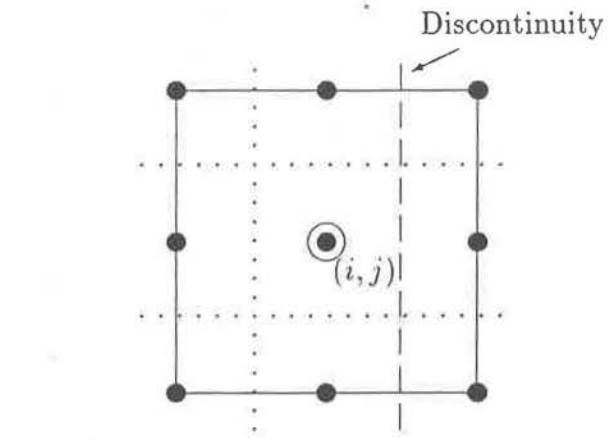
$$\frac{\partial p}{\partial n} = \frac{\partial q}{\partial n} = 0 \tag{3.32}$$

and

$$\frac{\partial z}{\partial n} = \mathbf{n} \cdot (p, q) \tag{3.33}$$

which are assumed to hold across discontinuities.

When no discontinuities are present we use the usual full weighting operator for restriction. However, in the presence of discontinuities the restriction operator follows a simple integration scheme over that part of the coarse grid cell lying on the same side of the discontinuity as the centre grid point. So, given the situation depicted in Figure 3.24,



● – fine grid point

◉ – grid point common to coarse and fine grid

Figure 3.24: Coarse grid cell with discontinuity.

for example, the residual $d^H_{i/2,j/2}$ is assigned to be

$$d^H_{i/2,j/2} = (4d^h_{i,j} + 2(d^h_{i-1,j} + d^h_{i,j+1} + d^h_{i,j-1}) + d^h_{i-1,j+1} + d^h_{i-1,j-1})/12. \tag{3.34}$$

For the prolongation operator, we use the usual bi-linear interpolation provided there is no discontinuity present. In latter cases, nearest neighbour interpolation is used. For example, in the situation depicted in Figure 3.25(a) the corrections $v^h_{i,j}$, $v^h_{i,j+1}$ and $v^h_{i,j-1}$ are assigned to be

$$v^h_{i,j} = (v^H_{i/2,j/2} + v^H_{i/2,j/2+1})/2 \tag{3.35}$$

$$v^h_{i,j+1} = v^H_{i/2,j/2+1} \tag{3.36}$$

$$v^h_{i,j-1} = v^H_{i/2,j/2} \tag{3.37}$$

while for Figure 3.25(b) we have

$$v^h_{i,j} = v^h_{i-1,j} = v^h_{i,j+1} = v^H_{i/2,j/2+1}. \tag{3.38}$$

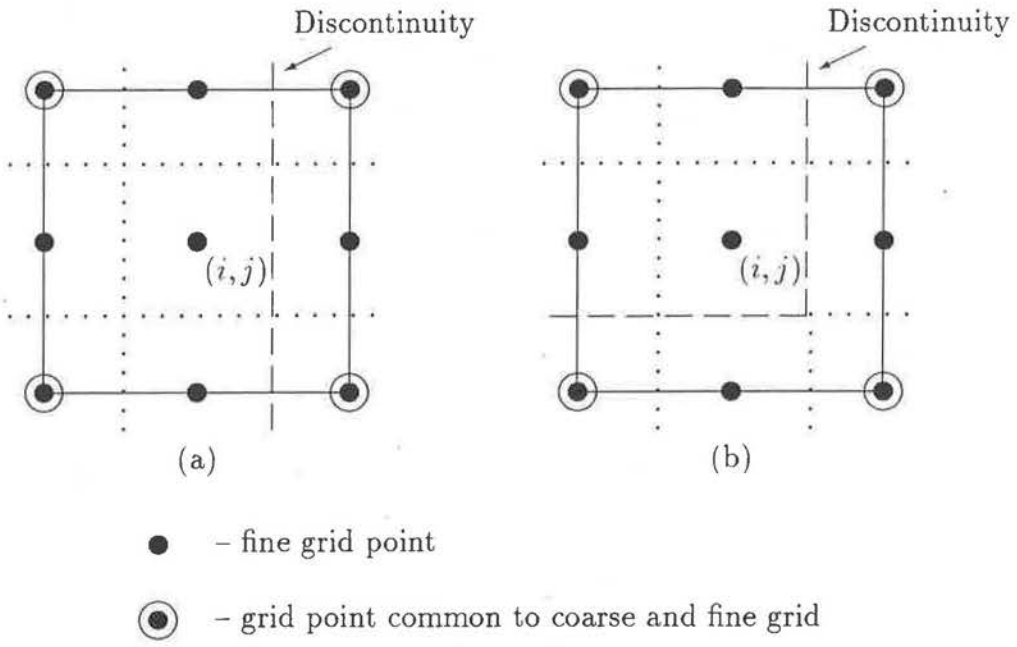


Figure 3.25: Fine grid cells with discontinuity.

In this manner we have a prolongation operator that is the adjoint of the restriction operator.

3.6.1 Synthetic Hemisphere and Vase

We now test the performance of the algorithm when given information about the location of discontinuities as discussed in the previous section. We again concentrate on the synthetically generated hemisphere and vase. Natural boundary conditions are employed at the interface of the object with the background while Dirichlet conditions are used at the boundary of the image (i.e. $z = p = q = 0$). We apply two extra relaxation sweeps per Gauss-Seidel iteration across points which border the interface. Experiment has shown that this improves the convergence rate without significantly increasing the cost of the computation. The number of W cycles performed on coarser grids is also increased so that the residual error is reduced below 0.1 before proceeding to a finer

level. Typically, no more than two W cycles are required.

We first test the algorithm on the hemisphere using the same parameters as used in Section 3.5.1. The algorithm in this case did not converge even after 10 W(2,2) cycles. It was hypothesized that poor convergence was due to the fact that there is some part of the sphere which is not illuminated in either image giving rise to a region where there is essentially no information. This hypothesis was tested by changing the reflectance map so that instead of taking

$$R(p,q) = \max\left\{0, \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2}}\right\} \tag{3.39}$$

we simply take

$$R(p,q) = \frac{1 + p_0 p + q_0 q}{\sqrt{1 + p_0^2 + q_0^2}\sqrt{1 + p^2 + q^2}} \tag{3.40}$$

so that *negative* brightness values are allowed in regions that are normally not illuminated. Hence, information is supplied over all parts of the surface. The availability of such information is of course physically unreasonable, but allowing ourselves this luxury will enable us to determine the extent to which a lack of information in non-illuminated regions is affecting the algorithm. With this modified reflectance map, the algorithm in fact converged. After only 5 W(2,2) cycles, the residual error was 1.175×10^{-4} and the error in the computed surface height was 3.742×10^{-2}. This result suggests that our hypothesis is correct. It should be noted that the error in surface height and the residual error are now measured only over the object and not over the background. Since the background is a simple flat plane, measuring errors over the background leads to figures which are overly optimistic.

Another, slightly more reasonable way of testing the hypothesis, is to attempt to reduce the area of the region on the sphere which is not illuminated by either image. One way of reducing this region to zero is to choose one of the light source directions

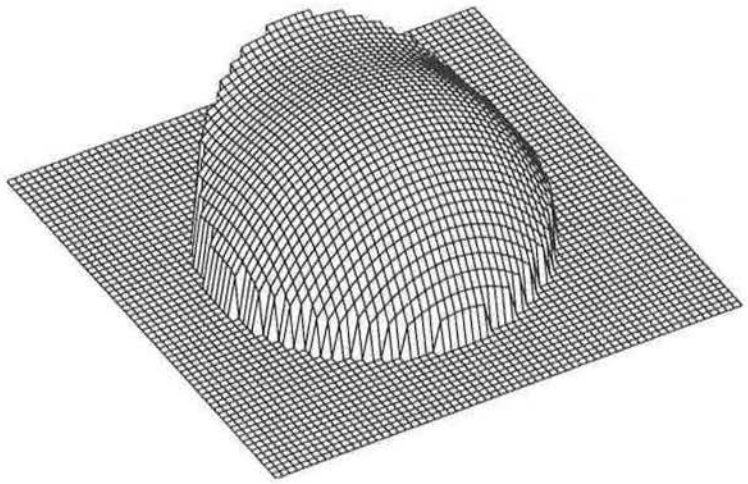
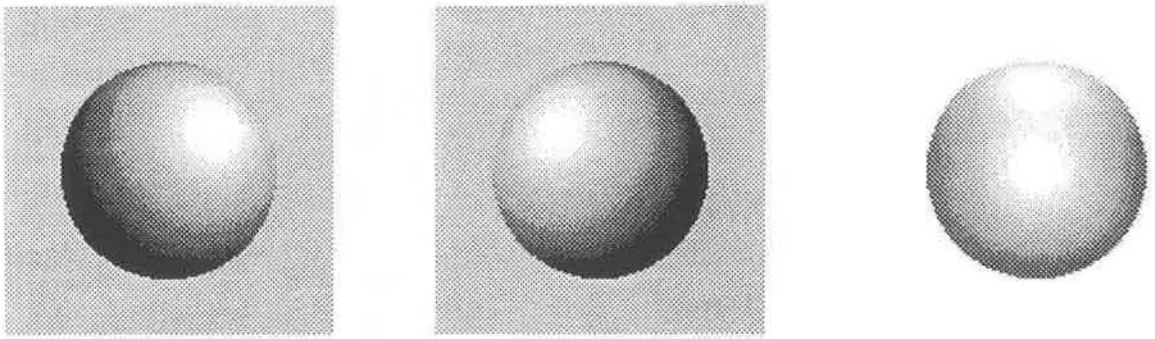


Figure 3.26: Computed Surface : Hemisphere from two images and silhouette.

to be overhead. However, it is well known that an overhead light source direction leads to difficulties with non-uniqueness as well as poor convergence. We therefore decided to choose the light source directions to be $(0.5, 0.5, 1)$ and $(-0.5, 0.5, 1)$. After 10 W(2,2) cycles, the residual error had been reduced to 3.768×10^{-6} while the error in the computed surface height was 5.191×10^{-2}. We performed 10 W(2,2) cycles since the residual error after only 5 cycles was not considered to be small enough. The computed surface height can be found in Figure 3.26 while reconstructed images are in Figure 3.27.

A rather obvious error is noticeable in the solution at the interface of the object and background for positive values of y. This feature lies between the two light source directions. We now demonstrate how such an error appears in the solution.

Figure 3.28 depicts level curves of the reflectance maps $R_1(p, q)$ and $R_2(p, q)$ corresponding to the light source directions used to illuminate the hemisphere in the above experiment. Due to the non-linearity of these reflectance maps, they are not sufficient to determine a solution for the gradient uniquely, however, at each point on the surface,



(a) (b) (c)

Figure 3.27: Images of computed surface: using original light source directions (left & centre), using overhead light source direction (right).

there are at most two possible solutions.

Now consider points on the surface lying in the plane $x = 0$. For the exact solution, such points have $p = 0$. Now, looking at the positive q-axis in Figure 3.28 we see that the two possible solutions that can arise from using two reflectance maps are very close together. It is therefore reasonable to suggest that the algorithm is simply choosing the wrong solution in this region. In this case, the preference for the wrong solution can be explained by the fact that the wrong solution is smoother than the correct solution since the first derivatives of p and q become infinite at the boundary of the hemisphere.

For the vase we also took the same parameters as in Section 3.5.1. Notice that for the given light source directions there is in fact no region on the surface of the vase which is not illuminated by either image. After 5W(2,2) cycles on the finest level the residual error was 1.283×10^{-2} which was not considered small enough to have confidence in the computed solution. The error in the solution at this point was 3.277×10^{-2}. It was therefore decided that due to the poor convergence rate ($\rho = 2.90$) 10W(2,2) cycles should be performed on the finest level. After this number of iterations the residual error was reduced to 8.647×10^{-5} while the error in the computed surface height had not

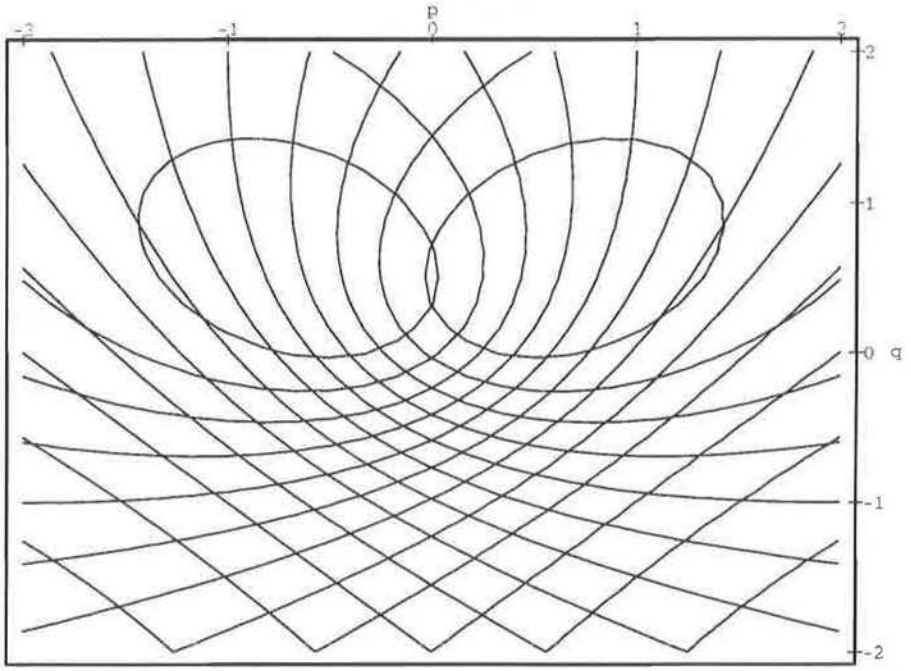


Figure 3.28: Iso–brightness contours in gradient space for Lambertian reflectance maps corresponding to light source directions $(0.5, 0.5, 1)$ and $(-0.5, 0.5, 1)$.

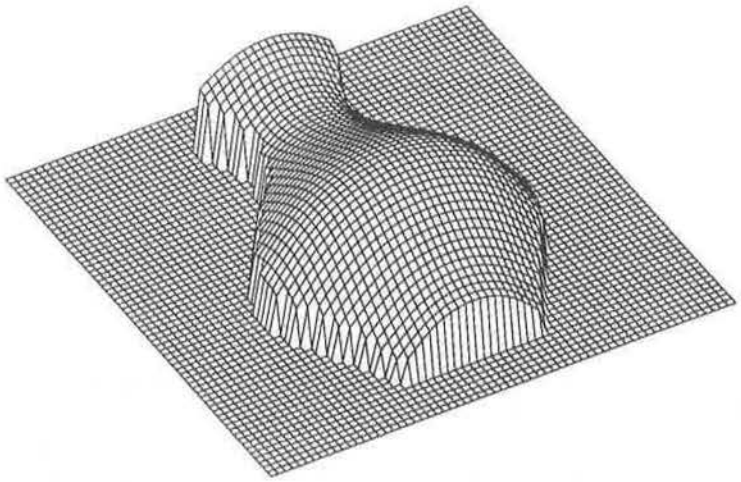


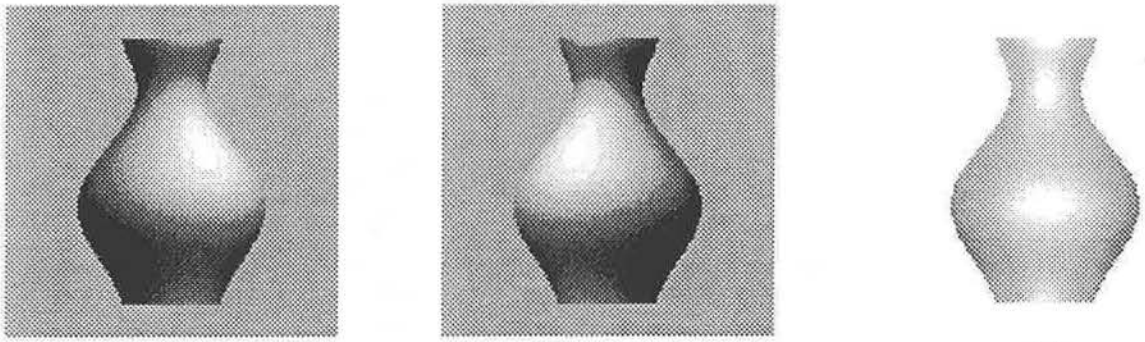
Figure 3.29: Computed surface: Vase from two images and silhouette.

changed much at 3.272×10^{-2} indicating that perhaps we could have had more confidence in the result computed after only 5W(2,2) cycles. The computed surface can be found in Figure 3.29 and reconstructed images are given in Figure 3.30.

At this point it must be noted that the computed error in the solution is not much better than that reported for earlier tests although the result appears to be substantially improved. This may be indicative of the importance of boundary information in the human visual recognition system. We should also recall that the error in surface height is computed only over the vase, if the background had also been included the error would be lower.

3.6.2 Three images as input

Finally, we consider the performance of the algorithm when three images are provided as input. Recall that Woodham [55] has shown that with three images and a silhouette file as input, and a known reflectance map (as we are assuming here) the method of



(a) (b) (c)

Figure 3.30: Images of computed surface: using original light source directions (left & centre), using overhead light source direction (right).

photometric stereo does an excellent job of retrieving the surface which gave rise to the images. Here, we test the performance of our algorithm given the same information.

For the hemisphere, we choose our parameters to be the same as for Section 3.5.1 except for the light source directions which are taken to be $(0.5, 0.5, 1)$, $(-0.5, 0.5, 1)$ and $(0, -0.5, 1)$. Again we avoid the temptation of choosing an overhead light source direction as this leads to poor convergence. After 5 W(2,2) cycles, the residual error was 8.165×10^{-5} and the error in computed surface height was 6.558×10^{-3} which in comparison to the maximum surface height of $1/3$ indicates an error of less than 2%. The computed surface is shown in Figure 3.31. The difference between the computed surface and the exact solution (shown in Figure 3.8) is barely noticeable.

For the vase we also choose our parameters to be as for Section 3.5.1. The additional light source direction is taken to be $(0, -0.5, 1)$. After 5 W(2,2) cycles the residual error was 3.374×10^{-5} while the error in the computed surface height was 4.738×10^{-3}. In comparison to the maximum surface height of 0.2855, this represents an error of less than 2%. The computed surface is shown in Figure 3.32.

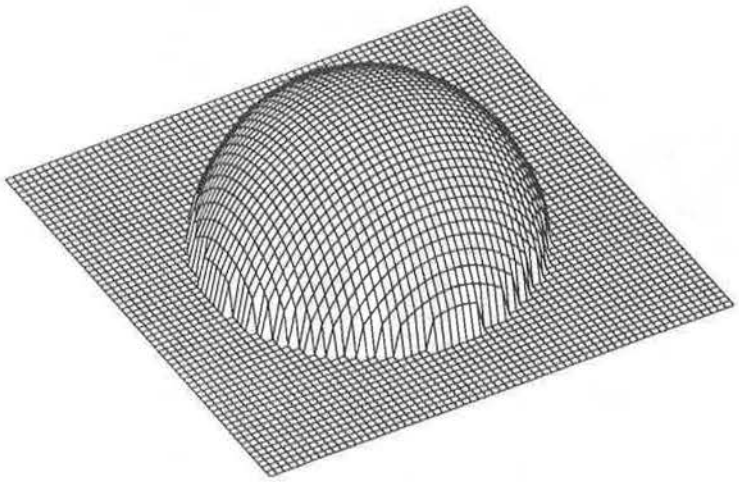


Figure 3.31: Computed surface: Hemisphere from three images and silhouette.

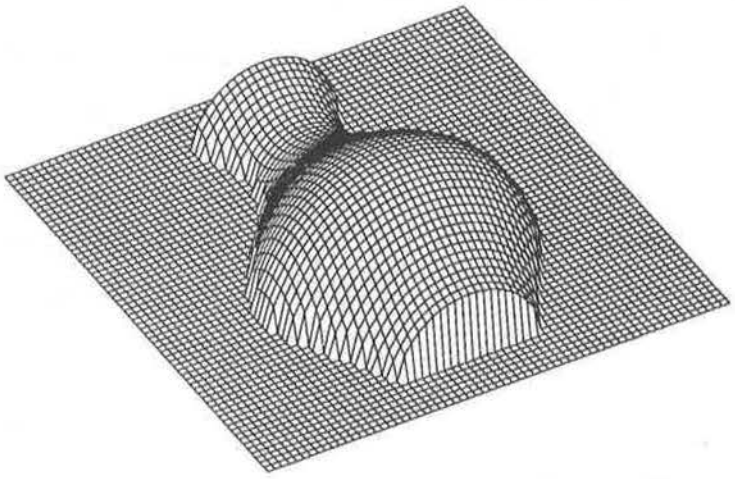


Figure 3.32: Computed surface: Vase from three images and silhouette.

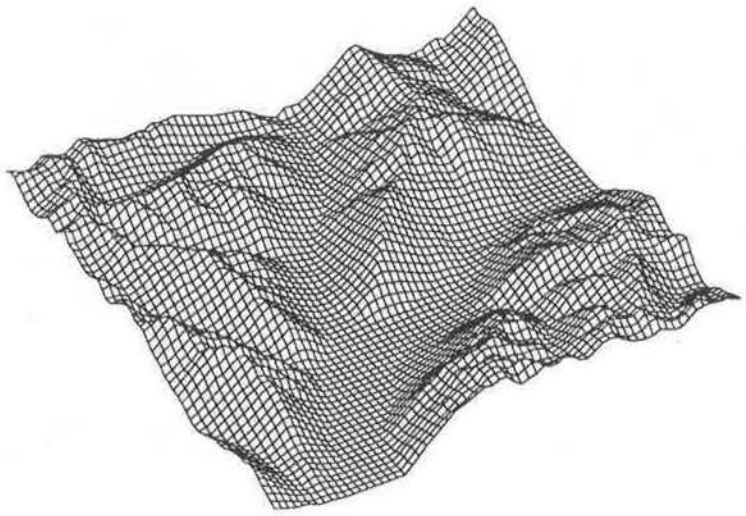


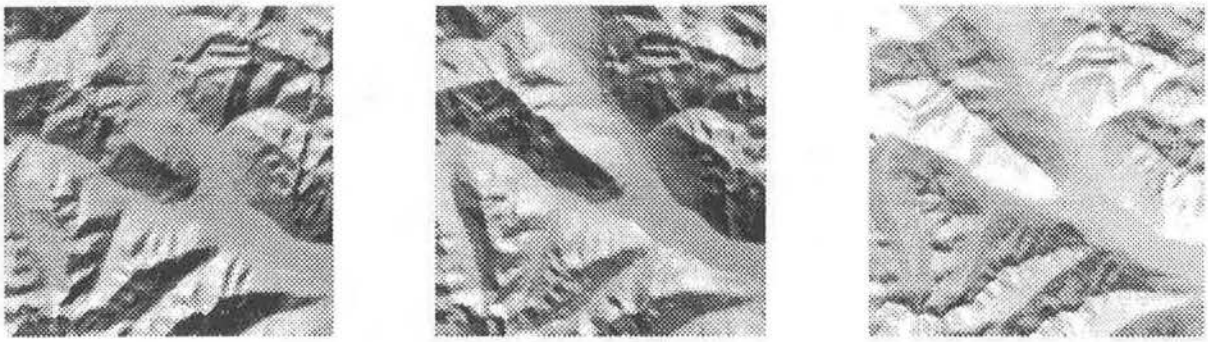
Figure 3.33: Exact surface: St. Mary Lake, British Columbia.

3.6.3 Synthetic Data from a Digital Terrain Model

We now present the results of applying our algorithm to rather more complicated data. We take a 129×129 grid of depth data from a digital terrain model of St. Mary Lake, British Columbia. [1] The surface is shown in Figure 3.33. The depth data in the model is sampled at a rate of 120 metres. We produce synthetic images from this depth data by estimating surface slopes numerically and using these as input to a Lambertian reflectance map. On the interior of the domain, we use second order central differencing to estimate the slopes but on the boundary we use only first order one sided differencing. Three images of the terrain illuminated from the light source directions $(0.5, 1, 1)$, $(-0.5, 1, 1)$ and $(0, -0.5, 1)$ are shown in Figure 3.34.

Given that we no longer have a single object in the field of view, we do not input a silhouette file. We use natural boundary conditions on all three variables and tie the

[1] The author would like to express his gratitude to Dr. Bob Woodham, Dept. of Computer Science, U. B. C. for making this data available.



(a) (b) (c)

Figure 3.34: Synthetic images of St. Mary Lake digital terrain model.

height down at the midpoint of the domain to the exact value. When only one image is provided as input, the algorithm converged only for very large values of λ ($\bar{\lambda} = 4000$). Given the complexity of the data and the fact that we have encountered difficulties previously when natural boundary conditions are used, the outcome is not particularly suprising.

The result of inputing the first *two* images shown in Figure 3.34 is shown in Figure 3.35. As for earlier tests, we took $\mu = 0.1$ and $\bar{\lambda} = 0.4$. The algorithm demonstrated good convergence. The final residual error was 1.313×10^{-5} with an error of 2.230×10^{-1}km in the computed surface height. Given that the maximum height in the exact data is approximately 2.5km, the error is just below 10%.

Given all three images as input, the algorithm computed the surface with a much smaller error of 6.980×10^{-2}km which is approximately 3% of the maximum height. The computed surface is shown in Figure 3.36. Notice that the computed surface is somewhat smoother than the original. This is to be expected given that the regularization parameter λ is non-zero on the finest level of the computational grid.

We can be very satisfied with these results given the complexity of the data.

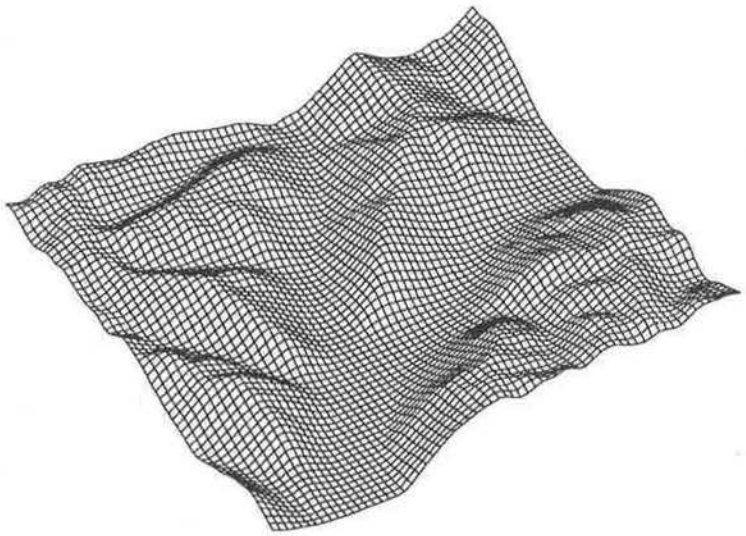


Figure 3.35: Computed surface: St. Mary Lake from two images.

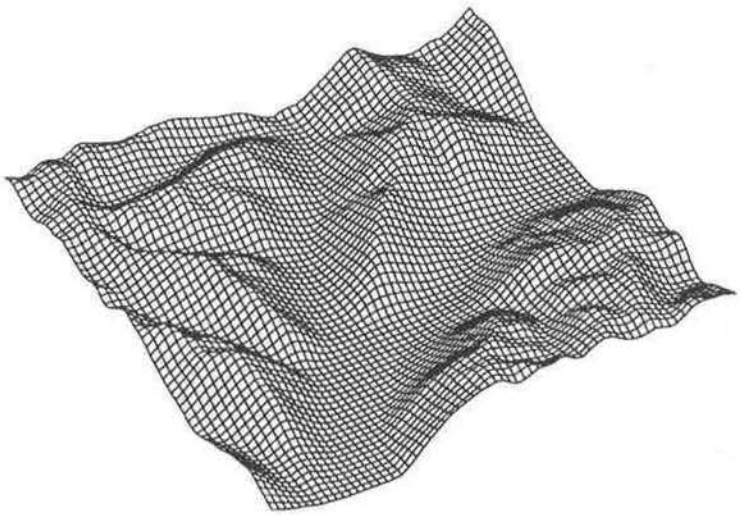


Figure 3.36: Computed surface: St. Mary Lake from three images.

3.6.4 Images with noise

We now investigate how our algorithm behaves in the presence of noise in the image data. We test this only for the case when three images are input since this case has so far proved to be the most robust.

We begin with the synthetic vase. We choose all parameters to be the same as those given in Section 3.6.2. We add digitization error to each of the three input images in the same way as was done in Chapter 2. The iteration converged after 5 W(2,2) cycles with residual error 2.436×10^{-5} and an error in surface height of 4.339×10^{-3}. The error in surface height is in fact smaller than the case when there is no noise present in the image data.

We conducted the same test on the St. Mary Lake data with similar results. After 5 W(2,2) cycles the iteration converged with residual error of 1.535×10^{-5} and an error in computed surface height of 7.002×10^{-2}. The error in the surface height is only marginally worse than in the case when there is no noise in the image data. Notice that in both tests, the presence of noise in the image data does not cause a significant deterioration in the multigrid convergence.

The variational algorithm clearly does not demonstrate the ill effects of digitization error as observed for the method of characteristics. This is not all that surprising since the accumulation of errors that occurs as we integrate along characteristics does not occur during pointwise relaxation in the Gauss-Seidel iteration.

We now investigate the effects of more severe pollution of the image data. In addition to the digitization error we now blank out a region in one of the images. For the vase we blank out a square region in the image created by illuminating the surface from the direction $(0, -0.5, 1)$ as shown in Figure 3.37.

Notice that the blanked out region in this image suggests to the algorithm that the

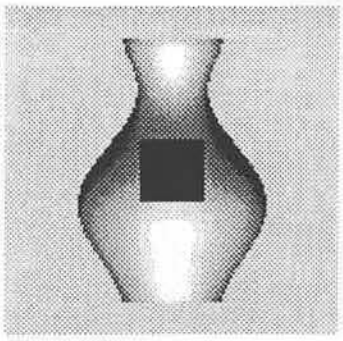


Figure 3.37: Polluted image: Vase

surface slopes in this region should satisfy the condition

$$1 + p_0 p + q_0 q < 0, \tag{3.41}$$

that is, $R(p, q) = 0$. This conflicts with the information supplied by the other two images in this region. Blanking out a region in one of the three images is *not* therefore equivalent to supplying information from only two images in that region. The blanked out region, rather than supplying no information to the algorithm, supplies incorrect information.

The iteration converged after 5 W(2,2) cycles with residual error of 3.590×10^{-5} and an error in computed surface height of 4.235×10^{-2}. The error is now an order of magnitude larger than in the case of no noise. The computed surface is shown in Figure 3.38.

Other tests, where the blanked out region crossed the edge of the vase or covered a particularly bright region of the image, resulted in non-convergence of the iteration. This is not particularly surprising since in the former case, the natural boundary conditions would be severely effected by inconsistent information from the images at the edge of the vase. When the blanked out region covers a bright region, this results in a particularly severe form of noise since we are replacing large, positive values of E with zero.

We repeated this test on the St. Mary Lake data. We blanked out a square region in the image illuminated from the direction $(0.5, 1, 1)$. This polluted image is shown in

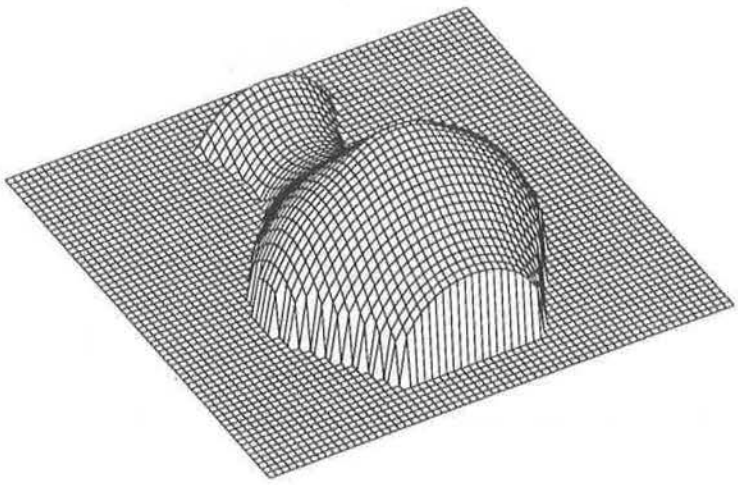


Figure 3.38: Computed surface: Vase from images with noise

Figure 3.39.

The iteration converged after 5 W(2,2) cycles with residual error 1.520×10^{-5} and error in the computed surface height of 4.916×10^{-1}. The computed surface is shown in Figure 3.40. The error in the surface height is an order of magnitude larger than in the case with no noise. This error, however, appears to be confined to the region where the image was blanked out.

Given the severity of the pollution in the image data, we can be pleased with the

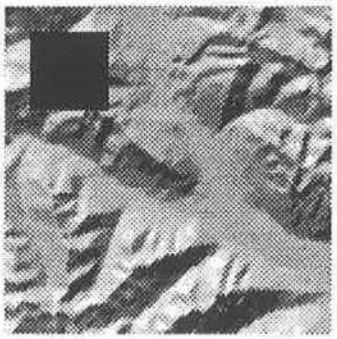


Figure 3.39: Polluted image: St. Mary Lake data

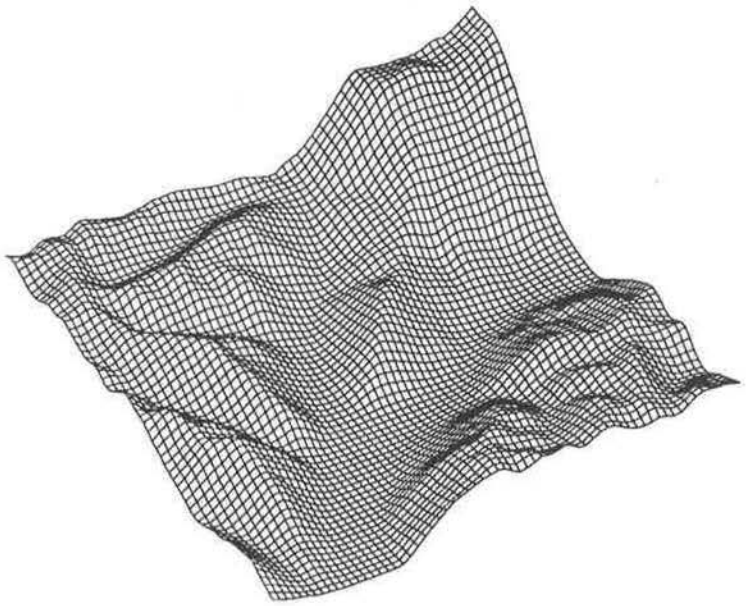


Figure 3.40: Computed surface: St. Mary Lake from images with noise

results generated by our algorithm. Note, however, that in cases where severe noise is known to exist in only one of the three images, it may be preferable to discard this image completely and use only the two consistent images. The errors reported in this section are slightly higher than for those arising when only two images of the surfaces are used.

3.7 Discussion

It could now be argued that given three images and a silhouette as input, the regularization applied through smoothing terms should no longer be required. However, it must be remembered that there will be some points on the surface which are illuminated by only one or two images. At such points, the regularizing terms fill in for the missing data. It must also be noted that despite the fact that we have some degree of global smoothing, the controlling parameter $\bar{\lambda}$ is small enough so as not to affect the computed solution in any serious way.

It is satisfying to see that given the same information as is required by photometric

stereo, our results are also excellent. Reasonable results can also be obtained in some cases when only two images are used. However, single image reconstructions have often (though not always) produced unsatisfactory results.

Our algorithm has also proved to be robust in the presence of simple digitization errors in the image data. In cases where more severe errors occur in the image data, the algorithm has computed the surfaces with reasonable accuracy.

Chapter 4

Upwinding Schemes

4.1 Introduction

We now turn our attention to another class of solution techniques which have so far not been considered in the shape from shading literature.

Our starting point is again the image irradiance equation

$$R(p,q) = E(x,y) \tag{4.1}$$

which we differentiate with respect to x and y to obtain

$$\left.\begin{aligned} R_p p_x + R_q q_x &= E_x \\ R_p p_y + R_q q_y &= E_y \end{aligned}\right\} \tag{4.2}$$

Here, we treat p and q as independent variables so that system (4.2) is singular, having no unique solution. However, making use of the integrability condition

$$p_y = q_x \tag{4.3}$$

we obtain the quasilinear system

$$\left.\begin{aligned} R_p p_x + R_q p_y &= E_x \\ R_p q_x + R_q q_y &= E_y \end{aligned}\right\} \tag{4.4}$$

Before proceeding, we consider the conditions under which systems (4.2) and (4.4) are equivalent. Subtracting equations (4.4) from equations (4.2) pairwise we get

$$\left.\begin{aligned} R_q(q_x - p_y) &= 0 \\ R_p(p_y - q_x) &= 0 \end{aligned}\right\} \tag{4.5}$$

We therefore see that on the continuous level, for which integrability is guaranteed, these equations are satisfied identically showing that a solution of (4.4) satisfies (4.2). However, on the discrete level, due to discretization error, integrability is not maintained exactly. We see from equations (4.5) that the equivalence of the two systems can be maintained in the absence of integrability by creating singular points where $R_p = R_q = 0$. We bear in mind this potential difficulty as we seek a numerical solution to this system. Note that applying a method of characteristics to (4.4) yields the method of characteristics discussed in Chapter 2. Here, however, we consider discretizations on a fixed, regular grid.

We observe a similarity between system (4.4) and the equations governing flow of an incompressible, inviscid, two dimensional fluid. This connection is established by considering a Lambertian surface illuminated from overhead so that

$$R(p,q) = \frac{1}{\sqrt{1+p^2+q^2}} = E(x,y) \tag{4.6}$$

which after some rearrangement yields

$$\tilde{R}(p,q) = p^2 + q^2 = \frac{1}{E^2} - 1 = \tilde{E}(x,y). \tag{4.7}$$

Notice that (4.7) is the Eikonal equation. We will therefore refer to $\tilde{R}(p,q)$ as the *Eikonal reflectance map*. Now, replacing R and E with $\tilde{R}$ and $\tilde{E}$ in (4.4) we get

$$\left.\begin{aligned} pp_x + qp_y &= \frac{\tilde{E}_x}{2} \\ pq_x + qq_y &= \frac{\tilde{E}_y}{2} \end{aligned}\right\} \tag{4.8}$$

which is analogous to

$$\left.\begin{aligned} uu_x + vu_y &= -p_x \\ uv_x + vv_y &= -p_y \end{aligned}\right\} \tag{4.9}$$

where u and v are the components of velocity in a two dimensional incompressible, inviscid fluid flow and p is the pressure. A difference between system (4.8) and system (4.9) is

that the pressure p in (4.9) is unknown and an additional divergence-free condition must be satisfied:

$$u_x + v_y = 0. \tag{4.10}$$

This allows us to write (4.9) in conservation form as follows:

$$\left.\begin{aligned} (u^2)_x + (uv)_y &= -p_x \\ (uv)_x + (v^2)_y &= -p_y \end{aligned}\right\} \tag{4.11}$$

Unfortunately, there is no equivalent divergence free condition for the shape from shading problem. Hence, discretizations based on conservation laws are not viable.

An alternative solution technique is to use upwinding schemes for advection equations, see Yavneh [58], Sidilkover & Ascher [48]. These methods employ one-sided, first and second order approximations to the derivatives of p and q appearing in system (4.4).

4.2 Upwinding Discretization Schemes

We begin by setting up a regular grid on the domain $[0,1]^2$ at the points (x_i, y_j), $0 \leq i, j \leq N$ where $x_i = ih$, $y_j = jh$ and $h = 1/N$. We assume that image data E^h_{ij} are specified at grid points and that Dirichlet boundary data are given on the surface slopes.

We use forward or backward differencing to discretize the first derivatives found in (4.4) depending on the direction of the "flow" at each grid point. For the shape from shading problem, this direction is equivalent to the direction of the characteristic strips discussed in Chapter 2, that is, $R_p\,\vec{i} + R_q\,\vec{j}$. Hence, for the situation depicted in Figure 4.1 we take a backward difference for x-derivatives and a forward difference for y-derivatives at the grid point (i, j).

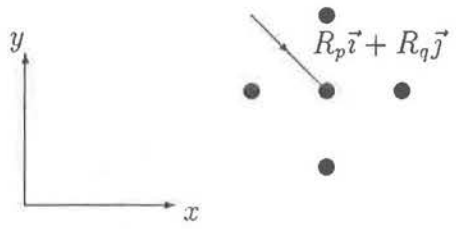


Figure 4.1: Characteristic direction at grid point (i, j).

This discretization may be represented compactly in the following way:

$$\left.\begin{aligned} |R_p^{ij}|\frac{(p_{ij} - p_{ew})}{h} + |R_q^{ij}|\frac{(p_{ij} - p_{ns})}{h} &= E_x^h \\ |R_p^{ij}|\frac{(q_{ij} - q_{ew})}{h} + |R_q^{ij}|\frac{(q_{ij} - q_{ns})}{h} &= E_y^h \end{aligned}\right\} \tag{4.12}$$

where

$$R_p^{ij} = R_p(x_i, y_j) \qquad E_x^h = (E_{i+1,j} - E_{i-1,j})/2h$$
$$R_q^{ij} = R_q(x_i, y_j) \qquad E_y^h = (E_{i,j+1} - E_{i,j-1})/2h$$

and

$$p_{ew} = \begin{cases} p_{i-1,j} & \text{if } R_p^{ij} > 0 \\ p_{i+1,j} & \text{if } R_p^{ij} < 0 \end{cases} \tag{4.13}$$

$$p_{ns} = \begin{cases} p_{i,j-1} & \text{if } R_q^{ij} > 0 \\ p_{i,j+1} & \text{if } R_q^{ij} < 0 \end{cases} \tag{4.14}$$

Similarly for q_{ew} and q_{ns}.

Notice that at points where $R_p = R_q = 0$, the system (4.12) is singular. Recall that such points are referred to as *singular points*. For the linear reflectance map, such points do not arise except in the degenerate case of overhead illumination when R_p and R_q are zero everywhere. For the Eikonal and Lambertian reflectance maps, these points are

points on the surface where the surface normal is parallel to the light source direction. Hence, such points are points of maximal (minimal, for the Eikonal map) brightness in the image $E(x, y)$. In the context of fluid flow, points of zero velocity are referred to as *stagnation points.*

It is easily seen that system (4.12) is equivalent to

$$\left.\begin{aligned} &- |R_p^{ij}|\frac{h}{2}\frac{(p_{i+1,j} - 2p_{ij} + p_{i-1,j})}{h^2} - |R_q^{ij}|\frac{h}{2}\frac{(p_{i,j+1} - 2p_{ij} + p_{i,j-1})}{h^2} \\ &+ R_p^{ij}\frac{(p_{i+1,j} - p_{i-1,j})}{2h} + R_q^{ij}\frac{(p_{i,j+1} - p_{i,j-1})}{2h} = E_x^h \\ &- |R_p^{ij}|\frac{h}{2}\frac{(q_{i+1,j} - 2q_{ij} + q_{i-1,j})}{h^2} - |R_q^{ij}|\frac{h}{2}\frac{(q_{i,j+1} - 2q_{ij} + q_{i,j-1})}{h^2} \\ &+ R_p^{ij}\frac{(q_{i+1,j} - q_{i-1,j})}{2h} + R_q^{ij}\frac{(q_{i,j+1} - q_{i,j-1})}{2h} = E_y^h \end{aligned}\right\} \tag{4.15}$$

which can be interpreted as a second order central difference approximation to (4.4) with added artificial viscosity. Notice that the amount of artificial viscosity depends on the size of R_p and R_q and is therefore not pointwise isotropic.

Yavneh [58] has shown that this scheme can, for certain flows, converge to a solution which is $O(1)$ away from the required solution. Specifically, flows for which the fluid recirculates yielding characteristics (streamlines) which form closed curves, may not be resolved using this simple upwinding scheme. As an alternative, Yavneh suggests a scheme which employs pointwise isotropic artificial viscosity. In experimental tests on recirculating flows, Yavneh shows that the isotropic artificial viscosity scheme does not yield the spurious results seen for the simple upwinding scheme. For the problem in

hand, this translates to the following discretization:

$$\left.\begin{aligned} &- \epsilon_{ij}\frac{(p_{i+1,j}+p_{i-1,j}+p_{i,j+1}+p_{i,j-1}-4p_{ij})}{h^2} \\ &+ R_p^{ij}\frac{(p_{i+1,j}-p_{i-1,j})}{2h} + R_q^{ij}\frac{(p_{i,j+1}-p_{i,j-1})}{2h} = E_x^h \\ &- \epsilon_{ij}\frac{(q_{i+1,j}+q_{i-1,j}+q_{i,j+1}+q_{i,j-1}-4q_{ij})}{h^2} \\ &+ R_p^{ij}\frac{(q_{i+1,j}-q_{i-1,j})}{2h} + R_q^{ij}\frac{(q_{i,j+1}-q_{i,j-1})}{2h} = E_y^h \end{aligned}\right\} \tag{4.16}$$

where

$$\epsilon_{ij} = \frac{h}{2}\max(|R_p^{ij}|, |R_q^{ij}|). \tag{4.17}$$

(An additional artificial diffusion term must be added near singularities.)

We now consider cases for which the shape from shading problem will yield characteristics which form closed curves and therefore correspond to a recirculating flow. For the linear reflectance map, recirculating flows are not possible since in this case the characteristics are straight lines. For the Eikonal and Lambertian reflectance maps, Oliensis [37] [38] has shown that characteristics follow the path of steepest descent (ascent, in the case of the Eikonal map) on the surface relative to a coordinate system in which height is measured in the direction of the light source. Hence, for these reflectance maps also, characteristics cannot form closed curves. We therefore conclude that we need not be concerned about the phenomenon observed by Yavneh [58] when simple upwinding is used in the case of recirculating flows. We therefore employ the discretization given in equations (4.15).

4.3 Regularization of Problem

In general, images contain singular points and so we expect to encounter points in the domain where our system will become singular. At such points, the flow of characteristics

can be placed into one of three categories: (i) a sink, towards which all characteristics in the neighbourhood of the point flow; (ii) a source, from which characteristics emanate; (iii) a saddle, for which two characteristics flow towards and two flow away from the point, see Saxberg [47], Oliensis [37][38]. At sinks and saddle points, characteristics flowing from different parts of the domain meet at the singular point. On the continuous level, they propagate the same solution to the point. However, if the system is perturbed, each characteristic could propagate a different solution yielding a non-unique solution in the neighbourhood of the singular point. At sources, characteristics flow away from the singular point towards the boundary (assuming there is no other singular point in the domain). If the system is perturbed, the solution propagated to the boundary may conflict with the boundary data. In general, we have an ill-posed problem when singular points are present in the domain. In such cases, we require some kind of regularization mechanism as is done in computational fluid dynamics. We bear in mind the possible ill-posed nature of the problem as we seek a numerical solution to our system.

Consider adding an artificial viscosity term to each of the equations in (4.4) to get:

$$\left.\begin{aligned} -\lambda\nabla^2 p + R_p p_x + R_q p_y &= E_x \\ -\lambda\nabla^2 q + R_p q_x + R_q q_y &= E_y \end{aligned}\right\} \tag{4.18}$$

The corresponding discrete equations are:

$$\left.\begin{aligned} - \quad & \lambda\frac{(p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{ij})}{h^2} \\ + \quad & R_p\frac{(p_{i+1,j} - p_{i-1,j})}{2h} + R_q\frac{(p_{i,j+1} - p_{i,j-1})}{2h} = E_x^h \\ - \quad & \lambda\frac{(q_{i+1,j} + q_{i-1,j} + q_{i,j+1} + q_{i,j-1} - 4q_{ij})}{h^2} \\ + \quad & R_p\frac{(q_{i+1,j} - q_{i-1,j})}{2h} + R_q\frac{(q_{i,j+1} - q_{i,j-1})}{2h} = E_y^h \end{aligned}\right\} \tag{4.19}$$

where the parameter λ controls the amount of added artificial viscosity. The addition of these terms essentially changes the problem being solved. We hope that a solution of

this problem in the limit as $\lambda \to 0$ corresponds to a solution of the original problem.

It must be noted that the addition of these terms changes the character of the equations from first order hyperbolic to second order elliptic. The solution of the second order elliptic problem requires specification of boundary conditions on p and q on the entire boundary. The original first order hyperbolic problem however, requires Dirichlet boundary data only on that portion of the boundary where the characteristics flow into the domain, i.e. on the inflow boundary.

Note that singularity at points where $R_p = R_q = 0$ is now avoided. By adding the artificial viscosity in (4.18) we have added a sense of direction where there was none previously. Before, integration could be carried out equally well in either direction along a characterisitic.

System (4.18) can now be compared to the equations governing the flow of an incompressible, viscous fluid:

$$-\nu\nabla^2 u + uu_x + vu_y = -p_x \tag{4.20}$$

$$-\nu\nabla^2 v + uv_x + vv_y = -p_y \tag{4.21}$$

Here, ν is the viscous parameter. Note that finding a solution to (4.18) for small λ is analogous to solving (4.21) for small ν. Flows for which the advection terms dominate the viscous terms are referred to as high Reynolds number flows.

4.4 Computation of Height from Gradient

Having computed the surface slopes, we now seek to compute the surface height. Since we do not expect the computed surface slopes to be exactly integrable, we seek the surface $z(x, y)$ whose first derivatives match p and q as close as possible in the least squares

sense. We therefore set out to minimize

$$\int\int_{\Omega}[(z_x - p)^2 + (z_y - q)^2]dxdy \tag{4.22}$$

on the domain Ω which is the domain of the image under consideration. Following the variational techniques outlined in Chapter 3, a necessary (and in this case sufficient) condition for such a minimum is that the following Euler equation be satisfied:

$$\nabla^2 z = p_x + q_y \tag{4.23}$$

We assume knowledge of the surface height on the boundary of the domain so that Dirichlet boundary conditions are specified on z. Note that this is not at all unreasonable given that we are assuming Dirichlet boundary data for p and q.

We solve (4.23) using a standard multigrid iteration with 5 W(2,2) cycles. This iteration has proved to be extremely robust, converging within the given number of iterations even when the data on the right hand side of (4.23) are not smooth. This will be verified for tests that will be conducted later in the chapter.

4.5 Results: Upwinding discretizations

At first, we choose to solve system (4.15) using a simple Gauss-Seidel iteration with lexicographic ordering on a 65×65 grid. Recall that system (4.15) is singular at points where $R_p = R_q = 0$. We therefore expect to have difficulty dealing with surfaces that give rise to such points.

We again report r.m.s. errors in the computed surface height and gradient and r.m.s. residual errors as defined in Section 3.3.

We begin by considering the simple case of a linear reflectance map. Pentland [43] has shown that in cases of oblique illumination, a linear reflectance map can be used to approximate the non-linear Lambertian reflectance map with reasonable accuracy. Linear

reflectance maps have the advantage that they do not, in general, give rise to singular points.

4.5.1 Linear Reflectance Map

We assume the linear reflectance map takes the form

$$R(p,q) = 1 + p_0 p + q_0 q \tag{4.24}$$

where (p_0, q_0) specifies the light source direction. We avoid the uninteresting case of overhead illumination which yields $R(p,q) \equiv 1$ for which system (4.15) is singular everywhere. We also resist the temptation of choosing one of p_0 and q_0 to be zero in which case the characteristics would be aligned with the computational grid. In such cases, if during the Gauss-Seidel iteration we sweep along grid points in the direction of the characteristics, the iteration acts as a direct solver and so only one iteration would be needed to solve the system. Instead we choose $(p_0, q_0) = (1,1)$ so that the characteristics are maximally non-aligned with the computational grid thereby considering a worst case scenario.

We begin by considering the surface

$$z = \sin(2\pi x)\sin(2\pi y) \tag{4.25}$$

on $[0,1]^2$. The exact solution is shown in Figure 4.2.

We set $\lambda = 0$ in (4.18) and performed 100 Gauss-Seidel iterations using the algorithm described above which yielded a residual error of 1.929×10^{-14}. We then computed the surface height using the multigrid algorithm described earlier for (4.23) which yielded an error in the surface height of 7.311×10^{-2} or 3.6%, given that the maximum range in surface height is 2. The computed surface is shown in Figure 4.3.

Given that we have only a first order differencing scheme, we can be quite pleased with the result. However, we must stress the great simplification that is gained from

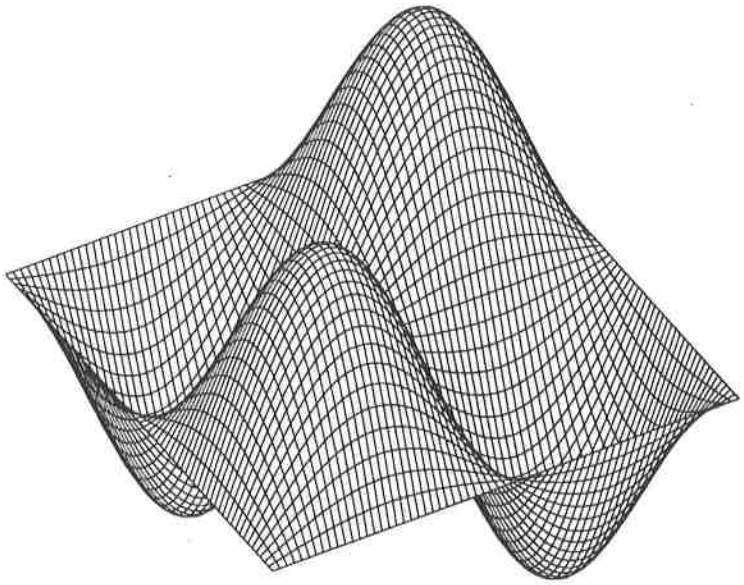


Figure 4.2: Exact solution: $z = \sin(2\pi x)\sin(2\pi y)$.

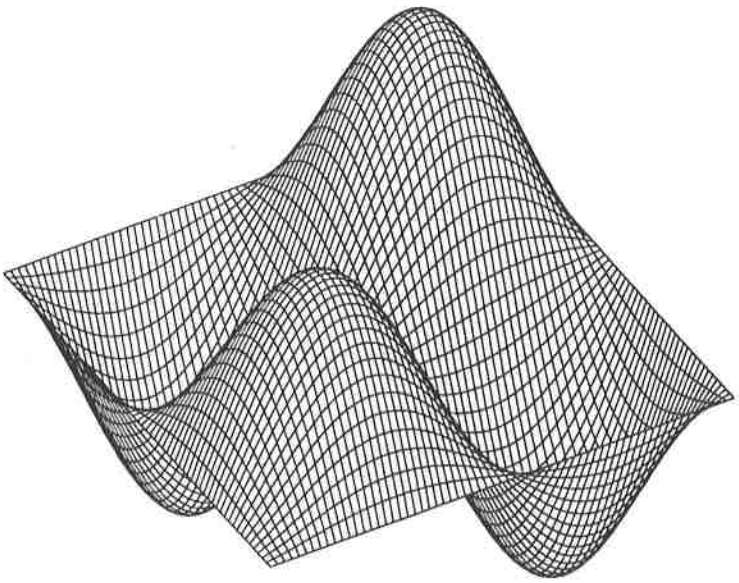


Figure 4.3: Computed surface, $z = \sin(2\pi x)\sin(2\pi y)$, Linear map.

using a linear reflectance map. Had we used an Eikonal or Lambertian reflectance map with overhead illumination we would have had 5 singular points in the interior and 8 singular points on the boundary of the domain. As we shall see, the existence of singular points in the domain yields problems which are less easily solved.

4.5.2 Eikonal Reflectance Map

Given the difficulties that we expect to encounter with the existence of singular points in the domain, we begin with a surface which gives rise to no singular points. For the Eikonal reflectance map, these are points where $p = q = 0$, that is, points where the surface has a horizontal tangent plane. We point out that just because the exact solution has no singular points, this does not guarantee that the iteration will not create singular points as it moves from the initial guess to the final converged solution. At such points we set

$$R_p = R_q = \epsilon, \quad 0 < \epsilon \ll 1. \tag{4.26}$$

We suggest a value of $\epsilon = 1.0 \times 10^{-6}$. The surface tested is given by

$$z = y \sin(\pi x), \quad x \in [0.1, 0.9], \quad y \in [-0.5, 0.5]. \tag{4.27}$$

Given the exact solution to the underlying continuous problem as an initial guess and taking $\lambda = 0$, the iteration converged after 50 iterations with residual error 3.975×10^{-12}. The surface height was computed from the resulting components of the gradient with an error of 2.091×10^{-3}. It would therefore appear that given an initial guess which is very close to the solution of the discrete problem, the suggested modification at singular points does not pollute the solution.

Next we try the initial guess $p = q = 0$ at all points on the interior of the domain which is reasonable in the absence of any other information about the surface to be reconstructed. In this case, initially, every interior point is a singular point. With $\lambda = 0$,

the scheme showed no signs of converging after 1000 iterations. The residual error was still $O(10)$.

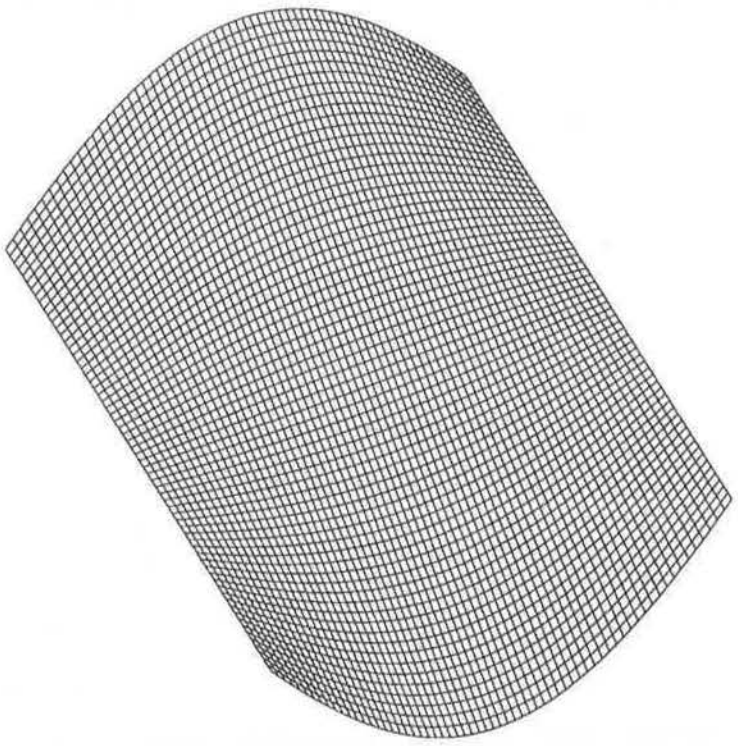
Given that we seek a solution to (4.19) for small λ and that in general we cannot expect to have a very good initial guess, we suggest a continuation algorithm in λ just as was employed for the variational techniques in Chapter 3. With the addition of the Laplacian terms we now use a Gauss-Seidel iteration with red-black ordering as this will yield more rapid convergence especially when λ is not small. We impose Dirichlet boundary conditions on p and q along the entire boundary of the domain. Again we considered the surface given in (4.27). The algorithm now converged successfully to a good approximation of the correct solution. The error in the computed surface height was 2.094×10^{-3}. The solution can be found in Figure 4.4 along with a table describing the performance of the algorithm during the continuation process.

Notice that the number of iterations required to reach convergence decreases with λ. We also see that the error in the components of the surface gradient is reduced with λ as expected, since smaller values of λ yield a problem which is closer to the one we are wanting to solve. The large number of iterations for $\lambda = 0.1$ can be drastically reduced using a multigrid method. However, here we concentrate on exploring the viability of the approach, not on tightening the efficiency of the algorithm.

Having successfully computed the solution to a problem for which there are no singular points, we now move on to problems which contain only one singular point. Consider the surface

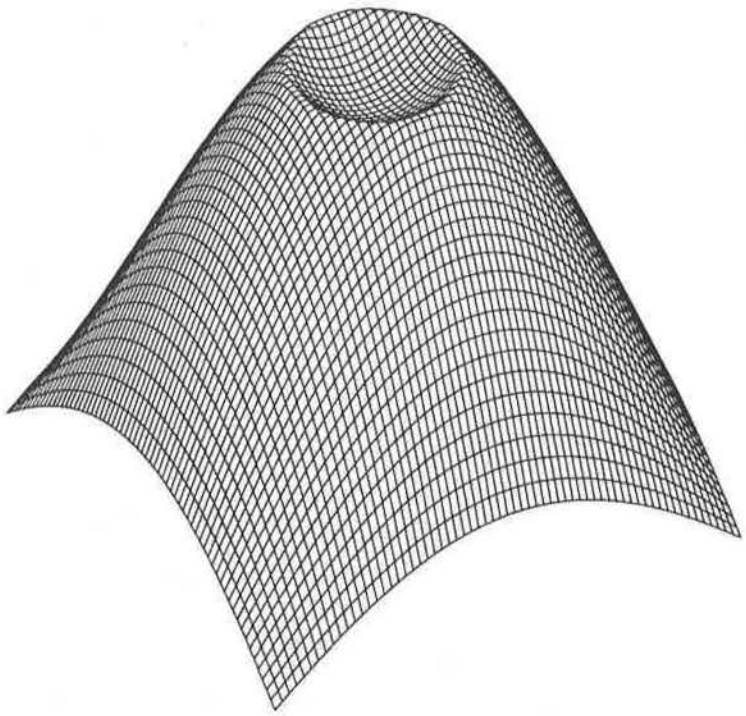
$$z = \sin(\pi x)\,\sin(\pi y), \quad x, y \in [0.1, 0.9] \tag{4.28}$$

which gives rise to a singular point at $(0.5, 0.5)$ – a local maximum on the surface. The computed surface and a table describing the continuation process can be found in Figure 4.5.



λ	0.1	1×10^{-3}	1×10^{-5}
Iterations	1500	250	50
Residual Error	1.803×10^{-3}	4.695×10^{-15}	8.242×10^{-14}
Gradient Error	1.139×10^{-1}	1.081×10^{-2}	9.310×10^{-3}

Figure 4.4: Computed surface: $z = y \sin(\pi x)$, $\lambda = 1 \times 10^{-5}$, Eikonal map.



λ	0.1	1×10^{-2}	1×10^{-3}	1×10^{-4}
Iterations	1500	500	100	100
Residual Error	1.228×10^{-7}	1.298×10^{-7}	1.670×10^{-14}	Diverged
Gradient Error	1.026×10^{-1}	9.291×10^{-2}	8.798×10^{-2}	–

Figure 4.5: Computed surface: $z = \sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-3}$, Eikonal map.

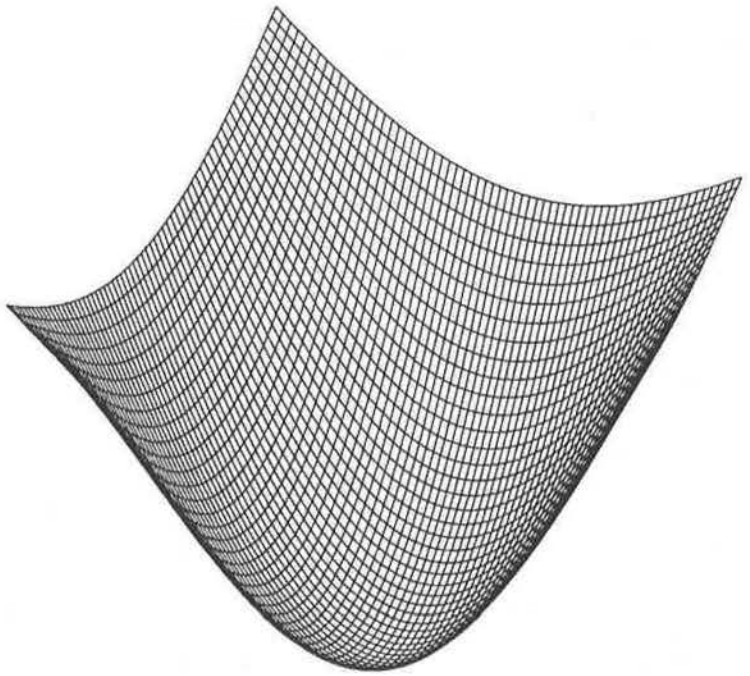
For this surface it was necessary to reduce λ more slowly and it was not possible to reduce λ as much as in the previous example. We suggest that this is linked to the presence of the singular point. The error in the computed surface height was 3.155×10^{-2}. The computed surface is clearly flawed. It is rather disturbing that the iteration has converged so readily to the wrong solution. It appears that the surface has been inverted in a neighbourhood of the singular point. Note that in the absence of boundary conditions, the reduced problem has two solutions – one given by z as in (4.28) and the other by $-z$. It is almost as though the iteration is choosing the wrong solution near the singular point.

We next replace z in (4.28) by $-z$ and recompute the solution. Note that this does not change the image, E. The results are shown in Figure 4.6. Curiously, the algorithm converges to a good approximation of the correct surface provided λ is not taken too small. Note in particular that the small inversion near the singular point is no longer present.

It could be hypothesized that the presence of the inversion in the solution near the singular point is linked to the nature of the singular point. The inversion occurred when trying to compute a surface which has a local maximum at the singular point. At such a point characteristics flow towards the singular point (a "sink"), whereas in the computation of a surface having a local minimum at the singular point, the flow of characteristics is away from the singular point towards the boundary, (a "source"). The next set of results reinforces this hypothesis although it will be seen that when computing a surface having a local maximum, the error in the solution does not always manifest itself in the same way.

Consider the surface

$$z = -(x^4 + y^4), \quad x, y \in [-0.5, 0.5] \tag{4.29}$$



λ	0.1	1×10^{-2}	1×10^{-3}	1×10^{-4}
Iterations	1500	500	100	100
Residual Error	1.296×10^{-12}	2.465×10^{-14}	2.342×10^{-14}	Diverged
Gradient Error	1.007×10^{-1}	2.483×10^{-2}	1.599×10^{-2}	–

Figure 4.6: Computed surface: $z = -\sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-3}$, Eikonal map.

which gives rise to one singular point at the origin at which the surface has a local maximum. The result of applying our algorithm to this problem is given in Figure 4.7. Notice that for this surface it was possible to decrease λ much further than for the previous tests while maintaining good, rapid convergence. However, the effort involved in reducing λ further than 1×10^{-4} did not seem to pay off in terms of a reduction in the error in the computed solution. It is expected that such a threshold will be reached in general, since there will be a point at which numerically, the effect of the Laplacian operator will not be felt for small, positive values of λ. The computed surface does not display the inversion in the solution observed in Figure 4.5. However, the presence of spurious ridges in the surface is clearly noticeable. The error in the computed surface is 6.575×10^{-3}.

Again we consider the problem of computing the inverted surface. The singular point at the origin now corresponds to a local minimum on the computed surface. The algorithm again correctly computes the surface with no noticeable deviation from the exact solution. The result is shown in Figure 4.8. The error in the computed surface is 1.351×10^{-3}.

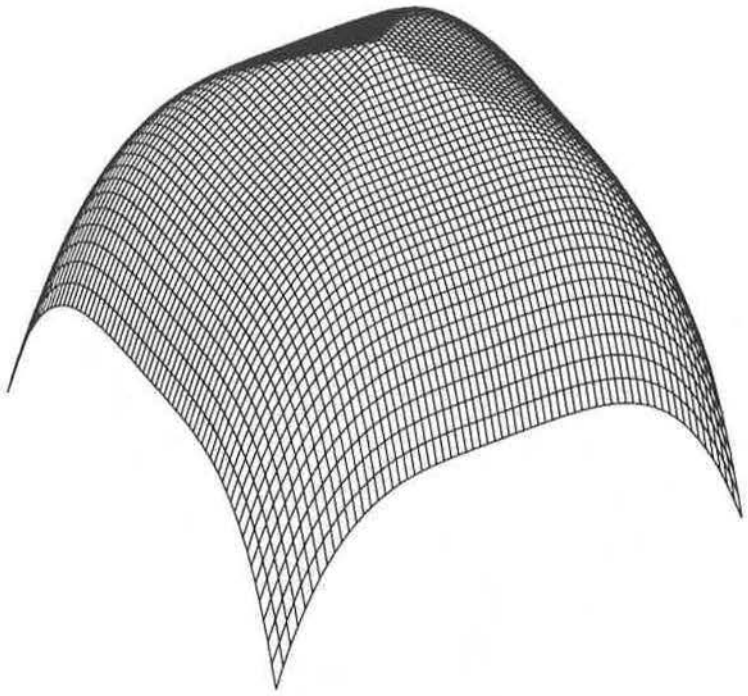
4.5.3 Lambertian Reflectance Map

We now turn our attention to the Lambertian reflectance map:

$$R(p,q) = \max\left\{0,\ \frac{1 + p_0 p + q_0 q}{\sqrt{(1 + p_0^2 + q_0^2)(1 + p^2 + q^2)}}\right\} \tag{4.30}$$

for which R_p and R_q are not linear in p and q as for the Eikonal reflectance map. We expect that this may add a degree of difficulty to the problem.

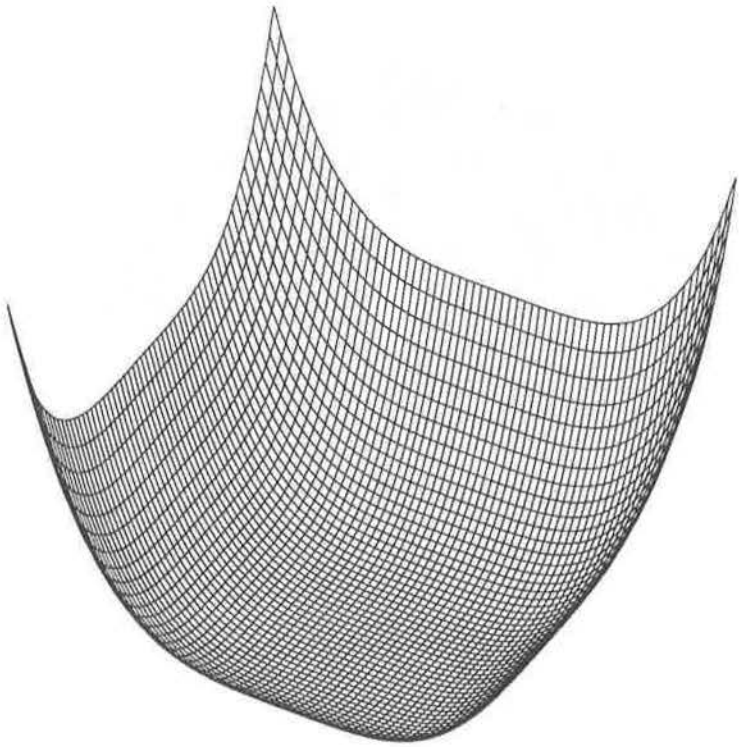
We repeat the tests on the surfaces considered in Section 4.5.2 which give rise to one singular point in the domain when illuminated from overhead. Given the connection between (4.30) when $p_0 = q_0 = 0$ (i.e. overhead illumination) and the Eikonal reflectance



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	1500	500	100
Residual Error	4.402×10^{-5}	3.451×10^{-9}	1.513×10^{-12}
Gradient Error	8.777×10^{-2}	4.584×10^{-2}	2.152×10^{-2}

λ	1×10^{-4}	1×10^{-5}	1×10^{-6}
Iterations	100	100	100
Residual Error	4.864×10^{-16}	4.436×10^{-16}	4.017×10^{-16}
Gradient Error	1.797×10^{-2}	1.759×10^{-2}	1.755×10^{-2}

Figure 4.7: Computed surface: $z = -(x^4 + y^4)$, $\lambda = 1 \times 10^{-6}$, Eikonal map.



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	1500	500	100
Residual Error	3.961×10^{-5}	1.010×10^{-5}	6.973×10^{-5}
Gradient Error	7.768×10^{-2}	2.922×10^{-2}	8.122×10^{-3}

λ	1×10^{-4}	1×10^{-5}	1×10^{-6}
Iterations	100	100	100
Residual Error	3.223×10^{-10}	2.198×10^{-15}	2.119×10^{-15}
Gradient Error	4.316×10^{-3}	3.763×10^{-3}	3.689×10^{-3}

Figure 4.8: Computed surface: $z = x^4 + y^4$, $\lambda = 1 \times 10^{-6}$, Eikonal map.

map (4.7) we may expect that the results will not differ from those of the previous section. However, these tests will shed an important insight on the class of surfaces which give rise to the spurious results observed in the previous section. First we try the surface

$$z = \sin(\pi x)\sin(\pi y), \quad x, y \in [0.1, 0.9]. \tag{4.31}$$

Recall that in the previous section, the surface appeared to be inverted in the neighbourhood of the singular point. The computed surface in this case can be found in Figure 4.9. Notice that the inversion in the surface is no longer evident. The error in the computed surface is 1.636×10^{-2}.

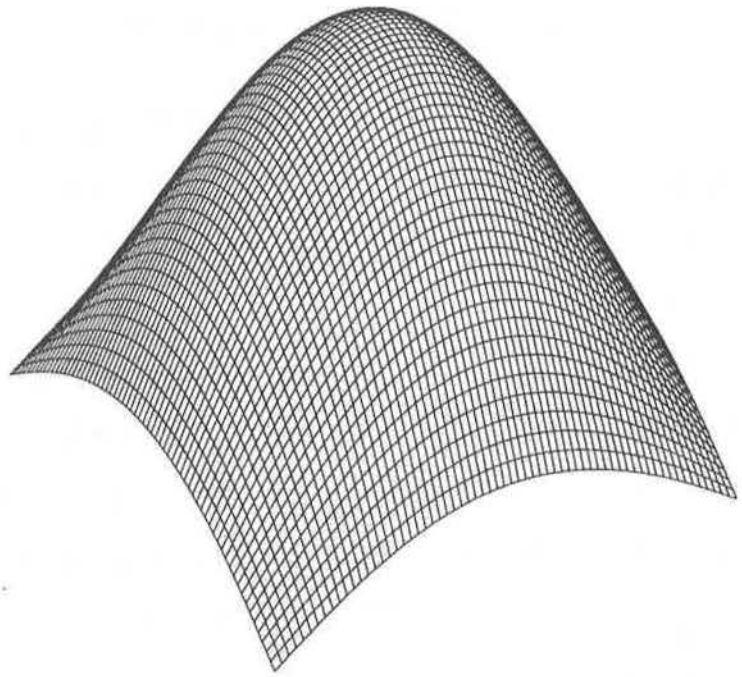
The result of replacing z by $-z$ and recomputing the surface can be found in Figure 4.10. For the Eikonal map, the surface in this case was approximated correctly with no evident flaws, however, for the Lambertian reflectance map we notice an inversion in the surface close to the singular point. In spite of the obvious flaw in the computed surface, the error is in fact smaller than in the previous test at 8.208×10^{-3}.

We also tested the surface

$$z = -(x^4 + y^4), \quad x, y \in [-0.5, 0.5] \tag{4.32}$$

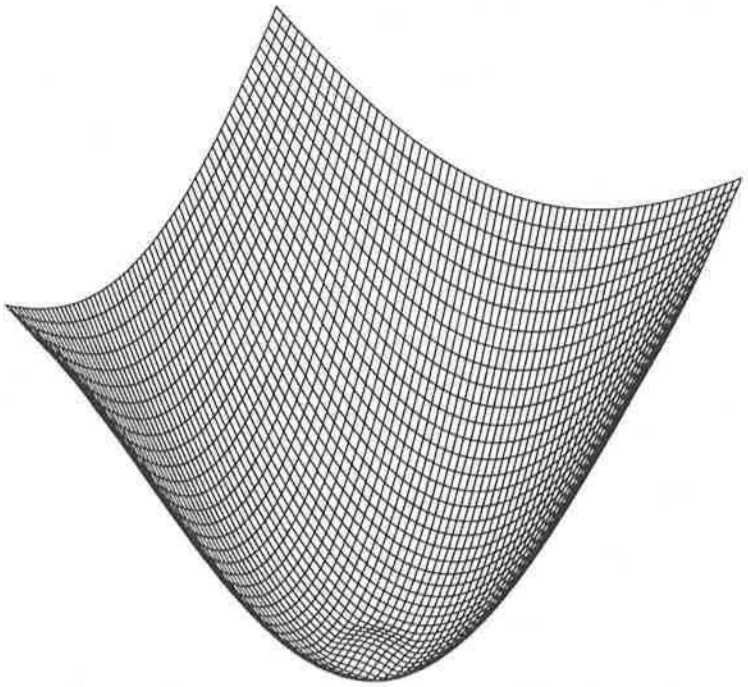
with similar results. The algorithm computed the surface with no obvious flaws in contrast to what occurred for the Eikonal reflectance map. When the surface was inverted, the algorithm produced a surface with creases in it, similar to those found in Figure 4.7.

It would therefore appear that the inversion and creasing of the surfaces is not linked to the nature of the *surface* at the singular point, that is, whether or not we have a local minimum or local maximum at the singular point, but rather to the nature of the *flow* of characteristics nearby the singular point. For both the Eikonal and the Lambertian reflectance maps we observed spurious solutions when the characteristics flow *towards* the singular point, i.e. when the singular point can be thought of as a sink in the flow.



λ	0.1	1×10^{-2}	1×10^{-3}	1.0×10^{-4}
Iterations	1500	500	100	500
Residual Error	1.323×10^{-3}	3.544×10^{-4}	1.096×10^{-8}	Not converged
Gradient Error	2.445×10^{-1}	1.622×10^{-1}	4.487×10^{-2}	–

Figure 4.9: Computed surface: $z = \sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-3}$, Lambertian map, overhead illumination.



λ	0.1	1×10^{-2}	1×10^{-3}	1.0×10^{-4}
Iterations	1500	500	100	500
Residual Error	1.332×10^{-4}	1.477×10^{-6}	4.490×10^{-9}	Not converged
Gradient Error	1.916×10^{-1}	8.499×10^{-2}	2.833×10^{-2}	–

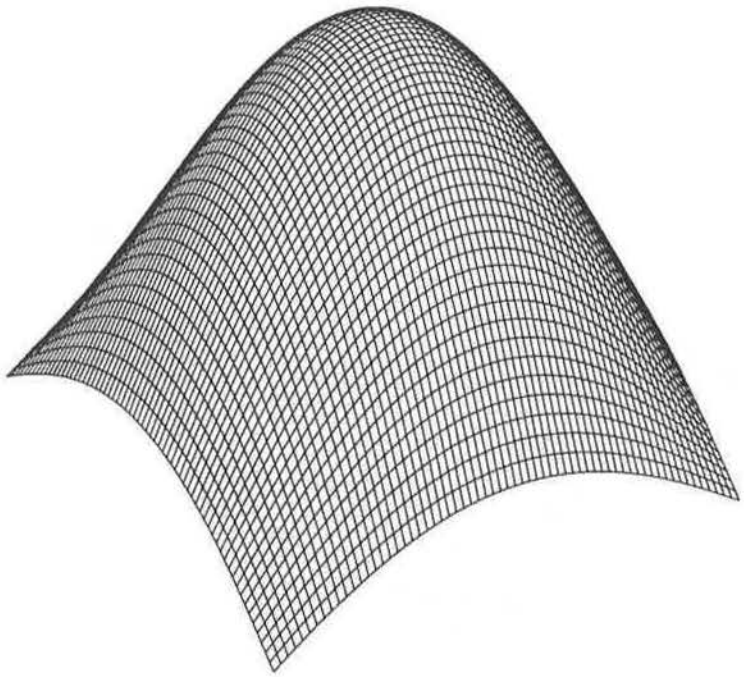
Figure 4.10: Computed surface: $z = -\sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-3}$, Lambertian map, overhead illumination.

We now hypothesize that our algorithm is prone to producing spurious results when the singular point is a *sink* in the flow of the characteristic strips. However, if the singular point is a *source* (so that characteristics flow *away* from the point), then the algorithm appears to produce reasonably good approximations to the solution.

We further test this hypothesis on surfaces that are not illuminated from overhead. In such cases, we define a new coordinate system having height measured in the direction of the light source. Oliensis [37][38] has shown that singular points now correspond to critical points of the surface with respect to this new coordinate system. He also points out that for the Lambertian reflectance map, characteristic curves follow curves of steepest descent with respect to this new coordinate system. Our hypothesis therefore suggests that surfaces having a local minimum with respect to the new coordinate system may not be computed correctly by our algorithm.

We begin by considering the surface (4.28) illuminated from the direction $(0.25, 0, 1)$. This gives rise to one singular point in the image at the point near $(0.525, 0.5)$. This choice of light source direction also ensures that the surface is illuminated everywhere so that we do not face additional difficulties due to regions in the domain in which no information is given about the surface. This point corresponds to a local maximum on the surface in a coordinate system which measures height in the direction $(0.25, 0, 1)$. This singular point can also be thought of as a source in the flow. The surface was computed with error 1.119×10^{-2} and can be found in Figure 4.11. As our hypothesis suggests, there are no serious errors.

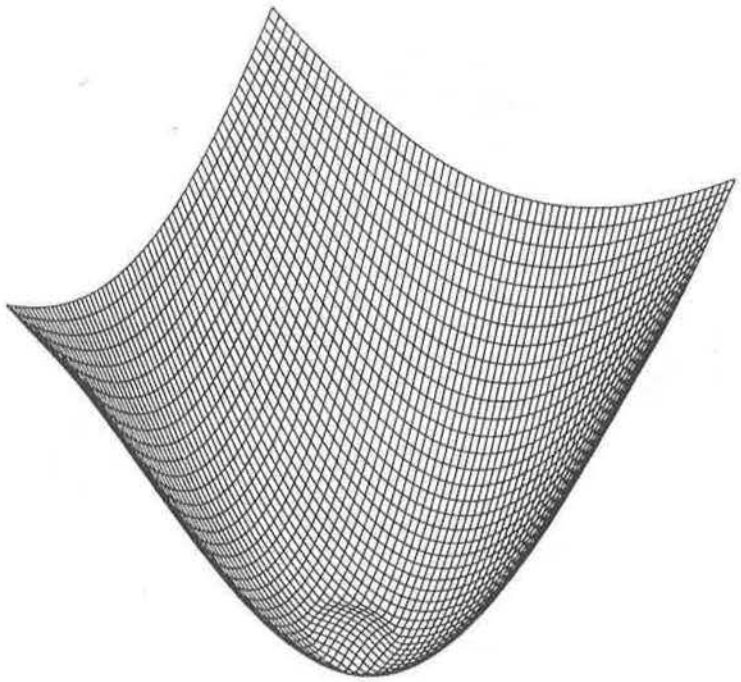
When the surface is inverted, there is now a surface minimum corresponding to the singular point which acts as a sink in the flow of characteristics. The surface was computed with error 6.539×10^{-3} which is again smaller than in the previous test, however, an inversion has appeared in the surface at the location of the singular point. The computed surface is shown in Figure 4.12. This observation fits our hypothesis.



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	2000	500	100
Residual Error	9.929×10^{-4}	8.608×10^{-4}	6.020×10^{-7}
Gradient Error	2.428×10^{-1}	1.688×10^{-1}	4.705×10^{-2}

λ	1×10^{-4}	1×10^{-5}
Iterations	200	100
Residual Error	7.436×10^{-5}	Diverged
Gradient Error	3.185×10^{-2}	–

Figure 4.11: Computed surface: $z = \sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-4}$, Lambertian map, light source direction $(0.25, 0, 1)$.



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	2000	500	100
Residual Error	6.305×10^{-4}	9.945×10^{-5}	4.134×10^{-6}
Gradient Error	1.922×10^{-1}	8.763×10^{-2}	3.208×10^{-2}

λ	1×10^{-4}	1×10^{-5}
Iterations	200	500
Residual Error	8.354×10^{-5}	Not converged
Gradient Error	2.352×10^{-2}	–

Figure 4.12: Computed surface: $z = -\sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-4}$, Lambertian map, light source direction $(-0.25, 0, 1)$.

In the previous test, the singular point was located fairly close to the point $(0.5, 0.5)$ at which the surface has a local minimum (with respect to the standard coordinate system). It could therefore be argued that it is the surface geometry at critical points of the standard coordinate system that is responsible for causing such errors in the solution. We address this issue by considering a slight modification to the surface just considered, namely

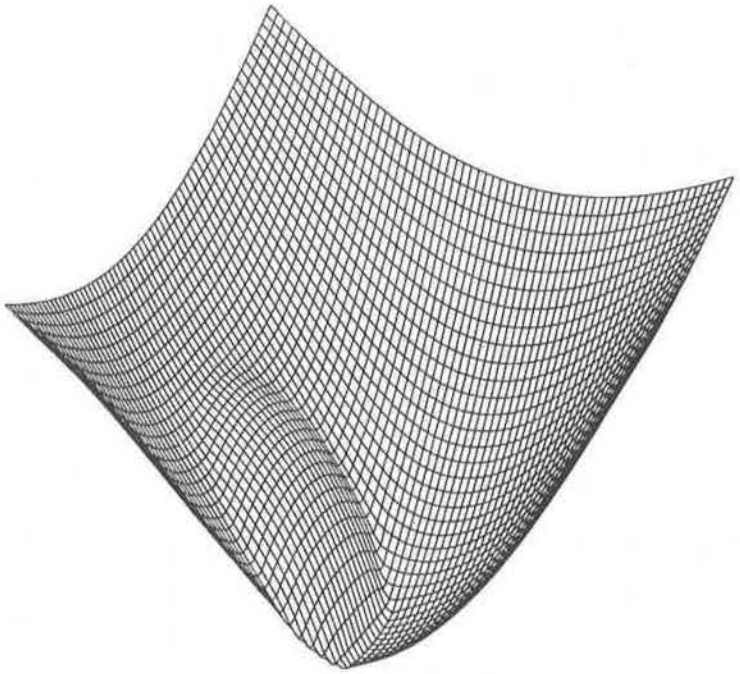
$$z = -\frac{1}{\pi}\sin(\pi x)\sin(\pi y), \qquad x, y \in [0.1, 0.9]. \tag{4.33}$$

The light source direction is given by $(-0.5, 0, 1)$ so that a singular point exists at the point $(0.6\dot{6}, 0.5)$ and the surface is illuminated everywhere. This point corresponds to a surface minimum with respect to a coordinate system where height is measured in the direction $(-0.5, 0, 1)$. The computed surface had a resulting error of 1.410×10^{-2} and is shown in Figure 4.13. The inversion in the surface is clearly evident and is located in the neighbourhood of the singular point $(0.6\dot{6}, 0.5)$.

Finally we consider the surface (4.29) in order to see how the creases in the surface are affected by oblique illumination. The creases occur only for the inverted surface, $-z$. The light source direction was given by $(-1/16, 0, 1)$ so that there is only one singular point at $(1/4, 0)$ and the surface is illuminated everywhere. The surface was computed with error 5.123×10^{-3} and is shown in Figure 4.14. The creases in the surface are again apparent but now they appear to be focussing at a point in the neighbourhood of the singular point $(1/4, 0)$.

4.5.4 Saddle Points

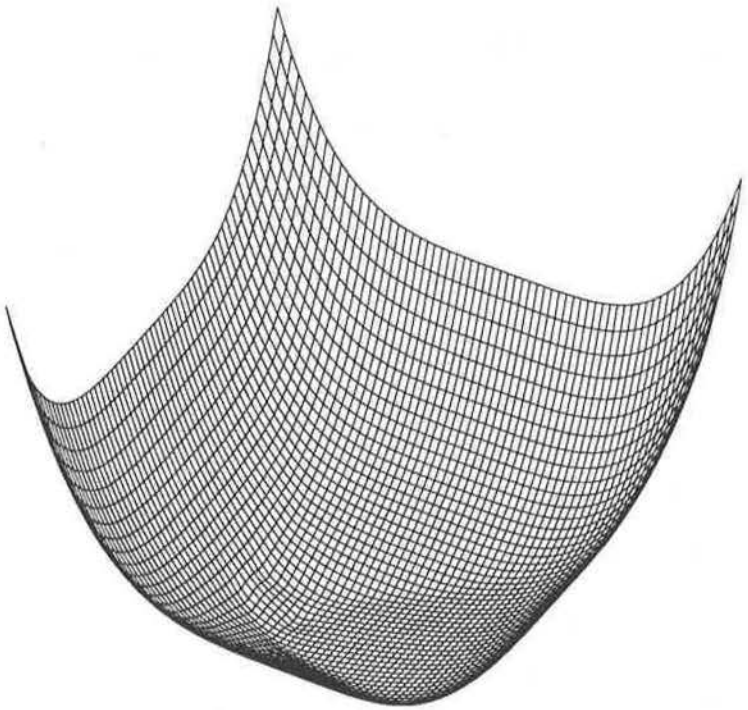
We have also tested saddle singular points where two characteristics flow towards and two flow away from the point. In this sense, saddle points exhibit some of the behaviour of both sources and sinks. In tests conducted on surfaces having a saddle point, creases



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	2000	1000	500
Residual Error	6.626×10^{-4}	2.471×10^{-5}	4.173×10^{-4}
Gradient Error	6.472×10^{-2}	3.150×10^{-2}	5.352×10^{-2}

λ	1×10^{-4}	1×10^{-5}
Iterations	100	500
Residual Error	2.542×10^{-12}	Not converged
Gradient Error	5.039×10^{-2}	–

Figure 4.13: Computed surface: $z = -\sin(\pi x)\sin(\pi y)$, $\lambda = 1 \times 10^{-4}$, Lambertian map, light source direction $(-0.5, 0, 1)$.



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	2000	500	200
Residual Error	2.951×10^{-4}	2.686×10^{-5}	1.259×10^{-6}
Gradient Error	8.862×10^{-2}	6.084×10^{-2}	2.395×10^{-2}

λ	1×10^{-4}	1×10^{-5}	1×10^{-6}
Iterations	100	300	300
Residual Error	7.133×10^{-6}	1.235×10^{-8}	2.947×10^{-14}
Gradient Error	1.706×10^{-2}	1.589×10^{-2}	1.579×10^{-2}

Figure 4.14: Computed surface: $z = x^4 + y^4$, $\lambda = 1 \times 10^{-6}$, Lambertian map, light source direction $(-1/16, 0, 1)$.

tend to appear in the computed surface.

4.5.5 Surfaces with more than one singular point

We now consider the "Mexican hat" surface:

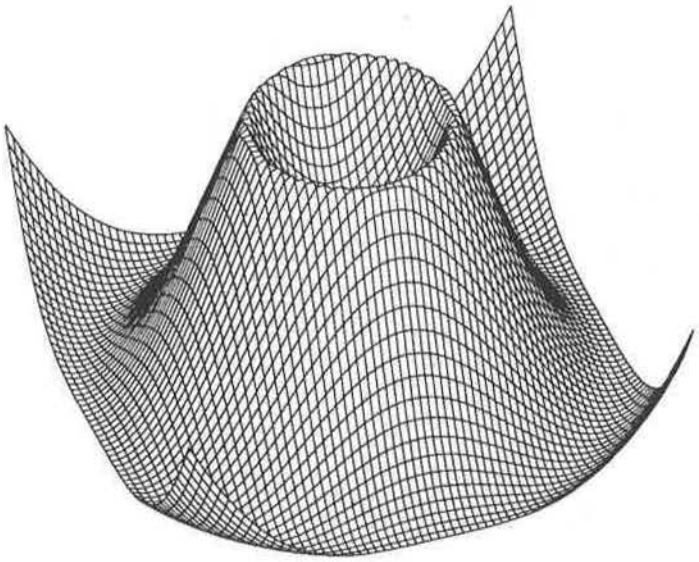
$$z = \frac{\cos(2\pi r)}{2\pi}, \quad r = \sqrt{x^2 + y^2}, \quad x, y \in [-0.5, 0.5] \tag{4.34}$$

considered in the previous two chapters. This surface, when illuminated using an Eikonal reflectance map, yields an isolated singular point at the origin and a "ring" of singularities at $r = 0.5$. The computed surface is shown in Figure 4.15. The error in the computed surface height was 4.326×10^{-2}. The singular point at the origin acts as a sink in the flow of characteristics while those at $r = 0.5$ act as sources. Notice that the solution is inverted in the neighbourhood of the sink.

We then replace z by $-z$ and recompute the surface. The result is shown in Figure 4.16. The error in the computed surface height was much larger than in the previous test at 2.296×10^{-1}. We now have a ring of sinks at $r = 0.5$. The inversion in the surface at $r = 0.5$ is clearly evident. This inversion results in the creation of another sink at the origin which yields yet another inversion in the surface near the origin. The computed surface is clearly unacceptable even though the exact surface is not particularly complicated.

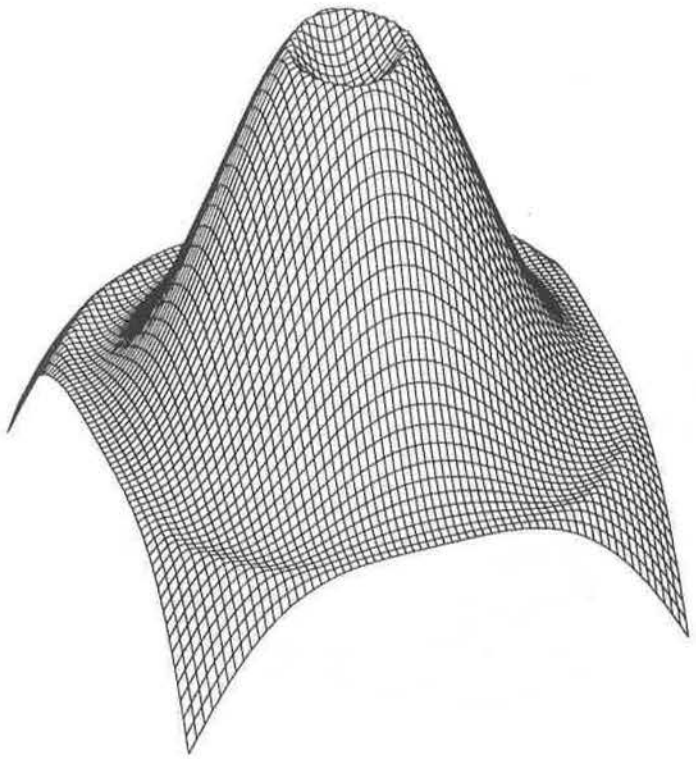
4.5.6 Discussion

Our experimental evidence has shown that in general, singular points give rise to problems for our computational algorithm. More specifically, when the singular point corresponds to a sink in the flow of characteristics then the computed surface appears either to be inverted in a neighbourhood of the singular point or to have creases on the surface which focus at the singular point. In the next section we seek to explain these phenomena.



λ	0.1	1×10^{-2}	1×10^{-3}	1×10^{-4}
Iterations	1500	500	200	500
Residual Error	4.685×10^{-5}	9.830×10^{-4}	6.868×10^{-5}	Not converged
Gradient Error	2.362×10^{-1}	1.257×10^{-1}	1.177×10^{-1}	–

Figure 4.15: Computed surface: $z = \frac{1}{2\pi}\cos(2\pi r)$, $\lambda = 1 \times 10^{-3}$, Eikonal map.



λ	0.1	1×10^{-2}	1×10^{-3}	1×10^{-4}
Iterations	1500	1000	200	500
Residual Error	5.358×10^{-6}	7.179×10^{-6}	5.957×10^{-5}	Not converged
Gradient Error	6.140×10^{-1}	6.630×10^{-1}	6.404×10^{-1}	–

Figure 4.16: Computed surface: $z = -\frac{1}{2\pi}\cos(2\pi r)$, $\lambda = 1 \times 10^{-3}$, Eikonal map.

4.6 The One Dimensional Problem

4.6.1 Problem formulation

We now seek an explanation for the curious results obtained in the previous section. We first consider a one dimensional analogue to our two dimensional regularized shape from shading problem, namely

$$-\lambda p'' + R_p p' = E', \quad x \in [0,1] \tag{4.35}$$

subject to the boundary conditions

$$p(0) = \alpha, \quad p(1) = \beta \tag{4.36}$$

where $p = p(x)$ and $E = E(x)$. We look for solutions to (4.35) in the limit as $\lambda \to 0$ and are therefore dealing with what is known as a singular perturbation problem, see Chang & Howes [14]. More precisely, we face nonlinear singular perturbation problems with turning points. Notice that in the limit as $\lambda \to 0$, equation (4.35) becomes

$$R_p p' = E' \tag{4.37}$$

which is known as the *reduced problem.* In our case, the reduced problem is the problem we actually wish to solve. We resist the temptation of seeking approximations to the conservation form $R'(p) = E'$ since it has no parallel in the two dimensional case. A solution to this equation cannot, in general, be expected to satisfy both boundary conditions specified in (4.36). Hence, boundary layer or interior layer phenomena can be observed in the solution of (4.35) subject to the boundary conditions (4.36).

The behaviour of the solution for very small λ can be defined in terms of solutions to the left and right reduced problems. We say that p_l (p_r) is a solution of the left (right) reduced problem if it is a solution of (4.37) subject to the left (right) boundary condition $p(0) = \alpha$ ($p(1) = \beta$).

In order to tie-in with the concept of characteristics, equation (4.37) can be written

$$\dot{p} = E' \tag{4.38}$$

$$\dot{x} = R_p \tag{4.39}$$

where a dot denotes differentiation with respect to a parameter ξ. Here $x(\xi)$ is a parametrization of the line segment $0 \leq x \leq 1$. We can now see that the term R_p specifies the direction of flow of the "characteristic". If $R_p > 0$ (< 0), $\forall x \in [0,1]$ then the characteristic curve flows from left to right (right to left) on the interval $[0,1]$. Of particular interest is the case where R_p changes sign at some point $x_0 \in (0,1)$. In this case, the characteristics will flow away from or towards the point x_0 depending on the sign of R_p on either side of x_0.

For a linear reflectance map, $R_p = p_0$, is a non-zero constant. Our problem now becomes

$$\lambda p'' + p_0 p' = E' \tag{4.40}$$

with boundary conditions

$$p(0) = \alpha, \quad p(1) = \beta. \tag{4.41}$$

The reduced problem in this case is

$$p_0 p' = E' \tag{4.42}$$

which has a fixed direction for the flow of characteristics determined by the sign of the constant p_0.

Let $0 < \delta \ll 1$ then for $p_0 > 0$

$$\lim_{\lambda \to 0} p(\lambda, x) \to p_l(x), \quad 0 \leq x \leq 1 - \delta < 1 \tag{4.43}$$

with a boundary layer at $x = 1$. However, for $p_0 < 0$

$$\lim_{\lambda \to 0} p(\lambda, x) \to p_r(x), \quad 0 < \delta \leq x \leq 1 \tag{4.44}$$

with a boundary layer at $x = 0$. Hence, if the characteristic direction is from left to right then the solution of the left reduced problem dominates whereas, if the flow of characteristics is reversed, the solution of the right reduced problem dominates.

In the case of a non-linear reflectance map, things are not as straightforward since now the flow of characteristics depends on the computed solution. Consider the Eikonal reflectance map for which $R_p = 2p$. Our problem now becomes

$$\lambda p'' + 2pp' = E' \tag{4.45}$$

so the coefficient of p', which determines the direction of flow of the characteristics, now depends on p. When $p > 0$, flow is in the direction of increasing x, while for $p < 0$ characteristics flow in the opposite direction. The case of $p = 0$ corresponds to a singular point or *turning point* in the language of singular perturbation phenomena. See Chang & Howes [14] for a discussion of the homogeneous form of (4.45). Note that we are *not* interested in linear turning point problems for which the flow of characteristics does not depend on the solution.

The reduced problem in this case is given by

$$2pp' = E' \tag{4.46}$$

which can be written

$$(p^2)' = E' \tag{4.47}$$

and hence integrated to yield

$$p^2 = E + C. \tag{4.48}$$

Here C is an arbitrary constant which, in general, can be chosen so as to satisfy only one of the boundary conditions. For the shape from shading problem, we choose the boundary conditions so that for $C = 0$ *both* are satisfied. We therefore have

$$p = \pm\sqrt{E}. \tag{4.49}$$

Assuming that there is only one singular point in the domain, the sign of p can be chosen so as to satisfy the boundary condition on each side of the singular point independently. So, for example if $p(-1) = \alpha > 0$ and $p(1) = \beta < 0$ we set $p = \sqrt{E}$ on the interval $[-1, s]$ and $p = -\sqrt{E}$ on the interval $[s, 1]$ where s is such that $p(s) = 0$, that is, s is the location of the singular point. In short, no interior or boundary layers appear in the exact solution of this singular perturbation problem in the limit as $\lambda \to 0$.

In the case of a Lambertian reflectance map, the above arguments can be generalized. The direction of the flow of characteristics is again governed by the sign of R_p which is now given by

$$R_p = \frac{p_0 - p}{\sqrt{(1 + p_0^2)(1 + p^2)^3}} \tag{4.50}$$

where p_0 specifies the light source direction.

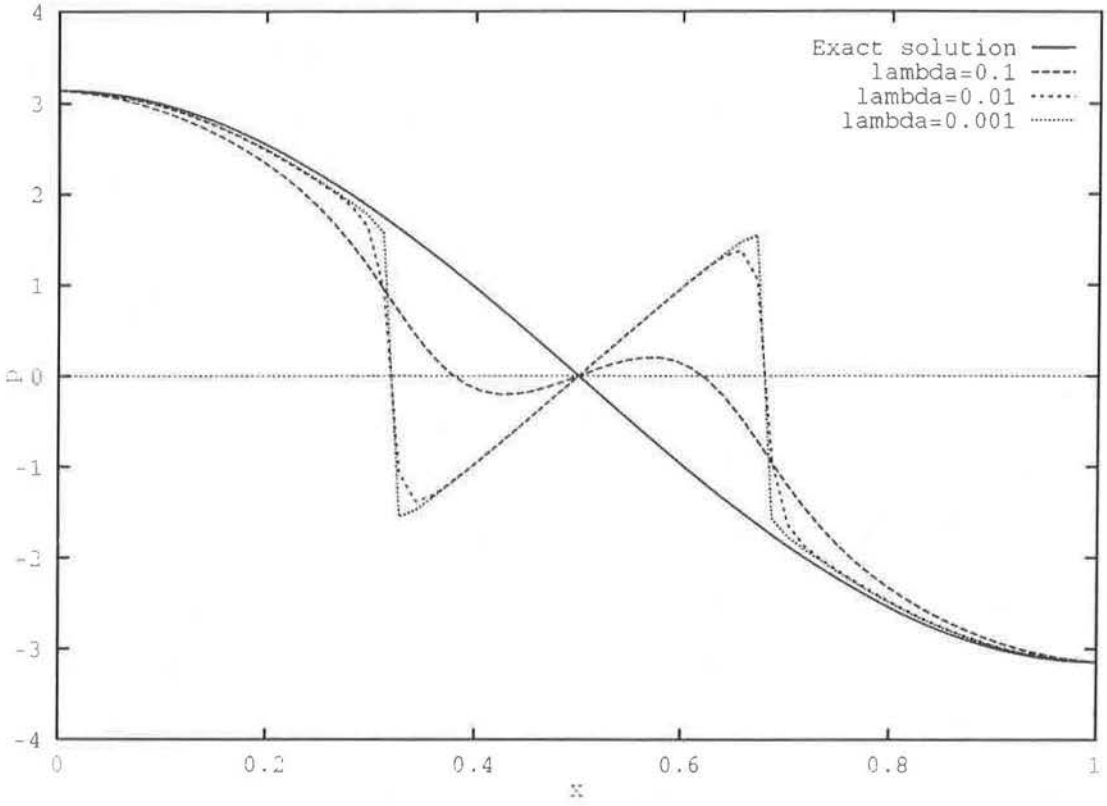
4.6.2 Results: One dimensional problem

We now test the one dimensional version of our algorithm on the problem outlined above. We begin by considering the case of the Eikonal reflectance map. We will focus only on those problems for which spurious results were obtained in the previous section. First consider the curve

$$z = \sin(\pi x), \qquad x \in [0, 1] \tag{4.51}$$

which has a local maximum, and therefore a singular point, at $x = 0.5$. This singular point corresponds to a sink in the flow of characteristics, that is, characteristics flow towards $x = 0.5$ from both sides.

We solve equation (4.45) using Gauss-Seidel with red-black ordering on a uniform grid with step-size $1/64$. We use a continuation algorithm in λ as for the two dimensional case. Initially we set p to zero on the interior of the domain. The result is shown in Figure 4.17.



λ	0.1	1.0×10^{-2}	1.0×10^{-3}	1.0×10^{-4}
Iterations	2000	100	100	2000
Residual error	2.862×10^{-4}	6.594×10^{-14}	7.908×10^{-11}	Diverged
Error in p	5.976×10^{-1}	1.152	1.191	–

Figure 4.17: Computed solutions : $z = \sin(\pi x)$, Eikonal map.

Notice the inversion in the solution in the neighbourhood of the singular point at $x = 0.5$. The behaviour of the computed solution to the one dimensional problem therefore appears to be analogous to the two dimensional problem.

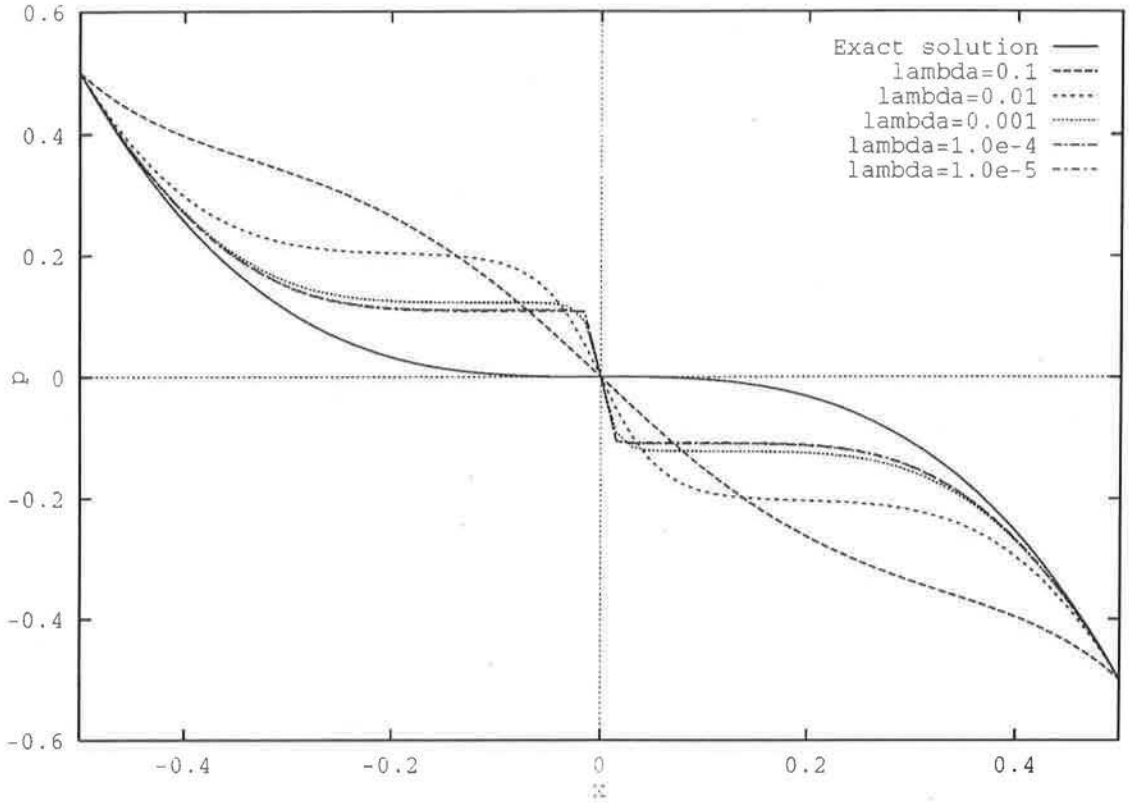
We also tested the curve $z = -x^4$ on $[-0.5, 0.5]$ using the same algorithm. The results are shown in Figure 4.18. This time an internal transition layer appears in the solution at the location of the singular point ($x = 0$). Such a transition layer causes a discontinuity in the first derivative of z and would therefore reveal itself as a "jump" in the slope of the computed curve. For the two dimensional problem, this would translate to a crease in the surface as was observed in the tests on the surface $z = -(x^4 + y^4)$.

Given that we are dealing with a non-linear iteration we now address the question of the choice of initial guess.

4.6.3 Choice of initial guess

For the tests conducted above, we used a zero initial guess. This may be a particularly bad choice since for the Eikonal map or Lambertian map with overhead illumination, every point in the interior of the domain is initially a singular point. We begin by starting with the exact solution as an initial guess and choose a very small value for λ. One may hope that under these circumstances, our algorithm will converge to a solution that is a good approximation to the exact solution. Note however, that the "exact" solution is exact for (4.37),(4.36), not the corresponding difference equations.

The result of applying this test to the curve $z = \sin(\pi x)$ on $[0, 1]$ with $\lambda = 1.0 \times 10^{-6}$ is shown in Figure 4.19. After 3000 iterations, the algorithm had not converged, the residual error was still O(10). We see that the computed solution, despite the very good initial guess, still demonstrates an inversion near the singular point. At least the iteration did not converge this time. The choice of initial guess would therefore appear not to be the cause of the problem.



λ	0.1	1×10^{-2}	1×10^{-3}
Iterations	1000	200	100
Residual Error	1.459×10^{-4}	1.215×10^{-5}	7.422×10^{-16}
Gradient Error	1.731×10^{-1}	1.272×10^{-1}	7.818×10^{-2}

λ	1×10^{-4}	1×10^{-5}
Iterations	100	100
Residual Error	4.203×10^{-16}	8.640×10^{-16}
Gradient Error	7.045×10^{-2}	6.960×10^{-2}

Figure 4.18: Computed solutions: $z = -x^4$, Eikonal map.

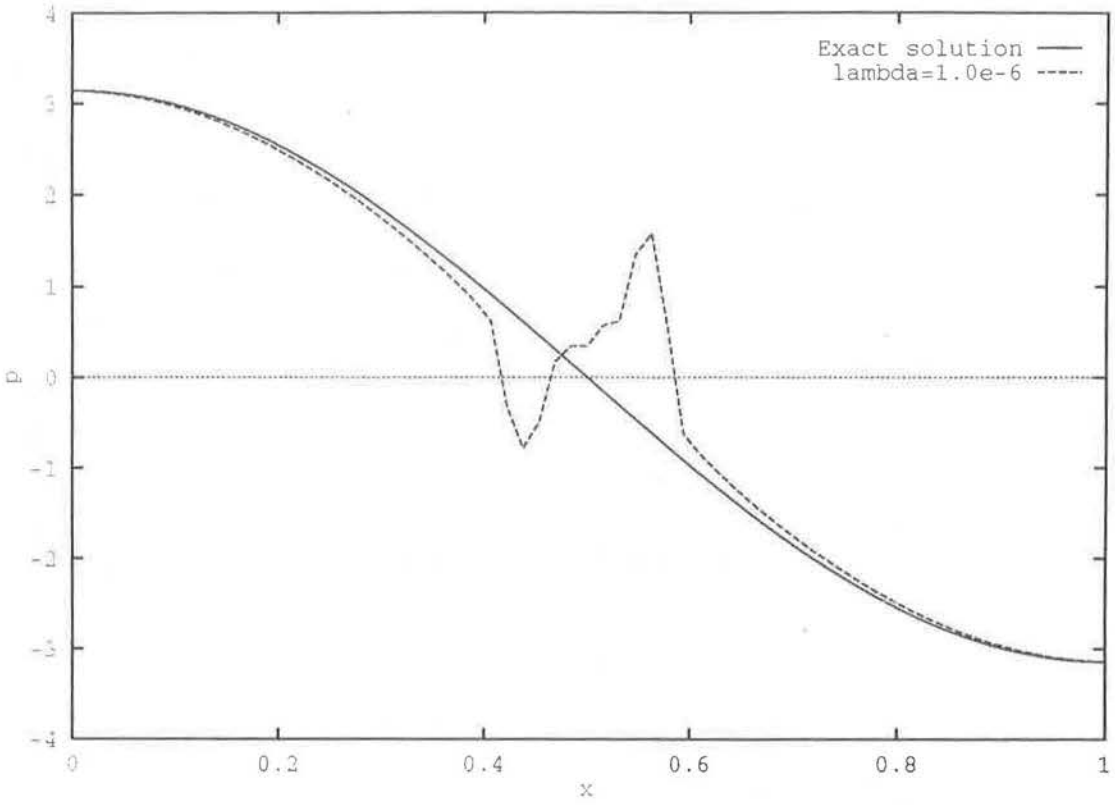


Figure 4.19: Computed solution, exact solution as initial guess, $z = \sin(\pi x)$, Eikonal map.

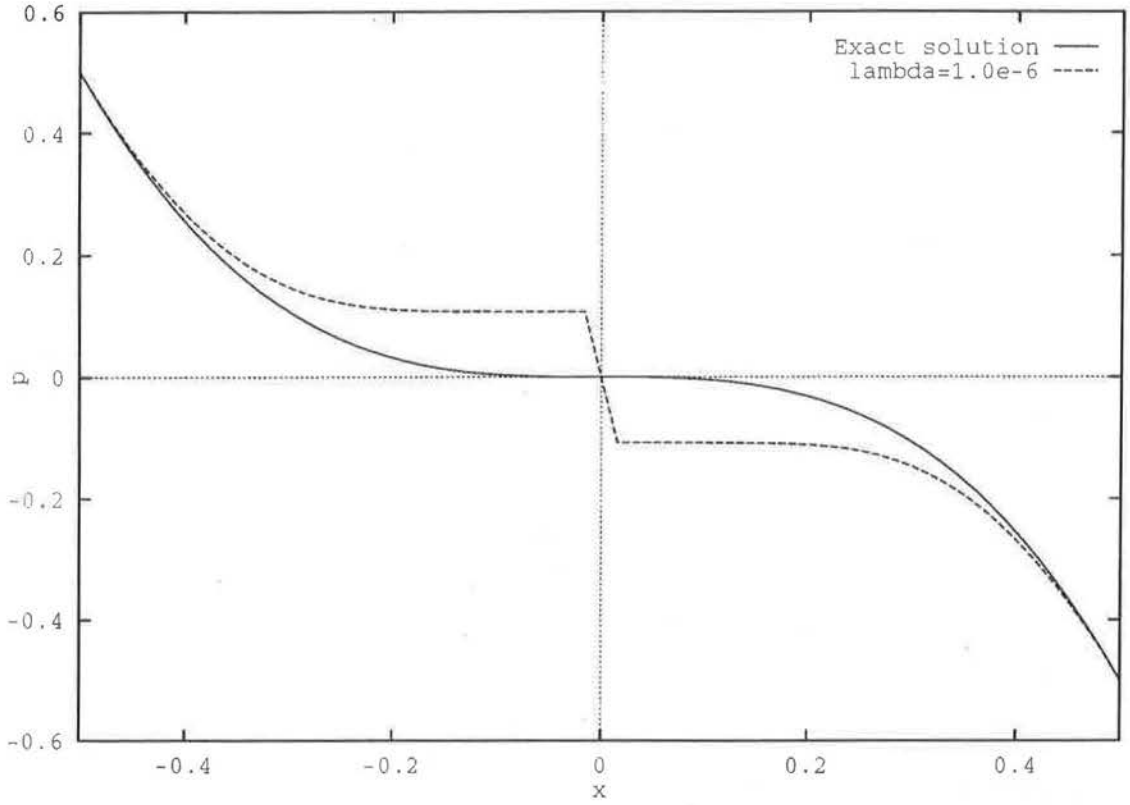


Figure 4.20: Computed solution, exact solution as initial guess, $z = -x^4$, Eikonal map.

We also tested the curve $z = -x^4$, $x \in [-0.5, 0.5]$ starting with the exact solution as initial guess and taking $\lambda = 1.0 \times 10^{-6}$. Again the computed solution shows a transition layer at the singular point. The algorithm in fact converged with residual error 7.558×10^{-16} after 100 iterations. The computed solution is shown in Figure 4.20.

We must now explain why, when given such a good initial guess and such a small value of λ, our algorithm moves so far away from the exact solution and may even converge to a poor approximation to the solution.

4.6.4 Analysis

We consider the equation

$$-\lambda p'' + 2pp' = E' \tag{4.52}$$

which when discretized using a simple upwinding scheme yields

$$-\lambda\frac{p_{i+1} - 2p_i + p_{i-1}}{h^2} + 2|p_i|\frac{p_i - p_{ew}}{h} = \frac{E_{i+1} - E_{i-1}}{2h} \tag{4.53}$$

Given that $E = p^2$, we see that the ratio of the artificial viscosity term to the other terms is

$$\frac{\lambda}{hp} \tag{4.54}$$

Hence, for the tests conducted in the previous section where $p = O(1)$, $h = 1/64$ and $\lambda = 1.0 \times 10^{-6}$ this ratio is 6.4×10^{-5}. It is therefore fair to conclude that the artificial viscosity term can effectively be neglected in our analysis, except in the neighbourhood of singular points where $p \simeq 0$.

Given that we started with the exact solution as initial guess, during the first iteration we are essentially solving the equation

$$2p_e p' = E' \tag{4.55}$$

where p_e is the exact solution. Assume without loss of generality that we wish to solve this equation on the domain $[-1, 1]$ and that the singular point is located at $x = 0$. We also assume that

$$p_e = \begin{cases} > 0 & \text{if } x < 0 \\ < 0 & \text{if } x > 0 \end{cases} \tag{4.56}$$

since this characterizes the curves for which our algorithm has produced spurious solutions.

Consider the left half of the domain. On this interval $p_e > 0$ and so characteristics flow from left to right. Choose a grid point x_i in this interval which is close to the singular

point. At this point we have

$$p_i = p_{i-1} + h\frac{E_i'}{2p_e(x_i)} \tag{4.57}$$

where

$$E_i' = \frac{E_{i+1} - E_{i-1}}{2h}. \tag{4.58}$$

A similar relation holds for p_{i-1} and so

$$p_i = p_{i-2} + h\sum_{k=1}^{2}\frac{E_k'}{2p_e(x_k)}. \tag{4.59}$$

Continuing in this fashion we have

$$p_i = p_0 + h\sum_{k=1}^{i}\frac{E_k'}{2p_e(x_k)}. \tag{4.60}$$

This can be interpreted as a discrete approximation to

$$p(x_i) = p(x_0) + \int_{x_0}^{x_i}\frac{E'(x)}{2p_e(x)}\,dx \tag{4.61}$$

which is the solution to the continuous problem (4.55). Hence

$$h\sum_{k=1}^{i}\frac{E_k'}{2p_e(x_k)} \simeq \int_{x_0}^{x_i}\frac{E'(x)}{2p_e(x)}\,dx = \int_{x_0}^{x_i} p'(x)\,dx \tag{4.62}$$

or, simply, on each interval

$$h\frac{E_i'}{2p_e(x_i)} \simeq \int_{x_{i-1}}^{x_i} p'(x)\,dx. \tag{4.63}$$

Now, when $p'' < 0$, as is the case for the curve $z = \sin(\pi x)$, on each interval we in fact have

$$h\frac{E_i'}{2p_e(x_i)} < \int_{x_{i-1}}^{x_i} p'(x)\,dx < 0 \tag{4.64}$$

and so we consistently compute discrete approximations which are larger in absolute value than they should be. As we approach the singular point, our estimate of p will therefore be smaller than the true value of zero. At the next iterate we have

$$h\frac{E_i'}{2p_i^{(1)}} \simeq \int_{x_{i-1}}^{x_i} p'(x)\,dx \tag{4.65}$$

where $p_i^{(1)}$ represents the discrete estimate of $p(x_i)$ from the first iterate. Now,

$$h\frac{E_i'}{2p_i^{(1)}} < h\frac{E_i'}{2p_e(x_i)} < \int_{x_{i-1}}^{x_i} p'(x)\,dx \tag{4.66}$$

and so the error resulting from the next iterate is of the same sign and larger in magnitude than the previous error. The argument can be repeated for all subsequent iterates and so we conclude that the absolute value of the error increases monotonically with the number of iterates.

This has a disastrous effect on the algorithm since the direction of flow of the characteristics is artificially changed. In essence, a singular point is created in the wrong place. The same phenomenon occurs in the right half of the domain except that now the discrete approximations are positive and larger than they should be. These artificial singular points create a region in the middle of the domain in which the flow of characteristics is reversed. This causes the inversion layer in the computed solution.

However, if $p'' > 0$, as is the case for the curve $z = -x^4$, on each interval we have

$$0 > h\frac{E_i'}{2p_e(x_i)} > \int_{x_{i-1}}^{x_i} p'(x)\,dx \tag{4.67}$$

and so we consistently compute discrete approximations which are too small in absolute value. As we approach the singular point, our estimate of p will therefore be larger than the true value of zero. For the next iterate,

$$h\frac{E_i'}{2p_i^{(1)}} > h\frac{E_i'}{2p_e(x_i)} > \int_{x_{i-1}}^{x_i} p'(x)\,dx \tag{4.68}$$

and so again we see that the next iterate produces an error of the same sign and greater in magnitude than the first iterate. This argument carries over to subsequent iterations and so the error again increases monotonically in absolute value.

The same occurs on the right half of the domain. A transition layer thus occurs at the singular point to connect the two solutions. This is exactly what is observed for the curve $z = -x^4$.

For the inverted curves (which correspond to surfaces in the two dimensional case where spurious results were not observed), the error does *not* increase monotonically as the iteration proceeds from the exact solution. A typical cancellation in errors resulting in a more random appearance, is therefore possible. This could explain why boundary layer effects are not as evident for these curves as interior layers are for the cases considered above.

In Figure 4.21 we plot the error between the computed solution and the exact solution at the point $x = -0.25$ for the curve $z = -x^4$. The error is plotted against the number of iterations performed starting from the exact solution and with $\lambda = 1 \times 10^{-6}$. Notice that the error increases monotonically as predicted.

In Figure 4.22 we repeat the test but for the inverted curve $z = x^4$. Notice that now the error does not increase monotonically and that upon convergence it is much smaller than in the previous test.

These arguments carry over easily to the case of a Lambertian reflectance map and overhead illumination provided $|p|$ is not too large. The main difference is that problems can now be expected to occur for curves satisfying

$$p_e = \begin{cases} < 0 & \text{if } x < 0 \\ > 0 & \text{if } x > 0 \end{cases} \tag{4.69}$$

assuming that the singular point is located at $x = 0$. In such cases

$$R_p(p_e) = \begin{cases} > 0 & \text{if } x < 0 \\ < 0 & \text{if } x > 0 \end{cases} \tag{4.70}$$

and so the singular point acts as a sink in the flow of characteristics. The requirement that $|p|$ is not too large stems from the fact that R_p decreases monotonically with p only for $|p| \leq 0.5$, approximately. The argument that for sinks, the error in the solution increases monotonically from the exact solution, requires R_p to be monotonic in p. Note that this requirement is met for the curves $z = \pm x^4$ on $[-0.5, 0.5]$.

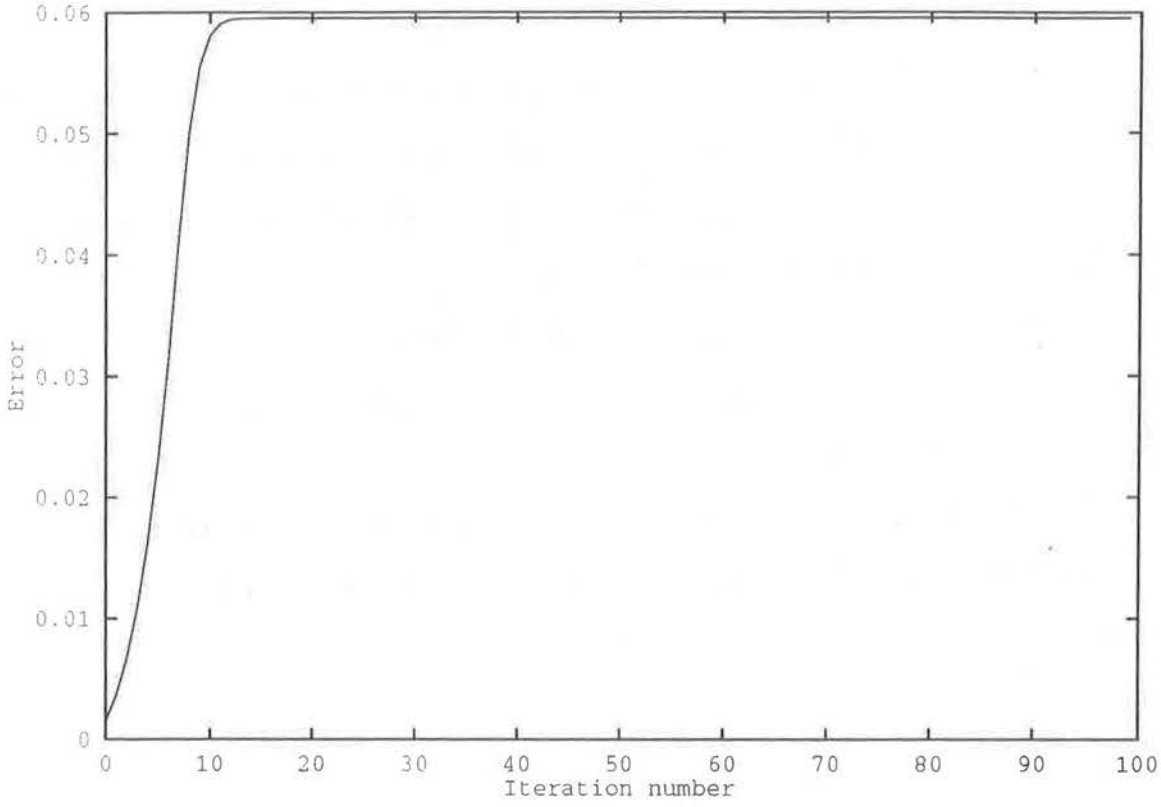


Figure 4.21: Error against number of iterations: $z = -x^4$, Eikonal map, exact solution as initial guess.

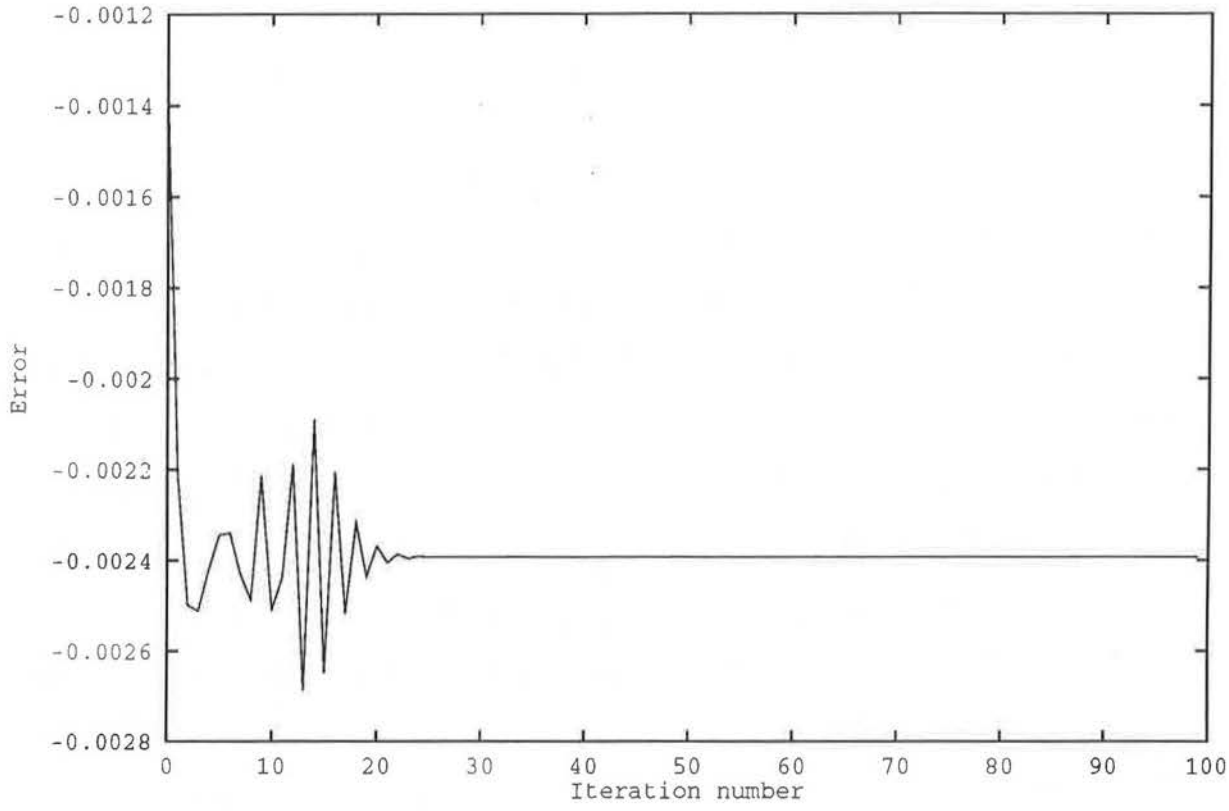


Figure 4.22: Error against number of iterations: $z = x^4$, Eikonal map, exact solution as initial guess.

The case of a Lambertian reflectance map with oblique illumination is less easily analyzed for the curves considered so far. The reason for this is that these curves no longer possess any symmetry properties about the singular point. In particular, the sign of $p''(x)$ is no longer consistent either side of the singular point – this feature was crucial to the earlier analysis. However, by constructing curves with the desired properties, we demonstrate that the same phenomena can occur in practice. Consider, for example, the curve

$$z = -\sin(\pi x) + \pi x \tag{4.71}$$

which when illuminated from the direction specified by setting $p_0 = \pi$, yields a singular point at $x = 0.5$. For this curve on the interval $[0, 1]$ we have $p'' > 0$ to the left of the singular point and $p'' < 0$ to the right of it. We may now expect to see artificial singular points created in the solution. These will show up as zeros of R_p. Due to the large discrepancy in the magnitude of R_p at either end of the interval $[0, 1]$ we choose to solve the problem on the interval $[0.3, 1]$. We start from the exact solution and set $\lambda = 1.0 \times 10^{-6}$. After 3000 iterations the algorithm had not converged. A graph of R_p for the computed solution is shown in Figure 4.23 while the solution is found in Figure 4.24.

An inversion in the sign of R_p in the neighbourhood of the singular point at $x = 0.5$ is clearly noticeable. This results in an anomaly in the solution at the same location.

Similarly we construct a curve which is expected to give rise to a transition layer at the singular point. Consider the curve

$$z = x^4 + 0.5x \tag{4.72}$$

on the interval $[0, 1]$. When illuminated from a direction specified by taking $p_0 = 0.5$, we obtain a singular point at the origin. It is easily verified that p'' is negative to the left of the singular point and positive to the right of it. Using the exact solution as an initial guess and taking $\lambda = 1.0 \times 10^{-6}$ the iteration converged after 100 iterations with residual

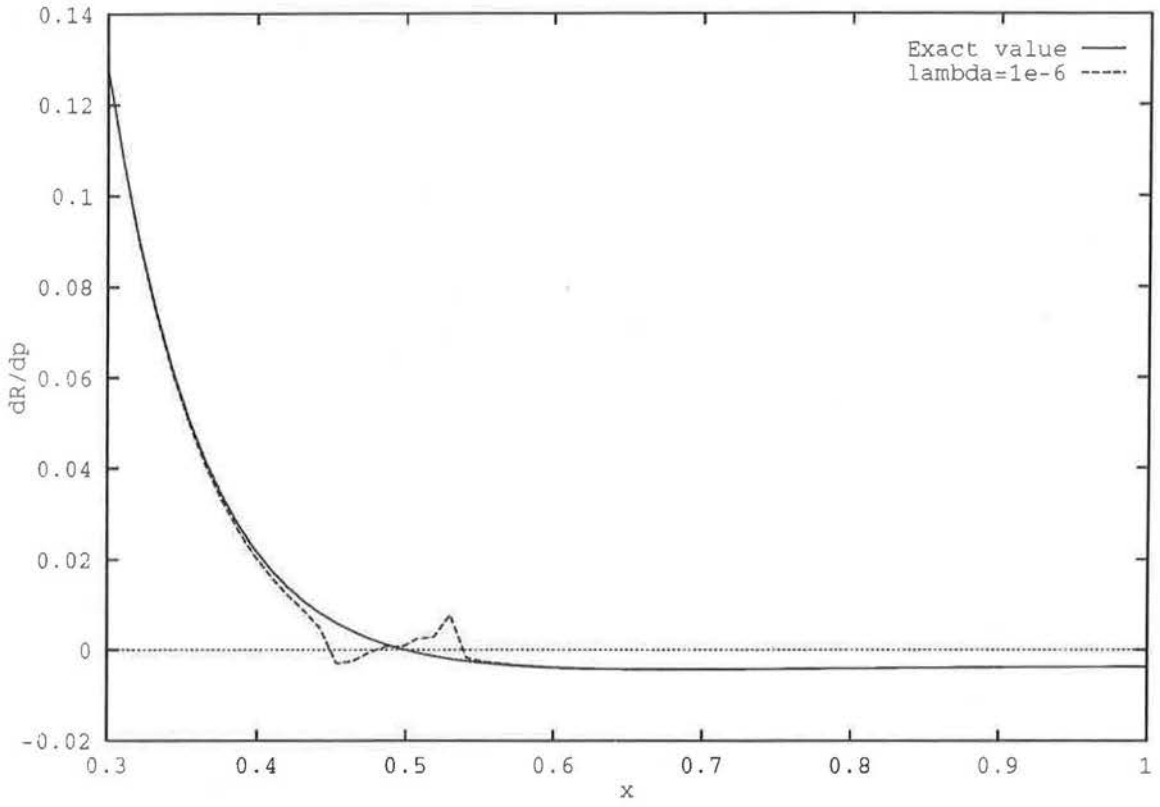


Figure 4.23: R_p for the computed solution, $z = \sin(\pi x) + \pi x$, Lambertian map, light source direction $(-\pi, 1)$.

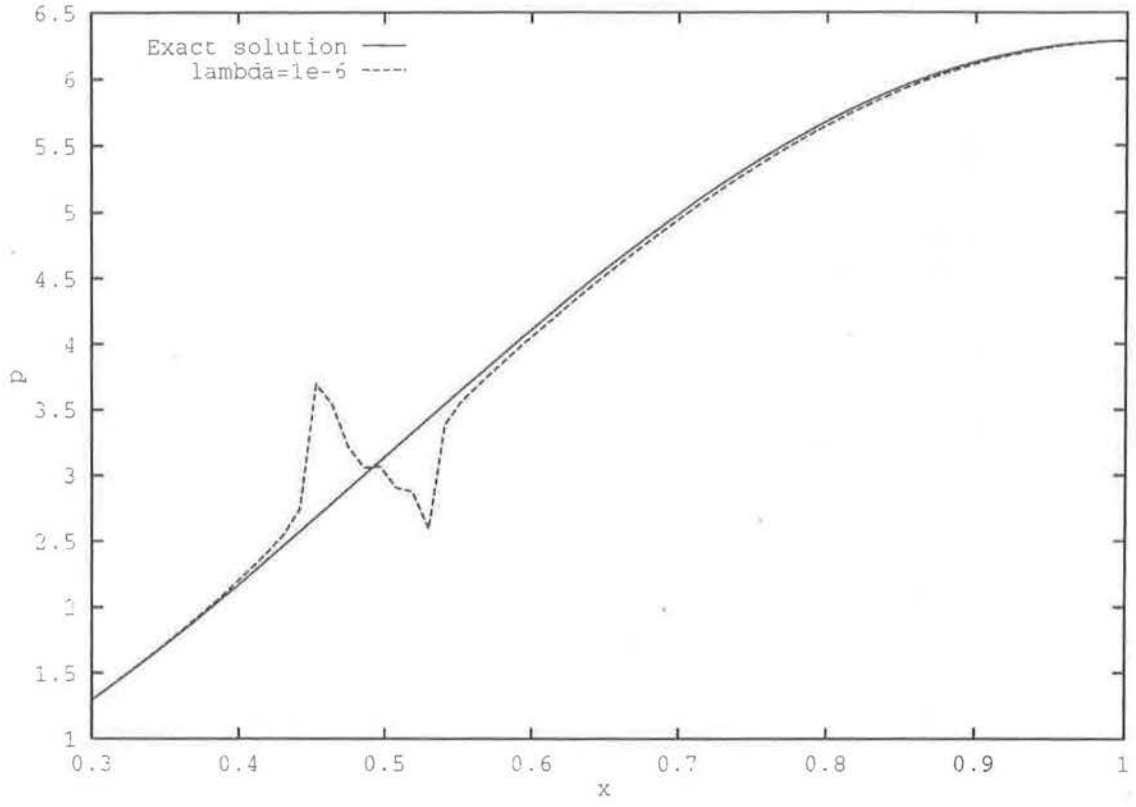


Figure 4.24: Computed solution : $z = -\sin(\pi x) + \pi x$.

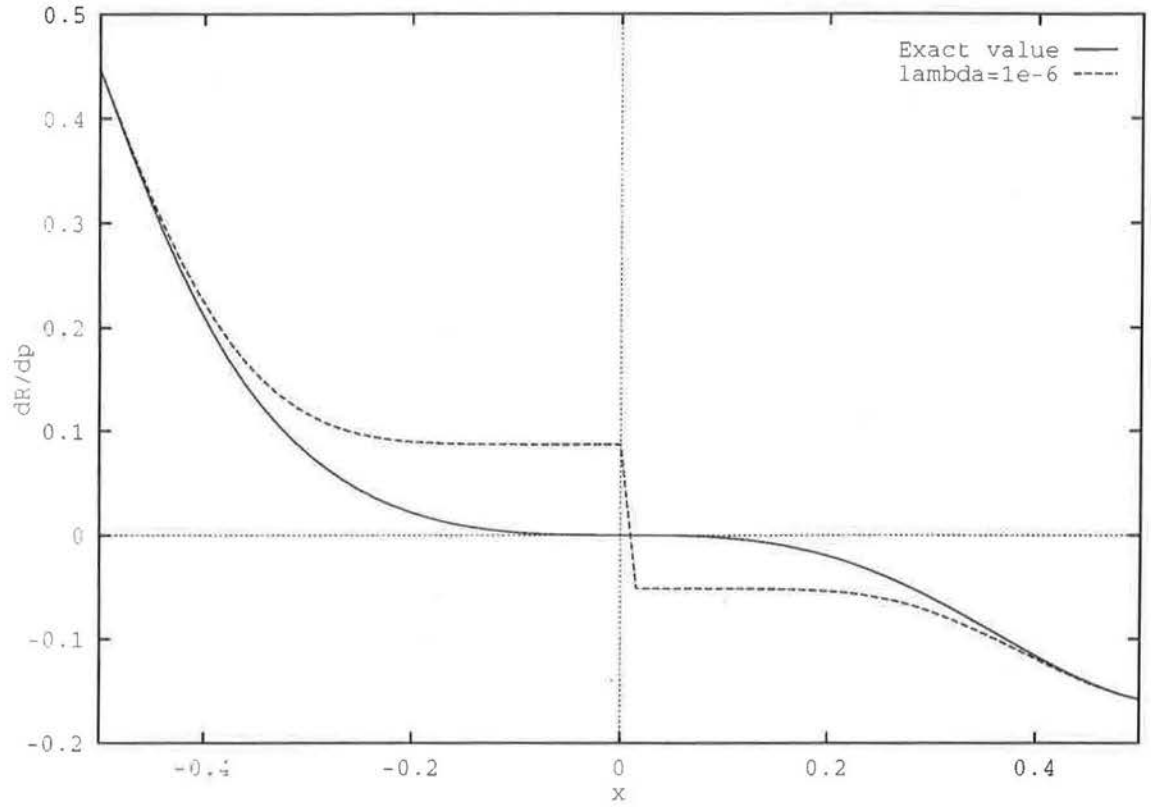


Figure 4.25: R_p for the computed solution, $z = x^4 + 0.5x$, Lambertian map, light source direction $(-0.5, 1)$.

error 5.733×10^{-16}. A graph of R_p for the computed solution is shown in Figure 4.25. The transition layer, located at the singular point, is evident in both R_p and in p. The graph of p is shown in Figure 4.26.

We have seen a strong similarity between the behaviour of the one dimensional problem and the two dimensional problem in the presence of singular points. We believe that the above explanation of the behaviour of the one dimensional algorithm also lies at the root of an explanation for the two dimensional case. For the latter case we also perform an integration of sorts along characteristics and we suggest that the propagation of errors along the characteristics is responsible for the inversions and creases observed in the two

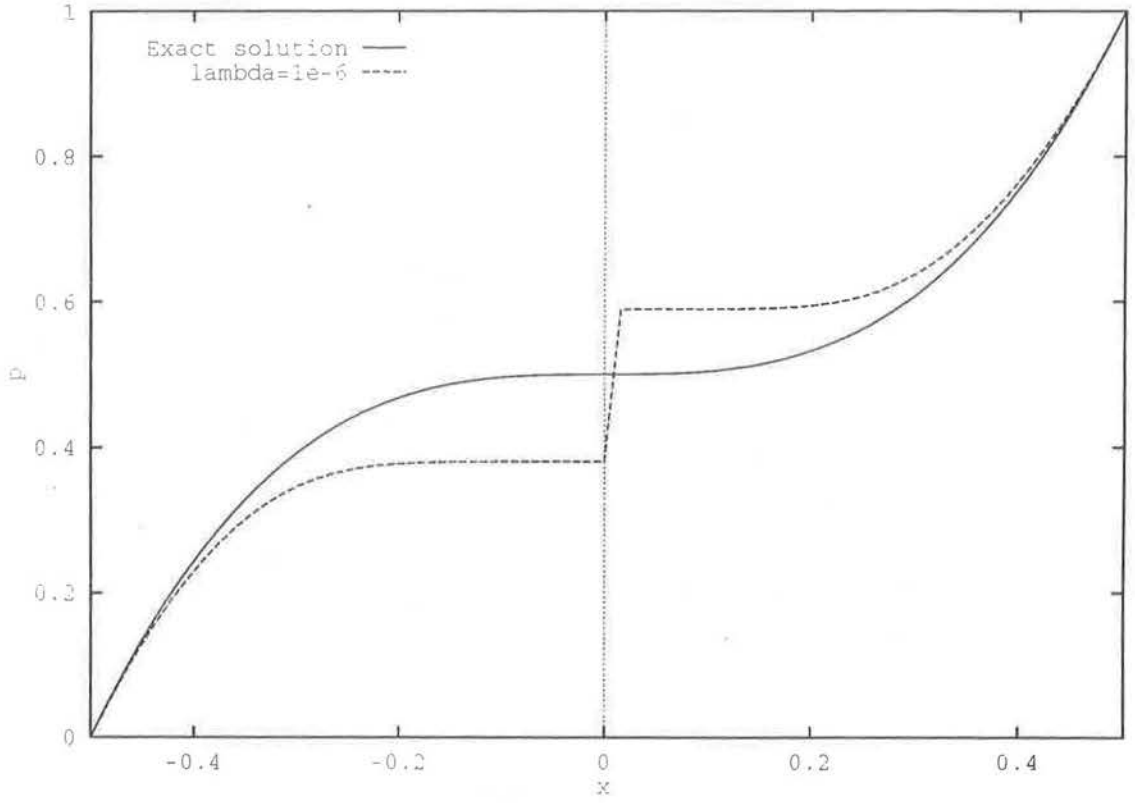


Figure 4.26: Computed solution : $z = x^4 + 0.5x$.

dimensional case.

4.6.5 Linearization

Consider the case of an Eikonal reflectance map for which we wish to solve the equation

$$-\lambda p'' + 2pp' = E'. \tag{4.73}$$

It could be argued that the iteration would benefit from a linearization of the term $2pp'$ about the previous iterate (as was done for the reflectance map in Chapter 3). Let $\tilde{p}$ and $\tilde{p'}$ denote the value of p and p' about the previous iterate then we have

$$pp' \simeq \tilde{p}\tilde{p'} + \tilde{p'}(p - \tilde{p}) + \tilde{p}(p' - \tilde{p'}). \tag{4.74}$$

Equation (4.73) then becomes

$$-\lambda p'' + 2\tilde{p}p' + 2\tilde{p'}p = E' + 2\tilde{p}\tilde{p'}. \tag{4.75}$$

When this equation is discretized, the coefficient of p_i at each grid point is

$$\frac{4\lambda}{h^2} + \frac{2|\tilde{p}_i|}{h} + \frac{\tilde{p}_{i+1} - \tilde{p}_{i-1}}{h} \tag{4.76}$$

where we have used central differencing to estimate $\tilde{p'}$. Notice that the first two terms are positive, however, the last could be positive or negative. In cases when the last term is negative near the singular point where $|\tilde{p}_i| = O(h)$, the coefficient (4.76) could become zero or negative leading to singularity or nonconvergence of the iteration. The linearization therefore works against the regularizing term. In general, therefore, such linearization is best avoided.

In particular, linearization would lead to difficulties for the curves $z = \sin(\pi x)$ and $z = -x^4$. Both of these curves have $p' < 0$ at the location of the singular point. Thus difficulties with linearization arise exactly when the methods considered earlier encounter difficulties. One saving grace of linearization is that it is likely to result in non-convergence, rather than convergence to the wrong solution in these circumstances.

For the curve $z = -\sin(\pi x)$ using the exact solution as an initial guess and taking $\lambda = 1 \times 10^{-6}$, the iteration without linearization does not converge. However, when the above linearization is implemented the algorithm does converge. In this case, $p' > 0$ at the singular point and so the linearization in fact adds stability to the iteration in that the additional term contributes to the diagonal dominance of the iteration matrix.

4.6.6 Alternative solution techniques

Given the apparent failure of our simple upwinding scheme in the presence of singular points we now consider alternative solution techniques.

Osher [42] presents a scheme for solving the time-dependent, two point boundary value problem

$$u_t = \lambda u_{xx} - (f(u))_x - E_x \tag{4.77}$$

with boundary conditions

$$u(a) = \alpha, \quad u(b) = \beta. \tag{4.78}$$

Such a scheme can be integrated to steady state (for which $u_t = 0$) to yield a solution to the equation

$$-\lambda u_{xx} + (f(u))_x = E_x \tag{4.79}$$

subject to the same boundary conditions.

We can write our one dimensional problem (4.35) in the form

$$-\lambda p_{xx} + (R(p))_x = E_x \tag{4.80}$$

for which Osher's scheme may now be implemented. This is a conservation form which we cannot extend to two dimensions, but we investigate whether casting the problem in this form would have alleviated the difficulty.

In the special case when $f(p)$ is convex with only one stagnation point, $\bar{p}$, such that $f'(\bar{p}) = 0$, Osher defines the functions

$$f_+(p) = f(\max(p_i, \bar{p})), \quad f_-(p) = f(\min(p_i, \bar{p})). \tag{4.81}$$

Notice that for the Eikonal map, $f(p) = p^2$ which satisfies the latter criteria. The term $(f(p))_x$ appearing in equation (4.80) is then discretized by

$$\frac{f_-(p_{i+1}) - f_-(p_i) + f_+(p_i) - f_+(p_{i-1})}{h}. \tag{4.82}$$

This yields an upwinding discretization. We omit further details of the implementation which may be found in [42].

We applied Osher's scheme to our problem in the case of an Eikonal reflectance map. For the curve $z = \sin(\pi x)$, the algorithm did not converge for $\lambda = 1.0 \times 10^{-3}$. The non-converged solution demonstrated an inversion in the neighbourhood of the singular point just as was observed for our simple upwinding scheme. For the curve $z = -x^4$, the algorithm converged for $\lambda = 1.0 \times 10^{-6}$ and produced a spurious transition layer at the singular point just as for our algorithm. It appears that nothing is to be gained from this approach.

4.7 Alternative Discretizations for the Two Dimensional Problem

More recently, Sidilkover & Ascher [48] have suggested a more complicated upwinding scheme for the two dimensional, incompressible Navier-Stokes equations. The scheme applies an upwinding discretization only when the advection terms are sufficiently large. When the advection terms are small (in particular, across a turning point), artificial viscosity is added and central differencing is used to discretize the advection terms.

For the equation

$$L(p, q) := -\lambda \nabla^2 p + 2pp_x + 2qp_y = E_x \tag{4.83}$$

the discrete differential operator, $L_h(p,q)$ becomes:

$$\begin{aligned} L_h(p,q) &= -\frac{(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) + (G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}})}{h} \\ &+ \frac{2p_{i,j}(p_{i+1,j} - p_{i-1,j}) + 2q_{i,j}(q_{i,j+1} - q_{i,j-1})}{2h} \end{aligned} \tag{4.84}$$

Where F and G are defined by

$$F_{i+\frac{1}{2},j} = F^b_{i+\frac{1}{2},j} + \xi_{i+\frac{1}{2},j} \cdot F^a_{i+\frac{1}{2},j} \tag{4.85}$$

$$G_{i,j+\frac{1}{2}} = G^b_{i,j+\frac{1}{2}} + \eta_{i,j+\frac{1}{2}} \cdot G^a_{i,j+\frac{1}{2}} \tag{4.86}$$

Here, a superscript a denotes aritificial viscous fluxes while b denotes a flux due to physical viscosity. The latter are given by

$$F^b_{i+\frac{1}{2},j} = \lambda \frac{p_{i+1,j} - p_{i,j}}{h}, \quad G^b_{i,j+\frac{1}{2}} = \lambda \frac{p_{i,j+1} - p_{i,j}}{h}. \tag{4.87}$$

The parameters ξ and η are given by

$$\xi_{i+\frac{1}{2},j} = \max\left\{0, \frac{|2p_{i+\frac{1}{2},j}| - 2\lambda/h}{|2p_{i+\frac{1}{2},j}|}\right\}, \quad \eta_{i,j+\frac{1}{2}} = \max\left\{0, \frac{|2q_{i,j+\frac{1}{2}}| - 2\lambda/h}{|2q_{i,j+\frac{1}{2}}|}\right\} \tag{4.88}$$

$$p_{i+\frac{1}{2},j} = \frac{p_{i+1,j} + p_{i-1,j}}{2}, \quad q_{i,j+\frac{1}{2}} = \frac{q_{i,j+1} + q_{i,j-1}}{2}. \tag{4.89}$$

The artificial viscosity is added through the terms

$$F^a_{i+\frac{1}{2},j} = \frac{1}{2}|2p_{i+\frac{1}{2},j}| \ (p_{i+1,j} - p_{i,j}) \tag{4.90}$$

$$G^a_{i,j+\frac{1}{2}} = \frac{1}{2}|2q_{i,j+\frac{1}{2}}| \ (p_{i,j+1} - p_{i,j}) \tag{4.91}$$

We first tried this scheme on the surface $z = -\sin(\pi x)\sin(\pi y)$ for which our simple upwinding scheme gives good results provided λ is not taken too small. For this scheme, it was possible to take λ as small as 1.0×10^{-6} and still maintain good convergence. The error in the computed surface was comparable to that produced by simple upwinding and there were no apparent flaws in the surface.

For the surface $z = \sin(\pi x)\sin(\pi y)$, however, which yielded an inversion in the solution for the simple upwinding scheme (Figure 4.5), the iteration did not converge for $\lambda = 1 \times 10^{-2}$. This behaviour is preferable to the earlier scheme because we can have more confidence in a scheme which does not converge when presented with a difficult problem than in one that converges to the wrong solution. Sidilkover & Ascher's scheme also diverged for the surface $z = -(x^4 + y^4)$ when $\lambda = 1.0 \times 10^{-3}$. For $\lambda = 1.0 \times 10^{-2}$, the iteration converged to a solution which yielded creases on the surface.

It would appear that schemes which are apparently successful in computing solutions of the incompressible Navier-Stokes equations, may fail for a substantial class of shape from shading problems, that is, problems containing a singular point of sink or saddle type. In the next section we point out some important differences between solutions to the Navier-Stokes equations and solutions to shape from shading problems.

4.8 Navier-Stokes vs. Shape from Shading

We now consider the conditions under which a solution (u, v) to the incompressible Navier-Stokes equations (4.9) is equivalent to a solution (p, q) of the shape from shading problem (4.8) which contains a singular point. We have seen that when no singular points exist, our simple upwinding scheme performs well. The lack of singular points corresponds to a lack of stagnation points for the incompressible fluid flow. Of more interest is the case when singular points (stagnation points) exist. Rotating or recirculating flows, which have closed streamlines (characteristic curves) surrounding a stagnation point, do not have a corresponding analogy in the class of shape from shading problems since we have seen that the latter cannot yield closed characteristic curves. This leaves stagnation points that have characteristics flowing towards or away from them, that is, sources, sinks and saddle points.

Consider a source located at the origin towards which all streamlines (characteristics) flow. Let C be an imaginary circle of radius $r > 0$ centred at the origin bounding a domain Ω. By the Divergence Theorem we have

$$\int\int_{\Omega} \nabla \cdot \mathbf{u} \, dxdy = \oint_C \mathbf{u} \cdot \mathbf{n} \, ds \tag{4.92}$$

where $\mathbf{u}$ is the velocity field and $\mathbf{n}$ is the outward unit normal to the curve C. Now, an incompressible fluid is divergence-free and so equation (4.92) tells us that the net flux of fluid across C must be zero. This clearly is not the case given that *all* characteristics cross C in the direction towards the origin. There is therefore a net flux of fluid across C into Ω. Sinks can therefore not occur for an incompressible fluid flow. By the same reasoning, sources cannot occur either. It is possible, however, for saddle points to exist. Consider the flow whose velocity components are given by

$$u = 6xy, \quad v = 3x^2 - 3y^2. \tag{4.93}$$

The corresponding shape from shading problem would be the surface whose height is given by

$$z = 3x^2y - y^3. \tag{4.94}$$

Notice that the flow is in fact divergence free and that there is a stagnation point corresponding to a saddle point at the origin.

We applied our simple upwinding scheme to this surface. The exact surface is shown in Figure 4.27. The continuation algorithm proceeded to a value of $\lambda = 1.0 \times 10^{-6}$ and achieved solid convergence. The computed surface, found in Figure 4.28, had obvious flaws.

We therefore appear to have constructed a problem which may give trouble for solvers of the incompressible Navier-Stokes equations. It is not clear to the author how large a class of problems is represented by this particular case. We point out that Sidilkover &

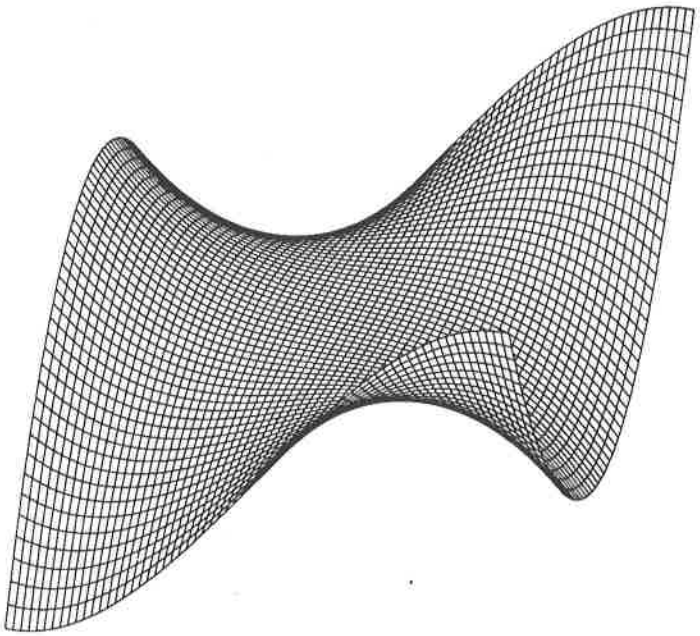


Figure 4.27: Exact solution: $z = 3x^2y - y^3$.

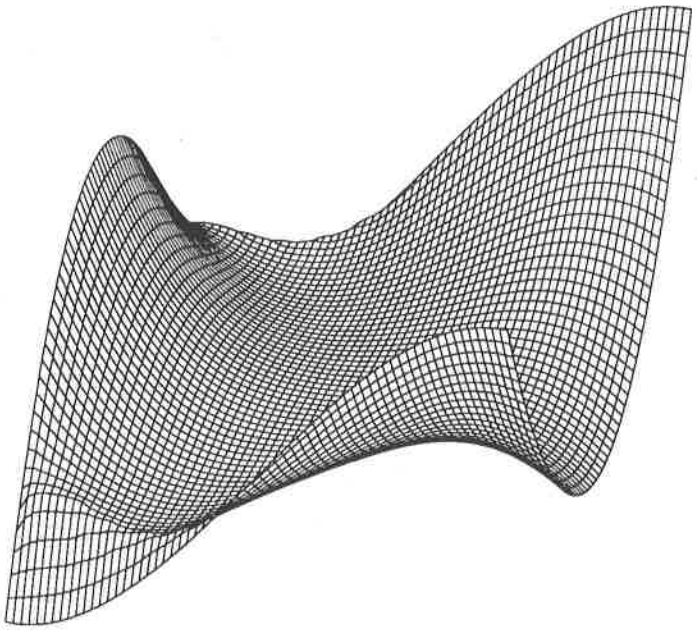


Figure 4.28: Computed surface: $z = 3x^2y - y^3$, $\lambda = 1.0 \times 10^{-6}$, Eikonal map.

Ascher's scheme did not converge in this particular case for λ smaller than 1.0×10^{-2} and that this behaviour is preferable to that of a scheme which solidly converges to the wrong solution. We also point out that the inviscid limit of a steady state incompressible Navier-Stokes flow may give an ill-posed problem, see Yavneh [58].

4.9 Discussion

We have considered a solution technique based on upwinding which is commonly used for solving the incompressible Navier-Stokes equations. The latter system is similar to the two dimensional shape from shading problem although there are important differences – we have seen that there are certain classes of problems that belong to only one or other of the systems.

Our simple upwinding scheme appears to be successful in the absence of singular points. The class of surfaces having no singular points is somewhat uninteresting, especially in the case of overhead illumination. Such surfaces would have no local maxima, minima or saddle points.

In the presence of singular points corresponding to sinks our scheme has often converged, even for small values of λ, to a solution which is not close to the exact solution in the sense that it contains nonrandom, spurious features. In such cases, the computed solution would be considered feasible had the exact solution not been known. This is particularly bad since, in general, we can have no confidence in the computed solution.

Other schemes based on simple upwinding have been tried and have also failed. Sidilkover's scheme has the advantage that it does not converge for values of λ which are reasonably large. This would indicate problems in the event that we did not know the exact solution and, as a result, we would be less likely to accept a spurious solution. However, the large class of problems containing singular points are still not handled

satisfactorily by this scheme.

We suggest that at the root of the problem is the fact that for the shape from shading problem, a singular point will have characteristics either flowing towards or away from the point. We have seen that the case when characteristics flow towards the point are particularly prone to difficulty. The requirement that solutions propagating from different parts of the domain converge at a singular point with the same solution is rather stiff. When characteristics flow towards the singular point we have no way of avoiding this point at which our system becomes singular. For the case of a source, any perturbation near the source will be propagated into the rest of the domain. This behaviour is quite different from the case of a recirculating flow where characteristics flow around the stagnation point and not towards or away from it.

Chapter 5

Conclusion

This thesis has sought to explore computational methods for the shape from shading problem. Our hope was to better understand the mathematical nature of the problem and to obtain robust, efficient methods for its solution. Our investigation took us in three directions.

In Chapter 2 we considered stable, implicit numerical schemes for the method of characteristic strips. This formulation of the shape from shading problem was one of the first to be considered historically. Earlier authors who had used less stable numerical schemes, reported a strong sensitivity of the solution to noise in the image data. We have shown that this sensitivity is also present for more stable numerical methods. The method of projected invariants, in which the solution at each step is projected onto the invariant manifold defined by the image irradiance equation, helped to reduce the effects of noise but not to the extent that we could be happy with the resulting computational procedure. Particular difficulties are associated with the existence of singular points where characteristics tend to converge. This can lead to characteristics crossing and therefore yielding multiple solutions. We considered images containing only one singular point and, based on the difficulties arising from this fairly simple case, concluded that the method would not be viable in the more realistic case of an arbitrary number of singular points.

In Chapter 3 we considered Horn's variational technique [22] as a basis for producing a fast multigrid solver for the shape from shading problem. Of all the formulations of

the shape from shading problem, including those of Chapters 2 and 4 and a variety of regularization techniques in the literature, this variational formulation was found to be the most promising. However, the iterative solution method used by Horn was rather slow. In order to obtain a fast algorithm, we first changed the discretization scheme. We used a finite volume discretization on a non-staggered grid. For this we developed a special variant of the nested multigrid method (FMG), where continuation in the regularization parameter λ is systematically applied within the multigrid nesting. In conducted tests, the resulting algorithm yields solutions which are both qualitatively and quantitatively similar to those produced by Horn's scheme, but in minutes rather than hours of computing time.

Horn's formulation was extended to allow for multiple image data and the incorporation of explicit knowledge of the location of discontinuities in surface height and orientation. The performance of the algorithm was tested for different choices of boundary conditions and differing numbers of input images. The algorithm's performance was least acceptable when discontinuities were present in the domain and natural boundary conditions were used for both the gradient and surface height. However, the performance under these circumstances was greatly improved when multiple images were used. We tested the algorithm on synthetic data ranging from a simple hemisphere to a complicated digital terrain model. When the data from three images was available to the algorithm, the results were excellent. We conclude that when given the same information as for photometric stereo, our algorithm produces comparable results.

A class of solution techniques, namely upwinding schemes for a flow formulation, which is new to the shape from shading literature was examined in Chapter 4. We took our lead from solution methods for the incompressible Navier-Stokes equations. The latter were shown to be of a similar form to a certain "flow" formulation of the shape from shading problem. We showed that in the absence of singular points the algorithm

performs well given Dirichlet boundary data on the surface slopes and data from only one image. However, certain problems arose in the presence of singular points, especially such points which can be viewed as sinks for the flow of characteristics. We explain that under these conditions the problem is ill-posed. We sought a solution to the problem bearing in mind the potential difficulties that can arise due to the ill-posed nature of the problem. We demonstrated that when a singular point is present in the domain, the algorithm is prone to produce spurious solutions. The latter occurred only for singular points which act either as sinks or saddle points in the flow of characteristic curves. Through a study of the analogous one dimensional problem, we justified the form of anomalies that are present in the computed solution. The most disturbing observation is that the error, which in general is unavoidable, accumulates in a particular fashion which does not appear random and causes noticeable spurious effects.

A comparison was made between the class of shape from shading problems and the class of problems arising from the flow of an incompressible fluid. We showed that singular points that act as sinks cannot occur in an incompressible fluid flow. The difficulties experienced for such shape from shading problems cannot therefore be encountered in the fluid mechanics setting. However, we showed that singular points which act as saddle points in the flow, may occur in both types of problems.

We have shown that when singular points are present in the domain, our algorithm can readily converge to a wrong solution which could be considered feasible if the exact solution were not known. It was seen that other algorithms based on upwinding also produce spurious results. The poor performance of these algorithms was attributed to the ill-posed nature of the problem when singular points are present. We therefore concluded that we can have no confidence in such an algorithm given that singular points are typically present in a very large class of shape from shading problems.

Of the three directions which we have pursued here, the two based on, or related

to, characteristic flow have yielded some very curious results. They did not yield hope for a robust algorithm for the shape from shading problem. The variational (and regularization) approach has yielded the most promising results. Of the various variational formulations, Horn's [22] is particularly promising. We have proposed a very efficient algorithm for the numerical solution of this formulation. It could be used to further explore the viability of shape from shading as a component of a computer vision system.

Bibliography

[1] Ascher U. & Carter P. "Multigrid Solution for Shape from Shading" in Proceedings Third European Conference on Multigrid Methods, GMD, Bonn, 1990.

[2] Ascher U. & Carter P. "A Multigrid Method for Shape from Shading", SIAM Journal on Numerical Analysis, **30** (1) pp102-115, February 1993.

[3] Blake A. & Zisserman A. *Visual Reconstruction*, MIT Press 1987.

[4] Brandt A. "Guide to Multigrid Development", in *Multigrid Methods*, eds. W. Hackbusch & U. Trottenberg, Proceedings of Conference on Multigrid Methods, Koln-Porz, 1981.

[5] Brandt A. & Dinar N. "Multigrid Solutions to Elliptic Flow Problems" in *Numerical Methods for Partial Differential Equations* ed. S. Parter, Academic Press 1979.

[6] Briggs W. *A Multigrid Tutorial* SIAM 1987.

[7] Brooks M.J. & Horn B.K.P. "Shape and Source from Shading", Proceedings 1985 International Joint Conference on Artificial Intelligence pp932-936, Los Angeles, CA, August 1985.

[8] Brooks M.J., Chojnacki W. & Kozera R. "Impossible and Ambiguous Shading Patterns", International Journal on Computer Vision 7:2, pp119-126, 1992.

[9] Brooks, M.J., Chojnacki W. & Kozera R. "Circularly symmetric Eikonal Equations and Nonuniqueness in Computer Vision", Journal of Mathematical Analysis and Applications, **165** (1) pp192-215, March 1992.

[10] Brooks M.J. "Variational Techniques and Open Problems in the Recovery of Surface Shape from Image Shading", manuscript. 1990.

[11] Bruckstein A.M. "On Shape from Shading", Computer Vision, Graphics, and Image Processing, **44** pp139-154, 1988.

[12] Bruss A.R. "The Eikonal Equation: Some results applicable to Computer Vision", Journal of Mathematical Physics **23**(5) 1982.

[13] Carrier G.F. & Pearson C.E. *Partial Differential Equations* 2nd edition, Academic Press, 1988.

[14] Chang K.W. & Howes F.A. *Nonlinear Singular Perturbation Phenomena: Theory and Application*, Applied Mathematical Sciences 56, Springer Verlag, 1984.

[15] Courant R. & Hilbert D. *Methods of Mathematical Physics* Vol I, Interscience 1953.

[16] Frankot R.T. & Chellappa R. "A method for enforcing integrability in shape from shading algorithms", IEEE PAMI **10** pp439-451, 1988.

[17] Gibbins D., Brooks M.J., & Chojnacki W. "Determining light source direction from images of shading", Discipline of Computer Science, School of Information Science & Technology, Flinders University, Adelaide, Australia.

[18] Hackbusch W. *Multi-Grid Methods and Applications* Springer-Verlag 1985.

[19] Hairer E., Norsett S.P. & Wanner G. *Solving Ordinary Differential Equations I: Non-stiff Problems*, Springer Series in Computational Mathematics **8**, Springer-Verlag, 1987.

[20] Horn B.K.P. "Obtaining Shape from Shading Information" in *The Psychology of Computer Vision* ed. P.H. Winston, McGraw-Hill 1975.

[21] Horn B.K.P. *Robot Vision* MIT Press 1986.

[22] Horn B.K.P. "Height and Gradient from Shading", IJCV **5** pp37-75, 1990.

[23] Horn B.K.P. & Brooks M. eds. *Shape from Shading* MIT Press 1989.

[24] Horn B.K.P. & Brooks M. "The Variational Approach to Shape from Shading", CVGIP **33** pp174-208, 1986.

[25] Horn B.K.P. & Sjoberg R.W. "Calculating the Reflectance Map", Applied Optics **18**(11), pp1770-1779, 1979.

[26] Horn B.K.P., Szeliski R.S. & Yuille A.L. "Impossible Shaded Images", IEEE PAMI **15** (2) pp166-170, February 1993.

[27] Ikeuchi K. & Horn B.K.P. "Numerical Shape from Shading and Occluding Boundaries", Artificial Intelligence **17** No.1-3 pp141-184, 1981.

[28] Kimmel R. & Bruckstein A.M. "Shape from shading via level sets", to appear.

[29] Kozera R. "On shape recovery from two shading patterns", International Journal of Pattern Recognition and Artificial Intelligence, **6** (4) 1992.

[30] Leclerc Y.G. & Bobick A.F. "The Direct Computation of Height from Shading", CVPR 1991.

[31] Lee D. "A provably convergent algorithm for shape from shading", in *Shape from Shading*, eds. B.K.P. Horn & M.J. Brooks, MIT Press, 1989.

[32] Lee K.M. & Kuo C.-C. J. "Shape from shading with a linear triangular element surface model", IEEE PAMI, to appear.

[33] Lee K.M. & Kuo C.-C. J. "Shape Reconstruction from Photometric Stereo", submitted: Journal of the Optical Society of America: A.

[34] Lions P.L., Rouy E. & Tourin A. "Shape from Shading, Viscosity Solutions and Edges", Numerische Mathematik, **64** (3) pp323-354, 1993.

[35] Malik J. & Maydan D. "Recovering three dimensional Shape from a Single Image of Curved Objects", IEEE PAMI **11**(6) 1989.

[36] Marr D. *Vision*, Freeman & Co. 1982.

[37] Oliensis J. "Uniqueness in Shape from Shading", IJCV **6**(2) pp75-104, 1991.

[38] Oliensis J. "Shape from Shading as a Partially Well Constrained Problem", CVGIP: Image Understanding, **54**(2) 1991.

[39] Oliensis J. & Dupuis P. "Direct method for reconstructing Shape from Shading", Proc. SPIE Conference 1578 on Geometric Methods in Computer Vision, San Diego, California, July 1991.

[40] Onn R. & Bruckstein A. "Integrability Disambiguates Surface Recovery in Two Image Photometric Stereo", IJCV **5**(1) pp105-113, 1990.

[41] Ortega J.M. *Numerical Analysis: a second course*, Classics in Applied Mathematics **3**, SIAM, 1990.

[42] Osher S. "Nonlinear singular perturbation problems and one sided difference schemes", SIAM Journal on Numerical Analysis **18** (1), Feb. 1981.

[43] Pentland A. "Linear Shape from Shading", IJCV **4** pp153-162, 1990.

[44] Poggio T. & Torre V. "Ill-posed problems and regularization analysis in early vision", A.I. Memo No. 773 MIT April 1984.

[45] Ramachandran V.S. "Perceiving Shape from Shading", Scientific American, Aug 1988.

[46] Rouy E. & Tourin A. "A viscosity solutions approach to Shape from Shading", SIAM Journal on Numerical Analysis, **29** pp867-884, 1992.

[47] Saxberg, B. "A Differential Geometric Approach to the Shape from Shading Problem", AI-TR1117, MIT, Cambridge, MA, June 1989.

[48] Sidilkover D. & Ascher U. "Multigrid solver for Navier-Stokes equations", Proc. Copper Mountain Multigrid Conference, April 1993.

[49] Sweldens W. & Roose D. "Shape from Shading using Parallel Multigrid Relaxation", Proceedings Third European Conference on Multigrid Methods, GMD, Bonn 1990.

[50] Terzopoulos D. "Image Analysis using Multigrid Relaxation Methods", IEEE PAMI **8**(2) 1986.

[51] Tourin A. "A comparison theorem for a piecewise Lipschitz continuous Hamiltonian and application to Shape from Shading Problems", Numerische Mathematik, to appear.

[52] Torre V. & Poggio T. "On Edge Detection", IEEE PAMI **8**(2) 1986.

[53] Vega O.M. & Yang Y.H. "Shading logic: A heuristic approach to recover shape from shading", IEEE PAMI, to appear.

[54] Woodham R.J. "A co-operative algorithm for determining surface orientation from a single view", International Joint Conference on Artificial Intelligence, MIT 1977.

[55] Woodham R.J. "Photometric Method for determining surface orientation from multiple images", Optical Engineering, **19**(1) 1980.

[56] Woodham R.J. "Analyzing images of curved surfaces", Artificial Intelligence **17** 1981.

[57] Woodham R.J. "Determining Surface Curvature with Photometric Stereo", Proceedings 1989 IEEE Conference on Robotics and Automation, Scottsdale, Arizona.

[58] Yavneh I. "Multigrid Techniques for Incompressible Flows", Ph.D. Thesis, Weizmann Institute, Rehovot, Israel, 1991.

[59] Zheng Q. & Chellappa R. "Estimation of Illuminant Direction, Albedo and Shape from Shading", IEEE PAMI **13** (7), June 1991.