J. Opt. A: Pure Appl. Opt. **8** (2006) 279–289 doi:10.1088/1464-4258/8/3/009

Phase reconstruction by the weighted least action principle

Chungmin Lee[1], **John Lowengrub**[2], **Jacob Rubinstein**[3] **and Xiaoming Zheng**[4]

[1] Mathematics Department, Indiana University, Bloomington, IN 47405, USA
[2] Department of Mathematics, University of California, Irvine, CA 92697, USA
[3] Mathematics Department, Indiana University, Bloomington, IN 47405, USA
[4] Department of Mathematics, University of California, Irvine, CA 92697, USA

E-mail: chunglee@indiana.edu, lowengrb@math.uci.edu, jrubinst@indiana.edu and xzheng@math.uci.edu

Received 22 September 2005, accepted for publication 18 January 2006
Published 14 February 2006
Online at stacks.iop.org/JOptA/8/279

Abstract
The problem of phase reconstruction from intensity is considered in the paraxial geometrical optics regime. The data consist of intensity measurements on two planes orthogonal to the direction of propagation. A new variational principle is used to find the ray mapping between the planes. The mapping is found by minimizing an appropriate function. The minimization process involves a nonlinear partial differential equation. A numerical algorithm for solving this equation is described. The phase is then integrated from the ray mapping. Finally, the method and code are examined by applying them to simulated data.

Keywords: phase sensors, least action principle

1. Introduction

Phase reconstruction is a central problem in wave propagation in general and in optics in particular. One of the common devices for phase measurement is the Hartman–Shack (HS) sensor. This sensor consists of a screen with a plurality of lenslets, and a detection screen. Each lenslet approximately focuses light at a spot on the detection screen. The centroid of the spot determines the average slope of the wavefront that is incident upon the lenslet. Originally developed for the purpose of astronomy optics, HS devices have been used in recent years in ophthalmology to estimate the deformation of wavefronts propagating outside the eye from a point source on the retina. In spite of the general success of HS sensors, they suffer from several drawbacks. In particular, their resolution is limited, since the lenslets cannot be made too small as this would result in undesired diffractive effects.

Since it is relatively easy to measure the wave's intensity, it is tempting to seek algorithms for phase reconstruction from intensity. Almost all intensity sensors are based on the transport of intensity equation (equation (1) below). This equation is considered as a two-dimensional partial differential equation for the phase ψ in terms of the intensity I. One serious difficulty is that (1) cannot be solved without boundary conditions, and no simple method for obtaining the needed conditions was found so far (although reference [1] suggests a method to obtain boundary data under certain assumptions on the phase). An alternative method for finding the phase of a wave from its intensity was proposed in [2]. Rubinstein and Wolansky developed there a new variational principle, called the *weighted least action principle* (WLAP), that generalizes the Fermat principle of least time. The main result is a cost function that provides, upon minimization, the ray mapping of the wave from information on its intensity. Since determining the phase from the rays is a simple problem, the WLAP forms an attractive option for phase reconstruction.

To apply the WLAP it is needed to minimize an unusual functional under certain constraints. The minimization can be done by the steepest descent method. This leads, however, to a system of coupled partial differential equations [3, 2]. In addition to the difficult problem of designing a numerical method for this system of equations, one has to tackle additional ad hoc difficulties, such as supplying adequate initial conditions for the steepest descent flow.

Therefore, a main goal of this paper is to examine the WLAP as a method for phase determination from intensity, including the introduction of accurate and robust numerical methods for implementing it. The WLAP is formulated for paraxial waves in section 2, where we also derive the minimization evolution equation. In section 3 we report on a number of simulations that we carried out to test the method and the code. In these simulations we addressed issues such as sampling resolution of the rays and resolution of the finite element mesh (for solving the nonlinear evolution equation). Our results are summarized and discussed in section 4. The numerical algorithm of the minimization problem is described in an appendix. This part is somewhat technical and may not appeal at once to all readers from the optics community; nevertheless, we chose to explain our method in some detail to assist those readers who wish to develop their own solvers.

2. Formulation

Consider a monochromatic wave in the paraxial geometrical optics approximation propagating in a uniform medium. Its phase (or, rather, its eikonal) function $\psi(x, z)$ and intensity $I(x, z)$ satisfy in the large k limit the following pair of equations:

$$\frac{\partial I}{\partial z} + \nabla\cdot(I\nabla\psi) = 0, \qquad (1)$$

$$\frac{\partial\psi}{\partial z} + \frac{1}{2}|\nabla\psi|^2 = 0. \qquad (2)$$

Here $x \in R^2, z$ is the direction of propagation and the differential operator ∇ is in the two-dimensional plane orthogonal to z. We assume that the intensity I has been detected at two screens: $z = Z_1$ denoted by P_1, and $z = Z_2$ denoted by P_2. The distance between the screens is $Z = Z_2 - Z_1$. The transport equations (1) and the eikonal equation (2) are considered in the interval $z \in [Z_1, Z_2]$. The problem is to reconstruct the phase $\psi(x, z)$ for all z from knowledge of $I(x, z = Z_1) := I_1(x)$ and $I(x, z = Z_2) := I_2(x)$.

Rubinstein and Wolansky showed in [2] that the problem above has a variational formulation. In their theory, one computes first the ray mapping $\bar{U}(x)$ between a point x on the screen P_1 and the intersection point between the ray leaving x and the screen P_2. Once the ray mapping $\bar{U}$ is known, the phase can be read off it. Any ray mapping must satisfy the standard intensity conservation requirement

$$I_1(x) = I_2(\bar{U}(x))|J(\bar{U})|, \qquad (3)$$

where $J(\bar{U})$ is the Jacobian of $\bar{U}$. Since the relation (3) appears frequently below, we designate a special notation for it: a mapping $U(x)$ is said to transport I_1 into I_2 (or to be an intensity transporting mapping) if U satisfies (3); we write then

$$U_{\#}I_1 = I_2. \qquad (4)$$

In [2] it was shown that the ray mapping $\bar{U}$ associated with the optical problem (1), (2), together with the intensity data I_1 and I_2, satisfies

$$\int \left|\bar{U}(x) - x\right|^2 I_1(x)\,dx \leqslant \int |U(x) - x|^2 I_1(x)\,dx$$
$$\forall U_{\#}I_1 = I_2. \qquad (5)$$

Before justifying the statement above, we proceed to analyse its application. The variational formulation (5) requires a numerical method for finding the optimal mapping $\bar{U}$. An attractive way to optimize (5) is to start with an admissible mapping $U(x, t = 0) = U_0(x)$, i.e. a mapping that transports I_1 into I_2, and to evolve it along the steepest descent of the functional

$$M(U(x, t); I_1, I_2) := \tfrac{1}{2}\int |U(x, t) - x|^2 I_1(x)\,dx. \qquad (6)$$

We emphasize that the steepest descent flow must be confined to the manifold of mappings U that satisfy the constraint (4). Notice that here t is a parameter for the steepest descent flow, and is not associated with any physical time. Indeed, a gradient flow for M was developed in [3]. To find the flow of $U(x, t)$ one needs at each time step to express U in terms of its Helmholtz decomposition

$$U(x, t) = \nabla P(x, t) + V(x, t), \qquad \nabla\cdot V = 0. \qquad (7)$$

Then the evolution equation for U, i.e. the steepest descent flow for (6) constrained by (4), is

$$\frac{\partial U}{\partial t} + \frac{V}{I_1}\cdot\nabla U = 0. \qquad (8)$$

Although the variational principle (5) was proved in [2] by two different methods, we shall prove it here again by a third method. The method we employ here is based on computing the first variation of the functional M. The advantage of this approach, and the reason why we bring a third proof for (5), is that the same computation of the first variation can be used to derive the flow (8), thus justifying the numerical approach that we shall use in the simulations in section 3.

Derivation of (5) and (8)

The first variation of M is computed using two key ideas. The first one is a useful decomposition of intensity transporting mappings U observed by Brennier [4]. Let U_b be a fixed mapping satisfying (3), and let $\mathcal{S}$ be the set of invertible mappings from the plane P_1 to itself that transports I_1 to itself, i.e. $S_{\#}I_1 = I_1$ for all $S \in \mathcal{S}$. Such mappings are called I_1 preserving. Then any mapping U transporting I_1 to I_2 can be written as $U = U_b \circ S^{-1}$ for some $S \in \mathcal{S}$. This statement has a simple physical interpretation: instead of looking for different intensity transporting mappings U, we fix one such mapping U_b and convert it to any other mapping U by using S to shuffle appropriately the points in the plane P_1.

The second idea, following a general theme due to Kantorovich, is to embed the functional M within a larger family of functionals. For this purpose we set

$$K := \tfrac{1}{2}\int\int |x - y|^2 \lambda(x, y)\,dx\,dy, \qquad (9)$$

where the density $\lambda(x, y)$, defined for points $x \in P_1, y \in P_2$, satisfies the conditions $\int \lambda(x, y)dx = I_2(y)$, and $\int \lambda(x, y)dy = I_1(x)$. The functional K reduces to M under the special choice

$$\lambda(x, y) = I_1(x)\delta(y - U(x)). \qquad (10)$$

We proceed to take the first variation of K. Fix an intensity transforming mapping U_b. All other intensity transforming U

are expressed as $U = U_b \circ S^{-1}$. For example, the new density λ becomes $\lambda^S(x, y) = I_1(x)\delta(y - U_b(S^{-1}x))$. The functional K now depends on S:

$$K(S) = \tfrac{1}{2}\int\int |x - y|^2 I_1(x)\delta(y - U_b(S^{-1}x))\,\mathrm{d}x\,\mathrm{d}y. \quad (11)$$

The advantage of the new formulation is that we now take variations only with respect to $S(x)$, with y kept fixed. It is convenient to introduce the parameter t for the family of mapping S to be varied, i.e. we write $S = S(x, t)$. Changing variables in (11) gives

$$K(S) = \tfrac{1}{2}\int\int |(S(x, t) - y|^2 \lambda(x, y)\,\mathrm{d}x\,\mathrm{d}y. \quad (12)$$

Therefore,

$$\frac{\mathrm{d}}{\mathrm{d}t}K(S) = \int\int (x - y)\cdot w(x, t)\lambda(x, y)\,\mathrm{d}x\,\mathrm{d}y, \quad (13)$$

where the velocity field w is defined by $w(x, t) = \frac{\mathrm{d}S(x,t)}{\mathrm{d}t}$. Returning to the original M-formulation we write the first variation in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}M = \int I_1(x)(x - U(x, t))\cdot w(x, t)\,\mathrm{d}x. \quad (14)$$

As a first application of the first variation formula (14), we use it to argue that the optimal mapping $\bar{U}$ is the ray mapping for the optical problem (1), (2). For this purpose set $t = 0$ in (14), and look for a condition on U that will equate the first variation of M to zero. Since S is I_1-preserving, the velocity $w(x, t)$ transports I on the plane P_1 according to $\frac{\partial I}{\partial t} + \nabla\cdot(Iw) = 0$. But I does not depend on t explicitly, and thus w must be conservative with respect to I_1, i.e.

$$\nabla\cdot(I_1 w) = 0. \quad (15)$$

Therefore, equating the first variation of M to zero implies that $(x - \bar{U})$ must be orthogonal to all divergence-free vector fields in order for $\bar{U}$ to be a critical point of the functional M. This means that there exists a function ψ_0 such that

$$\bar{U}(x) - x = Z\nabla\psi_0(x). \quad (16)$$

Consider now the Fresnel equations (1), (2) with initial condition on the plane P_1 given by the pair $(\psi(x) := \psi_0(x),\ I(x) = I_1(x))$. The rays are straight lines, and the eikonal relation between the phase and the rays implies that a ray starting at a point $x \in P_1$ will intersect the screen P_2 at the point $\bar{U}(x)$. Therefore, $\bar{U}(x)$ is indeed the ray mapping for these equations between the planes P_1 and P_2. Notice that the analysis above shows, in fact, that any critical point of M provides a ray mapping.

The second application of (14) is to derive a steepest descent flow for the minimization of M. Here we follow [3] and introduce the Helmholtz decomposition of $U(x, t) - x = \nabla P(x, t) + V(x, t)$ into (14) (notice that x is evidently a gradient, so the divergence-free part of $U - x$ is the same as that of U). Thanks to property (15) one obtains

$$\frac{\mathrm{d}}{\mathrm{d}t}M = -\int I_1(\nabla P + V)\cdot w\,\mathrm{d}x = -\int I_1 V\cdot w\,\mathrm{d}x. \quad (17)$$

Therefore, the velocity field

$$w(x, t) = \frac{V(x, t)}{I_1(x)} \quad (18)$$

provides the steepest descent flow for M. Notice that by the definition of V, we indeed have $\nabla\cdot(I_1 w) = 0$. Since S is transported by w, the standard theory of first order partial differential equations implies that it satisfies $S_t + w\cdot\nabla S = 0$. Recalling the relation $U = U_b \circ S^{-1}$, we obtain for U the transport equation (8).

3. Simulation results

In this section we report on a number of numerical simulations that we ran to test the algorithm and code. Lengths are measured everywhere in millimetres.

Test 1

A wave $u = \exp(ik\cdot 0.01(16x^4 + y^4))$ is given at the aperture (unit disc at the plane $z = 0$). The intensity is evaluated at the plane $z = 50$ and also at a number of other planes ($z = 40, 51, 60$). The Rayleigh–Sommerfeld diffraction integral was used to simulate the wave. We used a relatively low wavenumber $k = 200$. The domain D_1 in the plane $z = 50$ is selected to be a disc of radius 2. In the other planes the mapped domain is assumed to be circular too, with the radius chosen appropriately so that the captured intensity in the disc D_2 equals $\int_{D_1} I(x, y, z = 50)\,\mathrm{d}x$. In figure 1 we depict the intensity profile on the planes $z = 40$ (left) and $z = 50$ (right). The theoretical and computed phases are given in figure 2, and the map of the error in the phase computation is drawn in figure 3. The initial condition was generated by sampling the intensity over 4000 points.

A number of different runs are compared in table 1 below. In run (1a) we used the intensity at planes $z = 50, 51$ to find the phase. The FE solver had 3199 nodes. Run (1b) is based on the same data as run (1a), except that we deliberately shuffled the initial mapping $U(x, 0)$ by two sampling sectors. Runs (1c) and (1d) are similar to run (1b), except that the planes at $z = 40, 50$ were used in run (1c) and the planes $z = 50, 60$ were used in run (1d). For each run we give the initial and final (optimized) weighted global error in the x and y components of the rays. The weighted errors are denoted by $\|e_x\|_w$ and $\|e_y\|_w$, respectively. The weighted norm is defined by

$$\|e_x\|_w^2 = \sum_i |r_x^a(i) - r_x(i)|^2 I_1(x(i)) \Big/ \sum_i I_1(x(i)). \quad (19)$$

Here r_x is the actual x component of the direction vector r of a ray at the point $x(i)$, r_x^{a} is the approximated value of the same component of that ray (either initial or optimal), $I_1(x(i))$ is the intensity of the ray through $x(i)$ and the summation is over all the rays. The use of weighted norms is adequate since our method is based on intensity-related data; thus rays that carry more intensity should be given extra weight.

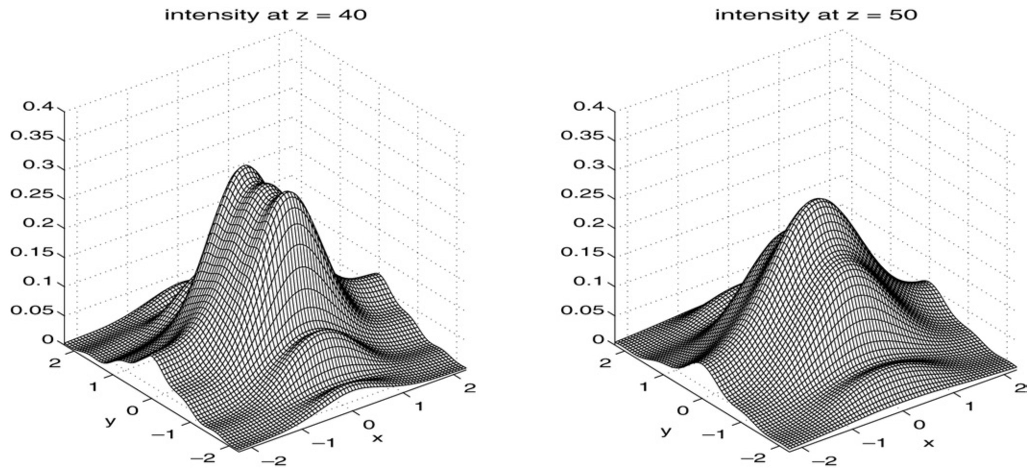

Figure 1. Intensity at $z = 40, 50$.

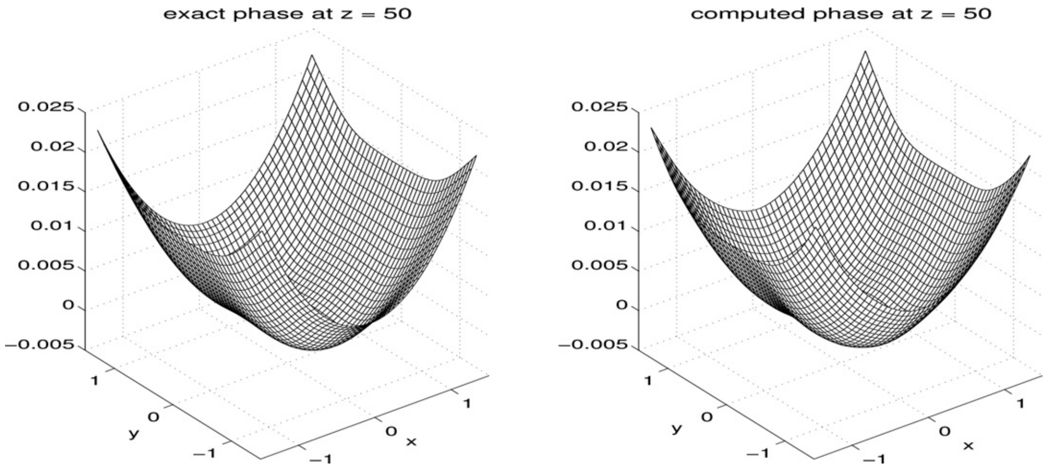

Figure 2. The exact phase (left) and computed phase (right).

Table 1. Initial and final error norms for the simulations in test 1.

Run number	Initial $\|e_x\|_w$	Initial $\|e_y\|_w$	Optimized $\|e_x\|_w$	Optimized $\|e_y\|_w$
(a)	0.014 530	0.013 683	0.013 598	0.011 582
(b)	0.131 118	0.117 722	0.015 453	0.013 135
(c)	0.023 340	0.022 275	0.002 176	0.002 167
(d)	0.031 384	0.027 638	0.001 264	0.002 142

Test 2

In the second test we simulated two waves in the large k limit. We ran two types of simulations. In both of them we computed the phase at the plane $z = 50$ from the simulated intensity at this plane and in some other plane. The domain D_1 was the unit disc. In one family of tests the intensity at $z = 50$ was uniform, while in the other family of tests the intensity on $z = 50$ was Gaussian. As part of the simulation, the unit circle on the plane $z = 50$ was mapped (by the rays) onto the other relevant plane. This determined the shape of the associated domain (D_2) at that plane. The phase at the plane $z = 50$ was (in all cases) elliptical $0.01((x/2)^2 + y^2)$. As in the previous test we sampled the intensities on each domain over 40 sectors and 100 radial points per sector to generate the initial condition for the steepest descent flow. In the current set of simulations we used 1196 FE nodes.

Figure 4 shows the intensity contours at the plane $z = 51$ (left) and at the plane $z = 60$ (right) for the case where the intensity at the plane $z = 50$ was exactly unity.

The error in the phase reconstruction depends on the planes used for finding the ray mapping. In the left part of figure 5 we draw the error for the case where the ray mapping was found from the intensities at the planes $z = 50, 51$. Similarly, on the right part of that figure we draw the corresponding error where the ray mapping was found from the intensities at the planes $z = 50, 60$.

A few examples of error norms are given in table 2. In run (2a) the intensity at the plane $z = 50$ is uniform, and the

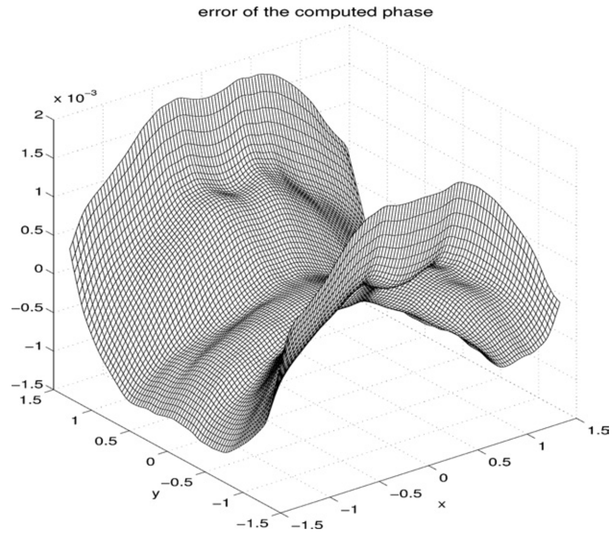

Figure 3. The error in the computed phase.

ray mapping is based on the simulated intensity at the planes $z = 50$ and 51. Run (2b) is similar, except that the intensity on the plane $z = 51$ was replaced by the intensity at the plane $z = 60$. Run (2c) is similar to (a) except that the intensity is Gaussian, and the initial data was deliberately shuffled by two sectors. Run (2d) is similar to run (2c) except that the second plane is, as in run (b), at $z = 60$.

Tables 1–3 provide the absolute errors in the ray direction before and after optimization. To demonstrate that the smallness of the error is significant even in the paraxial regime, we give in table 2a the relative error (e^{r}_x, etc) in the ray direction. The relative error should be computed with care. Near the aperture's centre the x and y components of the true rays are extremely small, so even a small actual error will result in an enormous relative error. Therefore, the relative error norms in table 2a (that correspond to run (2d)) were computed only for rays whose x and y direction components are larger than 0.005. These rays cover about 90% of the disc's area.

Test 3

The third test is similar to the second test, but the simulations are done with diffraction integrals (so they are exact). As usual we use the plane $z = 50$ as the central plane where the computations are done. The domain D_1 is chosen on that plane. Typically it is a circular domain. One purpose of test 3 is to examine a method for determining the shape of the boundary of the domain D_2. The image of the boundary of the basic domain D_1 under the ray mapping need not be circular when the phase is general. To find the shape of the boundary of D_2, a circular ring is placed on the plane $z = 50$. The image of the ring on the other plane is then determined. In actual optical measurements the image of the ring is not focused, of course, but in this report we idealized somewhat the situation and used a 'clean' form of the imaged ring. We used here an elliptical phase $0.01((x/2)^2 + y^2)$ and uniform intensity at the aperture. The wavenumber is $k = 200$, and the diffraction calculation used the Rayleigh–Sommerfeld integral.

In figure 6 we depict on the left the intensity distribution at the plane $z = 50$ and on the right its contour lines. An artificial ring is placed on that plane, and its contour is drawn with a broken line. Figure 7 is analogous to figure 6, except that the intensity here is evaluated at the plane $z = 60$. Again, notice the broken line, which is the image of the ring that was placed on the plane $z = 50$. It provides the shape of the boundary of D_2 that is the ray image of the boundary of the disc D_1 on the plane $z = 50$. The distribution of the 4000 sampling points is drawn in figure 8.

A number of numerical error estimates are given in table 3. In this table we compare the effect of working with different sampling points and different FE nodes on the accuracy. In all the runs in this table, the phase was computed from the simulated intensities on the planes $z = 50$ and 60. Runs (3a), (3b) and (3c) used 40 × 100 sampling points. The number of FE nodes was 498 in run (3a), 1196 in run (3b) and 1997 in run (3c). Runs (3d), (3e) and (3f) used 80 × 100 sampling points. The number of FE nodes was 498 in run (3d), 1196 in run (3e) and 1997 in run (3f).

4. Discussion

A number of conclusions can be drawn from the numerical simulations reported in section 4. Table 1 shows a significant improvement in accuracy by increasing the separation distance Z between the detection planes from unity to ten. This is the result of the lever principle. The ray direction is determined by the ratio of the ray deflection between the planes, i.e. by $\delta(x) = \bar{U}(x) - x$, and the separation distance Z between the planes. Hence, the error in the ray direction is a consequence of an error in the ray mapping. Therefore, increasing Z reduced considerably the direction error. We found that a further increase in Z does not have a marked effect on the error. This is probably due to the inaccuracy built into the calculation in this test by the assumption that the domain D_2 is circular just like D_1. In run (b) the initial error was deliberately made quite large, but the optimization process brought it down considerably, almost to the level of run (a), where the initial data was far more accurate. In a further test we averaged the ray directions of runs (c) and (d). We found no significant reduction in error through this step.

While in test 1 the simulation was done by exact diffraction integrals, we used in test 2 an approximate geometrical optics simulation. Thus, the error is purely due to the numerics. Indeed, runs (a) and (b) in test 2 are more accurate than the respective runs (a) and (d) of test 1, even though test 1 used many more FE nodes. Another advantage in the simulation of test 2 is that the domain D_2 is not assumed to be circular, but rather is given to the optimization code as part of the data. The accuracy of runs (a) and (b) in this test is of $\mathrm{O}(10^{-4})$ radians, which is rather remarkable, but one must recall that the data here are nearly ideal, which may not be the case in real experiments. A deliberate shuffling of the initial condition $U(x, 0)$ in runs (c) and (d) of test 2 gave rise to a large initial error. Although the error was significantly reduced by the optimization code, which strengthens our confidence in it, the optimized mapping in this case is not as good as in runs (a) and (b), that enjoyed better initial data. One might wonder about this sensitivity to initial conditions. Presumably, there is a unique minimizer that, since we are working with a fixed FE grid, should be recovered from any initial data. We

Table 2. Initial and final error norms for the simulations in test 2.

Run number	Initial $\|e_x\|_w$	Initial $\|e_y\|_w$	Optimized $\|e_x\|_w$	Optimized $\|e_y\|_w$
(a)	0.382 222	0.388 396	0.003 186	0.002 643
(b)	0.050 043	0.057 219	0.000 618	0.000 737
(c)	0.321 111	0.331 005	0.009 276	0.005 398
(d)	0.045 476	0.046 921	0.005 895	0.001 905

Table 2a. Initial and final relative error norms for the simulations in test 2.

Run number	Initial $\|e^{\mathrm{r}}_x\|_w$	Initial $\|e^{\mathrm{r}}_y\|_w$	Optimized $\|e^{\mathrm{r}}_x\|_w$	Optimized $\|e^{\mathrm{r}}_y\|_w$
(d)	2.350 944	4.290 766	0.223 338	0.096 255

Table 3. Initial and final error norms for the simulations in test 3.

Run number	Initial $\|e_x\|_w$	Initial $\|e_y\|_w$	Optimized $\|e_x\|_w$	Optimized $\|e_y\|_w$
(a)	0.011 188	0.014 992	0.008 515	0.010 371
(b)	0.011 007	0.014 799	0.008 899	0.009 575
(c)	0.010 944	0.014 745	0.003 167	0.003 374
(d)	0.005 685	0.007 573	0.004 544	0.005 197
(e)	0.005 598	0.007 468	0.004 721	0.004 791
(f)	0.005 564	0.007 442	0.002 181	0.002 415

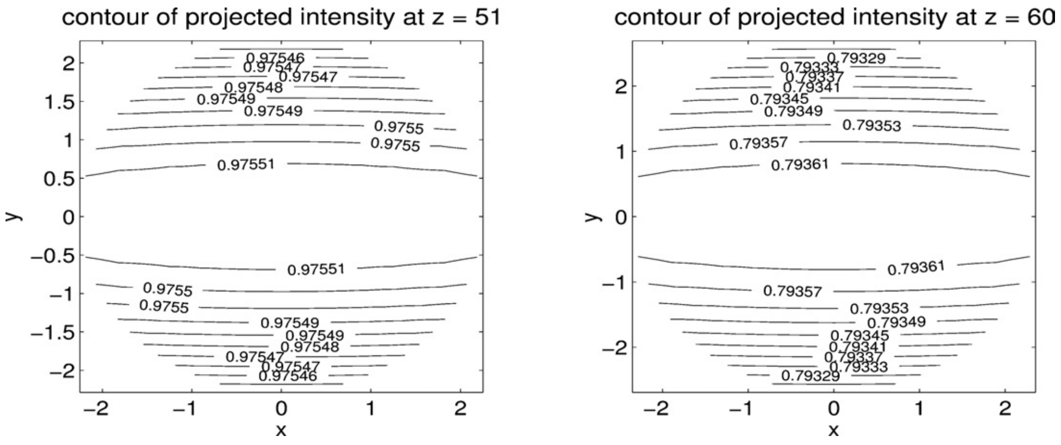

Figure 4. The intensity contours on the planes $z = 51$ (left) and $z = 60$ (right). The intensity is unity on $z = 50$.

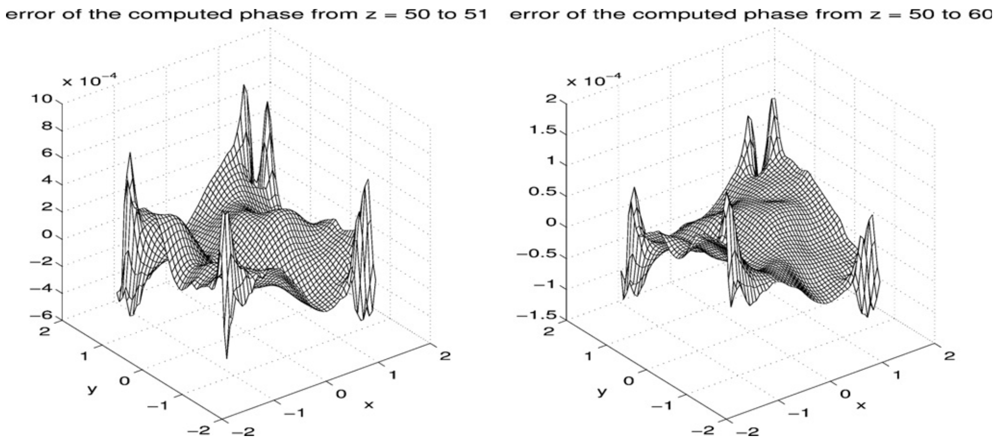

Figure 5. The error in the phase reconstruction (left—using the intensities at $z = 50$ and $z = 51$; right—using the intensities at $z = 50$ and 60).

have to recall, though, that the optimization problem for the functional M is not standard. One has to optimize M under the constraint (4). This means that the steepest descent flow for M proceeds along a specific manifold of functions. The

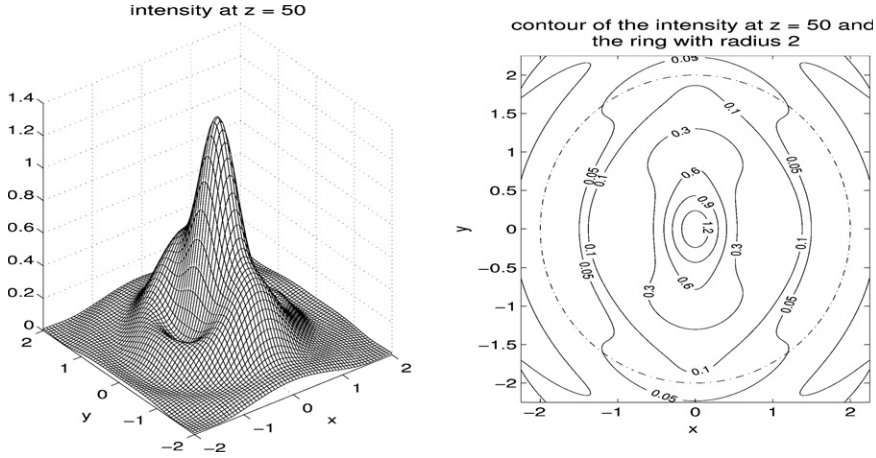

Figure 6. The intensity distribution on the plane $z = 50$ (left) and the contour map of the same intensity (right). An artificial ring is placed on the plane $z = 50$ and its contour is depicted by the broken line.

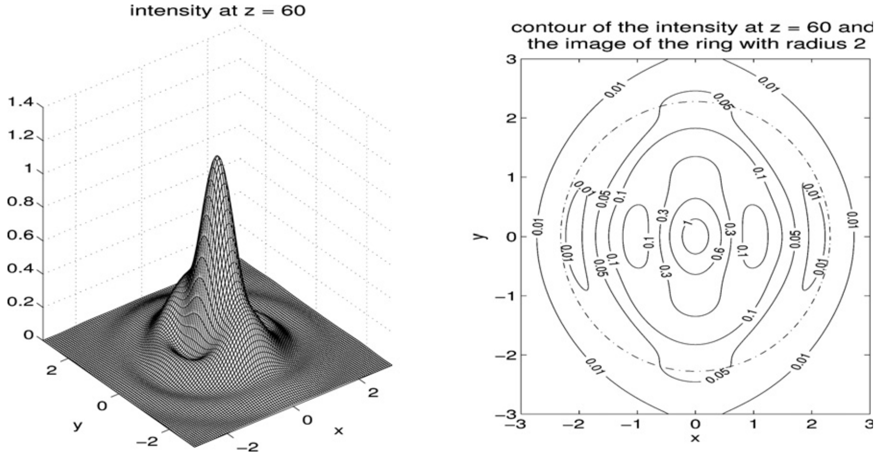

Figure 7. The intensity distribution on the plane $z = 60$ (left) and the contour map of the same intensity (right). The ring placed on the plane $z = 50$ is imaged on the plane $z = 60$ and its contour is depicted by the broken line.

manifold is determined by the initial conditions. Incidentally, the only role played by the intensity I_2 is in determining this manifold. Therefore, different initial conditions set up different manifolds, and this gives rise to different optimized solutions.

Test 3 has two main purposes. The first is to check in a controlled way the effect of increasing the number of sampling points and the number of FE nodes. The second goal is to simulate the ray mapping of a circular ring on the plane $z = 50$ on other planes. Runs (a)–(c) used the same number of intensity sampling points. Therefore, the initial error is essentially the same in all of them. Increasing the number of FE nodes improved the accuracy of the optimized mapping, but more tests are needed to determine the convergence rate. Doubling the number of intensity points halved the initial error in the ray directions. Again, increasing the number of FE nodes improved the accuracy of the optimized solution. We see in this table again that better initial data imply in general better optimized solutions under the same FE resolution. This is an important conclusion, since increasing the sampling resolution is a relatively cheap operation, in fact much cheaper than increasing the number of FE nodes. The ray map of the ring (broken circle in the right part of figure 6 on the plane $z = 50$) is mapped to the shape shown by the broken line in figure 7 on the plane $z = 60$. In real experiments we cannot expect such a nice clear image, and part of the work then would be to determine the shape of the ray-imaged circle for the diffracted image of the ring.

We summarize now our findings and further thoughts on the problem.

- The code that is explained in detail in the following appendix was demonstrated to be effective for the solution of the weighted least action problem. An important numerical issue that we realized is that one must perform the Helmholtz decomposition at each steepest descent step very accurately. This is essential in order to remain on the constrained manifold (4). Therefore, any solver should pay special attention to the Helmholtz decomposition.
- Good initial conditions are important too. Of course, since the phase is not known *a priori*, there is no way of being sure about the starting ray mapping. Nevertheless, one can assume that in practice the phase typically has no sharp gradients. Therefore, the main ingredient for good initial conditions is dense sampling. We used a very simple

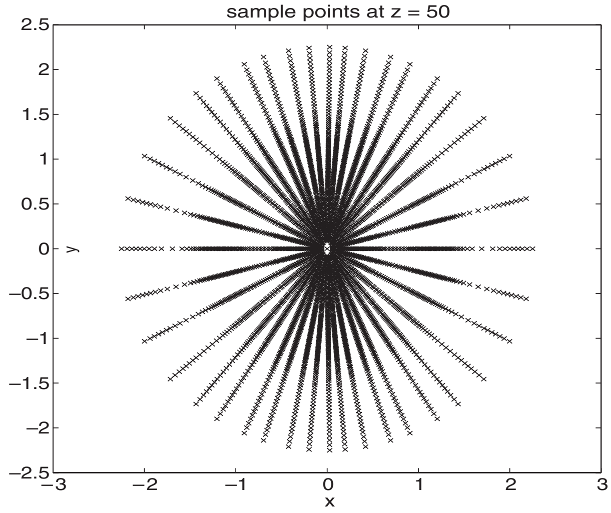

Figure 8. The sampling points consisting of 40 sectors with 100 points per ray.

sampling algorithm that is valid for circular domains. We also used a variant of it to noncircular domains. An outstanding problem is to develop a numerical method to generate appropriate sampling algorithm for general domains and to cases with highly nonuniform intensity.

- The number of rays is determined by the number of FE nodes. On the one hand, one would like to obtain the image of many rays to improve the phase resolution. On the other hand, increasing the number of FE nodes is computationally very costly. Therefore, it is useful to look for ways that increase accuracy or reduce running time. Here we see another advantage of starting with good initial data. If the initial mapping $U(x, 0)$ is close to the optimal solution, the running time would be short. It should be noted, though, that even with good initial conditions the current algorithm is too slow for real time applications such as adaptive optics.
- As was mentioned above the domain D_1 is typically circular. A good estimate of its ray-image D_2 is also needed. If no information beyond the intensities is given, one must make a guess about the shape of D_2, constrained only by the condition $\int_{D_1} I_1(x)\,\mathrm{d}x = \int_{D_2} I_2(x)\,\mathrm{d}x$. Since the intensity often decreases towards the boundary, the weight of the error induced by an incorrect shape of D_2 may not be severe. We examined here another option, namely to ray-image the shape of D_1 by placing a ring on the main plane and observe its image on the other plane.
- A number of researchers attempted to find the phase from intensity measurements using equation (1) alone. The equation is often called the transport of intensity equation (TIE) and the resulting phase reconstruction is called curvature sensing. Teague [11] seems to be one of the first people to propose to consider (1) as an equation for ψ, and not (as one might naturally do) as an equation for I. Thus, knowing I on two near-by planes enables the consideration of (1) as an elliptic second order equation for ψ. One of the difficulties in this approach is that one needs to supply (1) with boundary conditions [12, 13]; however, it is not clear how to get such conditions.

There are a number of important differences between the TIE and the WLAP methods. First, note that, in contrast to the TIE-based method, in the current (WLAP) method the separation Z between the detection planes need not be small. One advantage of this feature is that increasing Z decreases the error in the ray direction (since the ray direction is obtained by the ray deviation in the detection plane divided by Z). Another advantage of using relatively large Z is that the intensity variations are more appreciable with respect to the system's noise.

Another, related, advantage of the WLAP method is that while the TIE requires the differentiation of the intensity data (a step known to enhance measurement error) the WLAP integrates the data. Therefore, the WLAP method has the potential of reducing measurement errors. We have not yet explored this issue, though.

Finally, we point out that since the WLAP is based on a variational principle the solution is obtained by a minimization process. Such processes are typically quite stable numerically. On the other hand, the WLAP method, and in particular the algorithm that we used, are far more complex than the solution based on the TIE.

Acknowledgments

CL and JL were supported by NSF-DMS grant and JR was supported by an ISF grant.

Appendix. The numerical algorithm

In this appendix we describe a numerical algorithm to solve the optimization problem (5). The method is based on solving the evolution equation (8) towards a critical point of the energy function M. The two intensity distributions I_1, I_2 as described in the previous section are assumed to be given. The first step towards finding the ray mapping $\bar{U}$ is to set up initial conditions $U(x, t = 0)$ for the flow (8). Having done so, we shall present a finite element (FE) solver for the Helmholtz decomposition (7), and then evolve U according to (8). We now explain in detail the implementation of each of the steps.

Intensity distributions and initial conditions

The theory in section 2 was formulated for intensity distributions provided over the entire plane. In practice the distributions $I_i(x)$ are given only on some finite domain(s) in the plane. For simplicity, we assume first that I_i are both defined over discs. The compatibility condition $\int_{D_1} I_1(x)\,\mathrm{d}x = \int_{D_2} I_2(x)\,\mathrm{d}x = m$ is also assumed. The restriction to discs is particularly relevant to many optical applications. Extending the method described below to other domains can be done in a number of ways. For example, given a domain D, one can consider the smallest disc that bounds D, extend the intensity I, defined in D, to be zero in the part of the disc outside D, and proceed as before.

To find an initial intensity transporting mapping $U(x, 0)$ the distributions I_1, I_2 are discretized. Since the mapping is based on preserving intensities, one should construct discrete arrays such that each point (on its respective plane) carries an equal amount of intensity. Consider for example the plane

P_1. We first divide the disc into k_θ sectors such that the total intensity in each sector σ_j is $\int_{\sigma_j} I_1(x)\,\mathrm{d}x = m/k_\theta$. The set of sectors naturally defines a set of radii extending from the origin to the disc's boundary. Each radius divided into k_r points, such that the integral of I_1 between adjacent points along the radius is the same. This process defines a set $\mathcal{P}_1$ of $k = k_\theta \times k_r$ points (x_i^1, y_i^1) in D_1. A similar process is applied to the intensity I_2 in the disc D_2, resulting in a set $\mathcal{P}_2$ of points (x_j^2, y_j^2). It remains to define an association between the two sets of points defined above. This can be done by defining an ordering on these sets. For this purpose observe that there is some arbitrariness in the definition of the sectors. We must decide on an initial angle with respect to which to measure them. This degree of freedom can be removed by selecting the angle $\theta = 0$ as the radius for the first sector. Once the sectors in both planes P_1 and P_2 are defined uniquely, we can order the points in the two sets $\mathcal{P}_i$. The points along the first radius (defined by $\theta = 0$) are ordered first, and then the other points are concatenated, radius after radius. Once an ordering is defined, an association between the points in $\mathcal{P}_1$ and $\mathcal{P}_2$ is induced by relating two points with the same position in the ordered sets to each other. In some of the numerical tests that are described in the simulation section we constructed initial conditions in which a permutation was applied to the association above. The permutation gave rise to an initial condition that consists of an association between points that are far from each other in their respective domains. This enables us to test how the algorithm and code seek the minimum of M while starting from an initial mapping that is far from the minimum.

A finite element solver was used to solve the system (7), (8) on the discrete level. Accordingly, the disk D_1 is covered by a quasi-uniform triangulation $\mathcal{T}_h$. The algorithm developed by Persson and Strang [5] is used to generate the mesh consisting of unstructured triangular elements. In this algorithm, the triangle vertices are modelled as the nodes of a truss, and linear force–displacement equations are used to equilibrate the bars of the truss. The Delaunay algorithm is used to construct the mesh topology by defining the triangle edges. The domain geometry is defined implicitly using a distance function. This algorithm is easy to implement; we used the Matlab code developed in [5]. The nodes and mesh connectivity are then used by the FE solver to perform the Helmholtz decomposition (7) and the steepest descent evolution equation (8).

The Helmholtz decomposition

To perform the Helmholtz decomposition (7), where the normal component of V on the boundary ∂D_1 vanishes (i.e. $V \cdot n = 0$), a dual mixed method together with the MINI element basis is used (e.g. [6–8]). The MINI element basis is the smallest set of basis functions for which numerical stability can be achieved for the dual mixed method [6–8].

Equation (7) can be rewritten in weak form as follows. Let W and Q be vector and scalar test functions in $H_0(\mathrm{div}, D_1)$ and $L^2(D_1)$ respectively. Here, $H_0(\mathrm{div}, D_1)$ is the space of square integrable vector functions with square integrable divergence over the domain D_1. The subscript 0 denotes that the normal component of a function in $H_0(\mathrm{div}, D_1)$ is zero at the boundary ∂D_1. The space $L^2(D_1)$ denotes the space of square integrable functions. Then, given U the solutions P and V of equation (7) can be obtained as the solution pair $(V, P) \in H_0(\mathrm{div}, D_1) \times L^2(D_1)$ to

$$\int_{D_1} V \cdot W\,\mathrm{d}x - \int_{D_1} P\,\mathrm{div}\,W\,\mathrm{d}x = \int_{D_1} U \cdot W\,\mathrm{d}x, \quad \forall W \in H_0(\mathrm{div}, D_1), \tag{20}$$

$$\int_{D_1} Q\,\mathrm{div}\,V\,\mathrm{d}x = 0, \qquad \forall Q \in L^2(D_1). \tag{21}$$

To obtain a unique 'pressure' P we set its average to zero. Equations (20) and (21) are the weak (variational) formulation of the Helmholtz decomposition (7). To solve this system numerically, a discrete version of this variational formulation is developed. Let $\mathcal{T}_h$ denote the triangulation of the domain D_1. On each triangle $K \in \mathcal{T}_h$, the discrete test functions $W_h \in H_{0,h}(\mathcal{T}_h) \subset H_0(\mathrm{div}, D_1)$ are characterized in terms of MINI element basis functions [6–8]:

$$H_{0,h}(\mathcal{T}_h) = \left\{ W_h \in H_0(\mathrm{div}; D_1) : W_h = \sum_i a_i \eta_i(x), \forall K \in \mathcal{T}_h \right\}, \tag{22}$$

where a_i are (constant) coefficients and η_i are local vector valued basis functions in triangle K. For the MINI element basis functions, the degrees of freedom (nodes) are the triangle vertices and the triangle centroids. At interior nodes, there are two degrees of freedom (i.e. two basis functions) corresponding to the two coordinate directions. At boundary nodes, there is a single degree of freedom because V has zero normal component.

The MINI element basis vector functions η_i can be described in terms of barycentric coordinates as follows. On any triangle K, denote the barycentric coordinates by λ_1, λ_2, λ_3. That is, λ_i (where $i = 1, 2, 3$) is equal to unity at vertex i, is equal to zero at all other vertices of the triangle and is linear in between. For example, on a triangle with vertices located at $\mathbf{x}_1 = (-1, 0)$, $\mathbf{x}_2 = (1, 0)$ and $\mathbf{x}_3 = (0, \sqrt{3})$, the barycentric coordinates are

$$\begin{aligned} \lambda_1(\mathbf{x}) &= \tfrac{1}{2}\left(1 - x_1 - x_2/\sqrt{3}\right), \\ \lambda_2(\mathbf{x}) &= \tfrac{1}{2}(1 + x_1 - x_2/\sqrt{3}), \\ \lambda_3(\mathbf{x}) &= x_2/\sqrt{3}, \end{aligned} \tag{23}$$

where $\mathbf{x} = (x_1, x_2)$. Now, if the triangle vertex i does not belong to ∂D_1, then a basis function is introduced for each spatial coordinate. For the MINI element basis, the basis functions corresponding to this node consist of the (vector) linear polynomials $(\lambda_i, 0)$ and $(0, \lambda_i)$ where λ_i are the barycentric coordinates. If a triangle vertex j belongs to ∂D_1, then only one basis function is used because the normal component of U at vertex j should be equal to zero. In this case, the MINI element basis function corresponding to this node is the vector $\lambda_j s_j$ where s_j is the unit tangent vector at vertex j. Finally, since the triangle centroids do not belong to ∂D_1, two basis functions are introduced again for each spatial coordinate. In the MINI element framework, at a triangle centroid l, these basis functions take the form of the so-called bubble (cubic polynomial) functions: $(\lambda_1\lambda_2\lambda_3, 0)$ and

$(0, \lambda_1\lambda_2\lambda_3)$ for the x- and y-coordinates respectively where λ_i are the barycentric coordinates for the triangle with centroid l.

The discrete scalar test functions $Q_h \in L_h^2(\mathcal{T}_h)$ are mean zero, piecewise linear continuous functions, i.e.

$$L_h^2(\mathcal{T}_h) = \left\{ Q_h \in L^2(D_1) \cap C^0(D_1) : \right.$$
$$\left. Q_h = \sum_{i=1}^{3} a_i \lambda_i(x),\ \forall\, K \in \mathcal{T}_h \right\}, \tag{24}$$

where again the λ_i are the barycentric coordinates discussed above. With these choices, the discrete form of the system (20), (21) becomes the following.

Find $(V_h, P_h) \in H_{0,h}(\mathcal{T}_h) \times L_h^2(\mathcal{T}_h)$ such that

$$\int_{D_1} V_h \cdot W_h \,\mathrm{d}x - \int_{D_1} P_h \operatorname{div} W_h \,\mathrm{d}x$$
$$= \int_{D_1} U_h \cdot W_h \,\mathrm{d}x, \qquad \forall\, W_h \in H_{0,h}(\mathcal{T}_h), \tag{25}$$

$$\int_{D_1} Q_h \operatorname{div} V_h \,\mathrm{d}x = 0, \qquad \forall\, Q_h \in L_h^2(\mathcal{T}_h), \tag{26}$$

where U_h is the projection of U onto $H_{0,h}(\mathcal{T}_h)$. Standard FE arguments (e.g. [9, 7]) imply that equations (25), (26) are equivalent to a linear system of the form

$$\begin{aligned} AV_h + B^T P_h &= F, \\ BV_h &= G, \end{aligned} \tag{27}$$

for appropriately constructed matrices A, B and vectors F, G. We use the conjugate gradient method to solve this linear system.

It can be shown rigorously (e.g. see section 14 in [8]) that V_h and P_h are first order accurate approximations for V and P in the $H_0(\mathrm{div}, D_1)$ (i.e. $\int_{D_1} \left(|V|^2 + (\operatorname{div} V)^2\right) \mathrm{d}x$) and L^2 (i.e. $\int_{D_1} P^2 \,\mathrm{d}x$) norms respectively. In practice, however, we find that the algorithm is second order accurate for both V and P in the L^2 norm; in fact, P_h is found to be a second order accurate approximation in the maximum norm. Finally, the divergence of V_h in the weak sense (i.e. $\int_{D_1} Q_h \operatorname{div} V_h \,\mathrm{d}x$) is zero to within the error tolerance of the conjugate gradient iterative solver.

Evolving the conservation law

A discontinuous Galerkin (DG) method (e.g. see [10]), with linear basis functions on each triangle K, is used to solve the evolution equation (8). The triangulation $\mathcal{T}_h$ is the same as that for the Helmholtz decomposition.

The evolution algorithm is based on a conservative discretization of equation (8) in component form. Taking $U = (U_1, U_2)$, equation (8) can be rewritten as

$$\frac{\partial U_l}{\partial t} + \nabla \cdot (I_1^{-1} V U_l) = U_l \nabla \cdot (I_1^{-1} V),$$
$$\text{for } l = 1, 2. \tag{28}$$

The DG method is derived as follows. On each triangle $K \in \mathcal{T}_h$, multiply equation (28) by a linear (P^1) test function ζ and integrate over K. This gives

$$\int_K \frac{\partial U_l}{\partial t} \zeta \,\mathrm{d}x + \int_{\partial K} I_1^{-1} V \cdot n \widehat{U}_l \zeta \,\mathrm{d}s - \int_K I_1^{-1} U_l V \cdot \nabla \zeta \,\mathrm{d}x$$
$$= \int_K \zeta U_l \nabla \cdot (I_1^{-1} V) \,\mathrm{d}x, \qquad \text{for } l = 1, 2, \tag{29}$$

where $\widehat{U}_l$ is the numerical flux on the edge ∂K. For stability, the upwind flux is used:

$$\widehat{U}_i = \begin{cases} U_l^- & \text{if } V \cdot n \geqslant 0, \\ U_i^+ & \text{if } V \cdot n < 0, \end{cases}$$

where U_l^- denotes the value taken inside K, and U_l^+ taken from the neighbouring triangle. On each triangle K, the function U_l is assumed to be a linear function of the form $U_l = \sum_{i=1}^3 u_{l,i}(t)\lambda_i(x)$ where the λ_i (described in the previous section) are the basis functions for each node $i = 1, 2, 3$ and $u_{l,i}(t)$ are time-dependent coefficients. The U_l are not assumed to be continuous at nodes or across triangle edges. With these assumptions, equation (29) reduces to a first order ordinary differential equation in time for the $u_{l,i}(t)$. We show below how such potentially discontinuous solutions are made compatible with the Helmholtz decomposition solver.

A second order accurate Runge–Kutta scheme is used to discretize equation (29) in time. Let $u_{l,i}$ denote a time-dependent coefficient of U_l with $u_{l,i}^n \approx u_{l,i}(t_n)$. Then, for given $u_{l,i}^n$, the value at the next time step $u_{l,i}^{n+1}$ is obtained by

$$u_{l,i}^* = u_{l,i}^n + \Delta t \frac{\partial u_{l,i}}{\partial t}(t^n, u^n) \tag{30}$$

$$u_{l,i}^{**} = u_{l,i}^* + \Delta t \frac{\partial u_{l,i}}{\partial t}(t^n + \Delta t, u^*) \tag{31}$$

$$u_{l,i}^{n+1} = (u_{l,i}^n + u_{l,i}^{**})/2. \tag{32}$$

It can be shown that this algorithm is second order accurate in the L^2 and maximum norms. A detailed description and error analysis of the DG method can be found in [10].

Coupling the Helmholtz decomposition and evolution algorithms

First, the initial condition U_h^0 is obtained by projecting $U(x, t = 0)$ onto the finite element space $H_{0,h}(\mathcal{T}_h)$. That is, $U_h^0 = \sum_i a_i \eta_i(x)$ where the coefficients a_i are found by integrating $U(x, t = 0)$ against η_i and using that $\int_\Omega \eta_i \eta_j \,\mathrm{d}x = 0$ for $i \neq j$. Note that U_h^0 is a continuous function. Second, the discrete Helmholtz decomposition system (25), (26) is solved to obtain V_h^0 at triangle vertices and centroids. Third, the evolution equation is solved using U_h^0 and V_h^0 obtained from the discrete Helmholtz decomposition solver. Note that this requires a second discrete Helmholtz decomposition to determine V_h^*, from the intermediate step U_h^*, that is needed to evaluate $\frac{\partial u_{l,i}}{\partial t}(0 + \Delta t, u^*)$.

Because U_h^* is discontinuous, U_h^* is projected into the space of vector-valued continuous piecewise linear functions (e.g. $L_h^2 \times L_h^2$), where

$$L_h^2(\mathcal{T}_h) = \left\{ Q_h \in L^2(D_1) \cap C^0(D_1) : \right.$$
$$\left. Q_h = \sum_{i=1}^{3} a_i \lambda_i(x), \forall K \in \mathcal{T}_h \right\}, \tag{33}$$

in order to perform the second discrete Helmholtz decomposition to obtain V_h^*. The projected U_h^* is now a continuous L^2

function. The updated function after the full time step, U_h^1, is also analogously projected. The process is then repeated until the L^2 norm of V_h^n falls below a specified tolerance level, $\|V_h^n\|_{L^2} < \text{tol}$, where tol is typically set to 10^{-8}.

References

[1] Gureyev T E and Wikins S W 1998 On x-ray phase imaging with a point source *J. Opt. Soc. Am.* A **15** 579–85

[2] Rubinstein J and Wolansky G 2004 A variational principle in optics *J. Opt. Soc. Am.* A **21** 2164–72

[3] Angenent S, Haker S and Tannenbaum A 2003 Minimizing flows for the Monge–Kantorovich problem *SIAM J. Math. Anal.* **35** 61–97

[4] Brenier Y 1991 Polar factorization and monotone rearrangement of vector-valued functions *Commun. Pure Appl. Math.* **64** 375–417

[5] Persson P O and Strang G 2004 A simple mesh generator in Matlab *SIAM Rev.* **46** 329–45

[6] Arnold D N, Brezzi F and Fortin M 1984 A stable finite-element method for the Stokes equations *Calcolo* **21** 337

[7] Fortin M and Brezzi F 1991 *Mixed and Hybrid Finite Elements* (New York: Springer)

[8] Roberts J E and Thomas J M 1991 Mixed and hybrid methods *Handbook of Numerical Analysis* vol 2 *Finite Element Method (Part 1)* ed P G Ciarlet and J L Lions (Amsterdam: Elsevier Science)

[9] Atkinson K and Han W 2001 *Theoretical Numerical Analysis* (New York: Springer)

[10] Cockburn B and Shu C W 2001 Runge–Kutta discontinuous Galerkin methods for convection-dominated problems *J. Sci. Comput.* **16** 173–261

[11] Teague M R 1983 Deterministic phase retrieval: a Green's function solution *J. Opt. Soc. Am.* **73** 1434–41

[12] Roddier F 1998 Curvature sensing and copensation: a new concept in adaptive optics *Appl. Opt.* **27** 1223–5

[13] Gureyev T E and Nugent K A 1995 Phase retrieval with the transport of intensity equation II: Orthogonal series solution for nonuniform illumination *J. Opt. Soc. Am.* A **13** 1670–82