



<https://theses.gla.ac.uk/>

Theses Digitisation:

<https://www.gla.ac.uk/myglasgow/research/enlighten/theses/digitisation/>

This is a digitised version of the original print thesis.

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study,  
without prior permission or charge

This work cannot be reproduced or quoted extensively from without first  
obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any  
format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author,  
title, awarding institution and date of the thesis must be given

Enlighten: Theses

<https://theses.gla.ac.uk/>  
[research-enlighten@glasgow.ac.uk](mailto:research-enlighten@glasgow.ac.uk)

Analogies between light and matter waves:

From laser modes to phase holography

by

Graeme B. Whyte

DOCTOR OF PHILOSOPHY

in

The Faculty of Physical Sciences

Department of Physics and Astronomy

THE UNIVERSITY OF GLASGOW

April 26, 2007

© Graeme B. Whyte, 2006

ProQuest Number: 10800614

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10800614

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

GLASGOW  
UNIVERSITY  
LIBRARY:

# Abstract

With the creation of the laser in the 1960s, optics research gained a whole new type of coherent, well-behaved light with which to experiment. Later, similar matter was created; first in the form of super-fluids and then Bose-Einstein condensates (BECs).

A BEC as a whole behaves in many ways analogous to a monochromatic laser beam; the ultra-cold atoms in a BEC are the matter equivalent of photons in the laser beam. Researchers have built the BEC analogs of a number of optical components, including lenses and beam splitters. The work described in this thesis was inspired by the aim to investigate theoretically BEC analogs of effects known from laser physics and by developing a BEC analog of Fourier holography.

After introductions into lasers and BECs (chapter 1) and numerical methods for solving the differential equations governing the behaviour of both light and BECs (chapter 2), this thesis comes in two parts.

The first part is concerned with the BEC analogs of the formation of transverse laser modes (chapter 3) and an interferometer for sorting optical vortices (chapter 5), and a non-destructive method of Fourier-transforming a BEC (chapter 4).

The second part is about optical holography and optical tweezers. It

---

starts with a review of hologram-design algorithms (chapter 6). Originally inspired by optical Fourier holography, discussions about BEC Fourier holography counter-inspired a new algorithm for optical Fourier holography, namely a Gerchberg-Saxton algorithm for 3-dimensional light shaping (chapter 7). My work on the improvement of hologram-calculation software for optical tweezers is described in chapter 8.

## Publications

- G. Whyte, G. Gibson, J. Leach, D. Robert, M. Miles and M. Padgett. “An optical trapped microhand for manipulating micron-sized objects,” *Opt. Ex.* **14**, 12497 (2006).
- G. Gibson, L. Barron, F. Bec, G. Whyte and M. Padgett. “Optically controlled grippers for manipulating micron-sized particles,” *New J. Phys.* **9**, 15 (2007).
- L. Thomson, Y. Bosseil, G. Whyte, E. Yao and J. Courtial. “Super-Resolution in Holographic Optical Tweezers,” Submitted to *New J. Phys.* Sept. 2006.
- G. Whyte, P. Öhberg, J. Courtial. “Fourier Transform in Bose-Einstein Condensates,” *New J. Phys.* **8** 196 (2006).
- M. Padgett, G. Whyte, J. Girkin, A. Wright, L. Allen, P. Öhberg, and S. M. Barnett. “Polarization and image rotation induced by a rotating dielectric rod: an optical angular momentum interpretation,” *Opt. Lett.* **31**, 2205 (2006).
- J. Courtial, G. Whyte, Z. Bouchal and J. Wagner “Iterative algorithms for holographic shaping of non-diffracting and self-imaging light beams,” *Opt. Ex.* **14**, 2108 (2005).

- 
- G. Whyte and J. Courtial. “Experimental demonstration of holographic three-dimensional light shaping using a Gerchberg-Saxton algorithm,” *New J. Phys.* **7**, 117 (2005).
  - G. Whyte, J. Veitch, P. Öhberg and J. Courtial “Vortex sorter for Bose-Einstein condensates,” *Phys. Rev. A.* **70** 011603 (2004).
  - G. Whyte, P. Öhberg and J. Courtial. “Transverse laser modes in Bose-Einstein condensates,” *Phys. Rev. A.* **69** 053610 (2004).



# Contents

<b>Abstract</b> . . . . .	1
<b>Publications</b> . . . . .	3
<b>Contents</b> . . . . .	5
<b>List of Figures</b> . . . . .	11
<b>Acknowledgements</b> . . . . .	14
<b>1 Introduction</b> . . . . .	15
1.1 Layout of this work . . . . .	15
1.2 Bose-Einstein condensates . . . . .	16
1.3 Lasers and BECs . . . . .	17
<b>2 Numerical Theory</b> . . . . .	19
2.1 Gross-Pitaveskii Equation . . . . .	19
2.1.1 Dimensionless Formulation . . . . .	21
2.2 Plane wave decomposition . . . . .	22
2.2.1 Plane wave method details . . . . .	23
2.3 Numerical Details . . . . .	23
2.3.1 $k$ space differences . . . . .	24

2.4	Summary . . . . .	25
<b>I Some optical ideas applied to Bose-Einstein condensates</b>		<b>26</b>
<b>3</b>	<b>Transverse laser modes in Bose-Einstein condensates . . . . .</b>	<b>27</b>
3.1	Introduction . . . . .	28
3.2	Laser Cavities . . . . .	29
3.3	Bose-Einstein Condensate Cavities . . . . .	31
3.4	BEC focusing . . . . .	32
3.5	Eigenmode selection in laser resonators . . . . .	35
3.6	Creation of Hermite-Gaussian Modes . . . . .	36
3.6.1	Round trip losses . . . . .	37
3.6.2	Simulation Results . . . . .	37
3.7	Conversion into Laguerre-Gaussian-like modes . . . . .	43
<b>4</b>	<b>Fourier Transforming a Bose-Einstein Condensate . . . . .</b>	<b>49</b>
4.1	Introduction . . . . .	50
4.2	The Fourier Transform . . . . .	51
4.2.1	Simple examples . . . . .	52
4.2.2	Gaussian transforms . . . . .	53
4.3	Fractional Fourier Transform . . . . .	53
4.4	Fourier Transforms in Bose-Einstein Condensates . . . . .	55
4.4.1	Fourier Transforming with lens pulses . . . . .	55
4.4.2	Fourier Transforming by waiting . . . . .	57
4.5	Simulation Results . . . . .	61

---

4.6	Fourier filtering . . . . .	68
<b>5</b>	<b>Vortex sorter for Bose-Einstein condensates . . . . .</b>	<b>71</b>
5.1	Introduction . . . . .	72
5.2	Vortices . . . . .	73
5.2.1	Vortex Phase . . . . .	73
5.2.2	Laguerre-Gaussian modes . . . . .	74
5.2.3	Optical Vortices . . . . .	76
5.2.4	BEC Vortices . . . . .	76
5.3	Vortex sorter . . . . .	77
5.4	Bose-Einstein condensate vortex sorter . . . . .	79
5.4.1	Bragg pulse splitting . . . . .	79
5.4.2	BEC setup . . . . .	82
5.5	Rotation of BECs . . . . .	85
5.5.1	Summary . . . . .	89
<b>II</b>	<b>Shaping light</b>	<b>92</b>
<b>6</b>	<b>Hologram Design . . . . .</b>	<b>93</b>
6.1	Introduction . . . . .	93
6.1.1	Holography with restrictions . . . . .	94
6.1.2	Intensity holography . . . . .	94
6.1.3	Phase holography . . . . .	95
6.1.4	Fourier phase holography . . . . .	96
6.2	Superposition approximation algorithm . . . . .	98
6.3	Additional phase structure . . . . .	99

---

6.4	Gerchberg-Saxton algorithm . . . . .	101
6.5	Modified Gerchberg-Saxton algorithm . . . . .	105
6.5.1	Adaptive Additive algorithm . . . . .	105
6.5.2	Curtis-Koss-Grier algorithm . . . . .	106
6.5.3	3D Gerchberg-Saxton . . . . .	106
6.6	Direct Binary Search . . . . .	108
6.7	Simulated Annealing . . . . .	109
6.8	Summary . . . . .	110
<b>7</b>	<b>Three-dimensional light shaping . . . . .</b>	<b>111</b>
7.1	Introduction . . . . .	112
7.2	Details of our implementation of the 3D Gerchberg-Saxton algorithm . . . . .	114
7.2.1	Thickness of $k$ -space sphere . . . . .	114
7.2.2	Influence of numerical aperture . . . . .	118
7.2.3	Simulation parameters and procedure . . . . .	120
7.3	Experiment . . . . .	122
7.3.1	Calculation of the phase-hologram pattern . . . . .	124
7.3.2	Results . . . . .	126
7.4	Summary . . . . .	126
<b>8</b>	<b>Optical Tweezer Interfaces . . . . .</b>	<b>129</b>
8.1	Optical Tweezers . . . . .	130
8.1.1	The forces involved . . . . .	130
8.2	Indirect Manipulation . . . . .	133
8.2.1	Difficult-to-trap objects . . . . .	133

---

8.2.2	Scattering methods . . . . .	134
8.2.3	Indirect Manipulation . . . . .	134
8.3	Real Time hologram calculation . . . . .	134
8.3.1	Superposition approximation algorithm optimisations .	135
8.4	Human-Tweezers Interaction . . . . .	145
8.4.1	Joystick Control . . . . .	145
8.5	Simple hand-based interface . . . . .	147
8.5.1	Imaging the user's hands . . . . .	147
8.5.2	Image processing . . . . .	148
8.5.3	Trap movement . . . . .	150
8.5.4	Layout of the interface . . . . .	151
8.5.5	Uses of the interface . . . . .	151
8.5.6	Examples of applications of the interface . . . . .	153
8.6	Summary . . . . .	156
 <b>III Conclusions and Appendix</b>		<b>159</b>
9	<b>Conclusions</b> . . . . .	160
9.1	Bose-Einstein condensates . . . . .	160
9.2	Light Shaping . . . . .	161
10	<b>Outlook</b> . . . . .	163
A	<b>Software User's Guide</b> . . . . .	164
A.1	Introduction . . . . .	164
A.1.1	Who should use this guide . . . . .	164

---

A.1.2	The software . . . . .	164
A.2	Information flow . . . . .	165
A.2.1	Why client-server . . . . .	165
A.2.2	Entering details to the server . . . . .	166
A.2.3	Between the client and sever . . . . .	166
A.3	Logging In . . . . .	168
A.4	Working with jobs . . . . .	168
A.4.1	Viewing jobs . . . . .	169
A.4.2	Creating jobs . . . . .	172
A.4.3	Editing jobs . . . . .	175
A.4.4	Starting jobs . . . . .	179
A.4.5	Stopping jobs . . . . .	180
A.5	Components . . . . .	182
A.5.1	Parameters . . . . .	182
A.5.2	Adding new components . . . . .	183
A.6	Client software . . . . .	185
A.6.1	Installing . . . . .	185
A.6.2	Running . . . . .	185
A.6.3	Options . . . . .	186
A.6.4	Windows . . . . .	188
A.6.5	Running jobs . . . . .	193
A.6.6	Results . . . . .	196
A.6.7	Running unattended . . . . .	196
<b>Bibliography</b>	. . . . .	<b>198</b>

---

# List of Figures

3.1	Unfolding a laser cavity . . . . .	30
3.2	Focusing a BEC with a lens pulse . . . . .	33
3.3	Hermite-Gaussian modes and the local loss required to create them . . . . .	36
3.4	The structure of the BEC during the first few iterations in a cavity . . . . .	38
3.5	Example eigenmodes produced in a BEC cavity . . . . .	39
3.6	Changes in the structure as the non-linearity is increased . . . .	41
3.7	Profile of the center of a HG <sub>2,0</sub> mode . . . . .	42
3.8	Creation of non-symmetric modes . . . . .	44
3.9	Effects of cavity design on structure . . . . .	45
3.10	Conversion into a Laguerre-Gaussian like mode . . . . .	47
4.1	Optical system of Fourier Transforming . . . . .	52
4.2	Fourier transforms of a plane wave . . . . .	54
4.3	Optical systems for fractional Fourier transforming a light beam	56
4.4	The dynamics of 1D breathing-dipole modes . . . . .	59
4.5	Examples of non-interacting BEC Fourier transforms . . . . .	63
4.6	The shift of the Fourier transform as the non-linearity increases	65

---

4.7	Fraction of a BEC which is Fourier transformed as time passes	66
4.8	The performance of the Fourier transform as the non-linearity is increased . . . . .	67
4.9	Fourier filtering in BECs . . . . .	69
5.1	Line integral of phase around multiple vortices . . . . .	75
5.2	Optical system to sort vortices . . . . .	78
5.3	Schematic of a BEC system to sort vortices . . . . .	80
5.4	Results of vortex sorting in BECs . . . . .	84
5.5	Fraction of the BEC correctly sorted . . . . .	86
5.6	Different rotation methods . . . . .	88
5.7	Comaprison of rotation methods within the sorter . . . . .	90
6.1	Intensity-only and phase-only holography . . . . .	95
6.2	Imperfect phase imprinting and the global shift method . . . . .	97
6.3	Additional phase structures . . . . .	100
6.4	Projection onto convex sets . . . . .	102
6.5	The Gerchberg-Saxton algorithm . . . . .	103
6.6	3D representation of light . . . . .	107
7.1	Examples of shaped 3D light intensities . . . . .	115
7.2	The structure of laser light's $k$ -space in 3D . . . . .	116
7.3	Effect of the thickness of $k$ -space hemisphere on shaping performance . . . . .	119
7.4	The effect of the cone-angle on light shaping . . . . .	121
7.5	Experimental setup to perform 3D light shaping . . . . .	123
7.6	Extracting the phase from a $k$ -space hemi-sphere . . . . .	125



---

7.7	Experimental results of the 3D light shaping . . . . .	127
8.1	Gradient forces involved in optical trapping . . . . .	132
8.2	The phase add optimisation . . . . .	140
8.3	Performance of the phase add optimisation . . . . .	141
8.4	Methods of adding multiple phases using the phase-add optimisation . . . . .	143
8.5	Manipulating traps using a joystick . . . . .	146
8.6	Image processing required to process image of user's hands . .	149
8.7	The interface used to manipulate optical traps using the user's fingers . . . . .	152
8.8	Manipulating a single red blood cell with the user's hands . .	154
8.9	Manipulating a metal particle using beads controlled by the user's hands . . . . .	157
A.1	The login options . . . . .	167
A.2	The listing of the jobs stored on the server . . . . .	170
A.3	The new job wizard . . . . .	173
A.4	Editable and un-editable jobs . . . . .	176
A.5	List DLLs . . . . .	181
A.6	The options dialog . . . . .	187
A.7	The main window . . . . .	189
A.8	The job queue . . . . .	191
A.9	The component log . . . . .	192
A.10	The application log window . . . . .	194

# Acknowledgements

There are many people who have helped me throughout these last three years and I'd like to thank them all for their support and friendship. Of those there are several who I'd like to specially thank. Firstly, Dr Johannes Courtial for endless support, groovy ideas and his Germanic attention to detail. I would also like to thank Prof. Miles Padgett for his inspirational ideas, support and letting me loose on the optical tweezers, Dr Jonathan Leach for his exceptional experimental knowledge, and Dr Ben van Well and Laura Thomson for keeping me sane.

A big thanks to Dr Patrik Öhberg for his invaluable Bose-Einstein condensate knowledge and ensuring we weren't talking rubbish while exploring this field.

Finally a very special thanks to Cassie for her unwavering support throughout and her ability to look interested when listening to my physics-related rants.

# Chapter 1

## Introduction

### 1.1 Layout of this work

This work can be split into two separate sections: optical analogies within Bose-Einstein condensates (BECs), and light shaping.

As will be shown, laser light and BECs are very similar in their behaviour despite their very different origins. This similarity has inspired many optical elements to be adapted for use in BECs (for example, lenses<sup>1</sup>, axicons<sup>2</sup> and beam splitters<sup>3</sup>). This has led us to investigate these similarities in further detail by simulating more complex optical systems, firstly in the form of a resonator capable of generating higher-order modes (chapter 3) and then an interferometer to decompose a light beam into its orbital angular momentum components (chapter 5). The process of forming higher-order modes gives rise to the idea of generalised BEC shaping where imprinting a pre-calculated phase pattern results in the BEC evolving into a desired structure. Much work has already been done to investigate ways of manipulating a BEC into a non-standard form; however, much of this has been limited to specific simple patterns<sup>4</sup> or highly-periodic patterns<sup>5,6</sup>. We wished to adapt the field of optical holography and beam shaping in order to discover if such schemes would be viable.

---

In order to expand the available options, there is a need to Fourier transform a BEC as Fourier holography offers many advantages over other methods. Observation of the behaviour of modes within a standard BEC trap gave rise to the surprising result of Fourier transforming by waiting. In harmonic traps, some BECs perform Fourier transforms periodically. With this ability, a holographic system for shaping a BEC was feasible and a suitable optical system which could be converted was sought.

The search for methods of shaping light uncovered many possible systems and many algorithms (chapter 6) to calculate the required holograms. The intriguing possibility of shaping light into approximations to arbitrary 3-dimensional shapes was explored in simulation and then performed experimentally (chapter 7).

The expertise acquired during this time led me to become involved in the optimization of algorithms for use in real-time optical tweezers (chapter 8) and later the development of a novel user interface for manipulating optical traps intuitively in real-time

## 1.2 Bose-Einstein condensates

Bose-Einstein condensates (BECs) have become a hot topic since their experimental realisation in 1995<sup>7,8</sup> and over 5000 papers have been published in the field<sup>9</sup>. The condensation of bosons into the same state when cooled low enough was recognised by Einstein in 1925; however, the condensation temperature of  $\sim 100\text{nK}$  was too low for the technology of the time. The use of advanced optical and magnetic trapping techniques resulted in the successful

---

condensation of Rubidium<sup>8</sup> and Sodium<sup>7</sup> which lead to Eric Cornell, Carl Wieman and Wolfgang Ketterle being awarded the Nobel prize for Physics in 2001.

As the atoms or molecules which make up a BEC are cooled, the thermal de Broglie wavelength,  $\lambda = \sqrt{2\pi\hbar^2/k_B m T}$  becomes larger. As the wavelength becomes comparable to the spacing between the atoms or molecules, the behaviour of the ensemble is described by quantum statistics. The density of the condensate can be described by a single wavefunction having an amplitude and a well defined phase. In this way, Bose-Einstein condensates are very similar to monochromatic laser light, where the atoms and molecules play a similar role to the photons within a laser light beam. As we shall see in Chapter 2, the comparison between BECs and laser light can be understood in more detail than this qualitative description.

### 1.3 Lasers and BECs

As mentioned, the behaviour of BECs is very similar to that of light, in fact the free evolution (in time) of the Gross-Pitaveskii Equation (equation 2.8) representing a linear BEC is almost identical to the evolution (in space) of the wave equation of a monochromatic light beam. The atoms within a BEC act in a similar manner to the photons within a light beam with the exception that the atoms can interact with each other. This interaction leads to deviation from the laser light similarity and also adds complexity to their behaviour.

This has led to the investigation of the extent of the similarity and where

---

it breaks down. The addition of any interaction between the atoms automatically breaks the equivalence in the evolutions. However, very ‘light-like’ behaviour can be observed with a modest interaction.

---

# Chapter 2

## Numerical Theory

This chapter describes the background to the numerical simulation tools used to model the dynamics of Bose-Einstein condensates (BECs) and monochromatic laser light. It also compares the  $k$ -space representation of each system and describes their similarity and differences. The required software for simulating the laser light was previously developed<sup>10,11</sup>; however, the software to simulate BECs had to be created specifically for this work.

My role in this portion of the work was to develop the bespoke software required to simulate a BEC in an arbitrary trapping potential. This software was based on the description in section 2.1.1 and simulated using a fourth-order Runge-Kutta method<sup>12,13</sup>. This software was first developed in two and one dimensions using the software development package LabView<sup>14</sup> before being developed for full three dimensional simulations using Visual C++<sup>15</sup>.

### 2.1 Gross-Pitaveskii Equation

In order to compare optics simulations to the related world of Bose-Einstein condensates (BECs), a method of simulating the dynamics of each system is required. It has been shown<sup>16</sup> that for a weakly interacting BEC at a temperature of absolute zero, the Gross-Pitaevskii equation accurately de-

scribes the behaviour of a BEC. See Ref.<sup>13</sup> for a derivation and Ref.<sup>17</sup> for a more detailed derivation which includes higher order interactions which are neglected in the approximation used here.

The free evolution of a non-interacting BEC can be described by the Schrödinger equation and a trap can be approximated by including an additional potential (often harmonic in shape). However non-interacting BECs are rare (although Feshbach resonances can be utilised to tune the non-linearity to any value, including zero) and the most interesting comparisons between optics and BECs will result from interacting condensates. In cold alkali atoms (which are considered in this work) the interactions between atoms are dominated by *s*-wave elastic scattering<sup>18</sup> and if the spacing between the atoms is much smaller than the scattering length, the interaction can be described by a psuedo-potential which depends on the density of the condensate at that point. Using this approximation the dynamics of a BEC represented by  $\tilde{\psi}(\tilde{\mathbf{r}})$ , can be described by the Gross-Pitaevskii equation,

$$i\hbar \frac{\partial \tilde{\psi}(\tilde{\mathbf{r}}, \tilde{t})}{\partial \tilde{t}} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + \tilde{V}(\tilde{\mathbf{r}}, \tilde{t}) + N\tilde{U}_0 |\tilde{\psi}(\tilde{\mathbf{r}}, \tilde{t})|^2 \right] \tilde{\psi}(\tilde{\mathbf{r}}, \tilde{t}), \quad (2.1)$$

where  $m$  is the mass of one of the atoms,  $\tilde{V}(\tilde{\mathbf{r}}, \tilde{t})$  is an external potential,  $N$  is the number of atoms and  $\tilde{U}_0$  is the strength of the psuedo-potential.

Because of the non-linearity, the normalisation of the wavefunction is an important factor as scaling the wavefunction by  $\alpha$  scales the nonlinear term by  $\alpha^2$ . In this work the wavefunction is normalised as

$$\int |\tilde{\psi}(\tilde{\mathbf{r}}, \tilde{t})|^2 d\tilde{\mathbf{r}} = 1. \quad (2.2)$$



In many cases the only external potential is that of an axisymmetric harmonic trap, which is used to confine the condensate. This trapping potential is described in terms of its oscillation frequency  $\omega_i$  by

$$\tilde{V}(\tilde{\mathbf{r}}) = \frac{1}{2}m[\omega_r^2(\tilde{x}^2 + \tilde{y}^2) + \omega_z^2\tilde{z}^2]. \quad (2.3)$$

We can define the trap anisotropy as

$$\lambda = \frac{\omega_z}{\omega_r}. \quad (2.4)$$

In many cases this potential is representative of experimental trapping configurations and for numerical simplicity we often use a spherically symmetric trap with  $\omega_r = \omega_z$ .

### 2.1.1 Dimensionless Formulation

In order to minimise the number of parameters in the simulations and to keep calculated quantities close to unity to avoid numerical rounding errors, we choose the following dimensionless formulation of the Gross-Pitaveskii equation<sup>13,19</sup>:

$$\mathbf{r} = \left( \frac{\hbar}{2m\omega_r} \right)^{\frac{1}{2}} \tilde{\mathbf{r}}m \quad (2.5)$$

$$t = \omega_r \tilde{t}. \quad (2.6)$$

This gives us a trapping potential of

$$V_0(r) = \frac{1}{4}(x^2 + y^2 + \lambda z^2). \quad (2.7)$$

By substituting the scale factors (Equation 2.5 and Equation 2.6) we can obtain the dimensionless, time-dependent Gross-Pitaveskii equation

$$i\frac{\partial\psi(\mathbf{r}, t)}{\partial t} = \left[ -\nabla^2 + V(\mathbf{r}, t) + g|\psi(\mathbf{r}, t)|^2 \right] \psi(\mathbf{r}, t) \quad (2.8)$$

and normalise the wavefunction as

$$\int |\psi(\mathbf{r}, t)|^2 d\mathbf{r} = 1. \quad (2.9)$$

It can now be seen that the strength of the non-linearity is solely dependent on the parameter  $g$ , which in two dimensions is given by

$$g = 4\pi N \frac{a \hbar^2}{d m}. \quad (2.10)$$

$d$  is the thickness of the BEC in the third dimension (the  $z$  direction), which needs to be very small for the BEC to behave 2-dimensionally. This is usually ensured by tight  $z$  confinement by a harmonic trap that satisfies the condition  $\mu < \hbar\omega_z$ , where  $\mu$  is the chemical potential. Two-dimensional condensates have been observed in traps with a high axial anisotropy<sup>20</sup> which results in negligible dynamics in the axial direction.

## 2.2 Plane wave decomposition

When comparing Bose-Einstein condensates to optical systems, a numerical method of propagating a monochromatic laser beam is required. The chosen method was based on the plane wave decomposition<sup>21</sup> of a numerical representation of a field. Using this method each plane wave component can be propagated separately and efficiently and then superposed to give the resulting field. As can be seen, this method requires the principle of linear superposition and thus can only be used to draw comparisons with a linear ( $g = 0$ ) condensate. This method also does not include any form of trapping potential, and so direct comparisons between results from this method and BECs can only be drawn in the linear untrapped case. Despite this small

set of possible comparisons, direct comparison between the two simulations has provided valuable insight into the similarities between the two forms of waves.

### 2.2.1 Plane wave method details

Given a grid of discrete points in a spatial plane  $x,y$  representing a complex light field, its momentum distribution can be calculated using a Fourier transform<sup>21</sup>. Each element of the Fourier transform,  $\Psi(k_x, k_y)$ , represents a plane wave component with the transverse momentum given by  $k_x$  and  $k_y$  and the amplitude and phase of the element gives the weighting and phase offset of that plane wave component.

As a plane wave propagates, its phase advances in proportion to the distance and inversely with the wavelength along the axis of propagation. Thus the phase of each component should advance by  $\Delta z k_z$  where  $\Delta z$  is the distance propagated and  $k_z = \sqrt{(2\pi/\lambda)^2 - k_x^2 - k_y^2}$  is the wave-vector component along the beam axis (not in the direction of the plane wave).

## 2.3 Numerical Details

When performing the simulations, the field is represented on a discrete grid of equidistant complex numbers. The number of points in this grid,  $N$ , is simply  $N = N_x N_y N_z$  where  $N_i$  is the number of points in that direction, while the represented size of the grid is denoted  $R_i$  with the subscript denoting which axis. The grid in each direction varies from  $-R_i/2$  to  $R_i/2$  with each point being separated by  $R_i/(N_i - 1)$ ; this gives rise to no grid element lying at

the origin provided  $N_i$  is even.

The  $k_i$  space grid then varies between  $-\pi N_i/R_i$  and  $\pi(N_i - 1)/R_i$ , with the individual elements of the  $k$  space grid in the  $i$  direction,  $k_{i,j}$  being given by

$$k_{i,j} = \frac{2\pi}{R_i} \left( j - \frac{N_i}{2} \right) \quad (2.11)$$

where  $j$  is between 0 and  $N_i - 1$ .

### 2.3.1 $k$ space differences

The wavevector component in the direction of propagation (which determines the phase shift that component gains on propagation) is calculated as

$$k_z = \sqrt{k^2 - (k_x^2 + k_y^2)}, \quad (2.12)$$

where  $k = 2\pi/\lambda$ .

This differs from the BEC evolution (with time) in which the phase shift of the momentum space component is proportional to the energy of that component, which (in two dimensions) is

$$E_t = -(k_x^2 + k_y^2). \quad (2.13)$$

It can be seen that the BEC evolves with a parabolic  $E_t$  while light propagates with a hemispherical  $k_z$ . If  $k \gg \max(k_x)$  and  $k \gg \max(k_y)$ ,  $k_z$  for light approximates a parabola within the represented  $k$  space. When directly comparing light simulations to free propagation of BECs, the curvature of  $k_z$  needs to be the same. This can be achieved by using  $\lambda = 2\pi$  (for the same represented size,  $R_i$ ). In most BEC simulations considered here, the size of the dimensionless represented box,  $R_i$ , is of the order of 10. This combined

with the limitation of  $\lambda = 2\pi$ , leads to the represented light beam being of a similar size to its wavelength. The behaviour of such a small beam can be quite different to the everyday idea of light (for example, deviation of the focus of a light beam when tightly focused<sup>22</sup>).

## 2.4 Summary

It can be seen that the descriptions detailing the dynamics of a light beam can in some situations be almost identical to that of a BEC. In this way, the study of BECs is an extension and generalisation of light. However the everyday behaviour of light is not to be used as the standard. The situations when light is most similar to BECs, is restricted to the case where the width of the light beam is only a few wavelengths across. Such extremes can lead to unusual behaviour in light.

## Part I

# Some optical ideas applied to Bose-Einstein condensates

## Chapter 3

# Transverse laser modes in Bose-Einstein condensates

This chapter describes the initial simulations performed to compare Bose-Einstein condensates (BECs) to laser light by mimicking a laser resonator for BECs. A large portion of this chapter is taken from the previously published paper<sup>23</sup>. Section 3.1 describes the motivation behind such a project, leading into the background of laser cavities in Section 3.2 and a discussion of the eigenmodes of such systems in Section 3.5. The original portion of this work begins in Section 3.3 where the analogy of a laser cavity is conceived and the creation of Hermite-Gaussian-like modes are seen in Section 3.6. In order to demonstrate that the formed modes are similar to Hermite-Gaussian modes in more than just shape, we simulate the conversion of our Hermite-Gaussian-like modes into Laguerre Gaussian modes in Sections 3.7 using the familiar cylindrical lens mode converter<sup>24,25</sup>.

My role within this project was to design and implement the bespoke software required for the simulations and to analyse and present the results. I was also involved in guiding the research through discussions with the co-authors in which I played a key part. I was also responsible for drafting of the paper linked to this research and was involved in every stage including

submission and final publication.

## 3.1 Introduction

When considering how to shape a laser beam, often the process which creates the beam is overlooked and more complicated methods of beam shaping are employed. The majority of laser light is generated in a resonant feedback cavity. The design of such a cavity is often chosen to produce a particular longitudinal mode (a certain frequency, collection of frequencies or shape of pulse). The design of the cavity also affects the transverse shape of the emerging beam. This is often overlooked since the output of most ordinary laser cavities is Gaussian in cross-section, which for many applications is the desired profile.

As we have seen, BECs can in many ways be seen as the matter-wave equivalents to laser light, and so perhaps the best place to begin an investigation into their similarities would be the cavity which helps create this similarity. By simulating the equivalent of the individual elements which make up a cavity for a BEC, any similarities in the transverse mode should be noticeable. In the non-interacting case (the most similar to optics) the behaviour should be almost identical; however, the interesting cases occur when the atomic interactions are increased and the behaviour deviates from the linear optical case.



## 3.2 Laser Cavities

When considering creating laser light, it is difficult not to consider an optical cavity. In order to increase the output power of lasers, the light emerging from the gain medium is often reflected back into the medium using a mirror on both sides of the medium. This results in the light passing through the gain medium many times in an attempt to maximise the stimulated emission. Because of the coherent nature of the light within the cavity, longitudinal modes can form when the reflected light interferes with the un-reflected light; these longitudinal modes, and therefore the arrangement of the mirrors, is an important design characteristic of most lasers. The mirrors also play an important role in the transverse mode of a laser beam; the size, shape and imperfections of each mirror can introduce local loss to the transverse profile of the beam, which after several round trips can have a large influence on the profile throughout the cavity and on transmission from the system.

Figure 3.1a sketches one round trip of a light pulse (or alternatively a “slice” of a continuous-wave (cw) light beam) through a standard laser cavity. As can be seen, after one round trip, the light is back in the same plane in which it started and travelling in the same direction due to the two facing mirrors. By using curved mirrors, the light is focused upon reflection which minimises the diffraction losses due to light leaving the cavity.

When considering the intensity structure of the light as it propagates through the cavity, it is often helpful to decompose the structure into a superposition of the eigenmodes of the cavity. Such modes are unchanged after a complete round trip (except for uniform phase and or intensity modifications) and often the output of a laser can be described by a handful of such

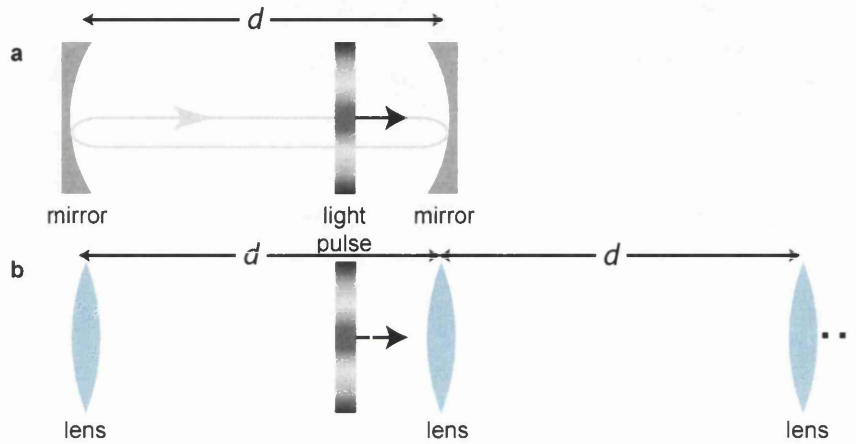


Figure 3.1: Laser resonator (a) and its unfolded lens-series equivalent (b).

A light beam passing through the lens-series equivalent is periodically being focused, just like light in the resonator is being focused every time it is reflected off one of the resonator's mirrors. A series of "lens pulses" – light pulses which exhibit a specific intensity profile and have a frequency far from absorption resonances of the condensate – can repeatedly focus a BEC in a similar way. For laser modes to form, localised absorption needs to occur in the resonator; this is not shown here.

eigenmodes. By introducing loss to specific undesired modes (for example by introducing a small aperture to suppress large modes), it is possible to change the dominant mode of the laser.

In order to simulate the transverse cavity dynamics, a cavity is represented in its unfolded form (figure 3.1b), where the two curved mirrors become a chain of pairs of lenses, separated by the length of the cavity. If the focal length of the lenses corresponds to the focal length of the mirrors and their separations are the same, the eigenmodes of the cavity will correspond to the eigenmodes of an infinitely long chain of lens pairs. This also holds for cases where local loss is introduced, assuming the loss is represented correctly in the unfolded cavity. In the majority of cases, the eigenmodes are formed within the first few round trips of the cavity and thus only propagation through the first few sets of lens pairs needs to be calculated to find the eignemode. This allows easy calculation of the beam resulting from  $N$  round trips, by passing the beam through  $2N$  lenses at a fixed separation. Localised loss then becomes  $2N$  apertures periodically placed through the system.

### 3.3 Bose-Einstein Condensate Cavities

As has already been discussed, Bose-Einstein Condensates are in many ways the matter-wave equivalent of laser light and as such should behave in a similar manner if placed inside a cavity. In particular it should be possible to excite higher-order modes of the cavity. In order to create an analogous system as well as to keep simulation times reasonable, we consider a

2-dimensional “pancake” BEC (formed experimentally when the trapping in the  $z$  direction is much stronger than in the  $x, y$  directions, forcing the BEC into a thin pseudo-2-dimensional state). The time evolution can be described by the 2D time-independent Gross-Pitaevskii equation

$$i\hbar \frac{\partial \psi(\mathbf{r}, t)}{\partial t} = \left[ -\frac{\hbar^2}{2m} \nabla^2 + \frac{1}{2} m \omega^2 r^2 + g |\psi(\mathbf{r}, t)|^2 \right] \psi(\mathbf{r}, t), \quad (3.1)$$

where  $m$  is the mass of each atom,  $\omega$  is the trap frequency and  $g$  is the nonlinear coefficient, which is given by

$$g = 4\pi N \frac{a \hbar^2}{d m}. \quad (3.2)$$

$d$  is the thickness of the BEC in the third dimension (the  $z$  direction), which needs to be very small for the BEC to behave 2-dimensionally. This is usually ensured by tight  $z$  confinement by a harmonic trap that satisfies the condition  $\mu < \hbar\omega_z$ , where  $\mu$  is the chemical potential. Note that  $g$  depends not only on  $m$  and  $a$ , the s-wave scattering length, but also on the number of atoms,  $N$ , in the BEC. As the number of particles is contained in the parameter  $g$ , the density is normalised according to

$$\int |\psi(\mathbf{r}, t)|^2 d\mathbf{r} = 1. \quad (3.3)$$

### 3.4 BEC focusing

During the investigation into BEC modes, the issue of a BEC’s reaction to focusing was raised. In order to focus a BEC into a small spot after time  $T_f$ , a simple optics analogy would require that a quadratic phase should be imprinted on the BEC of the form

$$\psi'(r, t = 0) = \psi(r, t = 0) \exp(ir^2/2T_f). \quad (3.4)$$

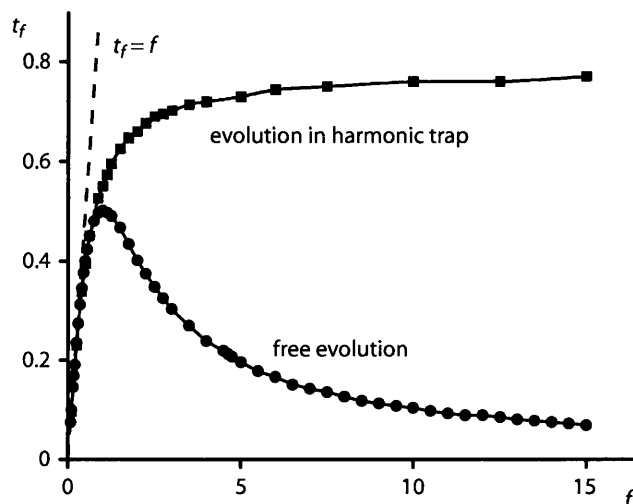


Figure 3.2: Focusing of a BEC by a lens pulse. A BEC in the ground state was focused using a light pulse of focal time  $f$ , which phase-shifts the BEC locally by  $r^2/(2f)$ . The graph shows the actual focusing time,  $t_f$ , which is the time it takes the BEC to become phase-flat after interaction with the lens pulse, as a function of  $f$ . Analogy with ray optics suggests that the BEC comes to a focus after a time  $t_f = f$ ; however, to our surprise we find that this is only the case for small values of  $f$ . In the range  $0 \leq g \leq 100$ , over which our simulations were performed, these results are virtually independent of  $g$ .

From geometric optics, the BEC should then come to a focus, and become phase flat, at  $t = T_f$ . As figure 3.2 shows the time after which a pancake condensate comes to a focus for both harmonically trapped and untrapped cases is only  $T_f$  for small values. For the untrapped case (the most similar to optics) this is a recognised optical effect which is only significant if  $\lambda f_{\text{opt}}/\pi\omega_0^2 > 1$  ( $\lambda$  is the optical wavelength,  $f_{\text{opt}}$  is the focal length as expected from geometrical optics, and  $\omega_0$  is the width of the beam at the focus), which is rarely the case in optics<sup>22</sup>. However in our BEC simulations this effect was significant when  $T_f > 1$ , which is often the case. For the harmonically trapped case, the trapping potential acts to focus the BEC which becomes the dominant focussing effect at longer local times. When the BEC is not trapped and is free to expand, the phase shift acts against the expansion. When  $T_F < 1$  the phase shift is strong enough to trap the condensate and it focuses in a similar way to the trapped condensate. However if  $T_F > 1$ , the expansion of the condensate is dominant. As  $T_F$  increases, the strength of the focusing pulse becomes less and the condensate requires less time to become phase flat.

Further investigation of BEC dynamics revealed that in 2 dimensions this effect does not depend on the non-linearity. When a condensate is held in an harmonic trap and a focusing lens pulse is applied, the time to focus no longer has a maximum; however at longer focal times (very weak focusing pulses), the natural focusing of the trap becomes the dominant contribution and the condensate comes to a focus after half a breathing-mode period (see Section 4.4.2 for details of the breathing modes of a BEC).

### 3.5 Eigenmode selection in laser resonators

One round trip is much more complicated than just some changes in propagation direction<sup>26</sup>: as the mirrors are curved, each mirror reflection also focuses the light. In addition, localised loss occurs in one or several planes, for example through apertures or specks of dust on mirrors, and while travelling between the mirrors the light beam diffracts. Some light beams – the resonator’s eigenmodes – are unchanged (apart from a uniform change in intensity or phase) after one round trip through a resonator. In the case of geometrically stable resonators, which we consider here, the eigenmodes can be divided into families (like the HG modes and the LG modes), whereby each family forms a complete orthogonal set, i.e. every light beam can, for example, be described as a superposition of HG modes. As already mentioned, the HG and LG modes can take on very recognisable shapes and have interesting features like high-charge vortices (see figure 3.3).

Experimentally, a pure resonator eigenmode is created in a laser by making the round-trip loss of all other eigenmodes higher than that of the desired eigenmode. This can be done, for example, by inserting absorbing cross-wires into the resonator at positions where the desired eigenmode is darker than the competing eigenmodes. For example, an absorbing wire in the central horizontal dark line of the  $\text{HG}_{2,1}$  mode shown in figure 3.3 would not significantly increase that eigenmode’s round-trip loss, but it would increase that of the  $\text{HG}_{2,0}$  mode. The fraction of the power in the undesired eigenmodes then falls off exponentially with round trip number, after a few round trips leaving only the pure lowest-loss eigenmode.

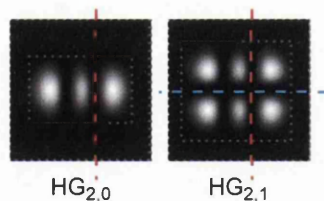


Figure 3.3: Hermite-Gaussian (HG) modes and possible configurations of aperture and absorbing wires which may produce them. The vertical wire (red) discriminates against non  $m = 2$  modes, however this does not restrict the  $n$  index. The addition of an aperture (dotted lines), or another wire allows individual modes to be selected.

### 3.6 Creation of Hermite-Gaussian Modes

One family of eigenmodes is the Hermite-Gaussian (HG) modes which display Cartesian symmetry and are described by two indices which represent the number of nodes in the field. HG modes form an orthogonal set and so any light beam can be described by a superposition of HG modes. There are several ways to preferentially select one HG mode. As the number of nodal lines increases, the extent of the field increases and by introducing a rectangular aperture, the higher-order modes will have much higher losses than lower-order modes. To select between lower modes, an absorbing wire could be placed along the nodal line of the desired mode, which then introduces losses to modes with displaced nodal lines. HG modes have a  $\pi$  phase change across the nodal lines and by introducing such phase jumps externally, the mode with matching phase profile will have a lower round trip loss. Often



these methods need to be combined to select the desired mode (for example there are an infinite number of modes with nodal lines at any given point, but an aperture slightly larger than the desired mode will increase losses of the higher-order modes).

### 3.6.1 Round trip losses

In the linear optical case the round-trip loss decreases with each round trip until the eigenmode of the cavity has been selected at which point the round trip loss becomes negligible. In the non-linear BEC case, during each round trip atoms are lost and the self-interaction potential is reduced. This results in the condensate 'seeing' a different potential with each round trip and no stationary eigenmode can form until the self-interaction becomes zero ( $g = 0$ ). For high non-linearities, oscillations in the pattern formed within the cavity were observed which are due to this effect of a changing self-interaction pattern. Figure 3.4 shows an example of the fraction of the condensate left after several round trips and the density pattern at the centre of the cavity at each round trip.

### 3.6.2 Simulation Results

By restricting the condensate to a box region with a thin absorbing line, the resulting modes resemble the optical Hermite-Gaussian modes of a laser resonator (figure 3.5). For a non-linear condensate the interaction of the atoms results in the pattern deforming (figure 3.6). In the case of repulsive interactions (positive  $g$  values) the distribution of atoms becomes wider and more diffuse, as shown in figure 3.7. It can also be seen that as the larger

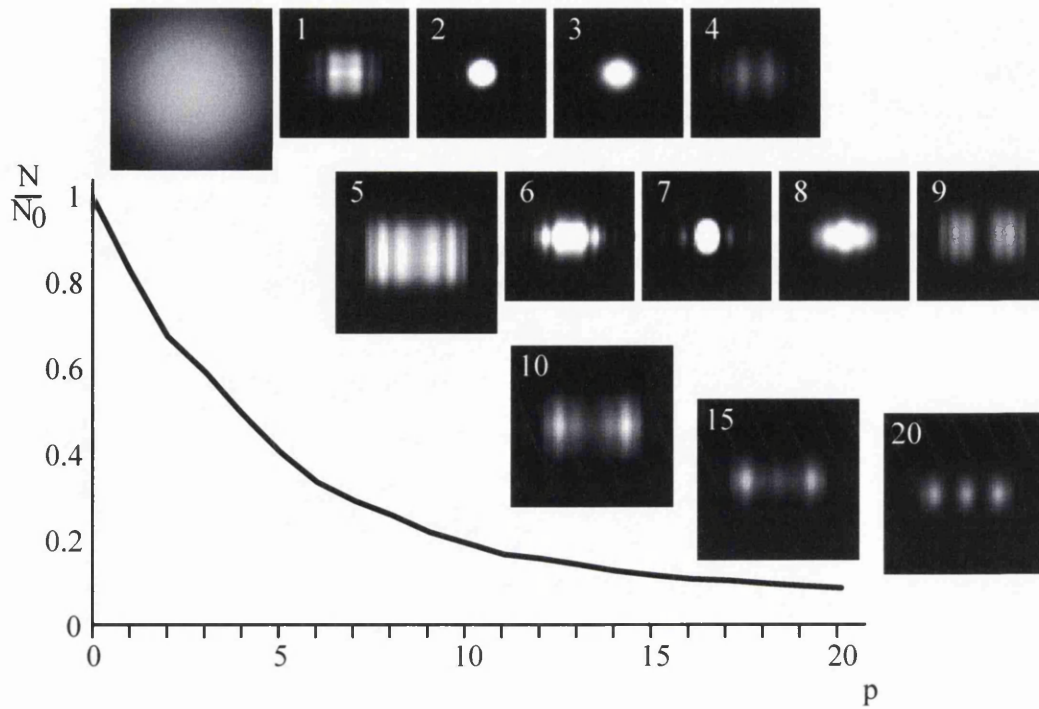


Figure 3.4: BEC after  $p$  round trips of a light-pulse resonator (the value of  $p$  for each frame is shown in the top-left corner), starting with the harmonic trap ground state ( $p = 0$ ), calculated for  $g = 100$ . The pulses of focal time  $f = 0.5$  are separated by  $L = 0.2$  (in dimensionless time units); the density patterns shown here are calculated a time  $L/2$  after the actual pulse. The curve at the bottom shows the fraction of the initial number of particles,  $N_0$ , that remains in the BEC after  $p$  periods of the light-pulse resonator. To demonstrate the loss, the density is not normalised in the frames in this figure; the full scale of grays represents in the frames for  $p = 0$  to 4 densities between 0 and 0.2, in the remaining frames between 0 and 0.05.

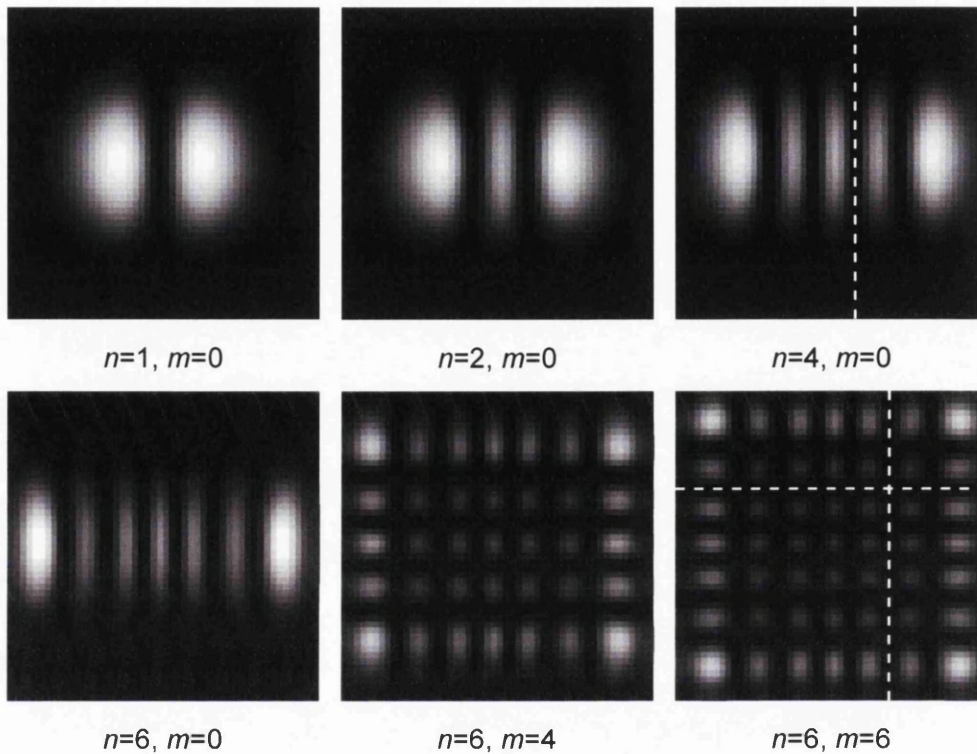


Figure 3.5: Density patterns of some eigenmodes of the standard light-pulse resonator with a non-linearity of  $g = 30$ . By varying the cross-hair position, different patterns – characterised here by their number of nodal lines in the horizontal and vertical direction,  $n$  and  $m$ , respectively – emerge, which strongly (but not exactly) resemble optical Hermite-Gaussian modes. These pictures show the BEC half-way between two lens pulses. A possible position of the cross-hair (white dashed lines) is shown in the two rightmost frames. The pictures show only the central  $6.25 \times 6.25$  units of the modelled area.

parts of the condensate (in figure 3.6, the two outermost parts) expand, the repulsion between the atoms result in the smaller lobes to be distorted from the rectangular symmetry. In the  $\text{HG}_{2,0}$  case this results in the centre lobe being 'pinched' and bulging outwards, for example. These two effects cause the resulting modes to become less Hermite-Gaussian in nature.

Figure 3.5 shows some of the stable patterns that emerge for a modest non-linearity ( $g = 30$ , typically corresponding to approximately  $10^3$  atoms of  $^{87}\text{Rb}$  in a  $20\pi$  Hz trap) by varying the localised absorption pattern – a 100% absorptive cross-hair of 1 pixel width placed in crossing node lines (see figure 3.5). Like all other pictures in this section (unless otherwise stated), this figure shows only the central  $6.25 \times 6.25$  units of the modelled area for clarity, and the grey values represent linear densities, ranging from 0 (black) to 0.2 (white). The patterns look very similar to Hermite-Gaussian modes, more so for small values of the non-linear parameter  $g$ , less so for large values (figure 3.6). In all cases, the eigenmodes have the tell-tale  $\pi$  phase difference between adjacent peaks, just like the optical HG modes.

The expansion of the higher non-linear condensates can be seen in Figure 3.7 which shows the cross-sectional line through  $y = 0$ . The distribution along the  $y$ -direction varies from a Gaussian in the linear case to a flatter, more diffuse distribution in the non-linear case. From the  $y = 0$  cross section, which cuts through all three lobes, the right-most node where the absorbing cross-wire lies is always a node of the resulting mode. However the second node on the left is further from the center in the linear case. This implies that the beam-waist size of the Hermite-Gaussian mode is larger in the linear case and has been shifted (as evident from the distribution close to  $x = 0$ ).

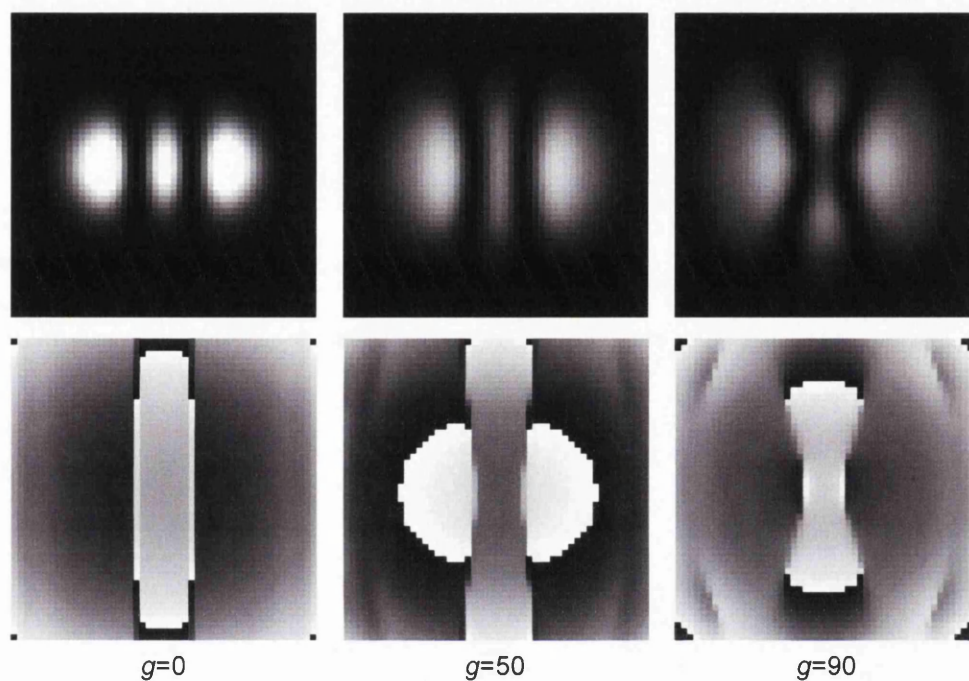


Figure 3.6: Density (top) and corresponding phase patterns (below) for various values of the non-linear coefficient,  $g$ . In all cases, the shape of the absorption region is a narrow vertical straight line at  $x = 0.5$  which corresponds to the null on the right of the centre lobe. Calculated using a resonator of length,  $L = 0.2$ , and a lens-pulse focal time of  $f = 0.5$ .

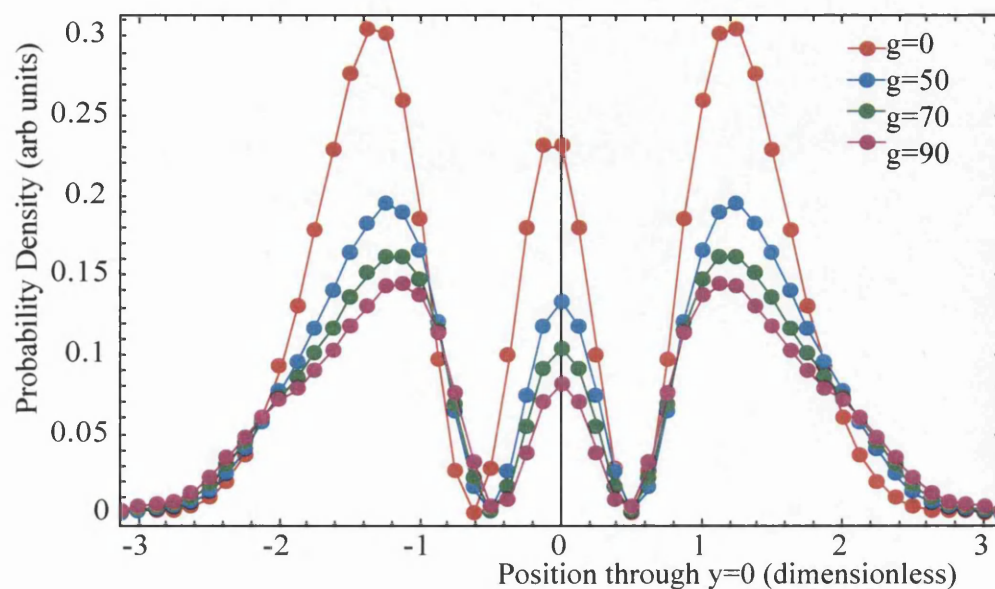


Figure 3.7: Cross-sectional profile through  $y=0$  of a HG2,0 mode for non-linear coefficient values,  $g = 0, 50, 70$  and  $90$ . Absorption is performed along  $x = 0.5$  which can be seen as the null.

This behaviour is consistent with non-linear modes having a larger physical extent for a given beam waist and therefore require a smaller beam waist to fit inside the same absorbing boundary.

When trying to form odd-numbered modes, the resulting phase pattern is anti-symmetric with a  $\pi$  phase dislocation across the origin. This results in problems in the simulations where the resonator is symmetrical. The resulting simulations would show an even-numbered mode forming in which the loss per round trip would converge. However after typically another few hundred iterations, the pattern would re-distribute itself and form the required odd-numbered mode with a lower round-trip loss. Figure 3.8 shows

the round trip loss during one such example. This behaviour is believed to arise from the build-up of random numerical errors in the simulation which seed the asymmetry. In order to improve the convergence of odd-numbered modes, additional random noise was added to the system after each round trip, which, as predicted, reduced the formation of a mis-matched even-numbered mode.

All the results shown so far were calculated for our standard resonator with a time between pulses – corresponding to the resonator length – of  $L = 0.2$ , and a lens-pulse focal time of  $f = 0.5$ . Figure 3.9 shows  $n = 1$ ,  $m = 0$  eigenmodes from different light-pulse resonators. In the optical analogy, altering resonator parameters while keeping the resonator geometrically stable (that is, satisfying  $L < 4f$ ) results in eigenmodes of a different size; the analogous effect in BEC light-pulse resonators can be seen in figure 3.9.

### 3.7 Conversion into Laguerre-Gaussian-like modes

To examine the similarities between the eigenmodes of light-pulse resonators and optical HG modes beyond the level of appearance, we investigate here the analogy of a conversion of optical HG modes, which have a rectangular symmetry, into circularly symmetric Laguerre-Gaussian (LG) modes (see section 5.2.2 for more details of Laguerre-Gaussian modes) under cylindrical focusing<sup>25</sup>. This conversion is due to a fairly subtle effect: the Gouy phase shift, which occurs when HG or LG modes pass through a focus. Like HG modes, LG modes are structurally stable. They are interesting because they

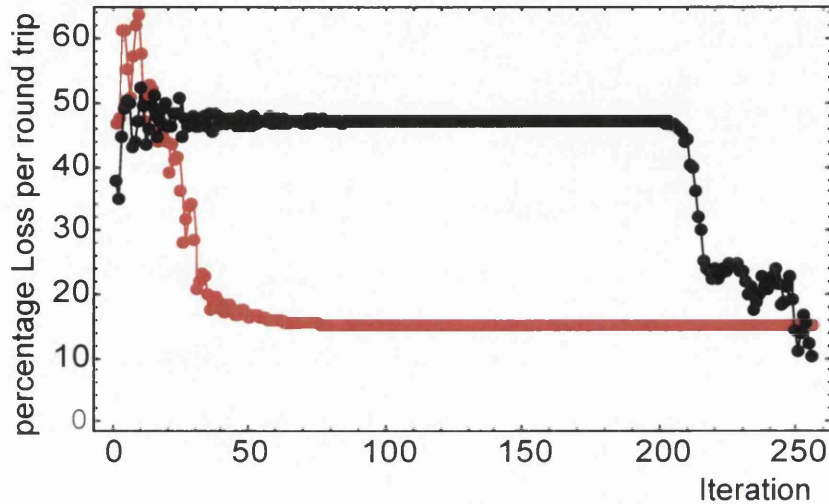


Figure 3.8: Percentage Round trip loss for HG4,0 (red) and HG5,0 (black) showing the temporary convergence of the odd numbered modes. Within the HG5,0 resonator a HG6,0 mode forms, but due to the mismatch between the mode and the resonator, experiences high loss. Once random numerical errors build up (aided by the re-normalisation of the wave-function), the symmetry of the system is broken and the HG5,0 mode can form. Because of the symmetry breaking requirement, the convergence time of a resonator for odd-numbered modes is much longer than a similar resonator for even-numbered modes



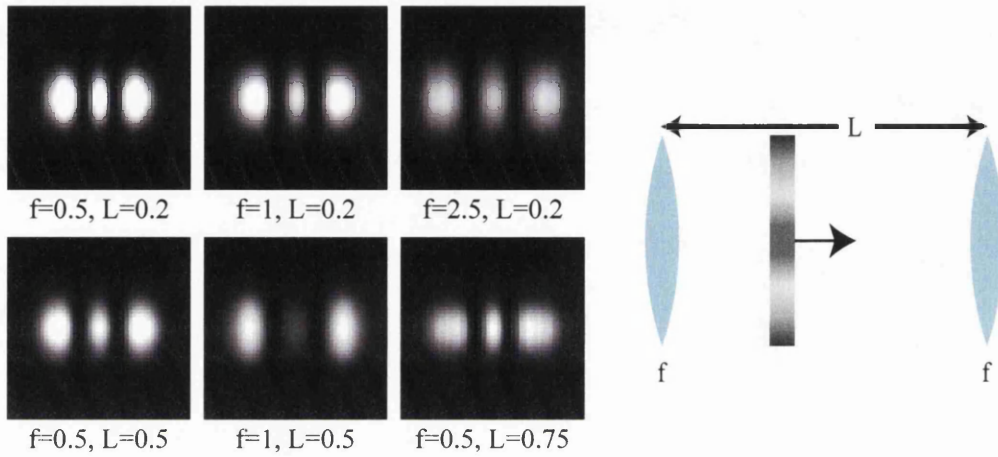


Figure 3.9: Density patterns for the output of different configurations of resonator, calculated for  $g = 0$ .  $f$  represents the focal time of the lens pulse,  $L$  the time between subsequent pulses (both in dimensionless units). For the particular choice of loss-pulse shape, the loss per period of the light-pulse resonator for these eigenmodes is, from top left to bottom right, 1.8%, 1.3%, 5.9%, 4.4%, 10.2%, and 9.7%.

can contain higher-charge vortices at their center.

In the optical case, the cylindrical focusing that converts HG modes into LG modes is usually performed with a pair of cylindrical lenses. By analogy, we use here a pair of cylindrical-lens pulses, i.e. far off-resonant laser pulses with an intensity profile that varies quadratically in one direction and which is constant in the perpendicular direction. Figure 3.10 shows some results that were calculated for  $g = 0$  and  $g = 50$ . The density patterns show strong signs of the ring structure characteristic of LG modes ( $p$  bright rings and a central bright spot if  $l = 0$ ,  $p + 1$  bright rings if  $l \neq 0$ ), although this ring structure is not perfect; instead, most of the density is contained in a distorted ring, and instead of containing a charge- $l$  vortex at the center, the wave function contains  $l$  vortices of charge 1. The resulting BEC is also not structurally stable. The conversion is therefore not perfect, but the emergence of vortices with a combined charge of  $l$  and (to a lesser degree) the conversion of the intensity structure into a distorted ring are indications that it is happening at least in part. In other simulations (not shown) we confirmed that this is also the case for different choices of  $g$  (including  $g = 0$ ), and that, in the appropriate cases, the shape of the BEC also shows signs of the multiple concentric rings characteristic of LG modes with a radial mode index  $p > 0$ .

In the optical case, the focal lengths of the cylindrical lenses and their positions need to be carefully matched to the parameters of the incoming HG mode to achieve good conversion into a LG mode. We tried to similarly optimise the parameters of the cylindrical-lens pulses. However, the similarity to a pure LG mode of the resulting density was always limited, which is not

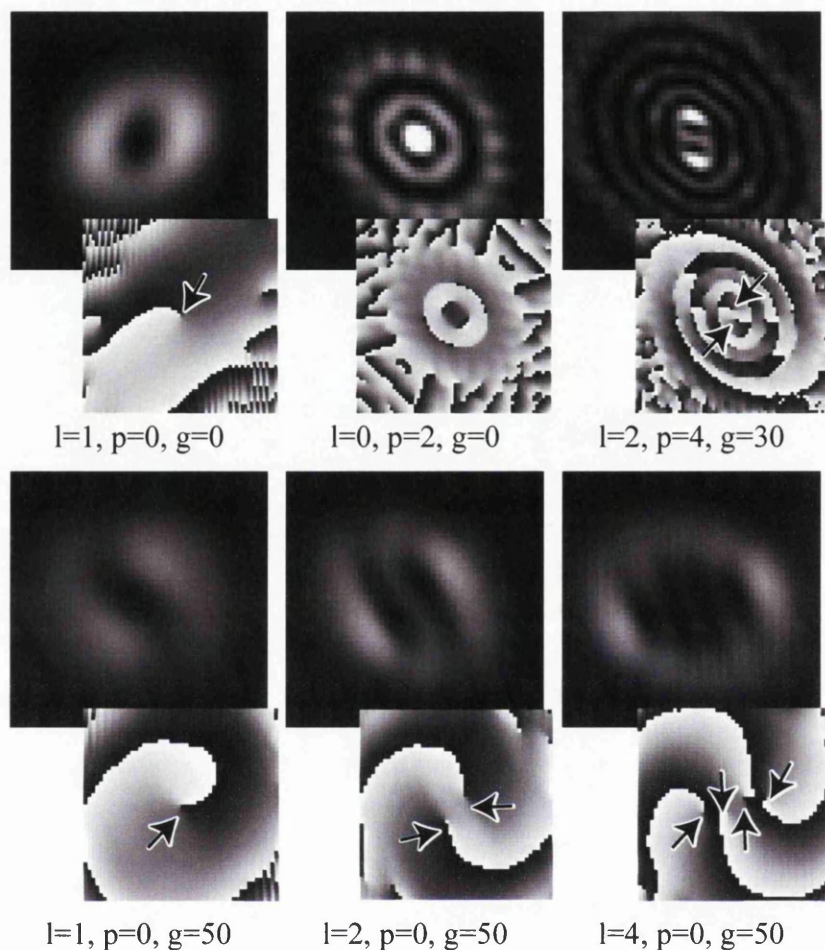


Figure 3.10: Modelled density (1st and 3rd picture row) and phase (2nd and 4th picture row; grey-level representation) immediately after a BEC initially in an eigenmode with  $n$  horizontal and  $m$  vertical nodal lines has interacted with a pair of cylindrical-lens pulses. From top left to bottom right, the parameters of the initial eigenmode and lens pair ( $f$ : focal time of each lens;  $s$ : separation between lenses; both were optimised in each case to give visually the best ring structure) are as follows:  $n = 1, m = 0, f = 0.46, s = 0.35$ ;  $n = 2, m = 2, f = 0.25, s = 0.18$ ;  $n = 6, m = 4, f = 0.25, s = 0.18$ ;  $n = 1, m = 0, f = 0.23, s = 0.35$ ;  $n = 2, m = 0, f = 0.23, s = 0.3$ ;  $n = 4, m = 0, f = 0.1, s = 0.11$ .

surprising as the BEC was initially not in a pure HG mode and as the evolution of the BEC is different from that of light. Finally, optimisation of the pulse parameters was not made easier by the complications in the focusing characteristics of a BEC mentioned earlier (figure 3.2).

## Chapter 4

# Fourier Transforming a Bose-Einstein Condensate

This chapter describes the numerical simulations regarding a Fourier transform within Bose-Einstein condensates (BECs). The first section describes the Fourier transform and some simple Fourier components which can aid in understanding the behavior of the Fourier transform. Section 4.3 describes a generalisation of the Fourier transform which can be adopted when discussing BECs in a harmonic trap, while Section 4.4 describes the various setups which may result in a Fourier transform in a linear system and Section 4.5 presents some of the results of the simulations for both linear and non-linear condensates.

My role in this work was to develop the bespoke software, to carry out the simulations, prepare and then analyse the data as well as helping define the direction of the work. I was also involved in the discussions which led to the work and in the writing and drafting of the paper which describes much of this work<sup>27</sup>. I have been lucky enough to present this work on several occasions during my PhD.

## 4.1 Introduction

When trying to shape a light beam into a desired pattern, loss of light is almost always an important consideration. For example, trying to create a small point of light from a large-diameter beam by passing it through an intensity mask with a pin hole will certainly give rise to a small spot of light behind the pin hole, but will also result in large portions of the light being lost due to absorption around the pin hole. One solution to this problem would be to focus the large beam using a converging lens and place the focus of the beam at the target plane. Such a system would transfer the majority of the light into the small spot, unlike the aperture case. This idea of using a lens to focus the light to a small spot forms the basis of the optical Fourier transform, and if the elements are arranged as shown in Figure 4.1, this results in the back focal plane corresponding to the Fourier transform of the front focal plane.

The higher power efficiency available with Fourier based shaping methods has resulted in a large literature dedicated to shaping a laser beam in the Fourier plane of an optical element and the calculation of the properties of that element to produce the required patterns. With the proven success and the wealth of knowledge available on Fourier holography, adapting it to BECs would bridge the gap and allow the BEC community to draw from the established holography techniques.

The first problem in trying to recreate Fourier holography is the ability to Fourier transform a BEC non-destructively. Many early BEC experiments have in effect Fourier transformed a BEC by turning off the trapping fields, allowing the condensate to expand, and finally imaging the condensate to gain

information about its momentum distribution, that is the modulus squared of its Fourier transform. This is analogous to propagating a laser beam into the far field, but as the light beam is (in principle) infinitely large in the far field, this is destructive and irrecoverable. For the same reasons switching off the trap of a BEC is also irrecoverable. By considering first the analogy of lens-based Fourier transforms and analogies with trapped BECs, we propose a method of Fourier transforming a BEC and investigate how the ability to Fourier transform a condensate changes as the atoms within the condensate begin to interact with each other.

## 4.2 The Fourier Transform

The Fourier transform has become a useful tool in many areas of research. Perhaps it is best known for its use in the time-frequency domain, where Fourier-transforming a time-varying signal results in the frequency distribution of that signal. In many applications where knowledge of a periodic signal is required, the Fourier transform forms an indispensable part of the analysis.

The Fourier transform also has an important role in optics: the Fourier transform of the field of a light beam gives the momentum-distribution and its structure represents the beam when propagated to infinity (the far field). The transform can be performed experimentally using a  $2f$  optical lens setup (see Figure 4.1), which can utilise some of the practical properties of the Fourier transform within an experiment without having to propagate the beam for large distances.

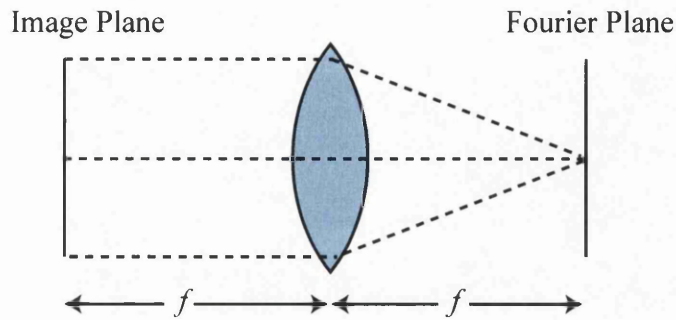


Figure 4.1: Optical  $2f$  system to Fourier transform the field in a plane of a light beam. The initial plane is located the focal length  $f$  in front of a lens, which causes the field in the plane  $f$  behind the lens is the Fourier transform of the initial plane.

### 4.2.1 Simple examples

The Fourier transform is a linear operation, as such in many applications it is easier to decompose a field into easier to manipulate components, Fourier transform those and then recombine the results. Here we look at the behavior of uniform (and infinite) plane waves and delta functions under a Fourier transform. As shown in figure 4.2, we consider plane waves to be uniform in amplitude and have an infinite extent. Plane waves do not change under propagation. As plane waves they have a uniform phase-gradient across the beam. Because of the uniform phase-gradient, the entire wave has the same transverse momentum and when considering the momentum distribution of the beam, there is only one component present. This gives rise to a sharp peak in Fourier space whose position is proportional to the sine of the angle at which the beam is travelling.



The nature of the Fourier transform means that the reverse also holds and the Fourier transform of an offset delta function is a plane wave with a uniform phase-gradient.

### 4.2.2 Gaussian transforms

Gaussian functions are a special case within Fourier theory: they are one of a handful of functions which do not change structure after Fourier transforming. The width of the resulting Gaussian is inversely proportional to that of the initial Gaussian and – in a similar manner to the plane-wave case – a uniform phase-gradient results in an offset of the resulting Gaussian. This nature of Gaussian functions makes them a useful set of functions to examine the behavior of the Fourier transform. In the limit of an infinitely narrow Gaussian, its Fourier transform is an infinitely wide Gaussian and we return to the previous examples of delta functions and plane waves.

## 4.3 Fractional Fourier Transform

One interesting property of Fourier Transforms is related to multiple applications of the transform. When the Fourier transform is applied twice to a function  $f(x)$ , although the structure of the function is the same, the axis is inverted, so the result is  $f(-x)$ . This can be regarded as a rotation by  $\pi$ . In order to retrieve the original function another two Fourier transforms are required (introducing a further rotation by  $\pi$ ). This allows us to understand the Fourier transform as a rotation of  $\pi/2$  in a space with  $x$  and  $k$  axes<sup>28</sup>; however this rotation is not limited to integer multiples of  $\pi/2$ . Gen-

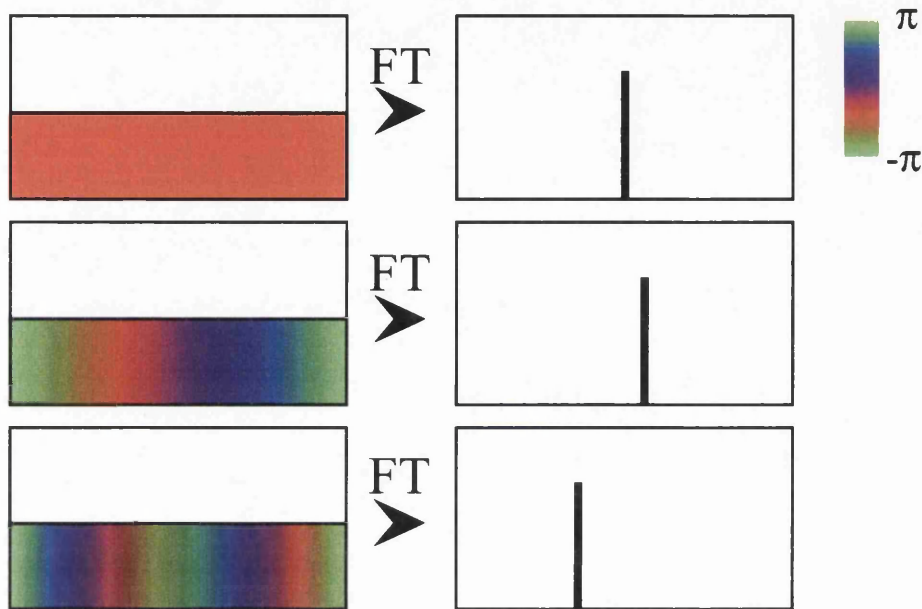


Figure 4.2: Examples of Fourier transforming a plane wave. In all examples the wave has a uniform amplitude and a uniform phase gradient. The phase of the wave at any point is represented by the colour. Top : A plane wave traveling perpendicular to the plane of interest Fourier-transforms to a centred delta function. Middle : If the plane wave is traveling at some angle toward the right, the delta function in the Fourier transform is displaced to the right. Bottom : The offset of the delta function is proportional to the angle at which the wave is traveling, and can also be negative.

eralisation to arbitrary rotation angles  $\alpha\pi/2$  leads to the fractional Fourier transform<sup>28</sup>,

$$\mathcal{F}_\alpha(f)(k) = \sqrt{\frac{1 - i \cot(\alpha)}{2\pi}} e^{i \cot(\alpha)k^2/2} \int_{-\infty}^{\infty} e^{-i \csc(\alpha)\omega x + i \cot(\alpha)x^2/2} f(x) dx \quad (4.1)$$

By substituting  $\alpha = 1$  the original Fourier transform is obtained, it has also been shown<sup>29</sup> that in the limit of  $\alpha \rightarrow 0$ ,  $\mathcal{F}(f) \rightarrow f$ .

A fractional Fourier transform does not simply correspond to the planes in the fractional positions between a Fourier lens and its focal plane. Figure 4.3 shows different methods of interpreting the fractional Fourier transform in optical systems. It can be seen that in order to generate a fractional Fourier transform with lenses, the system must consist of a series of lenses<sup>30</sup>. For example, to generate a  $\alpha = 0.5$  transform, two lenses must be equally spaced between the two planes and their power can be calculated by knowing that the combined focal length of the lens system has to be the distance between the initial plane and the  $\alpha = 0.5$  fractional Fourier plane.

## 4.4 Fourier Transforms in Bose-Einstein Condensates

### 4.4.1 Fourier Transforming with lens pulses

As was shown in Chapter 3, a BEC can be made to focus by applying a far off-resonant light pulse with a quadratically varying intensity, and although the relationship between the strength of the quadratic intensity of the pulse and

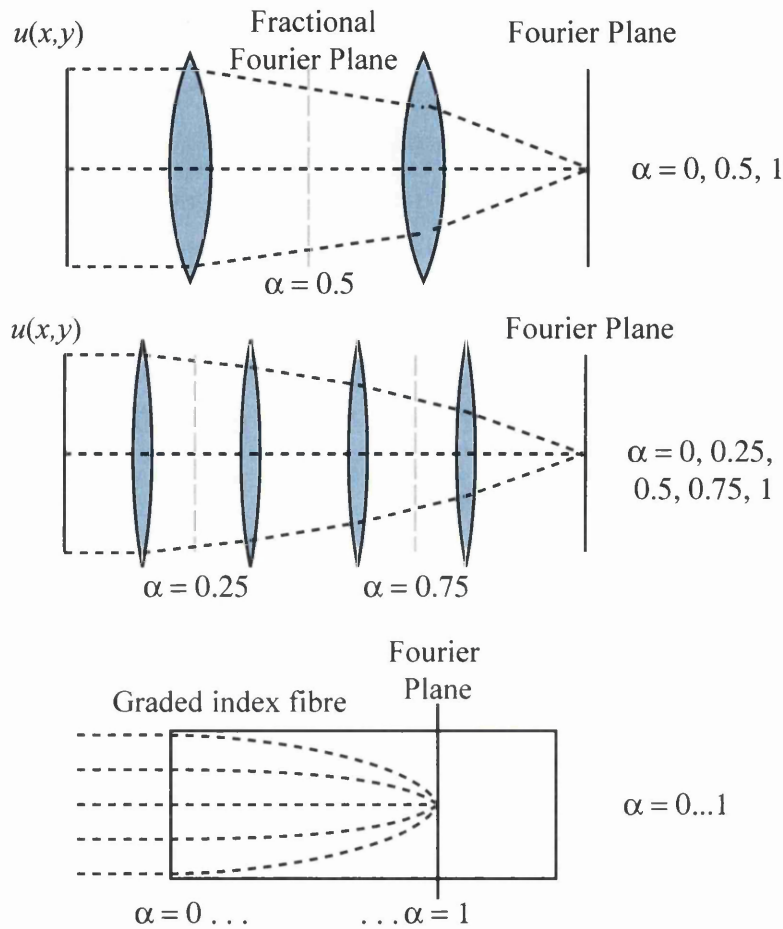


Figure 4.3: Optical systems to produce the Fractional Fourier transform of a light field  $u(x, y)$ . In the most straightforward case, in order to obtain a fractional Fourier transform of order  $\alpha$ , the original Fourier transform set-up is split into  $1/\alpha$  identical sections each containing a lens and propagation. The field half-way between the first and second lens is then the fractional transform of  $u(x, y)$ . If  $\alpha$  is a rational number, it is possible to construct such a system. The third system is that of a quadratically graded-index fiber, the refractive index across the fiber is that of a thin lens. Within such a system, the fractional Fourier transform order is the fractional distance covered between an image and Fourier plane.

the time to focus is not as straightforward as the optical case, the focusing behaves in the same manner.

When attempting to Fourier transform a BEC, the obvious first step is to map the optical case directly onto the BEC. In an experimental situation within optics, a beam is Fourier transformed using a familiar  $2f$  optical set-up which is laid out as shown in Figure 4.1. The lens is placed its focal length,  $f$ , behind the plane of interest and then at a distance  $f$  behind the lens, the complex field corresponds to the Fourier transform of the original plane. Such set-ups are often used for beam analysis where the inclination of the wavefront gives rise to a lateral shift of the pattern in the Fourier plane<sup>31</sup> and is also used in filtering momentum components from the beam.

The response of a BEC to a focusing laser pulse (as seen in Chapter 3) gives rise to added complications. In such a system, the time after which a condensate comes to a focus does not vary linearly with the focusing power of the imprinted phase and can excite additional dynamics within the condensate and complicate the system. These additional dynamics have a negative effect on the efficiency of the system, so another method of Fourier transforming was sought.

#### 4.4.2 Fourier Transforming by waiting

When seeking alternative methods for Fourier transforming the condensate, the additional dynamics associated with the harmonic trap became a focus of attention. By delving into the behavior of a trapped condensate, the similarities between the nature of these dynamics and the dynamics of light propagation in quadratic graded-index fibres became apparent.

### Dynamics of a trapped BEC

Most BEC traps are harmonic, at least locally. The corresponding shape of the ground state depends on the interactions. For a weakly interacting case it takes the form of a Gaussian, whereas if the chemical potential  $g\rho$  is larger than the ground-state energy of the harmonic trap,  $\hbar\omega$ , the density takes a parabolic shape, often referred to as the Thomas-Fermi solution. If the ground state is perturbed, interesting non-stationary modes can form<sup>32,33,34</sup>.

If a trapped BEC is initially in the shape of the ground state but displaced sideways from the trap centre it will keep its shape and size, but its centre of mass will oscillate about the trap centre with the trap period,  $T$  (the inverse of the trap frequency). Such a mode is called a dipole mode. If the width of the probability density is initially narrower than that of the ground state, the BEC expands and then contracts again – a so-called breathing mode. This oscillation happens over half a trap period for a weakly interacting gas, but the oscillation time changes with increasing interaction strength.

We consider here the dynamics of states of a non-interacting condensate that start off much wider than the ground state and which are positioned off-centre in the trap. Figure 4.4 shows the simulated dynamics of such states in a 1-dimensional (1D) BEC. The dynamics of the states contain the characteristics of both the dipole mode and breathing mode, so we call them *breathing dipole modes*.

Breathing dipole modes can be seen as approximations to Fourier components. An analytical solution of the 1D Gross-Pitaevskii equation with

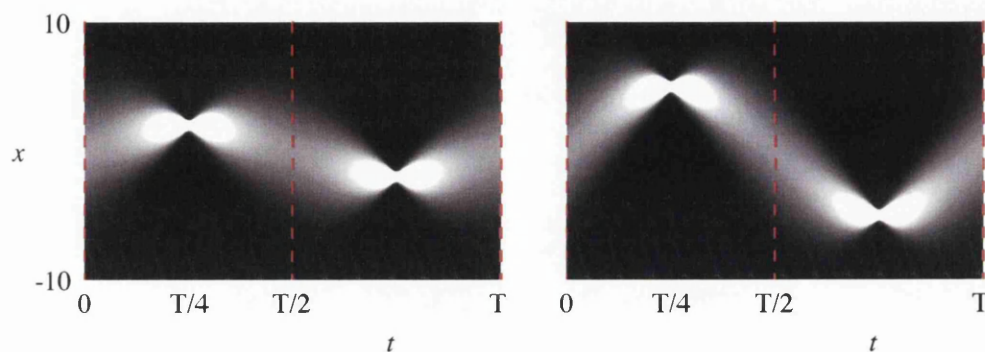


Figure 4.4: Dynamics of 1-dimensional breathing dipole modes – approximations to a BEC’s Fourier components. The graphs show a grayscale representation of the density distribution along the  $x$  axis as a function of time,  $t$ , calculated using equation (4.3). At  $t = 0$ , the wave function is a Gaussian of width 4 (in units of the dipole-mode width) with a transverse wave number  $k_x = 3$  (left) and  $k_x = 5$  (right). The time evolution of the probability density is plotted over one trap period,  $T$ . At  $t = T/4$ , the width of the BEC has shrunk to one quarter of the dipole-mode width (generally it is inversely proportional to the width at  $t = 0$ ) and its position is proportional to the transverse wave number at  $t = 0$ ,  $k_x$ . Every further quarter of a trap period corresponds to an additional Fourier transform.

starting conditions (at  $t = 0$ )

$$\psi_{\sigma, k_x}(x, 0) = \exp\left(-\frac{x^2}{2\sigma^2} + ik_x x\right), \quad (4.2)$$

which is a centred Gaussian of width  $\sigma$  with a linear phase gradient proportional to the transverse wave number  $k_x$ , gives for time  $t$

$$\psi_{\sigma, k_x}(x, t) = \left(\frac{i \sin t}{\sigma^2} + \cos t\right)^{-1/2} \times \exp\left(-\frac{x^2 \cot t + i\sigma^2(x^2 + k_x^2 - 2k_x x \csc t)}{2i + 2\sigma^2 \cot t}\right), \quad (4.3)$$

where length and time are respectively in units of  $\sqrt{\hbar/m\omega}$  and  $1/\omega$  of the harmonic oscillator. In these units, the trap period is  $T = 2\pi$ . For time  $t = T/4$  this simplifies to

$$\psi_{\sigma, k_x}(x, T/4) = A \exp\left(-\frac{(x - k_x)^2}{2\tau^2}\right) \quad (4.4)$$

(where  $A = \frac{\sigma(1-i)}{\sqrt{2}}$  and  $\tau = 1/\sigma$ ), which is a flat-phased Gaussian whose  $x$ -position is proportional to  $k_x$ , the phase gradient at time  $t = 0$ . If the width of the Gaussian at  $t = 0$ ,  $\sigma$ , is large, i.e.  $\sigma \gg 1$ , the width of the Gaussian at  $t = T/4$ ,  $\tau$ , becomes small, i.e.  $\tau \ll 1$ . This time evolution is approximately a Fourier transform, in which a uniform, inclined (with transverse wave number  $k_x$ ), plane wave transforms into a  $\delta$  function at a displacement proportional to  $k_x$ . In fact, in the limit  $\sigma \rightarrow \infty$  the breathing dipole mode behaves exactly like a Fourier component. This is the key observation which enables Fourier transforming by waiting in a trap.

Another quarter of a trap period later, at  $t = T/2$ ,  $\psi$  is back to a Gaussian peak of the original width  $\sigma$  and with a uniform phase gradient across it that



is equal and opposite to that of the wave function at  $t = 0$ :

$$\psi_{\sigma, k_x}(x, T/2) = i \exp\left(-\frac{x^2}{2\sigma^2} + i(-k_x)x\right). \quad (4.5)$$

The time evolution between  $t = T/4$  and  $t = T/2$  corresponds to another Fourier transform. As a Fourier transform is equivalent to an inverse Fourier transform combined with a flipping of the original distribution, two successive Fourier transforms, and therefore time evolution through  $T/2$ , correspond to a flipping of the original distribution.

So the time evolution through  $T/4$  of breathing dipole modes corresponds to the Fourier transform of individual Fourier components. For the time evolution through  $T/4$  of arbitrary superpositions of breathing dipole modes to correspond to a Fourier transform, the individual Fourier components have to keep their relative phase at the trap centre ( $x = 0$ ), which they do.

## 4.5 Simulation Results

To demonstrate the Fourier-transform property of a BEC's time evolution, we have numerically simulated<sup>13</sup> the time evolution of 2-dimensional (2D) and 3-dimensional (3D) non-interacting ( $g = 0$ ) BECs in isotropic harmonic traps by integration of the Gross-Pitaveskii equation. (The dynamics for anisotropic traps are considerably more complex and beyond the scope of this work.) We started with an initial wave function that is the (discrete) inverse Fourier transform of distinctive shapes: in the 2D case a photo frequently used in image processing, in the 3D case a shape that looks like a circle from one direction and like a square from another. Time evolution through  $T/4$

should then return the BEC into its original shape. Figure 4.5 clearly shows that this is the case in our simulation.

In order to further explore the nature of the time evolution of the BEC in terms of its Fourier transform, various 2-dimensional condensates were studied. The first investigations were performed using the discrete inverse Fourier transform of a easily identified pattern as the initial condition. Such simulations allowed ready identification of the wanted state and also made it easy to identify how well the pattern resembled the original. To quantify the efficiency of this Fourier transform, the fraction of the field which is in the desired state,  $f$  was defined as

$$f(t) = \sum \mathcal{F}(\psi(0))^* \psi(t) \quad (4.6)$$

where  $\psi(t)$  is the wavefunction at time  $t$  and  $\mathcal{F}(\psi(0))$  is the Fourier transform of the initial condensate.

Figure 4.7 shows how the fraction of the condensate in the final wanted state changes on propagation through  $T/2$ . It can clearly be seen that the fraction peaks around  $T/4$  and in the non-interacting case the fraction is near unity, however, it quickly falls from that value for higher values of  $g$ . A slight shift is visible as the non-linearity is increased from  $g = 0$ , although by  $g = 150$  additional structures have formed and moved the peak from this pattern. From the insets showing the state at the peak, it is quite clear that the condensate no longer resembles the wanted state: in fact the  $g = 100$  and  $g = 150$  cases seem to have an almost inverse relation to the wanted state.

Because of the non-linearity, the behavior of the condensate is very pattern-dependent, and considering the discrete Fourier transform of a photograph is a very unnatural initial condensate, this rapid drop in efficiency with in-

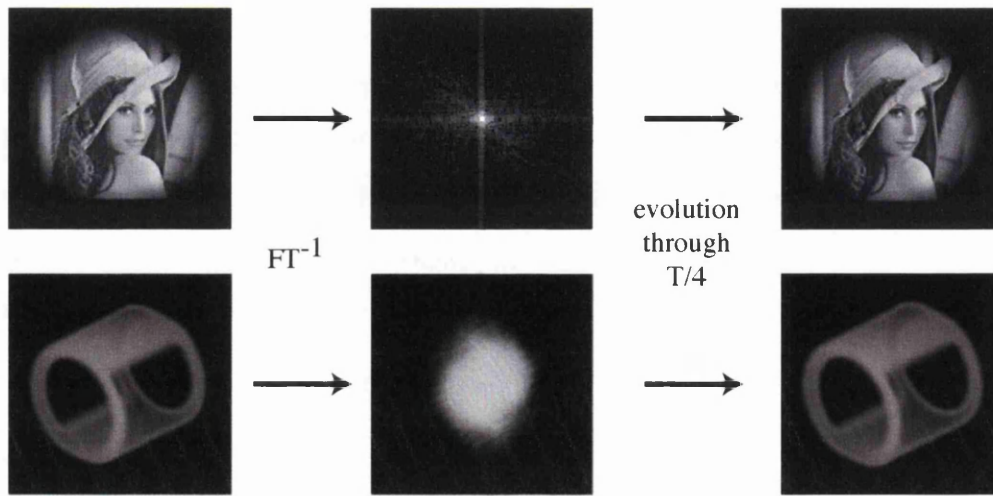


Figure 4.5: Simulated Fourier transform of non-interacting ( $g = 0$ ) trapped 2-dimensional (top) and 3-dimensional (bottom) BECs by waiting a quarter of the trap period, that is  $T/4$ . In both cases we set the initial wave function (centre; log plot over four orders of magnitude) to the (discrete) inverse Fourier transform of distinctive shapes (left) – a well-known photo in the 2D case and a shape that looks circular from one direction and square from another in the 3D case – and simulated their time evolution over  $T/4$ , using a split-step technique<sup>13</sup>. In the 2D case, the simulations were run on a grid of  $96 \times 96$ , representing a square of side length 34.7 dimensionless units. Density is represented as brightness (black represents zero density, white maximum density). In the 3D case, simulations were run on a  $64 \times 64 \times 64$  grid, representing a cube of side length 35 dimensionless units. Density is represented as white opacity in front of a black background; the graphs were created using the *Volpack* rendering software<sup>35,36</sup>.

creasing non-linearity is not unexpected. Figure 4.8 shows how this fraction changes as the non-linearity is changed for a series of different condensates. It is quite clear that in the cases of unnatural condensates (where the density is in the pattern of a photograph), the efficiency falls away very quickly. However the fall-off is much reduced in the cases of Laguerre-Gaussian states, which are given by

$$\Psi(r, \phi) \propto (r/r_0)^m e^{-(r/r_0)^2/2} e^{im\phi}, \quad (4.7)$$

where  $(r, \phi)$  are circular coordinates around the trap centre,  $r_0$  determines the size of the density doughnut, and  $m$  is the topological charge of the vortex. Single-charge ( $m = 1$ ) vortices are an eigenmode of many trap geometries and, once formed, can remain stable for the lifetime of the condensate<sup>37</sup>.

As discussed previously, the breathing mode period of a condensate is dependent on the non-linearity which breaks the link in the breathing mode and dipole mode periods, to further complicate the dynamics. Figure 4.6 shows the position of the peak in efficiency as the non-linearity is changed. Once again it is easy to see the difference between the Laguerre-Gaussian and the unnatural states, emphasising the dependence on the chosen initial state.

These results show that waiting for  $T/4$  does not always Fourier-transform the wave function of an interacting BEC perfectly, and that the detailed behavior depends on the wave function. General trends that can be seen in our limited-size sample are that  $f(t)$  has a maximum near  $T/4$ , that  $t_{\text{peak}}(g)$  is longer than  $T/4$  for  $g > 0$ , and that in Figure 4.8,  $f_{\text{peak}}$  generally falls off as  $|g|$  increases. However, our sample clearly does not represent all possible behaviors. Had we, for example, chosen the initial state to be stationary, then

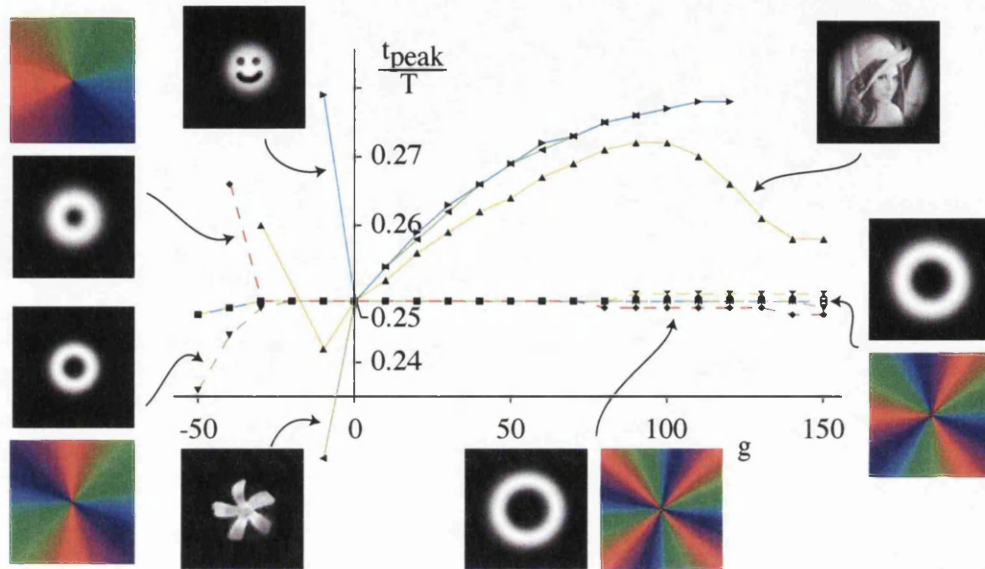


Figure 4.6: Time after which most of the BEC is in the Fourier-transformed state for various wave functions (insets) as a function of the interaction strength  $g$ . Only the data points for which more than 20% of the BEC are in the Fourier-transformed state are shown. Insets show a grayscale representation of the central  $20 \times 20$  dimensionless units of the density of the state  $P$  for each curve, and a colour representation (the colour range red  $\rightarrow$  green  $\rightarrow$  blue represents phases between  $0$  to  $2\pi$ ) of the phase, unless the phase is uniform.

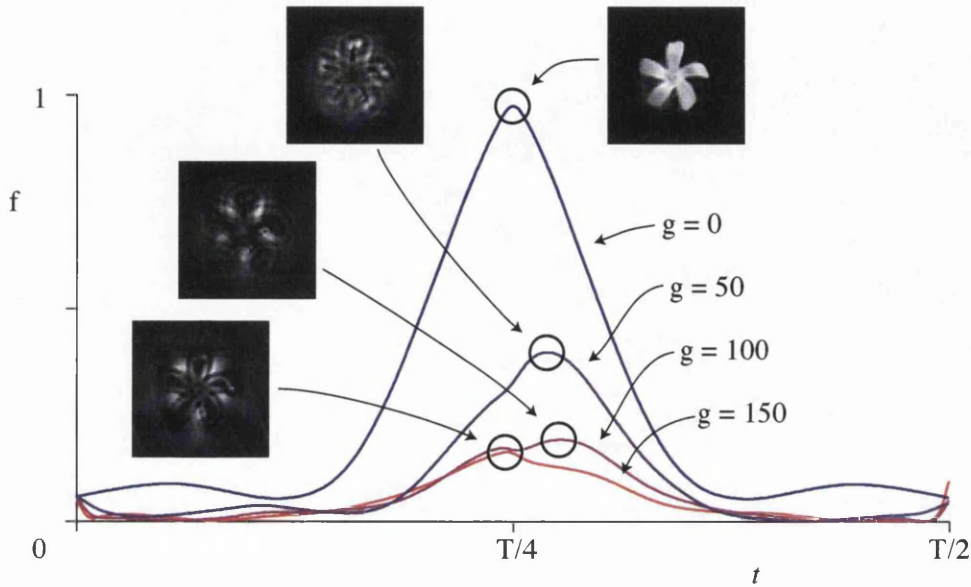


Figure 4.7: Time evolution of a BEC in a harmonic trap in terms of the Fourier transform of the initial wave function. The initial wave function ( $t = 0$ ) was set to the inverse Fourier transform of wave function  $P$  with a distinctive density distribution. The graph shows the fraction,  $f$ , of the wave function  $\psi(t)$  in the state  $P$ , calculated according to equation (4.6). The figure was calculated for various interaction strengths ( $g = 0, 50, 100$  and  $150$ ). In the interaction-free case ( $g = 0$ ), close to 100% of the BEC is in the Fourier-transformed state, i.e.  $P$ , at  $T/4$ . Insets show the density of the wave function corresponding to the peaks of the different curves, where for  $g = 0, 50, 100$  and  $150$  respectively 97.5%, 39.8%, 19.3% and 16.3% of atoms are in the state  $P$ . The central squares of side length 20 dimensionless units are shown. Like those for figures 4.6 and 4.8, the simulations for this figure were performed on a  $128 \times 128$  grid representing a square of side length 40.1 dimensionless units.

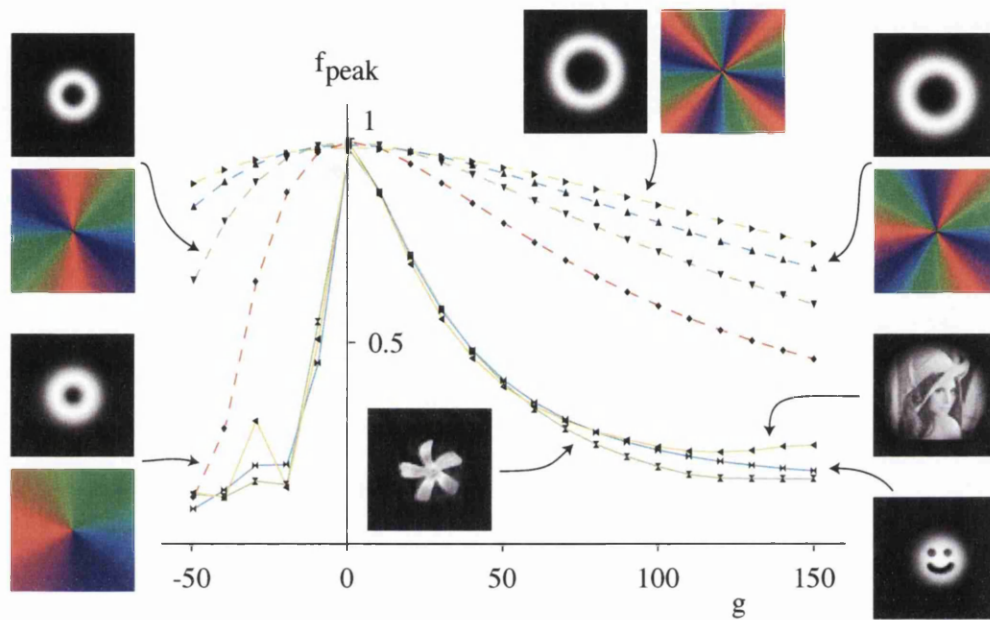


Figure 4.8: Maximum fraction in the state  $P$  of a BEC's wave function during simulated time evolution through  $T/2$ , starting with the Fourier transform of  $P$ , as a function of  $g$ , for various choices of  $P$ . The insets show a grayscale representation of the central  $20 \times 20$  dimensionless units of the density of the state  $P$  for each curve, and a colour representation of the phase (unless the phase is uniform).

the graph of  $f(t)$  would be a flat line and therefore not have a maximum near  $T/4$ . The detailed behavior also depends on the size of the states: for example, if the soliton states are stretched they are no longer solitons and their behavior will be different. Again, we chose the size of our states arbitrarily.

## 4.6 Fourier filtering

As an example for an application of the Fourier transform we consider here a Fourier-filtering technique borrowed from optics which allows arbitrary shaping of both a wave function's amplitude and phase when only the amplitude can be shaped directly. (There are also optical techniques for shaping phase and amplitude when only the phase can be shaped directly<sup>38</sup>, which, by analogy, would also work for BECs and make direct use of the various techniques for "phase imprinting"<sup>39,40</sup>.)

If a wave function with a uniform phase and arbitrary density distribution can be created, then part of it can be turned into an arbitrary state  $\psi_a$  as follows. The state

$$\psi_1 = \tilde{\psi}_a \exp(ik_x x) + \tilde{\psi}_a^* \exp(-ik_x x) + r \quad (4.8)$$

is a superposition of the state  $\tilde{\psi}_a$ , the inverse Fourier transform of  $\psi_a$ , travelling in the positive  $x$  direction (with additional wave number  $k_x$ ), its complex conjugate travelling in the negative  $x$  direction, and a real number  $r$ . As the sum of a complex number and its complex conjugate is a real number, this wave function is real everywhere, and if  $r$  is chosen to be sufficiently large it is also positive everywhere.



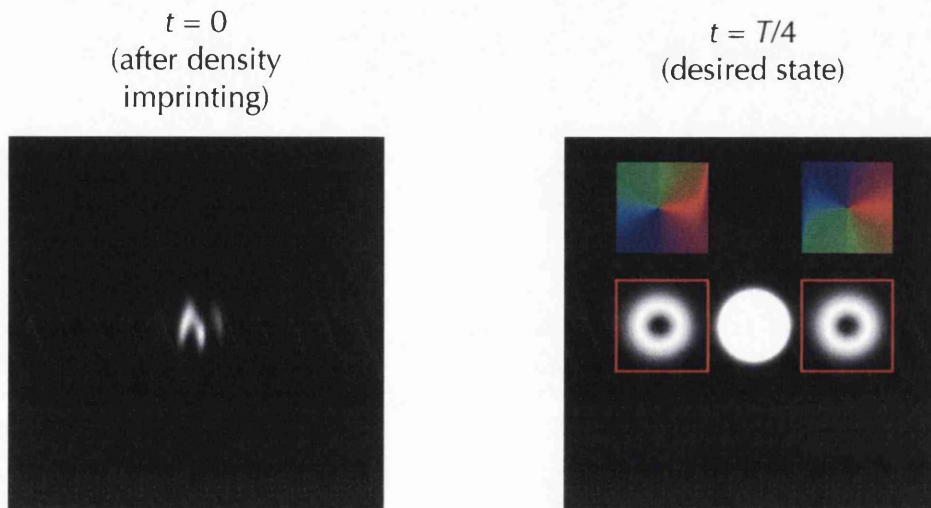


Figure 4.9: Shaping of a BEC in the Fourier domain. In this example the phase and density of part of a pancake BEC was shaped into a vortex at  $t = T/4$  after only the density of the BEC was manipulated a quarter of a trap period earlier ( $t = 0$ ). The state at  $t = 0$  was calculated according to equation (4.8). At  $t = T/4$ , about 85% of the remaining BEC is in the central peak, the rest is distributed equally between the doughnut-shaped vortices with circulation  $+1$  and  $-1$  on either side (about 7.5% each). A colour representation of the phase in the areas around both vortices, marked by red squares, is shown above the squares. The simulations for this figure were performed on a  $256 \times 256$  grid representing an area of  $28 \times 28$  dimensionless units.

The state  $\psi_1$  can therefore be created from a BEC ground state, which has uniform phase, by manipulating the density of the BEC to be  $|\psi_1|^2$ , provided this does not influence the phase distribution. This can be achieved by adding to the harmonic trap a light-induced dipole trap, whose shape can be engineered using holograms. A quarter of a trap period later, after the light-induced dipole trap has been removed but with the harmonic trap still present, the three terms in equation (4.8) have – as a whole and individually – Fourier transformed and separated spatially (provided  $k_x$  was chosen to be sufficiently large). The parts of the BEC that correspond to the last two terms can be filtered out using the same technique for removing parts of a BEC by using a shaped resonant laser beam to selectively transfer atoms into an untrapped state. The remaining part is the desired state  $\psi_a$ , positioned off-centre in the trap. Figure 4.9 shows an example in a 2D BEC in which vortices are created from a trap ground state by manipulating the density of the ground state and waiting for a quarter of a trap period.

This is an example of the more general idea of shaping only part of a BEC into a fully specified state and accepting that there is a remainder which might have to be removed from the trap in order not to interfere with the desired part at a later stage.

## Chapter 5

# Vortex sorter for Bose-Einstein condensates

This chapter describes the equivalent of an optical system which was developed to decompose a light beam into its orbital angular momentum components<sup>41</sup>. The optical system is based on interfering the un-rotated half of a split light beam with its rotated part. Section 5.1 describes the motivation for decomposing a light beam or BEC into its orbital angular momentum components. Section 5.2 describes some background to vortices and their behavior. Section 5.3 describes the optical setup of the vortex sorter and section 5.4 described the equivalent setup for BECs. Section 5.5 discusses several methods of performing the required rotation of a BEC.

My role within this project was to develop the bespoke software required for the simulations and to analyse and present the results. I was also involved in guiding the research through discussions with the co-authors in which I played a key part. I was also involved in the preparation of the paper which was published describing this work.

## 5.1 Introduction

One of the central characteristics of a superfluid such as a Bose-Einstein condensate (BEC) is the presence of quantized vortices. Vortices have been generated experimentally<sup>42,43</sup> by stirring the BEC, very similar to the rotating-bucket experiment in Helium<sup>44</sup>. The detection of vortices in BECs is typically made by a direct observation of the vortex core or by interference experiments<sup>45</sup>. In present experiments the orbital angular momentum of the condensate is known in principle, because the initial rotation frequency which stirs the cloud is known. However the distribution of the orbital angular momentum is not known. In this chapter we show how to split a condensate into its orbital angular momentum components.

Vortices have attracted considerable interest both experimentally and theoretically, mainly because of their inherent many-body character and the connection with fluid dynamics. In addition, optical vortices in single photons have recently been used to carry information – and in particular quantum information<sup>46</sup>. Optical vortices, special cases of light with orbital angular momentum, can carry huge amounts of information as there is in principle no limit to the quantized angular momentum acting as the information carrier. Light is an excellent carrier of information over large distances as the photons travel very fast and do not easily interact with each other.

For the same reason, photons are not very well suited for storing the information for longer times. This is where cold atoms, particularly BECs, would be better suited as a medium for storing information, especially quantum information. The instability of vortex states within BECs in harmonic traps is a potential problem, but it might be overcome with ideas such as the use

of pinning potentials<sup>47,48</sup>.

Transferring angular momentum to atoms would constitute a way to store information<sup>49</sup>. If such a storage device is to work with atoms we need a way to manipulate atomic states, and in particular vortex states, in an efficient and useful way. It is therefore important to know the mechanisms behind the vortex dynamics and more importantly how to manipulate the vortex states in order to be able to make any kind of readouts from the trapped quantum gas. This chapter discusses theoretically the application to BECs of ideas borrowed from optical vortex sorting<sup>41</sup>.

## 5.2 Vortices

Vortices form one of the standard states used when investigating the behaviour of our simulated systems. Vortex-research is an area of intense interest in both optics<sup>50</sup> and Bose-Einstein Condensates<sup>51</sup> with many aspects of their behaviour being studied.

### 5.2.1 Vortex Phase

The defining element of a vortex is its phase structure. Canonical vortices can be described as having a wavefunction of the form

$$\psi(r, \phi) = A(r) \exp(im\phi) \quad (5.1)$$

where  $A(r)$  is the real amplitude of the wavefunction and  $m$  is a non-zero integer. This gives rise to the phase having a linear dependence on the

azimuthal position and hence the momentum density flux

$$v = \nabla \arg \psi \quad (5.2)$$

having a single rotational component. It can be seen that if we integrate the phase around any closed loop,  $\sigma$  where the phase is well-defined, and assuming the wavefunction is continuous in space we have

$$\int_{\sigma} (\nabla \arg \psi) dr = 2\pi m, \quad (5.3)$$

where  $m$  is an integer. This quantised nature of orbital angular momentum implies that in order to get any rotation within a system, vortices must be involved, and hence a system without vortices can not rotate.

### 5.2.2 Laguerre-Gaussian modes

One of the most useful descriptions of vortices comes in the form of the Laguerre-Gaussian modes used in optics. They describe a family of structurally stable modes which are azimuthally symmetric and many of the modes contain vortices. They are a useful tool in optics as they are a complete, orthogonal set of modes and as such any arbitrary light beam can be described as a superposition of Laguerre-Gaussian beams.

A Laguerre-Gaussian mode with its beam waist at  $z = 0$  can be described by

$$\begin{aligned} LG_p^m(r, \phi, z) = & \frac{1}{\omega(z)} \left( \frac{r\sqrt{2}}{\omega(z)} \right)^{|m|} L_p^{|m|} \left( \frac{2r^2}{\omega^2(z)} \right) \exp \left( \frac{-r^2}{\omega^2(z)} \right) \\ & \times \exp \left( \frac{-ikr^2z}{2(z^2 + z_R^2)} \right) \exp(-im\phi) \exp(-ikz) \\ & \times \exp \left( i(2p + |m| + 1) \arctan \left( \frac{z}{z_R} \right) \right) \end{aligned} \quad (5.4)$$

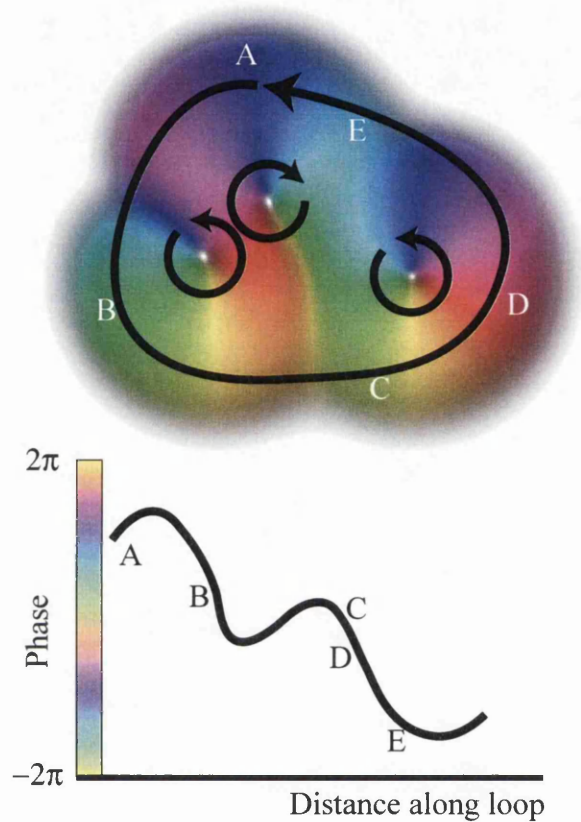


Figure 5.1: Top: An example of a wavefunction with three vortices (the phase is represented by the colour at each point using the sequence red  $\rightarrow$  green  $\rightarrow$  blue  $\rightarrow$  red to represent  $0 \rightarrow 2\pi$ ), one with clockwise rotation ( $m = 1$ ), the other two with anti-clockwise rotation ( $m = -1$ ). Also shown is a closed loop surrounding all three vortices. The figure on the bottom shows the phase along the loop as it passes through a series of points. As we move around the loop (anti-clockwise) we lose  $2\pi$  in phase as would be expected from adding up the charges of the vortices enclosed within the loop.

---

where  $m$  and  $p$  are the azimuthal and radial mode numbers respectively,  $z_R$  is the Rayleigh range of the beam,  $\omega(z)$  is the  $1/e$  radius of the beam and  $L_p^l$  is the associated Laguerre polynomial. With the exception of chapter 3, only the set of modes with  $p = 0$  is considered here.

### 5.2.3 Optical Vortices

Despite their seemingly strange behaviour, vortices are a very common occurrence, in fact the interference of 3 plane waves produces an abundance of vortices<sup>52</sup>. Their presence in speckle patterns has meant that many researchers have invested much time in finding ways to remove them. One of the most common ways of generating optical vortices is by using phase holograms of spiral phase plate (an optical element whose thickness varies linearly with azimuthal angle). Very pure vortex modes can also be generated using specially designed laser cavities<sup>53</sup>, holographic shaping or by transforming Hermite-Gaussian modes using cylindrical lens converters (see section 3.7 for more details).

### 5.2.4 BEC Vortices

Vortices are of great interest to the BEC community, with many experiments utilising many different methods to generate them. Their higher energy means that for most systems extra external energy must be provided to create a vortex. In many situations any created vortices will remain stable over long periods; however they can easily annihilate or drift outside the condensate.



One feature of vortices in BECs which is more prominent than in optics is the instability of higher charge vortices. High-charge vortices ( $|m| > 1$ ) in a harmonic trap are unstable and will quickly decay. The conservation of orbital angular momentum rules out a higher charge vortex simply decaying into a lower charge one. Instead the higher charge vortex will split into  $|m|$  single charge vortices which will often then rotate around either each other or the centre of the trap. This behaviour makes maintaining higher order vortices much more difficult and non-standard methods have been proposed to try and stabilise them<sup>54</sup>.

### 5.3 Vortex sorter

If a vortex (in light or in a BEC) of charge  $m = 1$  is rotated through  $180^\circ$  about its centre, it changes phase by  $\pi$  (and, in the simplest case, is unchanged in any other respect). If, on the other hand, a vortex of charge  $m = 2$  is rotated through  $180^\circ$ , its phase is unchanged. The two cases discussed above are in fact representative for all vortices with odd and even charges, respectively. This effect has been used in an optical two-arm interferometer which rotates the beams in the two arms with respect to each other to route vortices according to their charge into one of the interferometer's two exit ports<sup>41</sup>. Figure 5.2 shows a representation of such an interferometer. When the arms are re-combined, even-charge vortex components interfere constructively in one interferometer port and therefore exit the interferometer through it, while odd-charge vortex components interfere destructively in that port and therefore exit the interferometer through another port (in

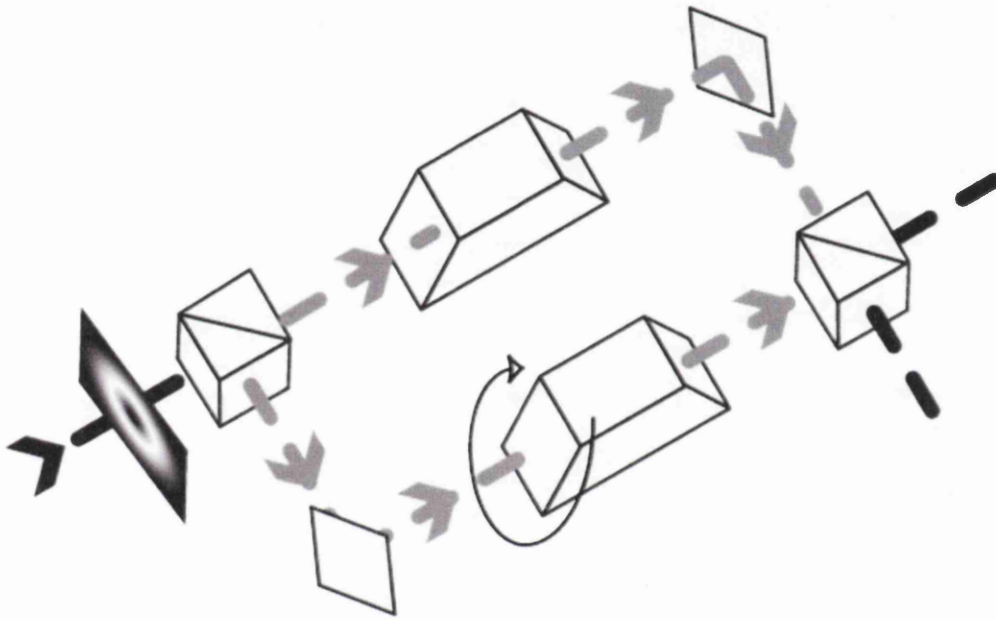


Figure 5.2: Schematic of the optical vortex sorter<sup>41</sup> which can decompose an input light beam into its vortex components. The optical system uses a Dove prism in each arm of the interferometer to introduce a rotation.

which even-charge vortices interfere destructively). The vortices exiting from the two ports can be sorted further in similar interferometers, but with different relative rotation angles<sup>41</sup>. For example, vortices with even charges ( $m = 0, \pm 2, \pm 4, \dots$ ) can be sorted into those whose charges are respectively integer and half-integer multiples of 4 ( $m \bmod 4 = 0$  or  $m \bmod 4 = 2$ , respectively). In some cases, uniform phase offsets in one arm are required<sup>55</sup>.

## 5.4 Bose-Einstein condensate vortex sorter

To simulate a BEC equivalent of the optical vortex sorter, suitable components have to be found. Bragg pulses offer the possibility to coherently split and recombine a BEC in the same manner as the beam splitter cubes used in the optical system.

The rotation of the BEC was modelled as two mirroring processes performed using cylindrical lens pulses which focus the condensate along one axis. Section 5.5 discusses other possibilities for rotating a condensate.

### 5.4.1 Bragg pulse splitting

The requirement of splitting the condensate into two spatially separate condensates can be achieved in a number of ways, including using a potential which produces multiple traps<sup>56</sup> or removing atoms from the trapped state and placing them into another trap. However recombining the condensate in the manner of a beam splitter is more difficult. One aspect of a beam splitter, which is often overlooked, is the two exit ports. Ignoring absolute phase and intensity scaling, the beam in one exit port is the sum of the two

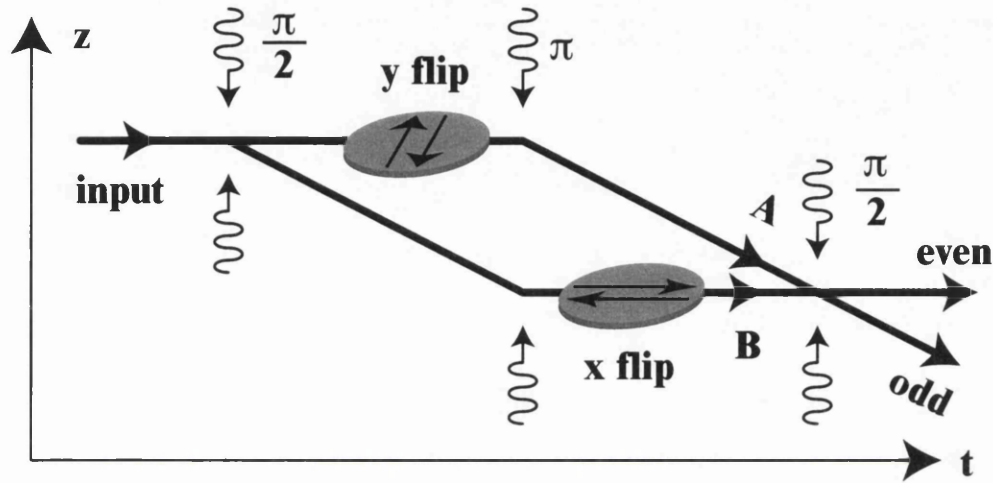


Figure 5.3: Schematic of a vortex-sorting Bragg-pulse interferometer. A stationary input BEC is first split by a  $\pi/2$  Bragg pulse (left; the horizontal axis is time). The two resulting BECs are in different momentum states and move apart. A  $\pi$  Bragg pulse swaps the momentum states, so that the two BECs move together again. We refer to the two different trajectories in this space-time diagram as two ‘arms’ of the interferometer. The BECs in the two arms are flipped vertically and horizontally, respectively, which corresponds to a relative rotation through  $180^\circ$ . A second  $\pi/2$  Bragg pulse mixes the two BECs such that even and odd vortex components have different momentum states – they exit the interferometer through different ‘ports’.

input beams, while that in the other port is their difference. (See equation 5.5 for a full formula.)

Bragg pulses have been used to create Mach-Zehnder interferometers similar to the vortex sorter system and have achieved almost 100% contrast<sup>57</sup>. A Bragg pulse is created by interfering two counter-propagating light beams with slightly different frequencies. This creates a “moving standing wave” which moves with a speed of one standing-wave frequency times the frequency difference. The interaction between the light and the atoms within the condensate acts to transfer photons from one of the interfering light beams to the other. Because the two light beams are of different frequencies, some atoms gain momentum in the direction of propagation while others lose momentum. The rate at which atoms are transferred between the stationary and the moving state is dependent on the populations of each state and, if the light beams are left on, results in the population oscillating between the two states.

The pulse length can therefore be chosen to transfer a given fraction of the population into the desired state. In this implementation two pulse lengths are used. These are described as  $\pi/2$  and  $\pi$  pulses. A  $\pi/2$  pulse acts in a similar way to a beam splitter cube and transfers the atoms according to

$$\begin{aligned}\Psi'_1 &= \frac{1}{\sqrt{2}}(\Psi_1 + \Psi_2) \\ \Psi'_2 &= \frac{1}{\sqrt{2}}(\Psi_1 - \Psi_2),\end{aligned}\tag{5.5}$$

where  $\Psi_1$  and  $\Psi_2$  are the stationary and moving states respectively. The  $\pi$  pulse swaps the populations of the two states and is used as a means of bringing the previously stationary state towards the previously moving state

to bring the two condensates together.

### 5.4.2 BEC setup

We model a 2-dimensional BEC that is split into two identical BECs, which are then rotated with respect to each other through  $180^\circ$ , and finally superposed. The wave function of the original BEC,  $\Psi$ , is split according to

$$\Psi'_1 = \Psi'_2 = \frac{1}{\sqrt{2}}\Psi; \quad (5.6)$$

the detailed physical splitting mechanism is not modelled. The two BECs are rotated with respect to each other through mirroring the two BECs with respect to the  $x$  and  $y$  axis, respectively. As explained in Section 5.5, this can be achieved through imprinting specific phases onto the BECs at specific times. In combination with the time evolution between the phase-imprinting events, this results in wave functions  $\Psi'_1$  and  $\Psi'_2$ . We simulate this time evolution according to the time-dependent Gross-Pitaevskii equation 2.8. Finally, again without modelling the detailed mechanism, the two wave functions  $\Psi'_1$  and  $\Psi'_2$  are superposed according to the equations

$$\Psi_{\text{even}} = \frac{1}{\sqrt{2}}(\Psi'_1 + \Psi'_2), \quad \Psi_{\text{odd}} = \frac{1}{\sqrt{2}}(\Psi'_1 - \Psi'_2). \quad (5.7)$$

This model can represent various interferometers, all of which are idealised in some respects. For example, a Bragg-pulse interferometer with rotation in the arms (Fig. 5.3), is idealised as follows. Firstly, the Bragg pulses are assumed to be perfect, that is acting according to equations (5.6) and (5.7). The  $\pi$  pulses, which are also required in the Bragg-pulse-interferometer scheme, and which swap the BECs between the two states, are also assumed

to be perfect (which is consistent with experiments in which fringe visibilities close to 1 were achieved in Bragg-pulse interferometers<sup>57</sup>). Secondly, the interaction between the BECs in the different arms is neglected. Unfortunately no experimentally realisable situation is exactly represented by this, but some are represented better than others. A Bragg-pulse direction that separates the planes of the two BECs, for example, should lead to less interaction between the BECs than a Bragg-pulse direction that moves the two planes across each other; however, applying the light pulses for rotation to the two arms separately is potentially difficult in this geometry. Thirdly, the arm length is just that required for rotation; we made no allowance for any additional time it might take for the BECs to separate sufficiently such that they can be rotated independently and subsequently recombined. However, in analogy to optical imaging, light pulses<sup>23</sup> might be able to return the BEC into an earlier state, thereby effectively shortening the arms. Other interferometer types that approximate our idealised model include for example those that split a BEC into two by putting different parts into different internal states<sup>45</sup> and manipulate the parts independently through phase imprinting with light pulses with different detunings.

Fig. 5.4 shows examples of sorting a BEC into its ‘even’ and ‘odd’ vortex components. In all our simulations the initial wavefunctions were 2D Laguerre-Gaussian (LG) functions of the form (see section 5.2.2 for full equation)

$$\Psi(\rho, \phi) \propto \rho^m e^{-\rho^2/2} e^{im\phi}, \quad (5.8)$$

with the exception of the last example in Fig. 5.4, which used a superposition of two such functions. LG functions are commonly used in optics (see, for

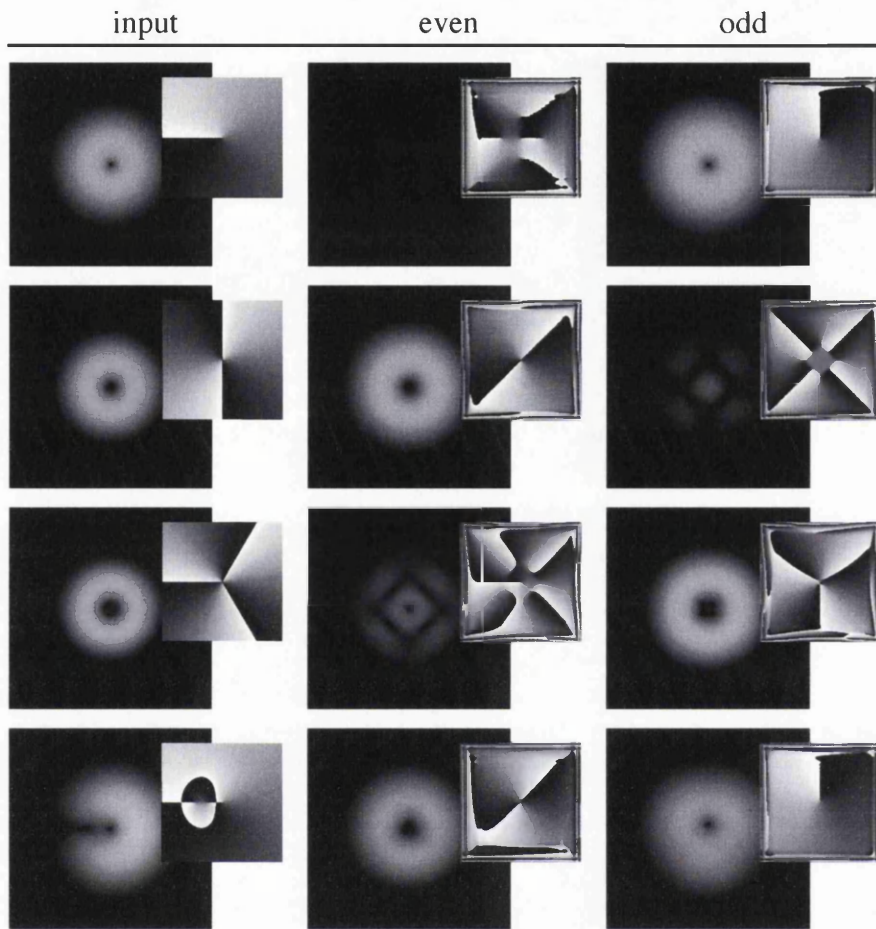


Figure 5.4: Simulated vortex sorting in an idealised interferometer. The columns show the probability density and phase (inset at reduced size) of the input BEC (left) and the BEC in the ‘even’ (centre) and ‘odd’ (right) output ports. The top three cases show sorting of pure vortices in  $m = 1$ ,  $m = 2$ , and  $m = 3$  states. The sorting can be seen to be not perfect: a small fraction can be seen to come out of the ‘wrong’ output port. The fourth case shows sorting of a superposition of vortices with charges  $m = 1$  and  $m = 2$ , which is split into its vortex components. This figure was calculated for  $g = 0$ .



example, equation (3) in Ref.<sup>24</sup> with  $p = 0$  and  $z = 0$ ). All the simulations in this chapter were performed over an area of  $14 \times 14$  (in units of  $\sqrt{\hbar/m_a\omega}$ ) on a  $256 \times 256$  grid of wave function amplitudes. Fig. 5.5 shows the fraction of the original BEC that exits the interferometer in the correct port – a measure of the quality of the sorting – as a function of the non-linear coefficient  $g$ . It can be seen that the scheme works better for small values of  $g$ . It can also be seen that the efficiency of the system at  $g = 0$  is not 1 (most noticeable in the higher charge vortices). This is due to the imperfect nature of the rotation (discussed in more detail in Section 5.5) and the instability of higher charge vortices<sup>58</sup>. Higher charge vortices are inherently unstable within BECs and will decay into multiple single charge vortices upon propagation (which can be seen in Figure 5.5).

## 5.5 Rotation of BECs

Several methods exist for setting a BEC into rotation, using for example Bragg pulses<sup>59,60</sup>, a careful arrangement of laser beams<sup>61,62</sup>, or an external magnetic field<sup>63</sup>. These methods were mainly aimed at rotating the BEC above a certain frequency to create vortex states. However the vortex sorter requires a method which can rotate the BEC through a given angle without otherwise affecting its state. Some of the methods for inducing rotational motion could be adapted for this purpose if the behaviour of the rotation is understood. However, instead we describe here a method based on an optical analogy<sup>25</sup>.

This method is based on the fact that mirroring in one axis (or plane in

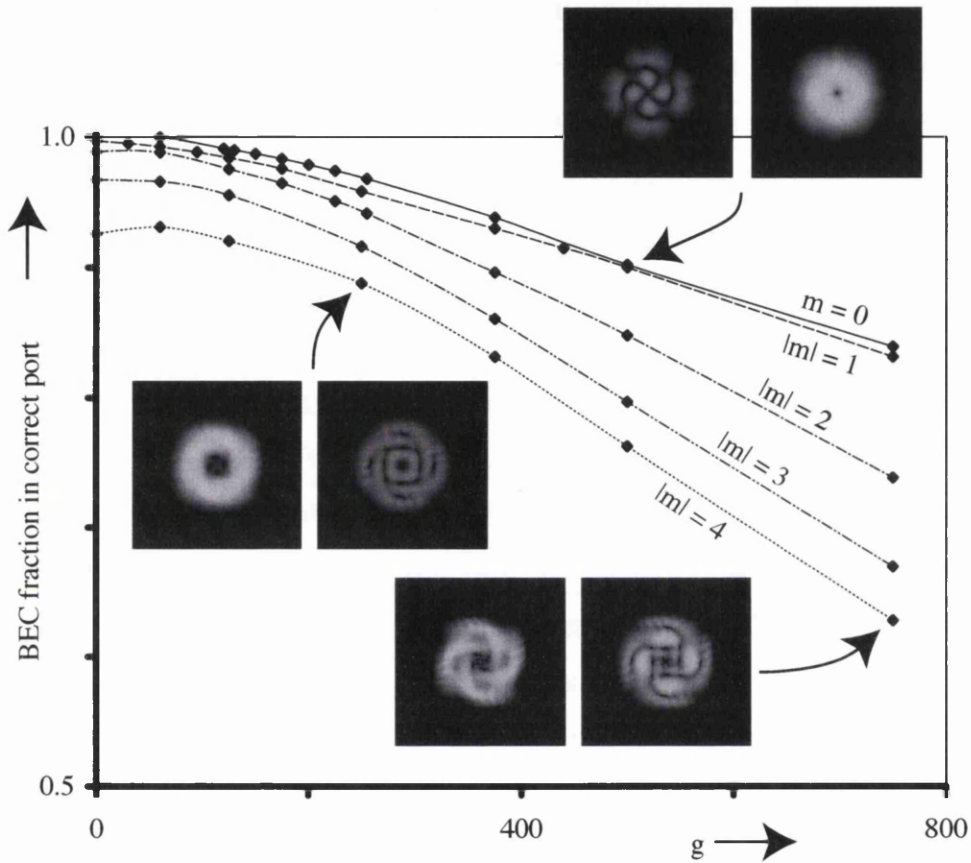


Figure 5.5: Fraction of the BEC in the correct port as a function of the non-linear coefficient,  $g$ , and for different values of the topological charge. Inset are the density cross-sections in the even (left) and odd (right) ports corresponding to some of the data points ( $m = 1, g = 500$  and  $m = 4, g = 250$  and  $750$ ).

3 dimensions), followed by mirroring in another axis, which is rotated with respect to the first axis by an angle  $\alpha$ , is equivalent to a rotation through an angle  $2\alpha$  about the intersection between the two axes. In analogy to mirroring of a light beam, which can be achieved with a pair of identical cylindrical lenses parallel to the mirror axis, each of focal length  $f$ , which are separated by  $2f$  (such a configuration is called a  $\pi$  mode converter<sup>25</sup>), a BEC can be mirrored by a pair of correctly separated cylindrical-lens pulses. These are far off-resonant light pulses which are similar to the lens pulses used in chapter 3 (see section 3.4 for details). These pulses have a transverse intensity distribution that is proportional to the thickness of the corresponding optical cylindrical lenses, that is the intensity falls off quadratically in one direction and is constant in the other. The effect of each cylindrical-lens pulse is a phase change proportional to the local intensity<sup>39,64</sup>: the cylindrical-lens pulses act like phase holograms of cylindrical lenses<sup>23</sup>. The phase change due to each lens pulse is  $r^2/(4t_f)$ , where  $r$  is the distance from the axis of the cylindrical-lens pulse and  $t_f$  is its focal time (the equivalent of the focal length in optical lenses). The chosen focal time for the simulations presented here was  $t_f = 0.03$  (in units of  $1/\omega$ ), which is one of the smallest focal times that satisfies the Nyquist criterion, and a time of  $t_d = 0.06$  between the lens pulses. Fig. 5.6 illustrates modelled examples of rotation of BECs through  $180^\circ$ .

This scheme does not work perfectly, not even in optics: a light beam (and, by analogy, a BEC with  $g = 0$ ) is mirrored perfectly only in the limit of cylindrical lenses with infinitely short focal lengths<sup>25</sup>. Obviously, this is not possible, and the result is imperfect mirroring that leads to asymmetry

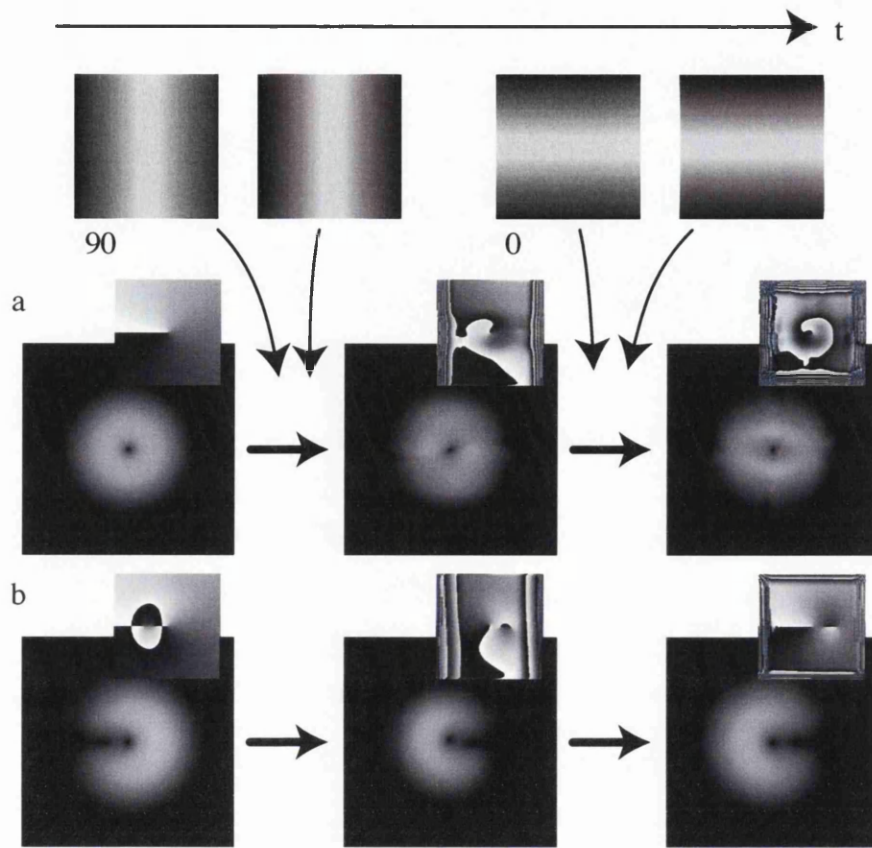


Figure 5.6: Rotating a BEC through  $180^\circ$  with a series of cylindrical-lens pulses. The intensity the pulses falls of quadratically and are shown at the top. The pictures show examples of simulated BECs before interaction with the light pulses (left), after interaction with the first pair of pulses (centre), and after interaction with the second pair (right). **a**: pure vortex with  $m = 1$ ,  $g = 500$ . The rotation can be seen more clearly in **b**, which starts off with a superposition of  $m = 1$  and  $m = 2$  vortices ( $g = 0$ ): after interaction with all pulses the probability density is indeed rotated through  $180^\circ$ . In both cases, the focal time of each cylindrical-lens pulse was  $t_f = 0.03$  (in units of  $1/\omega$ ); the pulses passed the BECs at times  $t = 0, 0.06, 0.06$  (the two middle pulses were merged into one), and  $0.12$ , respectively.

and vortex splitting. Another problem when using cylindrical-lens pulses to mirror BECs with  $g \neq 0$  is that the BEC can intermittently become focussed into a line, which greatly amplifies the non-linear effects, which in turn usually lowers the quality of the mirroring.

In the context of the vortex-sorting interferometer, it is only a differential rotation between the BECs in the two arms of the Bragg-pulse interferometer that is required. A better way of achieving such differential rotation is to apply the first two of the four rotation pulses shown in Fig. 5.6 to the BEC in one arm, and the other two to the BEC in the other arm; both BECs are mirrored, but with respect to different axes. As demonstrated above, this leads to good results for  $g \gtrsim 50$  for small values of  $|m|$ . Fig. 5.7 shows a detailed example of BECs that have been differentially rotated using twice mirroring in one arm on the one hand and mirroring in both arms on the other hand, and the resulting interference in an interferometer. The example shows a case in which the latter method works significantly better for  $g = 500$ .

### 5.5.1 Summary

In this chapter we have investigated the sorting of vortices in BECs using an interferometric technique. Our technique requires the BEC to be rotated, which we achieve with spatially inhomogeneous imprinted phases. If the non-linearity is strong, the efficiency of the rotation and therefore the efficiency of the sorting process is decreased, but with existing experimental techniques, such as exciting Feshbach resonances<sup>65</sup>, it should be possible to ‘tune’ the non-linear coefficient  $g$  to a value suitable for reliable vortex sorting.

The techniques presented here are based on methods used in conventional

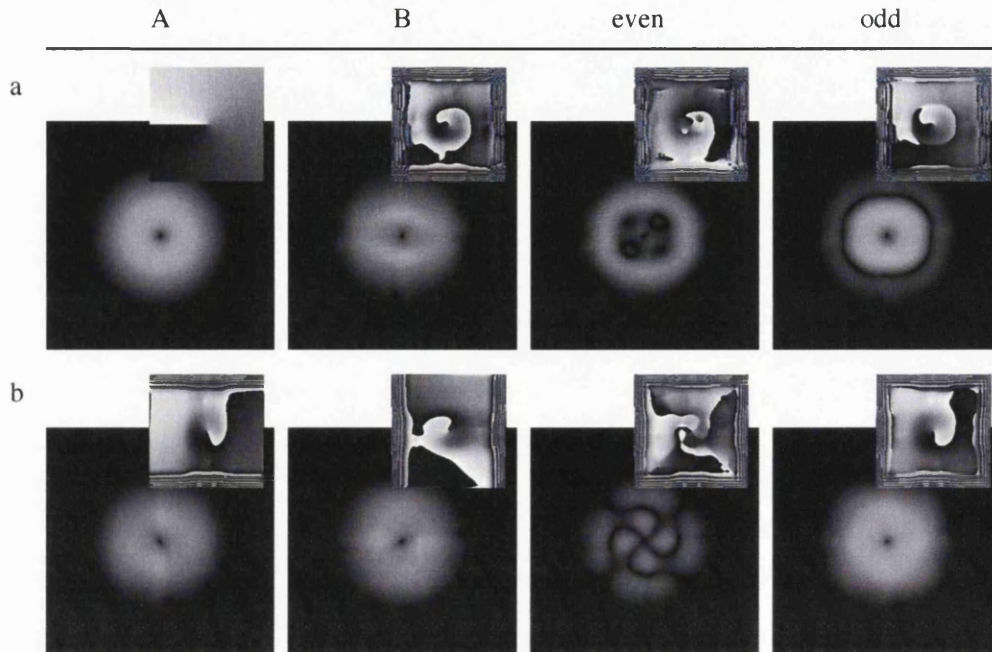


Figure 5.7: Simulation of a BEC vortex with topological charge  $m = 1$  in Bragg-pulse interferometers that (a) rotate the BEC in one arm through  $180^\circ$  and (b) flip the BEC in each arm, but about axes that are  $90^\circ$  rotated with respect to each other. From left to right, the four columns show the BECs in the first ( $A$ ) and second ( $B$ ) arm after rotation/flipping has taken place, and in the even and odd output ports. The figure is calculated for  $g = 500$ . Note that in both cases a sizable fraction of the BEC wrongly ends up in the even port (32.4% and 10.1%, respectively), but this fraction is much smaller in the interferometer that flips the BEC in both arms (compare Fig. 5.5).

optics. When these methods are transferred to BECs, complications arise, but also some intriguing new possibilities. In particular non-linearity – the origin of most complications – is important whenever information is not only to be stored, but also to be processed in computations.

## Part II

# Shaping light



# Chapter 6

## Hologram Design

This chapter describes several algorithms which can be used to generate holograms to shape a light beam. The need for such algorithms is discussed and the various schemes which can be used to shape a light beam.

The motivation for this investigation was based on adapting holography to Bose-Einstein condensates (BECs). This required an algorithm which would be capable of accurately incorporating the physical restrictions of a BEC. The investigation into the differing forms of holographic algorithms led me to become involved in the development of software for use in holographic optical tweezers. The optical tweezers require a light beam to be shaped into multiple spots and the spots to be manipulated in real time.

### 6.1 Introduction

Because the field in a single plane of monochromatic light can be used to describe the field in any other plane, controlling the field in one plane can give rise to a desired pattern in another plane. This gives rise to the possibility of creating a desired pattern of light in a plane which could not be modified directly by modifying the light in a more convenient plane. For any given target field, the corresponding field in an earlier plane can be calculated<sup>21</sup>.

---

If that field can be fully controlled, the problem is solved. However, full control over the field is rarely possible. When only a subset of modifications is possible, more complex algorithms have to be used.

### 6.1.1 Holography with restrictions

When trying to holographically shape a laser beam, it is important to take into consideration any physical restrictions present in the system before calculating the hologram. Because most hologram calculation algorithms find a hologram that produces an approximation to the target pattern, it is important that the shortcomings of the system are implemented in the algorithm to ensure the resulting hologram is the best compromise between the target pattern and the physical constraints.

The most common restrictions include using holograms which can only modulate the phase or the intensity of the light, the finite resolution of the hologram, the numerical aperture of the imaging system and the initial conditions. Many algorithms can incorporate these restrictions; however, some may then not converge to a stable solution, and with some it is not possible to implement all the restrictions which will alter the choice of algorithm.

### 6.1.2 Intensity holography

One of the simplest methods to shape a light beam into an arbitrary pattern is to place a mask with the desired pattern into the beam (and image that plane into the plane of interest). Acceptable results can often be obtained by creating a mask which can be partially transmitting. This system is restrictive in that the target image must be producible on a mask and the

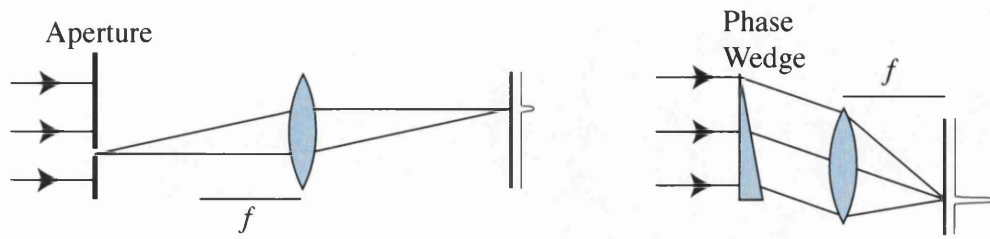


Figure 6.1: Schematic diagram of two optical setups to shape a uniform intensity laser beam into a single off axis spot. In the case of an intensity mask (left), a large portion of the beam is blocked by an aperture. This aperture is then imaged into the target plane to obtain a single spot. This can also be achieved by modifying the phase of the incoming beam (right). In this case a phase wedge is used, which changes the direction of propagation of the beam. Fourier transforming the plane of the wedge, a single off axis spot is created. The phase wedge method has the advantage that all the light from the original beam is present in the spot.

mask by its nature absorbs the light, which can result in unacceptable power loss.

### 6.1.3 Phase holography

It is also possible to consider holograms which only modulate the phase of a beam. This leads to almost lossless holography as the phase modulation does not alter the amplitude of the beam. Figure 6.1 schematically compares intensity mask shaping and phase holography. For our purposes, it is not the phase structure, but the pattern of the intensity in the target plane that

---

is of interest. In order to create an intensity structure from a phase-only hologram, the light has to be propagated from the plane of the hologram.

#### 6.1.4 Fourier phase holography

When considering where the hologram should be placed relative to the plane to be shaped, it is often desirable to be able to create patterns which are most dissimilar to the initial unmodified beam. It can be seen that the plane immediately in front of the hologram, regardless of the phase imprinted, will be very similar to the unmodified beam and as you move further from the hologram plane, the possibility for structure change becomes greater. By this process, the most dissimilar plane will be at infinity, i.e. the far field. The far field can be imaged into the back focal plane of a lens. If the hologram is placed in the front focal plane, the fields in the two planes are then Fourier transform pairs.

Phase holography can be achieved by locally altering the optical path length seen by the beam in proportion to the phase pattern. Modern liquid crystal technology has allowed the use of interactive holograms in the form of spatial light modulators (SLMs), in which the pattern can be changed at video frame rates. This has seen a rediscovery of many phase-holography techniques in a wide range of applications<sup>66</sup>.

Due to the imperfect nature of the hologram devices, the diffraction efficiency is rarely 100%. This results in a fraction of the light having a wrong phase shift. The resulting phase shift is an integer multiple of the original pattern which creates multiple diffraction orders, and so the beam leaving the hologram is a superposition of these orders. Unfortunately all these or-

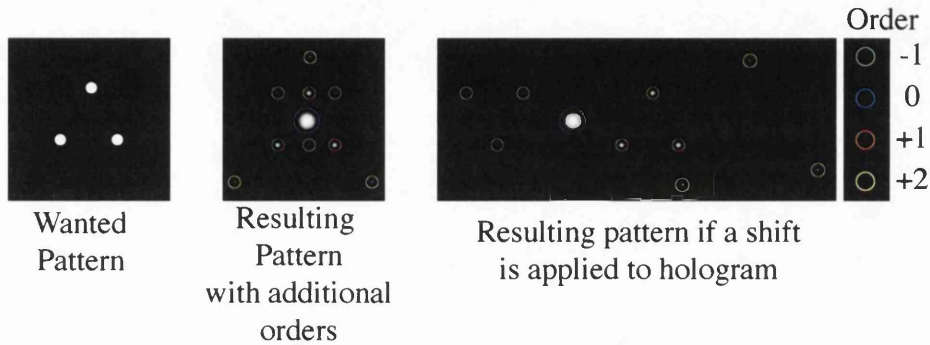


Figure 6.2: Representation of the resulting patterns of an imperfect hologram. If the phase modulation is most exactly matched to the wavelength of light, i.e. imprinting a  $\pi$  phase shift results in a  $0.9\pi$  shift on the light, the power transferred into the desired pattern is lower and this power is moved into other diffractive orders. If the pattern is used 'as calculated', the center of all the orders are coincident which can result in degradation of the pattern. Shown on the left is the desired triangular pattern of spots. The centre image demonstrates the overlapping of the different orders. If an additional 'shift' grating is applied, the diffractive orders have a transverse momentum proportional to the order number as shown on the right. If the momentum is large enough, the first order which contains the wanted pattern can be isolated from the other orders, which can be filtered out with a spatial filter.

---

ders normally travel along the same beam path and thus overlap. Figure 6.2 shows the superposition of the wanted intensity pattern (designated the first or +1 order) with the unmodulated beam (zeroth order), and several other unwanted orders. In order to separate these beams, an additional grating is often applied. This results in each diffracted order having a different angular direction, making it easier to separate them and filter out the unwanted beams as can be seen in the right of Figure 6.2.

## 6.2 Superposition approximation algorithm

If the pattern required is relatively simple, for example a series of discrete spots, the hologram to generate the pattern can be created by combining the holograms which produce the separate elements<sup>67,68</sup>. The hologram for each spot can be calculated as the sum of a linear phase wedge in the  $x$ -direction, a similar wedge in the  $y$ -direction, and the phase hologram of a lens. The two wedges result in the beam being shifted in proportion to the sine of the angle of the wedge; the lens can be seen to shift the Fourier plane corresponding to the field without the lens hologram and thus shifting the position of the spot in the  $z$ -direction.

In order to calculate the hologram for multiple spots, each of the holograms must be converted into a field by applying the hologram to a uniform field. These fields can then be added which results in the field in the hologram plane which will produce the spots in the target plane. However, as we concentrate on phase holography here and therefore the amplitude of the field cannot be changed, the amplitude of the superposition is ignored and

---

the phase of the calculated field becomes the hologram. This method has the advantage that it is very fast due to the individual holograms being easy to calculate and the non-iterative nature of the algorithm.

This method does not produce perfect results as the amplitude information is discarded and no prior information about the actual amplitude of the initial beam is used. Another shortcoming of this method is that the relationship between each element is static, and the pattern can often be improved if the phase between the spots can be altered iteratively. The first iteration of the Gerchberg-Saxton algorithm (see Section 6.4) calculates the same result, which it then refines further. Often it is possible to use this algorithm to seed one of the other more complicated iterative algorithms with an approximate answer which can be refined, thus combining its speed with the generality of more complex algorithms.

### 6.3 Additional phase structure

An interesting adaptation of the superposition approximation algorithm is the post-addition of the resulting hologram with an extra phase feature, for example, a phase vortex as shown in figure 6.3. This results in the final pattern in the Fourier plane being transformed by the Fourier transform of the additional phase feature. This is very useful if all the target structures have a common Fourier phase structure. For example, if all the spots are at the same  $y$  position, the holograms to offset the spots in the  $x$  direction can be calculated and combined (with complex addition) and then finally added to the  $y$  offset hologram. This method can speed up the calculations,

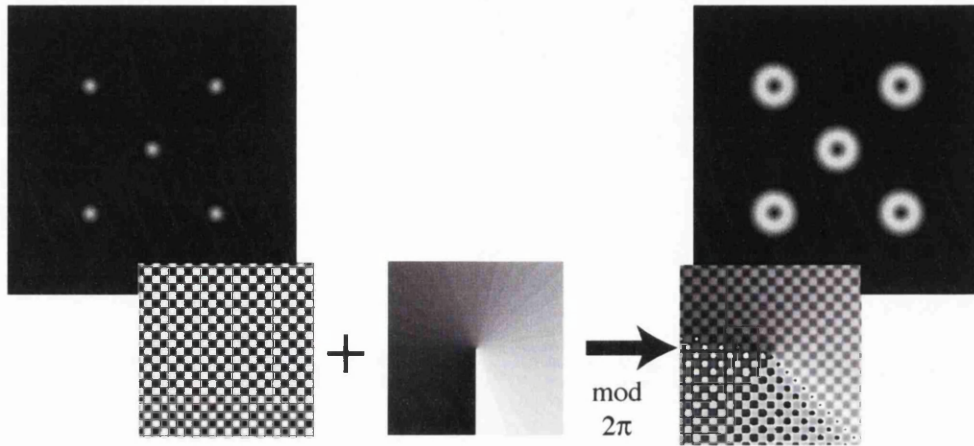


Figure 6.3: Example of including additional phase structures to obtain an array of vortices. If the wanted pattern all have the same phase in the hologram plane, the pattern can be created by calculating the hologram to create single spots in the desired locations and adding the phase of a single element of the pattern. This results in each of the spots obtaining the phase pattern. The hologram to produce the array of five spots (shown in the leftmost inset) is added to the phase of a vortex (middle inset) to produce a hologram which creates five vortices in the Fourier plane.



however it is limited to cases where the spots have something in common. The Curtis-Koss-Grier algorithm (see Section 6.5.2) uses a similar method to allow the shaping of individual spots into different structures.

## 6.4 Gerchberg-Saxton algorithm

The Gerchberg-Saxton (GS) algorithm<sup>69</sup> is an iterative method originally developed for recovering the phase of an electron or light beam from its intensity distributions in two transverse planes. It can also be applied to shape a light beam, specifically to calculate the phase pattern which light in one plane would require to form an approximation to any desired intensity pattern in a second plane. As the phase pattern required in the first plane can be imprinted onto the light beam with a phase hologram, this allows two-dimensional (2D) holographic light shaping. If the requirements in the two planes cannot be met simultaneously within what is allowed by the laws of diffraction, the GS algorithm finds a useful compromise to reconcile these conflicting requirements<sup>70</sup>. For numerical simplicity, the second plane is usually chosen to be the far field of the first plane; mathematically, the fields in the two planes are then Fourier transforms of each other. Here we restrict ourselves to this case.

We write the intensity distribution of the unshaped light beam in the first plane – the hologram plane – as  $I_H$  and the target intensity distribution in the Fourier plane as  $I_T$ . Both  $I_H$  and  $I_T$  are functions of  $x$  and  $y$ , which are stored in the computer as 2D arrays of real-valued numbers. The light fields in the two planes are represented by 2D arrays of complex numbers,  $u_n^H$  in

Solutions which fulfil  
restriction 1

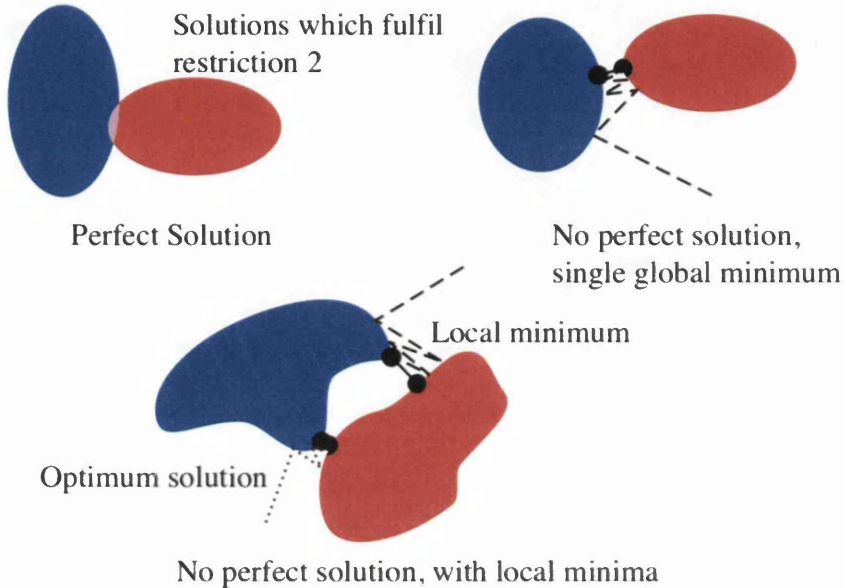


Figure 6.4: Diagram illustrating the projection onto convex sets algorithm.

Within the phase space of all possible solutions, the sets of solutions which fulfil the various restrictions are subsets of that space. If as is shown in the top left figure, the two sets overlap, there are perfect solutions which are common. However in many cases these sets do not overlap and the goal of the algorithm is to find the optimal solution which is given by the shortest distance between the two sets. During an iteration, the algorithm forces the solution onto one of the sets. This places the next solution onto the set at the closest point to the original. As the algorithm progresses, it alternatively projects the solution onto each set of solutions until the error no longer decreases. Because the projection find the closest point in the other set, the error can never increase. If the subset of solutions is not convex (bottom), the algorithm, depending on its initial conditions, may converge to a local minimum and not find the optimal solution.

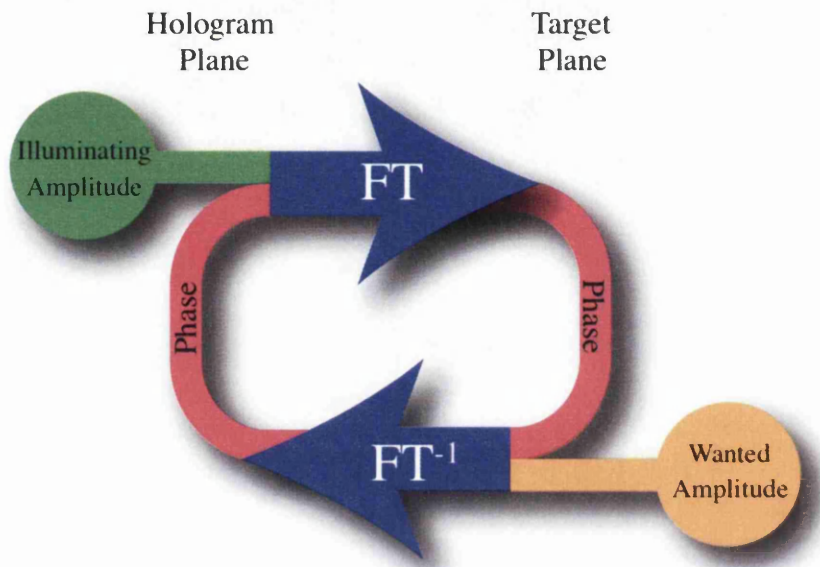


Figure 6.5: Representation of an iteration of the Gerchberg-Saxton algorithm. The algorithm begins in the plane of the hologram (top-left) where the illuminating amplitude is combined with the phase of the previous iteration and then is Fourier transformed into the target plane. The resulting phase of the Fourier transform is combined with the amplitude of the desired field and inverse-Fourier transformed into the hologram plane where the next iteration begins.

the hologram plane and  $u_n^T$  in the Fourier plane; the subscript  $n$  indicates the iteration number. The initial phase distribution in the hologram plane,  $\varphi_0^H$ , is set to any arbitrary distribution, often uniform, i.e.

$$\varphi_0^H \equiv 0. \quad (6.1)$$

One iteration of the algorithm, which calculates an improved phase distribution in the hologram plane,  $\varphi_n^H$ , from the previously calculated phase distribution,  $\varphi_{n-1}^H$ , then progresses as follows:

$$u_n^H = \sqrt{I_H} \exp(i\varphi_{n-1}^H). \quad (6.2)$$

$$\varphi_n^T = \arg(\text{FFT}(u_n^H)) \quad (6.3)$$

$$u_n^T = \sqrt{I_T} \exp(i\varphi_n^T) \quad (6.4)$$

$$\varphi_n^H = \arg(\text{FFT}^{-1}(u_n^T)) \quad (6.5)$$

Equation (6.2) calculates the initial field in the hologram plane, from which equation (6.3) then calculates the phase distribution in the target plane,  $\varphi_n^T$ . It uses a Fast Fourier transform (FFT), or more generally any discrete Fourier transform. Equation (6.4) combines this phase distribution in the target plane with the target intensity,  $I_T$ , giving the field  $u_n^T$ , from which equation (6.5) then calculates the corresponding phase distribution in the hologram plane,  $\varphi_n^H$ . Over a number of iterations, the actual intensity in the target plane,  $I_n = |\text{FFT}(u_n^H)|^2$ , converges to an almost diffraction-limited approximation of the desired intensity there,  $I_T$ ;  $\varphi_n^H$  is the corresponding phase-hologram pattern needed to produce this field in the target plane.

## 6.5 Modified Gerchberg-Saxton algorithm

The Gerchberg-Saxton algorithm as presented above is a very powerful and general algorithm. It is one example of the class of algorithms termed projection onto convex sets. These algorithms assume that the sets of answers which fulfil the restrictions are convex, however if these sets are not convex (as in the Gerchberg-Saxton case) the algorithms may not find the globally optimum solution as shown in figure 6.4.

However by further modifying how the restrictions are applied, the effect of local minima can be suppressed and thus better results can be obtained. The main method of improvement is to incorporate a priori knowledge into the algorithm to help eliminate local minima in the cost function which correspond to sub-optimal solutions and which the algorithm might find instead of the optimal solution.

### 6.5.1 Adaptive Additive algorithm

The adaptive additive algorithm can be seen as a generalised Gerchberg-Saxton algorithm which uses both the phase and amplitude of the field in the wanted space. Instead of replacing the amplitude, as the Gerchberg-Saxton algorithm does, the adaptive additive algorithm mixes the wanted pattern with the result of the Fourier transform, and Equation 6.4 becomes

$$u_n^T = (\alpha\sqrt{I_T} + (1 - \alpha)|\text{FFT}(u_n^H)|) \exp(i\varphi_n^T) \quad (6.6)$$

where  $\alpha$  is the chosen mixing parameter. It can easily be seen that for  $\alpha = 1$  the algorithm becomes the original Gerchberg-Saxton algorithm.

### 6.5.2 Curtis-Koss-Grier algorithm

The Curtis-Koss-Grier algorithm<sup>71</sup> is another modification of the Gerchberg-Saxton algorithm. The first change to the Gerchberg-Saxton algorithm is in the mixing of the wanted amplitude with the result of the Fourier transform, in a similar manner to the adaptive additive algorithm and Equation 6.4 becomes

$$u_n^T = \sqrt{I_T} \left[ (1 - \alpha) + \alpha \frac{\sqrt{I_T}}{|\text{FFT}(u_n^H)|} \right] \exp(i\varphi_n^T) \quad (6.7)$$

where  $\alpha$  is the chosen mixing parameter. It can easily be seen that for  $\alpha = 0$  the algorithm becomes the original Gerchberg-Saxton algorithm.

The second modification is a generalised Fourier transform which includes an additional spot-specific kernel term (and its associated inverse transform) which allows additional phase modifications to be applied to each spot. This method is similar in to the convolution method but is applied to each spot separately.

The generalised Fourier transform allows the structure of each spot to be altered independently while the mixing which takes place allows the algorithm to converge quickly.

### 6.5.3 3D Gerchberg-Saxton

The fields in the two planes in the GS algorithm can also be seen as a field in a real-space plane  $(x, y)$  and its  $k$ -space  $(k_x, k_y)$  representation. This allowed Shabtay<sup>72</sup> to adapt the GS algorithm to 3D: instead of dealing with fields in two planes, the algorithm deals with fields in two volumes. One field is a 3D real-space representation of the part of the beam that is to be shaped,

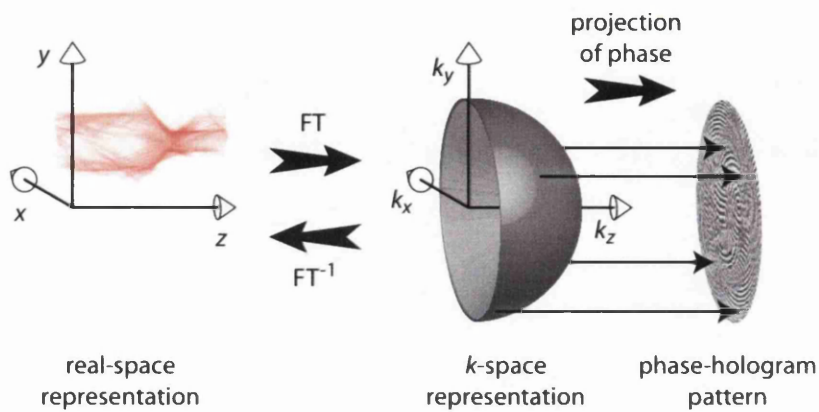


Figure 6.6: Schematic representation of a monochromatic light beam in real space (left) and  $k$  space (right). In  $k$  space, a monochromatic light beam is restricted to the surface of a hemisphere of radius  $k_0 = 2\pi/\lambda$ . A projection of the phase on this hemisphere into a plane (see section 7.3.1) provides a phase-hologram pattern suitable for beam shaping.

the other is its 3D  $k$ -space representation, which needs to be consistent with the beam's wavenumber spectrum. The two fields are 3D Fourier-transform pairs (Figure 6.6). Chapter 7 uses this to demonstrate the algorithm's ability to calculate holograms to shape 3D light fields. Note that the 2D Gerchberg-Saxton algorithm usually shapes light in the Fourier plane, while the real-space plane contains restrictions such as the intensity profile of the unshaped beam, whereas the 3D GS algorithm shapes light in the real-space volume while the Fourier volume contains the physical constraints.

As discussed in the introduction, the field in a single plane completely describes a monochromatic light beam. This is reflected by the fact that the

$k$ -space representation of such a beam is confined to a surface. In monochromatic light of wavelength  $\lambda$ , the wave-vector component in the direction of propagation,  $k_z$ , is related to the values of  $k_x$  and  $k_y$  through the equation

$$k_0^2 = k_x^2 + k_y^2 + k_z^2, \quad (6.8)$$

where  $k_0 = 2\pi/\lambda$ . This forms a sphere of radius  $k_0$  in 3D  $k$  space – an Ewald sphere<sup>72</sup>. A single laser light beam has a direction of propagation in the positive  $z$  direction only, hence the wave vectors are limited to the half of the sphere with  $k_z > 0$  (figure 6.6).

## 6.6 Direct Binary Search

One of the most general methods for calculating holograms is the direct binary search<sup>73</sup> (also known as random binary search) which is a brute-force method for calculating a hologram which will produce an approximation to the wanted field in another plane. If it is known how each pixel of the hologram affects the field in the plane of interest, it is possible to alter each pixel randomly and to determine if that change resulted in a better approximation to the wanted pattern or not. Based on this result, the change to the hologram can be kept or disregarded. If only the good changes are kept, the pattern converges to an approximation to the wanted pattern.

This method has the advantage that the calculation of the error can be chosen depending on the desire of the user. This allows, for example, more important areas of the wanted pattern to be weighted higher than others and for phase information to be included. This added flexibility comes at the cost of speed. Since every iteration only affects one pixel, which may not



---

actually change, it takes many iterations before the error reduces significantly. Because of these issues, a direct binary search is often combined with a faster but less flexible algorithm which first calculates a rough approximation to the desired pattern and then the direct binary search tries to improve it further.

## 6.7 Simulated Annealing

Unfortunately the direct binary search suffers from local-minima problems even if the algorithm is allowed to run infinitely long. The result is the best that can be achieved when compared to all the possibilities which differ by a single pixel, but it is not necessarily the optimum result. One method for trying to overcome this problem is simulated annealing<sup>74</sup>, which keeps some changes which increase the error in the hope of avoiding local minima. Simulated annealing takes its inspiration from the behaviour of atoms when crystallising during annealing. During annealing the temperature is slowly lowered, such that at any time the system can be considered in thermal equilibrium. As the temperature is lowered, the thermal energy available to each atom is reduced, thus restricting some of the possible transitions between potential wells. However, by reducing the temperature at the correct rate, atoms can still move to occupy lower potential states before the thermal energy is reduced. By analogy, any system can use a similar method to try to avoid local minima. In place of the temperature an allowable error variable can be used, which is slowly lowered throughout the calculation. This allows some errors to be incorporated into the solution with smaller and smaller errors being allowed as the calculation progresses until at the equivalent of

$T = 0$ , the algorithm performs as the standard underlying algorithm.

By incorporating simulated annealing, the convergence rate of the algorithm will reduce. It also adds the additional complication of the cooling rate parameter, which has to be optimised for each application individually.

## 6.8 Summary

In order to shape a beam, a hologram capable to approximating the desired pattern has to be calculated. The imperfect nature of applying holograms results in the resulting structure deviating from the desired pattern. In many cases this can be improved at the cost of computational power. Iterative algorithms can be used to find closer approximations by adapting the pattern to accommodate the physical restrictions present in the system.

When choosing an algorithm to calculate a hologram pattern, there is generally a compromise between speed of execution and efficiency. The use of the system will often limit the choice of algorithm, for example the superposition approximation algorithm can only be used to calculate arrangements of discrete spots, while (with 2006 technology) the iterative algorithms can not calculate high-resolution holograms at a real-time frame-rate.

---

## Chapter 7

# Three-dimensional light shaping

When trying to compare Bose-Einstein condensates and light, the question of dimensionality often arose. Monochromatic laser light travelling in the positive  $z$  direction say, is in many ways two dimensional: knowledge of such a beam in one plane allows the entire beam to be reconstructed. A three-dimensional representation therefore does not add any information, while Bose-Einstein condensates are three-dimensional and cannot occur in a ideal two-dimensional system<sup>75</sup>. These restrictions have resulted in most of the comparisons being done in two dimensions and assuming the BEC is tightly confined in one dimension, leading to a flat ‘pancake’ condensate. However the discussions on 3D light gave rise to interesting ideas about the nature of monochromatic light fields in 3D and the ability to shape a beam into arbitrary 3D patterns. The outcome of these discussions was the algorithm outlined in Section 6.5.3. Unfortunately, this same algorithm was independantly discovered<sup>72</sup> as a method of establishing what 3D intensity shapes monochromatic light beams can take. We have extended this into a hologram-calculation algorithm, which we demonstrated numerically and experimentally.

This chapter also describes some of the details which must be considered when trying to implement the 3D Gerchberg-Saxton algorithm which may be ignored when considering purely the algorithm. Section 7.1 gives a brief introduction to the 3D nature of monochromatic light fields before discussing the details of our implementation of the algorithm in Section 7.2. Finally the experimental setup is described and the experimental results are presented in Section 7.3.2.

My role in this work was to design and implement the bespoke software to implement the algorithm. I was involved in the discussions on the nature of light in 3 dimensions and how best to represent that in a simulation. I carried out the experimental side of the project single-handedly, including the setting up of the experiment and all result taking. I was heavily involved in writing and drafting the paper of this work<sup>76</sup> and was lucky enough to present it at several conferences.

## 7.1 Introduction

Any monochromatic light beam propagating in free space is a three-dimensional (3D) field and its intensity distribution forms 3D patterns. At the same time, the field cross-section in any one transverse plane completely determines the field everywhere else. This dependency of the entire beam on its cross-section in a single plane forms the basis of beam propagation methods (for example Ref.<sup>21</sup>), but it also restricts the possible 3D shapes a monochromatic light beam's intensity can take.

Chapter 6 discusses many methods of shaping a monochromatic light

beam in 2D, and shaping light beams into non-trivial but restricted 3D shapes. Other methods of creating non-trivial 3D patterns of light have been created by utilising phenomena such as the Talbot effect<sup>77</sup> or spiral-type beams<sup>78</sup>. It is also possible to shape a light beam into 3D configurations of bright spots<sup>68</sup>, whereby the spots can be shaped individually<sup>71</sup>; these techniques are important in the field of holographic optical tweezers. 3D beam shaping into arbitrary shapes has been demonstrated using computationally intensive direct search methods<sup>173</sup>, but continues to be a challenge.

The recent algorithm by Shabtay<sup>72</sup> finds beams whose shape is an approximation within physical limits to any arbitrary 3D target shape. We apply this algorithm to calculate examples of light beams shaped in 3D. From the beams' Fourier-space representations we calculate phase-only hologram patterns that allow us to create the beams experimentally.

As discussed previously, the field in a single plane completely describes a monochromatic light beam. This is reflected by the fact that the  $k$ -space representation of such a beam is confined to a surface. In monochromatic light of wavelength  $\lambda$ , the wave-vector component in the direction of propagation,  $k_z$ , is related to the values of  $k_x$  and  $k_y$  through the equation

$$k_0^2 = k_x^2 + k_y^2 + k_z^2, \quad (7.1)$$

where  $k_0 = 2\pi/\lambda$ . This forms a sphere of radius  $k_0$  in 3D  $k$  space – an Ewald sphere<sup>72</sup>. We consider here light moving in the positive  $z$  direction only, hence the wave vectors are limited to the half of the sphere with  $k_z > 0$

---

<sup>1</sup>Direct search methods allow shaping not only of the light intensity, to which we are restricted here, but also of the phase. They are therefore more versatile than the technique we discuss here.

(figure 6.6). As shown previously, using this 3D  $k$ -space representation of the light beam within a Gerchberg-Saxton algorithm allows the algorithm to find an approximation to the wanted 3D intensity structure.

Figure 7.1 shows examples of light shaped using this program. The algorithm can clearly produce intensity distributions in which the target shapes can be recognized. The resulting shapes are not exactly the desired shapes, reflecting the limitations due to the  $k$ -space restrictions (discussed in more detail in Section 7.2) and the algorithm.

## 7.2 Details of our implementation of the 3D Gerchberg-Saxton algorithm

In this section we explore some of the details of the implementation of the algorithm. We use the error<sup>79</sup>

$$\epsilon_n = \sqrt{\frac{\sum_{x,y,z} |\sqrt{I_n} - \sqrt{I_T}|^2}{\sum_{x,y,z} I_T}} \quad (7.2)$$

to quantify the similarity between the target intensity distribution,  $I_T(x, y, z)$ , and the intensity distribution the algorithm produces after  $n$  iterations,  $I_n(x, y, z)$ .

### 7.2.1 Thickness of $k$ -space sphere

A purely monochromatic beam restricts the allowed  $k$ -space elements to a spherical surface. In our numerical simulations 3D  $k$  space is discretised, and it is not immediately obvious how to represent the spherical surface that corresponds to a monochromatic light beam in this discrete space.

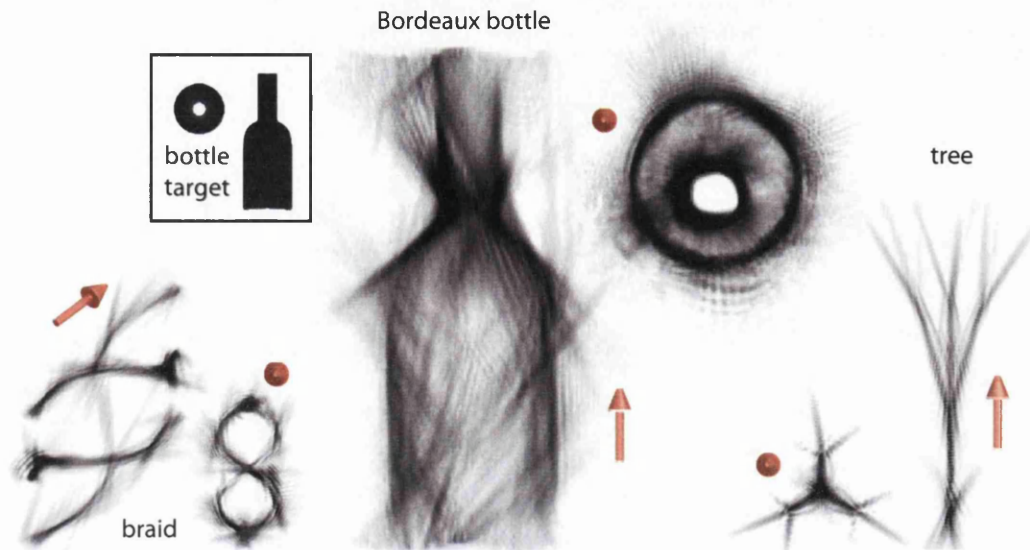


Figure 7.1: Examples of 3D light intensities shaped by the 3D Gerchberg-Saxton algorithm. High intensities are shown dark. Each light beam is shown from two directions; the propagation direction of the beam is indicated by a red arrow (the three smaller views look into the beam). In the case of the Bordeaux-bottle beam, two corresponding views of the target intensity distribution are also shown. The beams were calculated for a  $k$ -space cone angle  $\alpha = 90^\circ$  (see figure 7.4).

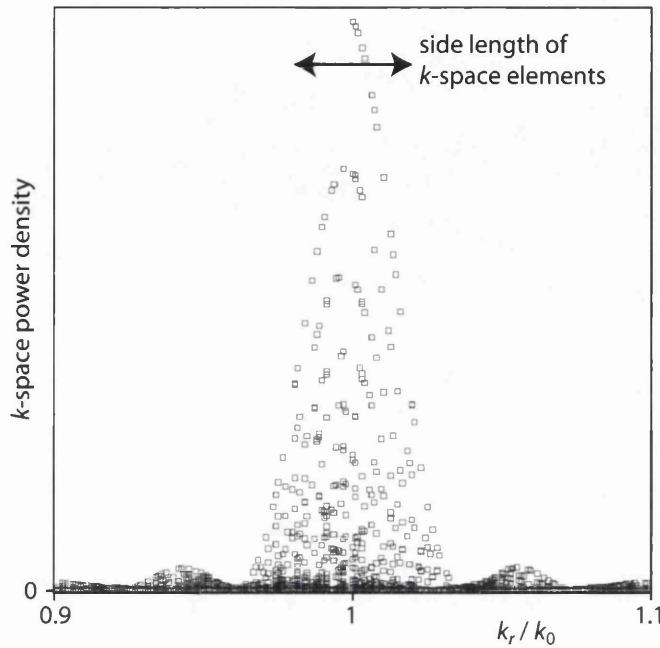


Figure 7.2: Example of the representation of a monochromatic light beam in discrete  $k$  space. The plot shows the powers in all discrete- $k$ -space elements with  $0.9k_0 \leq k_r \leq 1.1k_0$  (where  $k_0 = 2\pi/\lambda$ ) as a function of the radial wave number,  $k_r$ , calculated for a tightly focussed monochromatic Gaussian light beam. The discrete  $k$ -space representation was found by numerically calculating the light beam on a  $64 \times 64 \times 64$  grid representing a cube of side length  $25\lambda$  (where  $\lambda$  is the wavelength of the light), which was then 3D-Fourier-transformed. Whereas in continuous  $k$  space all the power would be at  $k_r = k_0$ , in discrete  $k$  space the power is distributed within a few  $k$ -space-element widths of  $k_0$ . Because the edges of the represented cube act like hard-edged slit apertures in the  $x$ ,  $y$  and  $z$  directions, the  $k$ -space distribution is that of continuous  $k$ -space, convolved with  $\text{sinc}^2$  functions in the  $k_x$ ,  $k_y$  and  $k_z$  directions (it is, of course, also discrete).



In an effort to understand this better we investigate here one aspect associated with the discrete nature of a light beam whose 3D light field and  $k$ -space distribution are represented on cubic grids of points. Figure 7.2 shows the radial cross section of the discrete- $k$ -space representation of a simulated monochromatic light beam. This  $k$ -space representation was calculated by taking the three-dimensional discrete Fourier transform of the beam's spatial representation over a discrete 3D volume. The spatial representation was calculated using a standard beam-propagation algorithm<sup>2</sup>, starting with the field cross-section of a Gaussian beam close to the beam waist and propagating it into 63 further, equally-spaced, planes. In addition to a peak centred at  $k_0 = 2\pi/\lambda$ , the radial profile of the resulting discrete- $k$ -space-hemisphere distributions has other distinct features. We believe these additional features to be due to the beam being represented only in a cube and the represented beam therefore having a top-hat profile in the  $x$ ,  $y$  and  $z$  directions, which in turn leads to the  $k$ -space distribution being widened, more precisely being convolved with a sinc function in the  $k_x$ ,  $k_y$  and  $k_z$  directions.

In our simulations we use perhaps the simplest form of the  $k$ -space hemisphere: for each represented pair of  $k_x$  and  $k_y$  values, the power in the discrete  $k_z$  value closest to  $(k_0^2 - k_x^2 - k_y^2)^{1/2}$  is set to one, that of all the other  $k_z$  values to zero. Such a  $k$ -space hemisphere is exactly one element thick in the  $k_z$  direction; the radial power profile would be a top hat of width 1 element. This is clearly different from the arguably 'natural' representation shown in

---

<sup>2</sup>Instead of using the quadratic approximation used in ref.<sup>21</sup>, we use the exact relationship, equation (6.8), to calculate  $k_z$  from  $k_x$  and  $k_y$ . We also use an absorbing boundary<sup>80</sup> (which is later removed) to avoid effects due to the periodic boundary conditions implicit in the discrete-Fourier-transform-based beam-propagation algorithm.

figure 7.2, but it works well enough to produce good experimental results (see below) and simplifies the extraction of phase-hologram patterns (see section 7.3.1).

It is worth discussing very briefly the case of thicker hemispheres. Figure 7.3 shows results from the 3D GS algorithm with a  $k$ -space hemisphere with a Gaussian radial power profile of variable width. It can be seen that an increase in the thickness of the hemisphere leads to lower errors. The reason for this is that a thicker  $k$ -space hemisphere implies more non-zero  $k$ -space elements whose phase the algorithm can alter. Physically, a  $k$ -space hemisphere of non-zero thickness corresponds to polychromatic light<sup>3</sup>; the represented field is a snapshot of the light field at one particular instant. The time evolution of the instantaneous field in coherent polychromatic light can be controlled by an extended GS algorithm that incorporates a time dimension added to the real-space representation of the field and a frequency dimension added to the  $k$ -space representation. This can be seen as an extension of the shaping of the time-resolved field of short pulses, which has previously been demonstrated experimentally<sup>81</sup>.

### 7.2.2 Influence of numerical aperture

Any experimenter realising 3D light shaping at some point has to make an implicit or explicit choice about the range of directions of the plane-wave components that superpose in the target volume. This is a choice of the numerical aperture (NA) of the system and it depends on the solid angle

---

<sup>3</sup>Note that the  $k$ -space components with  $k_r \neq k_0$  are not evanescent waves as they correspond to purely real values of  $k_z$ .

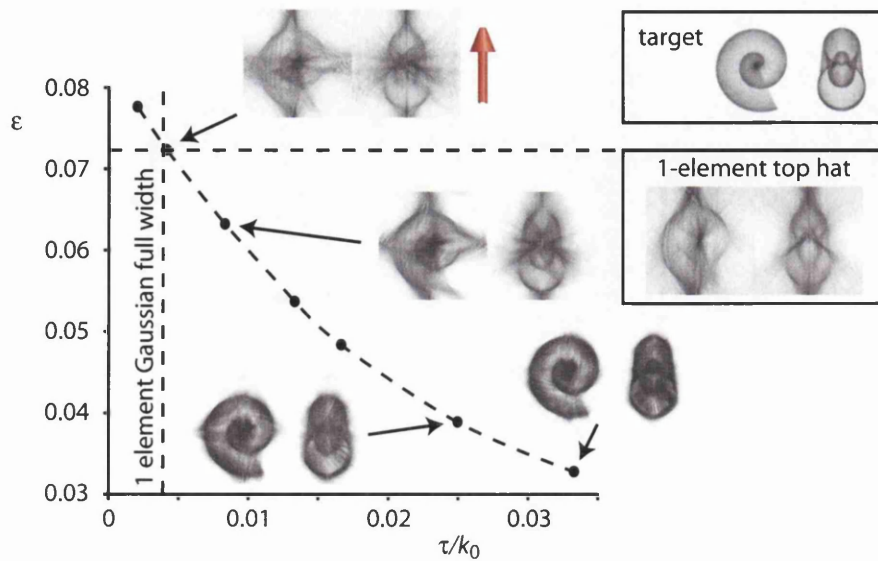


Figure 7.3: Effect of the thickness of the  $k$ -space hemisphere. The points show the error,  $\epsilon$ , corresponding to light beams shaped with the 3D GS algorithm using  $k$ -space hemispheres of various Gaussian half-thicknesses,  $\tau$ . More precisely, the power in each  $k$ -space element with  $k_z > 0$  is set according to its distance from the origin according to the equation  $p(k) = \exp(-(k - k_0)^2/(2\tau^2))$ , where  $k_0 = 2\pi/\lambda$  is the radius of the hemisphere; the power of  $k$ -space elements with  $k_z \leq 0$  is set to zero. The light beams corresponding to some of the points are also shown, each from two different perspectives; the propagation direction (red arrow) is always in the vertical direction. Also shown are the target shape (top inset on the right) and the light beam resulting from a single-element top-hat radial profile (lower inset on the right), which has an error comparable to a  $\tau = 0.0041$ , which for the grid size used ( $128 \times 128 \times 128$ ) corresponds to a Gaussian full width of approximately 1 element.

spanned by the directions of light rays reaching the target volume. Usually the light-ray directions form a cone of angle  $\alpha$  (figure 7.4a), which in the simplest case is determined by the size of the aperture of the last optical component and the distance between that component and the space where the beam is to be shaped. As each  $k$ -space point corresponds to parallel light rays with a specific direction, which is given by the gradient of the phase, such a cone of light-ray directions restricts non-zero  $k$ -space values to the section of the  $k$ -space hemisphere that lies within an angle  $\alpha/2$  of the  $k_z$  axis (figure 7.4b).

Figure 7.4c demonstrates the effect of varying the angular size of the  $k$ -space-sphere segment on the resulting intensity distributions. It can be seen that a larger cone angle allows the possibility of generating a 3D light intensity that resembles the target intensity more closely. The reason for this is that the larger range of values of  $k_x$ ,  $k_y$  and  $k_z$  in such a beam widens the range of structure sizes that can be present in the beam.

### 7.2.3 Simulation parameters and procedure

Our particular implementation of the 3D GS algorithm, running on a dual-processor 2.5 GHz Pentium Xeon desktop computer, takes several days to converge using a resolution of  $256 \times 256 \times 256$ . For this reason we initially test the generation of new shapes using a lower resolution of  $64 \times 64 \times 64$ . In addition to each iteration taking less time at this lower resolution, the algorithm requires fewer iterations to converge.

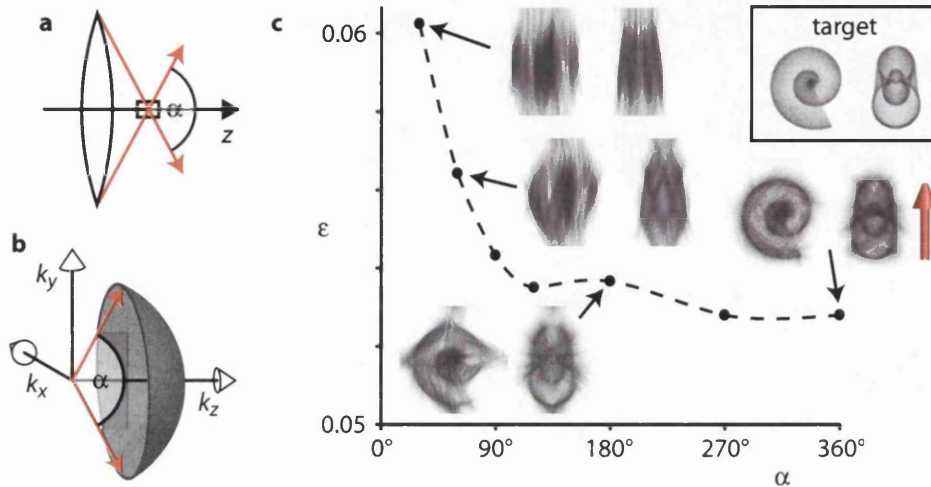


Figure 7.4: Geometry of the cone of light-ray directions and its effect on 3D light shaping. The cone of light angles is determined by the geometry of the setup, in the simplest case the aperture radius of the last lens in the system and its distance from the position where the light is to be shaped (a). The cone angle,  $\alpha$ , is also the angle of the  $k$ -space-sphere segment (b). The larger the cone angle, the smaller the error  $\epsilon$  between the desired and calculated intensity structure (c). The errors in c were calculated for the example of a shell-shaped target intensity, shown in the box in the top right corner. Also shown are the calculated intensity structures corresponding to some of the data points.

### 7.3 Experiment

We have created examples of light beams with a shaped 3D intensity structure experimentally, specifically some of the beams shown in figure 7.1. The experimental setup is outlined in figure 7.5a. It creates the shaped beam around the Fourier plane of a phase hologram, which was calculated from the  $k$ -space representation of the beam found by the 3D GS algorithm (see section 7.3.1).

Our hologram patterns were calculated for beams whose  $k$ -space representations have a relatively large cone angle of  $\alpha = 90^\circ$ . With our phase hologram, which has a height of approximately 20mm, this cone angle could be realised by using a Fourier lens with a very short focal length of approximately 10mm; this would create the shaped beam in a volume of approximate size  $0.1\text{mm} \times 0.1\text{mm} \times 0.1\text{mm}$ . However, in our experiment we use a Fourier lens with a much longer focal length of  $f_1 = 600\text{mm}$ , which corresponds to a significantly smaller cone angle of  $\alpha \approx 2^\circ$ . Our setup can be seen as a combination of a  $f = 10\text{mm}$  Fourier lens that shapes the intensity in a cubic target volume of side length  $\approx 0.1\text{mm}$ , and two more lenses that image this target volume: a lens with  $f = -10\text{mm}$  in the same plane as the 10mm Fourier lens, and the 600mm Fourier lens we actually use. Because of the imaging characteristics of this lens pair, the image of the original target volume gets stretched to a size of approximately  $6\text{mm} \times 6\text{mm} \times 500\text{mm}$ .

In our experiment, an expanded beam from a HeNe laser was reflected off a computer-controlled phase hologram in the form of a phase-only spatial light modulator (SLM)<sup>82</sup>. To deal with imperfections in the SLM, a blazed diffraction grating was added to the phase-hologram pattern. This resulted

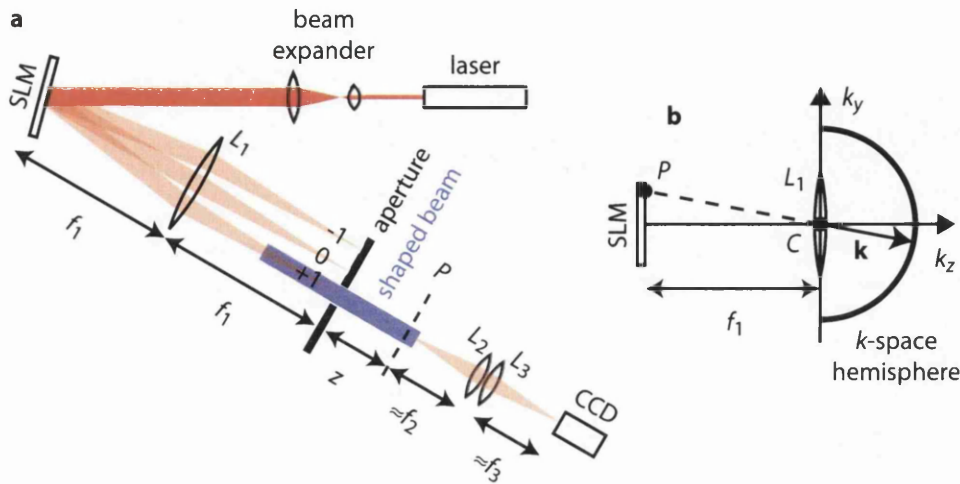


Figure 7.5: (a) Schematic of the experimental set-up. A collimated HeNe-laser beam illuminates a phase-only spatial light modulator (SLM)<sup>82</sup> before passing through a Fourier lens ( $L_1$ ). The SLM displays a phase pattern that shapes the light beam in a volume around the SLM's Fourier plane. An additional blazed phase grating displayed on the SLM directs the shaped beam into the grating's +1st diffraction order; the other orders – caused by imperfections in the SLM's phase response – are filtered out by a Fourier-plane aperture. The volume in which the light beam is shaped is shown in blue. A CCD, with the help of imaging lenses  $L_2$  and  $L_3$ , records the intensity in a number of planes across this volume. In our experiment  $f_1 = 600\text{mm}$ ,  $f_2 = 1350\text{mm}$  and  $f_3 = 200\text{mm}$ . (b) Geometry of the correspondence between position on the SLM and on the  $k$ -space hemisphere. Each point light source  $P$  in the front focal plane of the Fourier lens  $L_1$  gives rise to a uniform plane wave whose  $\mathbf{k}$  vector is parallel to the line from  $P$  to  $C$ , the centre of  $L_1$ .

---

in the desired beam travelling in the direction of the additional grating's first diffraction order; imperfections in the phase response of the SLM led to additional diffraction orders, which were filtered out by the aperture in the hologram's Fourier plane. We used a lens pair to image a plane at a variable distance  $z$  behind the aperture onto a CCD. The transverse magnification was slightly less than one. Intensity cross-sections corresponding to different planes taken across the shaped volume were later combined into volume data, which were visualized with bespoke 3D-viewer software based on the *VolPack* volume rendering library<sup>35,36</sup>.

### 7.3.1 Calculation of the phase-hologram pattern

The 3D GS algorithm finds phase values for the different points of a monochromatic beam's  $k$ -space hemisphere. Points on the  $k$ -space sphere correspond to infinite uniform plane waves; the direction of the  $k$  vector is the direction of the phase-front normal. Experimentally, a finite uniform plane wave can be created from a point light source in the front focal plane of a lens, whereby the position of the point light source determines the direction of the phase-front normal. Each point in a phase hologram that is placed in the front focal plane of a lens therefore corresponds to a point on the  $k$ -space hemisphere and controls its phase.

Figure 7.5b shows the geometry of the correspondence between points on the SLM and on the  $k$ -space hemisphere. In order to calculate the exact position of the point on the SLM that controls a given point on the  $k$ -space hemisphere, the point on the hemisphere should be projected through the centre of the lens into the SLM plane. However, we find that a simpler parallel



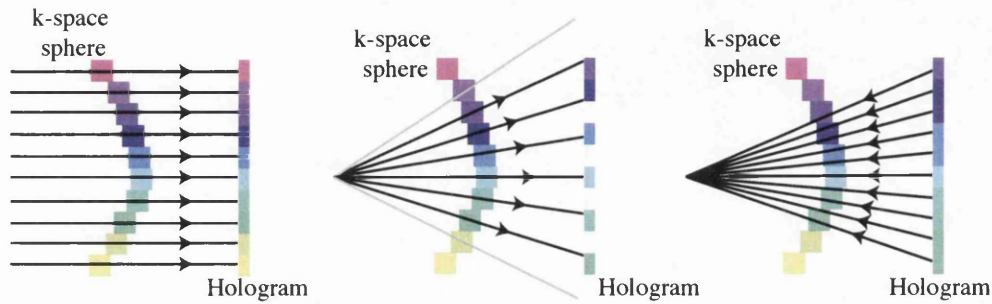


Figure 7.6: Illustration of the three methods proposed to extract the hologram from the  $k$ -space hemisphere. collinear projection (left) calculates the hologram by projecting each element of the  $k$ -space sphere onto the hologram based on its  $xy$  position. Outwards radial projection (middle) calculates the position of each  $k$ -space sphere element by projecting a line from the centre of the sphere through the centre of the element onto the hologram. If the hologram has a high number of pixels, this method can result in the hologram having empty pixels. Inwards radial projection (right) overcomes this by calculating the corresponding element of the  $k$ -space sphere for each pixel of the hologram by a similar method to the outwards radial projection.

---

projection as indicated in figure 7.6, which gives a phase-hologram pattern that is flipped and distorted, works very well. The distortions are small for points representing plane waves travelling at small angles with respect to the  $z$  direction, as is the case in our experiment. We calculate the phase-hologram pattern as the phase of the sum over all  $k$ -space elements with the same values of  $k_x$  and  $k_y$ . In the limit of an infinitely thin  $k$ -space hemisphere, this parallel projection is equivalent to calculating the phase cross-section of the 2D Fourier transform of the shaped beam in the Fourier plane (i.e. at  $z = 0$ ).

### 7.3.2 Results

Figure 7.7 shows some of our experimental results. The experimental patterns do not exactly match those calculated by the 3D GS algorithm. We believe this is mostly due to residual astigmatism in the optical system, imperfections in the SLM and imperfect alignment of the intensity data from different planes to 3D volume data. In any case, the results clearly demonstrate that the experiment works.

## 7.4 Summary

We have used an algorithm first described in reference<sup>72</sup> to calculate physically realisable light beams with a shaped intensity distribution. We have extended this algorithm to generate phase holograms to shape light beams into the given intensity distribution. The theoretical effects of various parameters of the algorithm were investigated. We used this algorithm to create

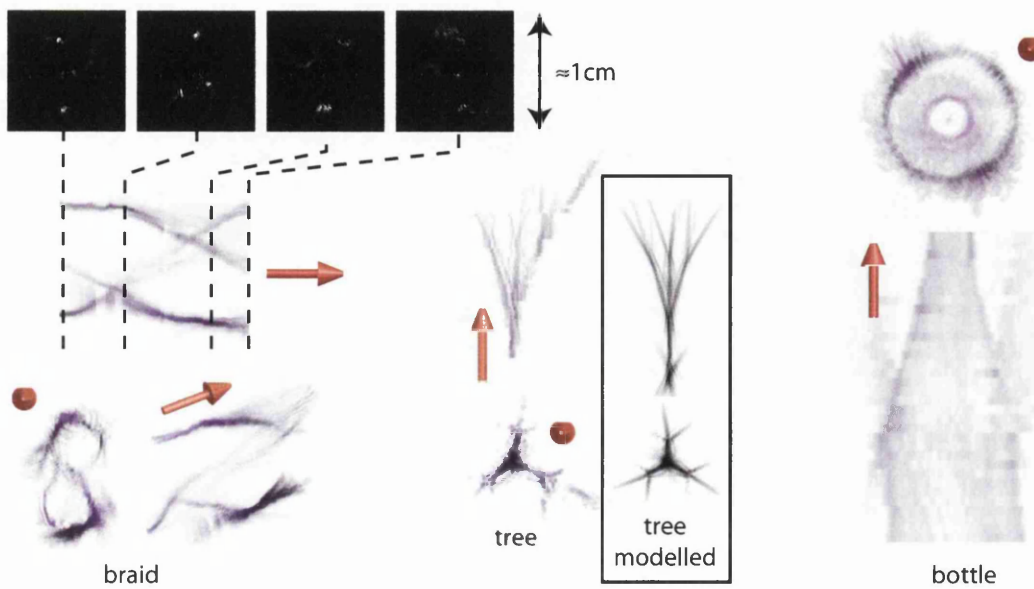


Figure 7.7: Experimental results. Left: three different projections of the experimentally recorded 3D intensity distribution of one period of braided bright lines (purple = brightest, white = darkest). Like in figure 7.1, red arrows indicate the propagation direction of the light beam. The intensity was collected over a volume of size  $\approx 10\text{mm}$  in the transverse directions and  $\approx 590\text{mm}$  in the longitudinal direction. The top row shows four (out of 60) experimentally recorded intensity cross-sections, which were later combined in the computer into volume data. Centre: two projections of the 3D intensity distribution over a volume of approximate size  $10\text{mm} \times 10\text{mm} \times 510\text{mm}$  of an experimentally generated light beam in the shape of a tree, together with modelled results (boxed). Right: two projections of the shoulder region of a bottle-shaped beam (different from the one shown in figure 7.1) over a volume of approximate size  $10\text{mm} \times 10\text{mm} \times 320\text{mm}$ .

examples of such light beams in an experiment using a phase hologram calculated from the beam's Fourier-space representation.

## Chapter 8

# Optical Tweezer Interfaces

This chapter describes some of the computational problems involved with implementing a holographic optical tweezers system. The main thrust of this work is in developing methods to manipulate optical traps in a simple and intuitive manner. Section 8.1 gives an introduction to optical tweezers and the forces involved. Section 8.2 describes some of the cases where a single optical trap cannot be used to effectively manipulate a particle and proposes methods to trap in these situations. In order to give a response time sufficient for interactive use, the calculation of the holograms which are required to create the traps has to be fast. Section 8.3 describes some of the optimisations which were implemented in order to speed up the calculation.

The second part of this chapter goes on to describe methods of interacting with the optical tweezers through various controls. In section 8.5 I describe a simple user interface which allows users to pick up microscopic objects using optical traps controlled by the user's hands. The details of the required experimental equipment and the software are mentioned and some results obtained from using the interface are presented.

My role in this work was in optimising the existing hologram calculation algorithms in order to allow real-time interaction. This got me involved in joystick 'claw' experiment<sup>83</sup> where the optimised software was used to

calculate the holograms to allow a joystick to manipulate four optical traps. The extension of this work was in creating a user interface which could use the user's hands to manipulate the optical traps. I developed this interface and performed its integration into several optical tweezers systems. Using the interface I was able to obtain a series of results of manipulating objects which are difficult to manipulate with conventional optical tweezers.

## 8.1 Optical Tweezers

Optical tweezers manipulate microscopic object using light. Since their early development in the 70s by Ashkin and others<sup>84,85,86</sup>, optical tweezers have become an invaluable tool in manipulating and probing microscopic systems. Optical tweezers have been used for everything from arranging 3D lattices<sup>87</sup> and creating motors<sup>88</sup> and pumps<sup>89,90</sup> to in-vitro fertilization<sup>91</sup>, detecting malaria<sup>92</sup> and even stretching DNA<sup>93</sup>.

### 8.1.1 The forces involved

When light reflects from a surface, the momentum of the light changes direction. To ensure the conservation of momentum, the reflecting object must gain momentum, so-called radiation pressure. This combined with the momentum gained when light scatters off the object is called the scattering force. The scattering force due to incident light of intensity  $I_0$  and wavelength  $\lambda$  on an object of radius  $r$  which is smaller than the wavelength of the light, has been calculated as<sup>66</sup>

$$F_{\text{scat}} = \frac{I_0}{c} \frac{128\pi^5 r^6}{3\lambda^4} \left( \frac{N^2 - 1}{N^2 + 2} \right)^2 n, \quad (8.1)$$

where  $N$  is the ratio of the refractive index of the particle to the index  $n$  of the medium.

One of the simplest system to hold a particle using light is to balance this radiation pressure with gravity and hold the particle in suspension, so-called optical levitation<sup>85</sup>. It was discovered during this experiment that another effect must be influencing the particle. When the weakly focussed laser beam was turned on, the particles would move towards the brightest part of the beam. If radiation pressure was the only effect, the force due to the brightest part of the beam would be higher and the particle would be repelled from that area.

By considering the particle as being a transparent object rather than a perfect mirror, it is possible for the light to be refracted through the particle if its refractive index is different from that of the surrounding medium. Figure 8.1 illustrates this effect and shows the forces for several cases. It can be seen that if the surface through which the light exits the object is not parallel to the entry surface, there is a net change of momentum of the light which must be balanced by the particle. The resulting force is called the gradient force and is given by<sup>66</sup>

$$F_{\text{grad}} = \frac{-n^3 r^3}{2} \left( \frac{N^2 - 1}{N^2 + 2} \right) \nabla(|E|^2). \quad (8.2)$$

By creating a tightly focussed light beam, the intensity gradient can become very high and in such a case, a particle of high refractive index immersed in a lower-refractive-index material will feel a force towards the small high-intensity spot and may be trapped there. If the trapping force is high enough, moving the light beam will move the particle. In this way it is possible to trap and arrange particles into 3D structures<sup>87</sup> which can be manipulated<sup>94</sup>.

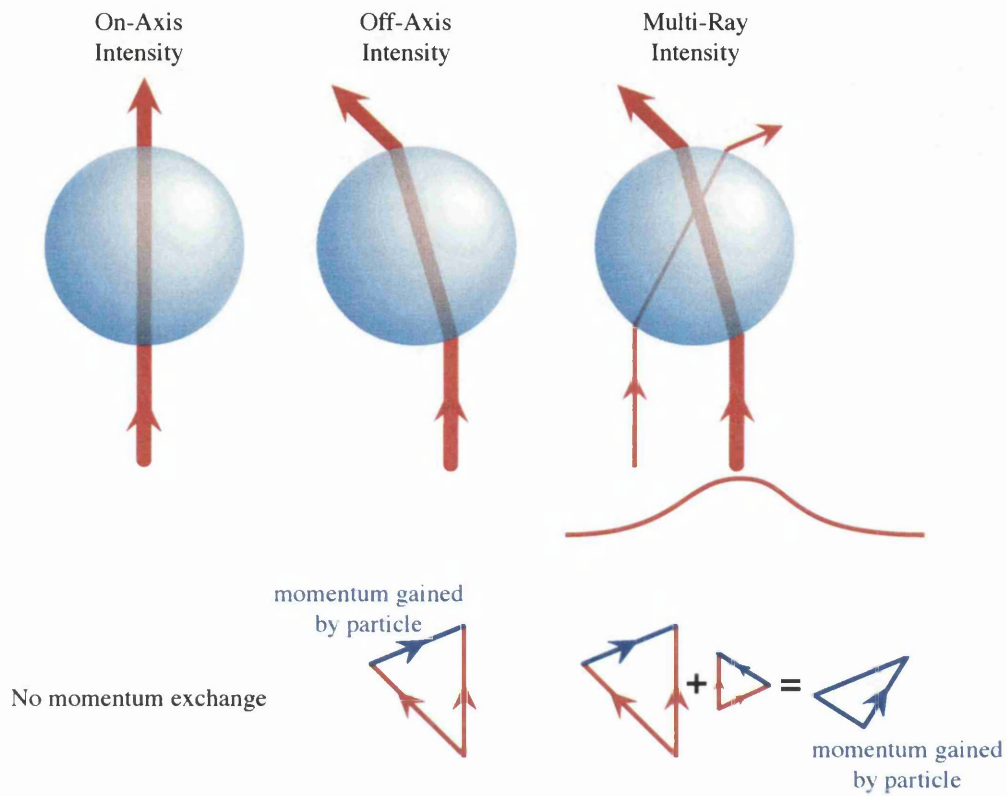


Figure 8.1: Representation of the gradient forces present upon a dielectric particle immersed in a medium of lower refractive index. Left shows the stable case where there is no force upon the particle as the light does not refract. Centre shows the case when the stable case is disturbed and the particle is off the axis of the light beam. There is a net force on the particle towards the right as the light is refracted and gains momentum to the left. If several rays are considered as the right case shows, the net force on the particle is towards the highest intensity.



## 8.2 Indirect Manipulation

### 8.2.1 Difficult-to-trap objects

Conventional tweezers require that the gradient force (Equation 8.2), which acts in the direction of the greater light intensity, is much larger than the scattering force (Equation 8.1), which always points in the direction of light propagation.

There are many situations where the gradient force is not the dominant force. For example, if a material absorbs light at the wavelength used, or if the focussing power of the trapping lens is not high enough and hence the gradient of the intensity is too low. If the beam is propagating upwards, it is possible for scattering to balance gravity; such optical levitation can be observed experimentally<sup>85</sup> and the particle can be held in place by the light.

One of the proposed applications of optical tweezers is in the creation of micro-machines<sup>95</sup> where the nanometre-scale resolution of the manipulations can orientate and combine components. Unfortunately these proposed micro-machines can have metallic components which do not trap effectively. Because light does not transmit through metallic objects, there can be no refraction and hence no gradient force.

There has been success in moving and orientating metallic particles utilising plasmon resonances<sup>96</sup>. However, the frequency of the light has to be chosen carefully to ensure that it is resonant, and this frequency depends on the shape and structure of the individual particles. This can be useful in selecting particular particles with a determined geometry and can thus ensure consistency in the selection processes, but is limited when trapping

---

objects of slightly different shape or size.

### 8.2.2 Scattering methods

It is possible to use only the scattering force to manipulate an object. This involves creating a small dark patch of light surrounded by a ring of high intensity which pushes the object away from the high intensity and holds it within the dark region. This ring can be created using beams with annular intensity rings<sup>97,98</sup> and have no intensity along the beam axis or by scanning of a beam focus to create a time-averaged ring of light<sup>99</sup>.

### 8.2.3 Indirect Manipulation

It is also possible to use trappable objects to interact with untrappable objects through contact. For example by trapping silica beads in dynamic traps, these beads can be brought into contact with an untrappable object and can manipulate it. Surrounding an object with silica beads can therefore constrain the object to a small volume. Simultaneous movement of the beads results in the object moving with the beads.

## 8.3 Real Time hologram calculation

To allow interactive manipulation of optical traps, the calculation of the hologram required must be performed in real time. This limits the available algorithms (for a summary of algorithms see Chapter 6) which can be used effectively. Due to their relatively long iteration times (tenths of seconds to

seconds), iterative algorithms (Gerchberg-Saxton, direct binary search) become too slow to provide real-time interaction for high-resolution holograms. The algorithm chosen to calculate the holograms was the superposition approximation (see Section 6.2). However this algorithm if implemented as described<sup>100</sup> results in suboptimal performance. By optimising the calculation, much higher resolution holograms can be calculated in real-time.

### 8.3.1 Superposition approximation algorithm optimisations

The superposition approximation algorithm (see section 6.2) has been used extensively in optical trapping, first demonstrated by Reicherter et al.<sup>67</sup> for spots in one plane and later by Liesener et al.<sup>68</sup> in three dimensions.

The main optimisation steps were 1. removing redundant calculations; 2. increasing the speed of the algebra; and 3. calculating only the phase. The details of each of these steps are outlined below along with their improvements to the speed of the algorithm.

#### Removing redundant calculations

A hologram of a phase gradient in the  $x$  direction results in the focus of the laser beam in the Fourier plane of the hologram to be shifted in proportion to the phase gradient. By combining a shift in the  $x$  direction with a similar gradient in the  $y$  direction, the focus can be shifted to any position within the Fourier plane. It is also possible to shift the position of the focus in the axial direction by adding the phase hologram of a lens to the  $x$ - and  $y$ -shift holograms. This method requires calculating three holograms for each spot

and combining them to produce the single hologram for the spot. However, an off-centre spot can also be calculated using a single off-centre lens which results in only a single hologram being needed (either an off-centre lens if the spot is out of the Fourier plane, or a  $x, y$  shift if the spot is in the Fourier plane). This method provides an improvement despite the added complexity of the calculation due to the large time taken to add large arrays of numbers together.

### Increasing the speed of the algebra

Calculation	Double Precision	Single Precision
Lens Hologram	5010	4276 (85.3%)
Combined $xyz$ Hologram	45734	40921 (89.5%)
Adding two 256x256 arrays	13296	1765 (13.3%)
Multiplying two 256x256 arrays	12921	1843 (14.3%)

Table 8.1: Comparison of the time in milliseconds to perform various hologram-related calculations. The number in brackets is the percentage of the double precision (64 bit) calculation time. Each of these calculations were performed 10000 times (on a dual-processor 2.5GHz desktop PC with 1GB RAM) to ensure the calculation of processing time is negligible and each set was repeated 3 times and averaged.

With the speed of modern computers, many calculations are performed with high precision to reduce the impact of numerical rounding errors. However this additional accuracy comes at the cost of increased memory usage

---

and reduced processing speed. When performing calculations on floating-point numbers, modern desktop-computer processors perform the calculations using 80 bits of information and then truncate this to the accuracy desired (64 bits for double-precision number, 32 bits for single-precision numbers), so although little improvement in calculation speed can be achieved by using a lower accuracy, the gains in memory usage and transporting the data within the computer can give a noticeable effect when large arrays of numbers are used. These effects can be seen in Table 8.1. It is important to note that, as mentioned, there is only a small effect in using 32 bit floating point numbers in place of the previously used 64 bit numbers during many calculations; however when the large arrays of numbers are manipulated, an approximately 6 fold improvement can be obtained.

The speed of data transport within the computer, combined with the limitation of the spatial light modulator (SLM) to only allow 256 levels of modulation, led to storing the data in 8-bit format for as much of the processing as possible.

By calculating the components of the hologram and performing all the manipulations using unsigned-8 bit integer numbers, significant gains can be made at the expense of rounding errors. However, because the superposition requires the addition of fields, the 8-bit hologram data have to be converted to floating-point numbers to facilitate the complex addition and then back into 8-bit data. Once the individual hologram calculation and preparation for displaying is 8 bit, the conversion to complex numbers and the complex addition becomes the bottleneck in the hologram algorithm.

### Phase-only calculation

When calculating a phase-only hologram for multiple spots using the superposition algorithm, the amplitude of the superposition is discarded by the algorithm and only the phase is used. This gives rise to the possibility of increasing the processing speed by not calculating the amplitude information.

Figure 8.2 describes an optimisation which allows the phase of the sum of two equal-magnitude vectors to be calculated without any trigonometry. Figure 8.3 shows the timing for the different methods of extracting the phase of a superposition of fields with uniform amplitude. The process of adding two phases using complex addition can be described as

1. real part<sub>a</sub> =  $1 \cdot \cos \theta_a$
2. imaginary part<sub>a</sub> =  $1 \cdot \sin \theta_a$
3. real part<sub>b</sub> =  $1 \cdot \cos \theta_b$
4. imaginary part<sub>b</sub> =  $1 \cdot \sin \theta_b$
5. real part = real part<sub>a</sub> + real part<sub>b</sub>
6. imaginary part = imaginary part<sub>a</sub> + imaginary part<sub>b</sub>
7. phase =  $\arctan(\text{imaginary part} / \text{real part})$

where  $\theta_a$  and  $\theta_b$  are the two phases to be added. By looking at the expanded procedure for adding two complex numbers, the complexity of the computation is much larger than might be expected. The five trigonometric functions are computationally intensive and because they operate on floating point

numbers, the time taken to perform this addition becomes the bottleneck in the algorithm.

Complex Addition		Phase-Add Optimisation	
Pros	Cons	Pros	Cons
Very General	Slow	Very Fast	Uniform amplitude
Any number of phases	5 trigonometric functions	No trigonometry	Only add 2 phases at once
Scales linearly with N			Only add $2^N$ phases equally

Table 8.2: The relative pros and cons of the two methods of calculating the phase of a superposition of uniform amplitude fields.

### Phase-only addition

In order to reduce the computational load of the complex addition, I investigated other methods of calculating the phase from the complex addition complex numbers with the same modulus. Figure 8.2 shows a representation of the addition of two complex numbers which is in terms of their Argand vectors and the basis of the phase-add optimisation which is used in the hologram calculation.

The phase-add method utilises the fact that the resultant phase of a complex addition is the average of the two phases if the amplitude of the numbers are equal (with an additional phase shift in some cases). This reduces the computational load for a single element calculation to between one and three

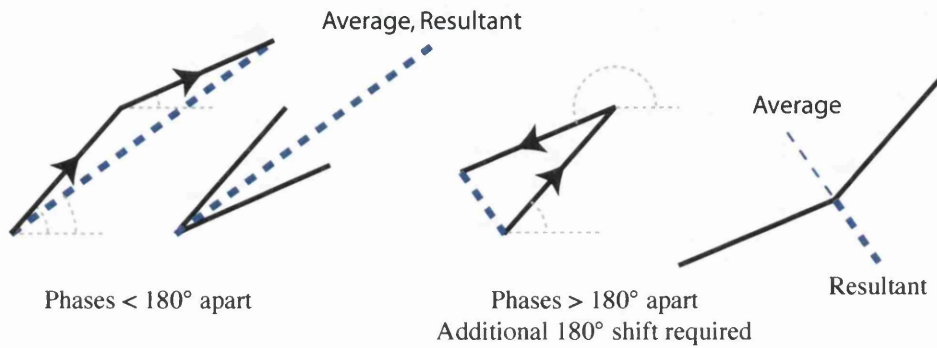


Figure 8.2: Argand-diagram representation of the vector addition of two complex numbers with equal amplitudes. Left: When the two phases are within  $\pi$  of each other, the phase of the resulting vector can be calculated as the average of the phases of the two inputs. Right: When the difference between the two phases is greater than  $\pi$ , the average becomes  $\pi$  out of phase with the result. This requires the phase-add method to include a comparison to allow the additional phase shift, if the difference of the two phases is greater than  $\pi$ .



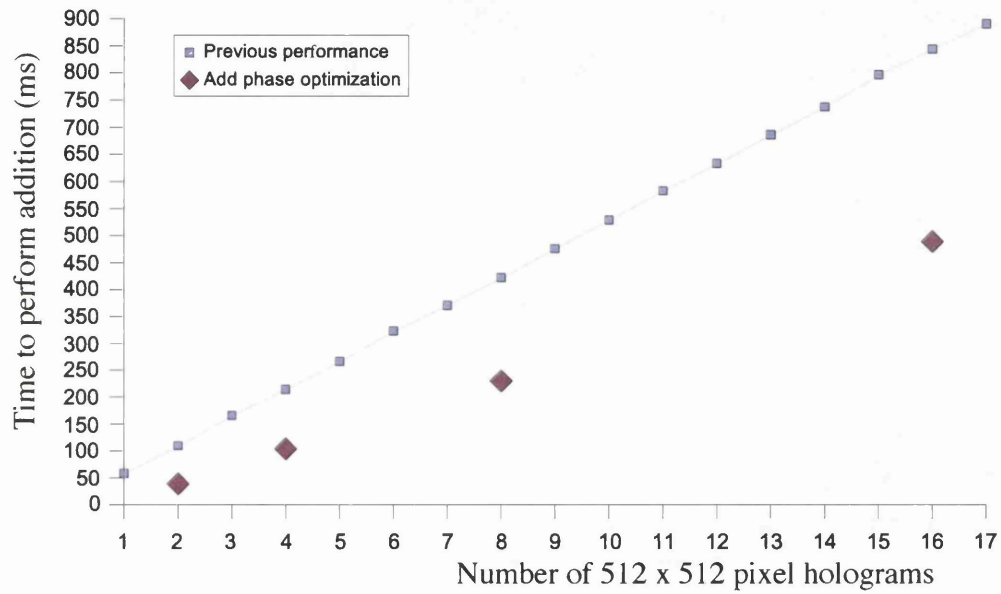


Figure 8.3: Graph showing the time to calculate a single 512 x 512 element hologram from the addition of multiple holograms. The timing for the previous implementation is designated by squares and the phase add optimisation is designated by diamonds.

additions, a single divide (or numerical bit-shift) and two comparisons. In terms of computational load, the calculation of the resulting phase using the phase add method does not require any computationally intensive trigonometric functions (unlike calculation using complex addition which requires five trigonometric operations) and avoids the unnecessary calculation of unwanted amplitude information.

### Limitations of the Phase-Add optimisation

Table 8.2 summarises the limitations of the phase-add optimisation which are discussed in more detail below.

Because the optimisation only works for two equal amplitude fields, this places some restrictions on the relative weighting of the component fields. Figure 8.4 shows the required arrangement of phase additions and the distribution of the resulting power. In the simplest case of two hologram patterns being added, only one phase addition is required for each pixel and the power is evenly distributed between the two patterns. If three phases are to be added, two phase additions per pixel are required, firstly between the first two phases and then between the resultant phase and the third phase. This results in the third phase being given the same power as the sum of the first two giving rise to a power distribution of  $\frac{1}{4}, \frac{1}{4}, \frac{1}{2}$  respectively. For four phases, three phase additions are required and in their optimum configuration, the first and second phases are added and the third and fourth phases are added before the result of each addition being added to give the resultant phase. This gives a uniform power distribution between all the patterns.

It can be seen that in order to achieve uniform power distribution between

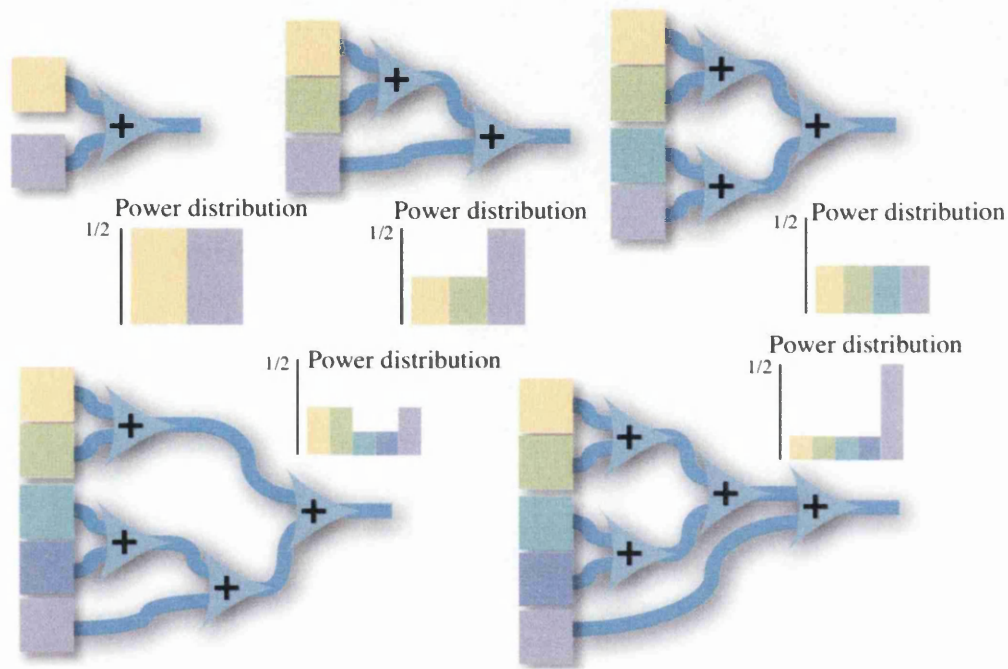


Figure 8.4: Representation of the methods of adding multiple holograms using the phase-add optimisation. The limitation that the inputs to each addition are equally weighted can only be achieved if there are  $2^N$  holograms (for integer values of  $N$ ). If the number of holograms is not  $2^N$ , the sequence in which the additions are performed determines the distribution of the power (insets).

---

all the component patterns, there have to be  $2^N$  patterns. This places a large restriction on the possible configurations of patterns. If a non-uniform distribution of power is required, the distribution is determined by the exact order of the additions and can lead to unexpected results. For example adding a new fourth spot on the same position as the third spot results in the power distribution remaining the same.

In order to maximise the uniformity in power between the component powers, the additions have to be performed in a tree structure. This results in a non-linear dependence on the number of patterns, and hence the additional speed improvements for each addition can be negated by the larger number of additions required.

To maintain the uniformity of the patterns with any number of component holograms, the program has several dedicated optimised calculations for patterns made up of 2, 4, 8, 16 and 32 patterns; all other combinations are performed using the slow but general complex addition. It could also be possible to optimise the addition to the nearest power of 2 and then use complex addition to add the other patterns. The main drawback of such a system is the time taken to transform the integer data (0...255) to the equivalent floating point numbers ( $-\pi... \pi$ ). This, combined with the added complexity of the required comparisons, was judged to be too much for the small gains and was left un-implemented.

---

## 8.4 Human-Tweezers Interaction

As the development of optical tweezers makes the system usable and useful outwith a physics lab, the requirement of reliable and easy-to-use systems becomes more important. One of the key factors in the ease of use of such a system is the user interface as this is the part of the system which receives the most use. The two most common ways of communicating the desired trap positions to the computer are entering the trap positions using the keyboard and dragging representations of the traps around using the mouse. Both of these methods are easy to implement but can only manipulate one object (either a trap, or group of traps) at one time and only one and two degrees of freedom respectively.

### 8.4.1 Joystick Control

Several recent optical-tweezers interfaces use a joystick to manipulate the traps. This also limits manipulation to one object at a time. However, with some designs of joystick there are many degrees of freedom which can be manipulated simultaneously, and often more than a dozen buttons. This allows many separate manipulations of an object to be performed at once.

One such interface<sup>83</sup> uses a joystick to manipulate a pattern of four traps in a square pattern. Figure 8.5 illustrates the degrees of freedom used by the interface and the associated manipulation. By using many of the joystick inputs, all the foreseen manipulations are controllable using the joystick and once the experiment is underway, the user does not need to take his or her hand from the joystick.



Figure 8.5: Representation of the degrees of freedom used by the joystick interface and the manipulation used by each. The  $xy$  position of the joystick represents the velocity of the centre of the four traps, the twist of the joystick relative to the base gives rise to a rotation of the square pattern with the speed of rotation being proportional to the angle of twist. Two buttons on the top of the joystick controls the size of the square in the  $xy$  plane and another four buttons on the base control the  $z$  height and the height difference between the sets of spots  $\delta z$ .

---

The main disadvantage of the joystick is that it can only manipulate a single fixed pattern and all the manipulations have to be pre-programmed into the software. If, for example, transforming the square pattern into a rhombus is a useful transform in one application, the user would have to pre-plan such a transform and implement it in software before beginning the experiment. To get around such difficulties, other methods of interacting with the optical tweezers were required.

## 8.5 Simple hand-based interface

In order to create an easy-to-use and intuitive interface for holographic optical tweezers, more independent degrees of freedom are required. One method<sup>101</sup> uses a virtual-reality glove to control a trap based on the position in 3D space of the user's fingers, and although this provides higher degrees of freedom, the virtual-reality hardware is expensive and requires specialist knowledge to implement the software. A single glove is also inherently limited to five independent objects, which can be extended by additional gloves thereby quickly becoming very expensive and more complex to implement.

A simpler system based on imaging the user's fingertips should provide the same degrees of freedom without the high cost, complex hardware interaction and limitation to 5 objects.

### 8.5.1 Imaging the user's hands

The webcam used for imaging is fixed above the bench and pointed down at the bench. The small aperture ensures the depth of field is sufficient so that

the volume in which the user's hands move is in focus. The bench below the camera is covered with black cloth to remove any detail of the bench top which might interfere with the image processing. By using black gloves the hands of the user can become indistinguishable from the background and by attaching map pins with bright heads to the glove's fingers, the finger tips have high contrast against the background.

This results in the image containing several bright discs against a dark background which makes the image processing straightforward. By using only one camera, obscuring the map pins can become an issue and the image processing can not maintain the positioning of the associated trap.

By placing a webcam close to the volume where the user's hands will be, the imaged hands will appear large in the captured image and the apparent size will be dependent on the distance from the camera. If they are close enough, this size difference will be measurable.

### 8.5.2 Image processing

Figure 8.6 shows the image processing steps required to translate the image into trap positions. The first step is to convert the colour image into a 1-bit black-and-white image which is done with a user-customisable threshold to allow for changing light levels. This image is then searched for circular patterns which will include the pin heads. In order to treat correctly discs that are (partially) obscured, the circle finding algorithm tries to identify partial and overlapping circles; this can wrongly identify (discretisation) artefacts as circles. In order to limit the influence of artefacts, the resulting circles are filtered further to eliminate very small and very large circles; this removes



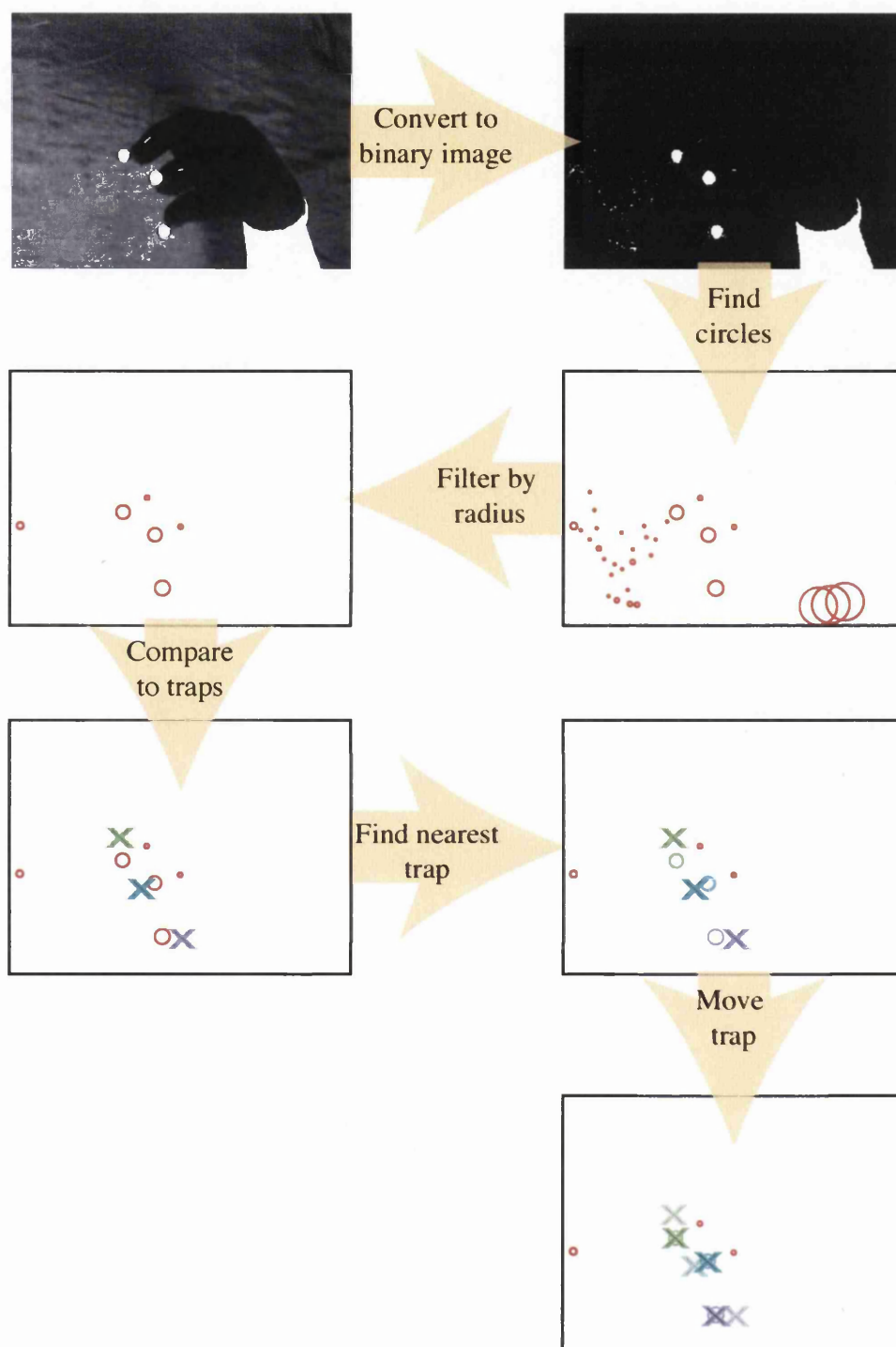


Figure 8.6: Flow diagram of the image processing required to transform the live image to map pin-positions and then into trap positions.

single-pixel noise and many large unwanted objects, such as the user's arms.

The resulting circles are then treated as corresponding to pin-head positions. The radius of each of the circles is used as a measure of the pin head's  $z$  (height) position and the location of the centre of the circle is used as the  $x$  and  $y$  (transverse) position. These positions are compared with the trap positions of all the known optical traps and the nearest map pins and traps are identified. If the distance between the pin head and the optical trap is smaller than a pre-defined distance called the "trap snap" distance, the pin head's position is assumed to be the trap's desired new position. In order to improve the ease of use, the distance between the trap and the pin head can be calculated as the distance in 3D, the  $xy$  distance or the  $z$  distance; this facilitates matching a pin head to a trap.

### 8.5.3 Trap movement

To ensure that the position of the trap does not jump too far in one iteration, there is a user customisable "maximum trap movement" parameter which allows the user to limit the distance the trap can move in a single iteration. If the pin head has moved from the trap position further than is allowed by this parameter, the trap is moved by the maximum distance over several iterations towards the current pin head position. The software does not maintain a history of previous positions of the pin heads. The trap will always move towards the current position, thus tracing a path quickly does not mean the trap will follow that path. This functionality was chosen to eliminate spurious movement causing the traps to trace an unwanted pattern. Careful use of the "maximum trap movement" parameter allows the optical

---

traps to keep hold of the trapped particles by ensuring the traps move within an upper speed limit.

#### 8.5.4 Layout of the interface

To ensure the visual feedback element of the interface is utilised, the main display of the interface is composed of the live image from the optical tweezers with information overlaid. Figure 8.7 shows the interface in use; the inset shows the information overlaid onto the live image.

A small section of the interface contains the live image of the user's hands with the position of the traps overlaid to aid tracking. This area of the interface can be switched between the live image of the user's hands, the uncropped image of the tweezers and the various one-off configuration options.

The final element of the interface is the  $z$  positioning. On the bottom left of the main image, the  $yz$  position of the traps and the map pins are plotted in red and white respectively. This allows quick and easy identification of the height of the traps relative to the map pin positions.

#### 8.5.5 Uses of the interface

Because of the nature of the interface, it is most suited to applications where real-time, interactive manipulation of the traps is required. One of the most important elements of the interface is the simultaneous independent manipulation of multiple traps, which is ideally suited to collecting multiple particles and bringing them together. The interactive nature allows the user to compensate for the small movements or re-orientations of the trapped objects in

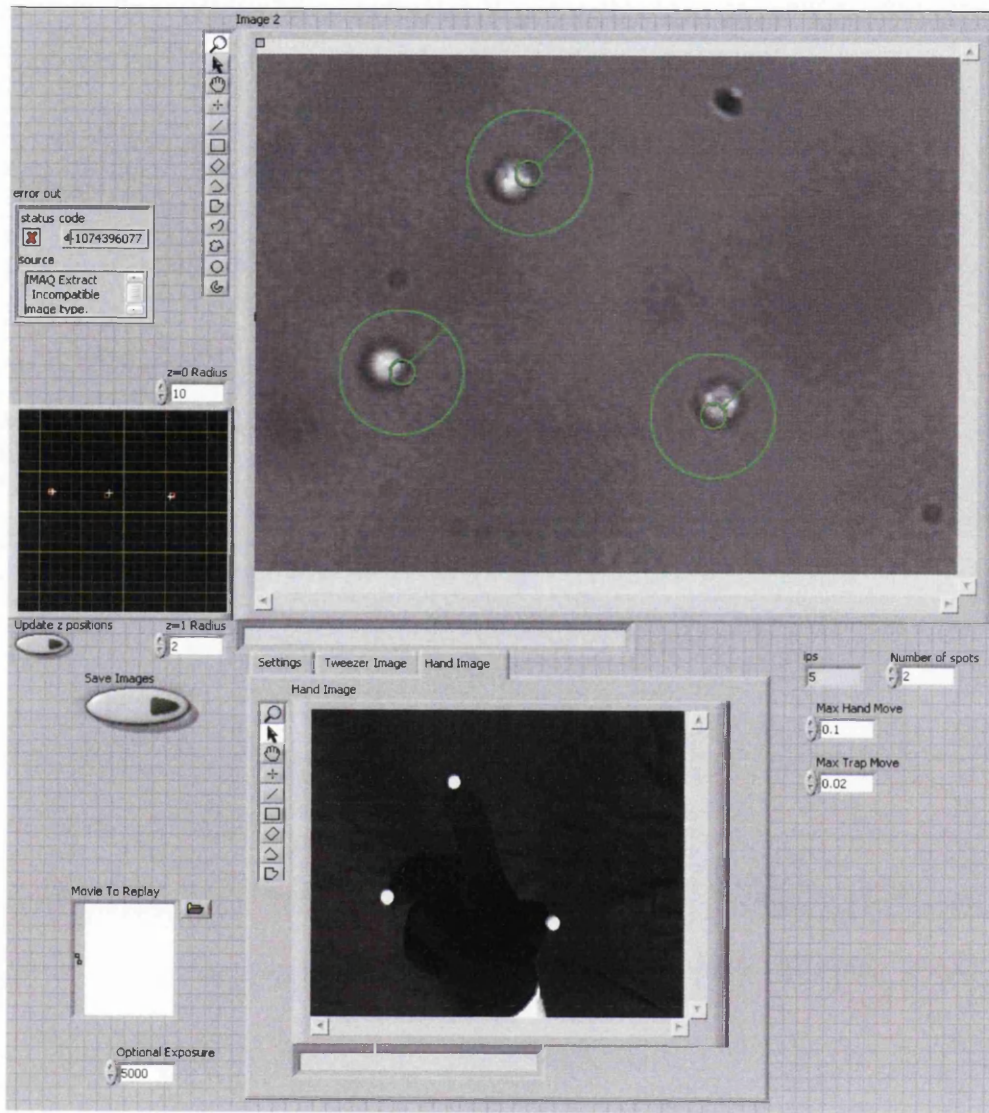


Figure 8.7: Screen capture of the visual interface of the hand-based interface.

The main image which consists of a cropped section of the live microscope image of the optical tweezers with data relating to the trap and pin-head locations overlaid. The image in the bottom shows a live image of the user's hands and the graph in the middle left is a  $yz$  plot of the trap (red) and map-pin positions (white), showing the relative heights.

an intuitive manner which gives rise to the possibility of building macro-scale objects by assembling the components by hand.

The multiple traps can be used to either trap multiple objects or to hold extended, anisotropic objects in a given orientation. If a single trap is used to hold such anisotropic particles, they will align themselves with the single trap to minimise their energy. If the object is extended, the orientation may play an important role in the experiment and by using multiple traps, greater control of the orientation can be gained. By using the interactive interface, no prior knowledge of the arrangement of traps is needed, which can be adjusted to any changes.

### 8.5.6 Examples of applications of the interface

To demonstrate the potential of such an interface, several difficult-to-manipulate objects were manipulated using the user's hands. The first example is using three traps to directly manipulate a single red blood cell. In this demonstration, the ability to compensate for small movements by the blood cell allows the orientation of it to be controlled.

The second example uses silica beads to indirectly trap and manipulate a metallic particle. The metallic particle will scatter away from light and so cannot be trapped directly. By using the hand interface to manipulate four silica beads, it is possible – in fact, relatively easy – to surround a metallic particle and move it.

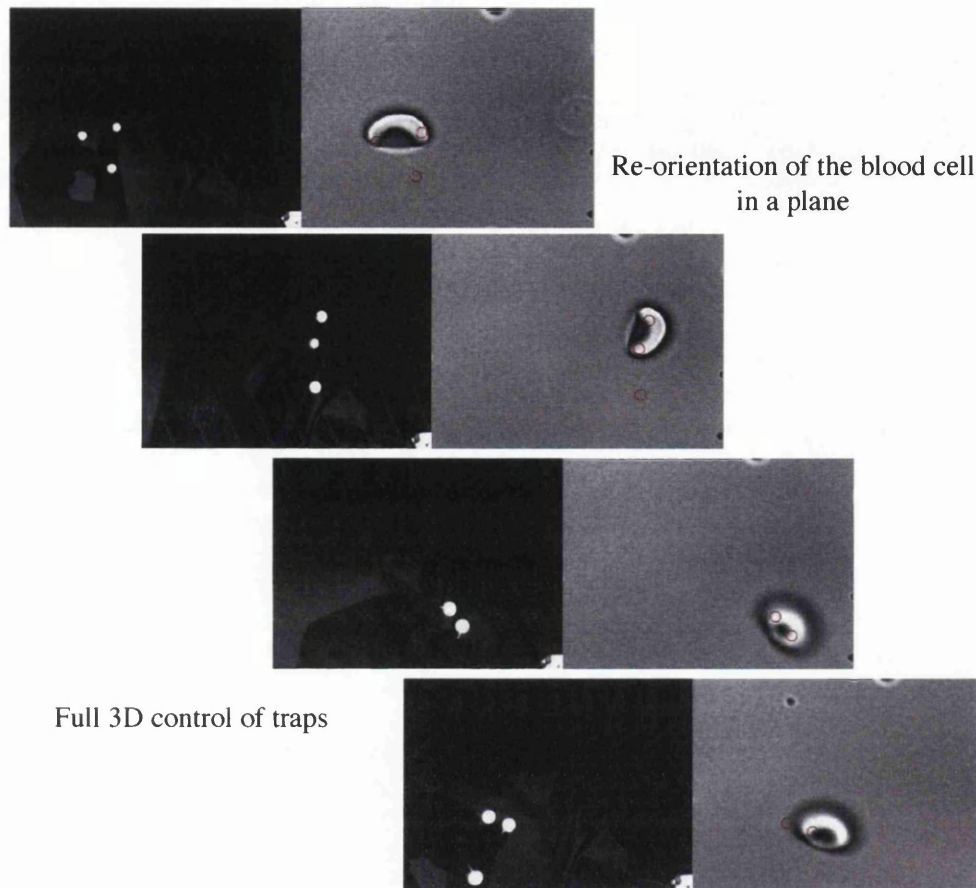


Figure 8.8: Selected frames from a movie showing the manipulation of a single red blood cell using the hand-based interface. Images of the user's hands (left of each frame) show the position of the map-pins used to determine the positions of the user's fingertips. The microscope images (right of each frame) show the trapped red blood cell moving with the traps. The red circles indicate the positions of the map-pins overlaid onto the microscope image. The top two frames demonstrate the ability of the system to re-orientate the red blood cell using multiple traps. The bottom two frames demonstrate this can be done in 3D space.

### Manipulating a single red blood cell

As mentioned, one of the possibilities of manipulating multiple traps is the ability to control an extended object in a more controlled way than if only one trap was used. To demonstrate this, a single red blood cell was manipulated using two traps, each controlled by a fingertip. The large size of the blood cell ( $\sim 15\mu\text{m}$ ) and its anisotropic structure made it a useful test. Sample images taken during a series of manipulations is shown in figure 8.8. In the lower images of figure 8.8 show the red blood cell being re-orientated in a higher  $z$  plane, which can be seen by the larger apparent size of the pin-heads and the out-of-focus red blood cell.

In a single trap, the cell will try to minimise its energy by placing as much of itself in the beam as possible. In extreme cases, the cell will fold itself up to increase the overlap between the cell and the beam<sup>102</sup>. If the cell maintains its form it will tend to centre itself and align its long axis with the beam axis (the orientation of the short axis is outwith the control of the experiment).

By using two closely separated traps, the centre of the cell is then held between the two traps and the direction of the short axis is along the line of the traps. The long axis is again along the direction of the beam. Thus by manipulating the centre of mass of the traps, the position of the cell can be altered, and by altering the relative positions of the traps, the orientation of the cell can be changed.

By introducing a third trap, it should be possible to alter the direction of the long axis of the cell and fully manipulate the orientation of the cell. Unfortunately in our experiment the trapping spot in the axial direction was

---

too extended to allow such manipulations. However slight tilting can be achieved, showing the system, with further improvements, may be able to perform such manipulations.

### **Indirect manipulation of a metal particle using multiple silica beads**

When a metallic particle is near a focussed light beam, the scattering force of light reflecting from the surface is large and pushes the particle away from the high intensity. Because of this, conventional optical traps cannot hold metallic particles and other methods are required (as discussed in section 8.2). By using the hand interface, several silica beads (which can be trapped effectively) can be brought into contact with a metal particle and used to manipulate it.

Figure 8.9 shows an example of such indirect manipulation. The user positions his hands to orientate the silica beads into a square around the metallic particle before moving the beads into contact with the metallic particle and moving it across the screen.

## **8.6 Summary**

By careful design and implementation of holographic algorithms, real-time interactive interfaces can be realised and frame rates of upwards of 10 frames per second can be achieved on desktop PCs. When manipulating many particles it is useful to have many degrees of freedom; however, intuitively using those degrees of freedom requires a well-designed interface.



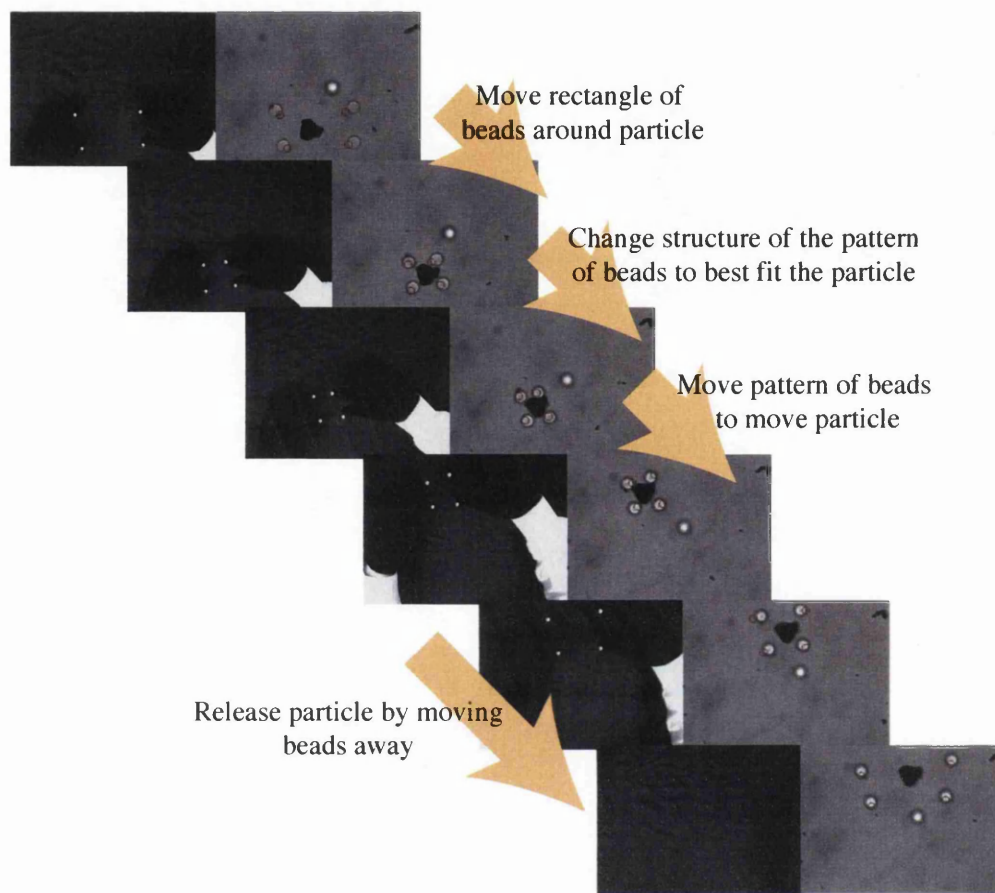


Figure 8.9: Selected frames from a movie showing the manipulation of a hard-to-trap metal particle. Images of the user's hands (left of each frame) show the position of the map-pins used to determine the user's fingertip position; microscope images (right of each frame) show the trapped silica beads (the white discs with dark border) which can be used to manipulate the metal particle, which can also be seen (dark irregular shape). The ability to quickly and intuitively alter the arrangement of the traps can be seen as the beads are brought into contact with the metal particle.

An intuitive interface can be based on the position of a user's fingertips to control the position of optical traps. Such an interface can be created using a single camera which images high-contrast objects attached to the tips of the user's fingers. Analysis of the images can give information on the 3-dimensional position of the fingertips which can be used to position optical traps using holographic optical tweezers. This interface can be used to interactively and intuitively manipulate multiple particles. These can in turn be used to manipulate objects which can not be trapped effectively using conventional optical tweezers.

## **Part III**

# **Conclusions and Appendix**

---

# Chapter 9

## Conclusions

### 9.1 Bose-Einstein condensates

The analogies which can be drawn between laser light and BECs are both wide ranging and powerful. An array of diverse optical systems can be implemented in BECs ranging from lenses and resonators to prisms and interferometers. Many of the systems perform well even with modest non-linearities, and some generalisations can be drawn. Better understanding of such analogies will allow rapid development of a range of BEC systems based on their optical counterpart.

Chapters 3 and 5 investigated the similarities by modeling examples of optical systems in BECs. Chapter 3 demonstrated that modes similar to Hermite-Gaussian modes can be created in a BEC which is confined in a resonator. The similarity to a Hermite-Gaussian mode decreased as the non-linearity of the condensate was increased. However notable characteristics of the Hermite-Gaussian modes were still maintained in the non-linear system. Most notably the extended nodal structure of the resulting mode.

Chapter 5 simulated a BEC in a more complex optical system, a vortex sorter. The aim of such a system is to decompose a the input field into a two sub-sets of the possible orbital angular momentum (OAM) states. Such a

---

sorting process could be repeated indefinitely operating on different sub-sets until the required level of sorting has been achieved. In the idealised sorter which was modeled, the fidelity of a non-linear BEC was again higher than might be expected. The possibility of more accurate rotation of a non-linear condensate could further improve the sorting efficiency.

Chapter 4 investigates the ability to Fourier transform a BEC. This would allow the use of optical Fourier holography techniques to be applied to BECs which would allow straightforward, almost arbitrary state preparation. It was discovered that, similar to graded-index fibres, the propagation of a linear BEC in a harmonic trap will Fourier transform the BEC by waiting for a quarter of a trap period.

## 9.2 Light Shaping

Chapter 7 investigated a method of shaping a monochromatic light beam into an arbitrary 3D intensity pattern. The method was based on the same assumptions as those made by Shabtay<sup>72</sup> which was built-on and generalised to allow for hologram generation. The method was tested by experimentally shaping a light field using a very simple setup, showing its potential strengths.

With optical tweezers becoming more widespread as a tool, some of the problems facing their development become problems of optimisation. In many cases the components of research tweezers systems are comprised of commercially available, and well understood, components. Equally the basic calculations required to produce holographic optical tweezers are well understood. Nevertheless there are large and important gains to be made by

optimising each component. Once the trapping strength is sufficient to hold extended objects and the calculations can be performed in real-time, the potential of optical tweezers can be more fully realised.

It was this goal which lead to creating a fingertip based tweezers interface described in chapter 8, which can be used to manipulate particles and cells using the user's fingers as guides for the position of the optical traps. Such a system could find many widespread uses in applications which require an intuitive and user-friendly means of manipulating optical traps in real-time.

---

# Chapter 10

## Outlook

Much of this work has been laying the foundations for further research. Considerable time has been spent in designing and developing the bespoke software and exploring initial questions. Understanding the similarities of BECs and laser light allows many light-based phenomena to be applied to BECs quickly and offers an alternative interpretation of BEC behaviour.

Specifically, the ability to Fourier transform a BEC leads to the possibility of using well established Fourier holography techniques to be adapted in order to shape a BEC. There are several major considerations in such a system, including the experimental system for imparting the hologram and insuring the algorithm incorporates any short-comings in that system. The possibility of shaping a non-linear condensate is also of great interest, as this is where the analogy with laser light no longer holds. However, the computational load of calculating such a hologram, the strong dependence on the non-linearity and the ability to match the non-linearity between the model and experiment may limit the usefulness of such a system.

Research into creating an algorithm to calculate the optimum hologram to shape a BEC into a desired state is ongoing and is being combined with research into methods of applying a hologram to a BEC which could result in Fourier holography being used for state preparation.

# Appendix A

## Software User's Guide

### A.1 Introduction

Welcome to the user guide for the Optics group's BEC client-server software. This guide describes the basic principles of the software and how to effectively use it. It does not cover details of the underlying software or how to develop new plug-ins or additional software

#### A.1.1 Who should use this guide

This guide is intended for people who will be using the software to enter simulations to be run either by their own computer, or by a dedicated cluster. It is also suited to a new developer who wishes to gain an overview of the user-end of the software and the flow of information before beginning the Developer Guide.

#### A.1.2 The software

The software can be broadly split into two parts: the client and the server. The client part of the software calculates the simulation based on information given to it by the server. The server on the other hand cannot perform



---

calculations and only stores to settings for a simulation and maintains the list of simulations to run.

When entering a new simulation to run, it will be through the server interface which will enter the simulation into the list of things to run. Whenever a client then connects, the server picks the most appropriate simulation and gives it to the client.

## A.2 Information flow

When you enter a new simulation to run, the details of that simulation have to be passed around between the client and the server in order for it to be calculated. The client periodically checks the server if any new details have been entered, and if so, downloads them to be run. The client also request any additional files which are required to run the simulation. Once the client has finished, it informs the server and sends information about the simulation back to the server.

### A.2.1 Why client-server

A client-server model allows the task of controlling the simulations and running the simulations to be separated. This means that you can create a list of several jobs (or several hundred) while a simulation is also running and the new jobs will be processed in turn. By separating the two tasks, also results in better stability. If one of the simulations crashes the software, the list of tasks still to be done is maintained and can be continued. However the two main benefits of client-server is that the client and server do not need

---

to be on the same computer (or in the same country) and that there doesn't need to be a one-to-one ratio of clients and servers running. In-fact it is best configured if there is a single server and many clients running simultaneously.

### **A.2.2 Entering details to the server**

If you wish to enter the details of a new simulation, it must be done via the server software. The server maintains a list of all the possible simulation steps as well as all the available options which can be set for each of those steps.

Once the simulation steps and the options for each step have been chosen, the simulation is entered into the list of simulations to be run by the clients. Every-time a client connects, the server checks this list and picks the most appropriate simulation and gives its details to the client before removing it from the list.

### **A.2.3 Between the client and sever**

When the client requests a new simulation, the server chooses the most appropriate one and sends the details to the client. The client then checks to ensure that it can perform the simulation and requests and required files from the server. Once the client has finished the simulation, it reports back to the server and sends any logged information which the user might want to see.

## a. Login Dialog



The dialog box is titled "Prompt" and contains the following text and controls:

- An information icon (i) followed by the text: "Enter username and password for 'My Realm' at http://opt1"
- A label "User Name:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A checkbox with the text: "Use Password Manager to remember this password."
- Two buttons: "OK" and "Cancel".

## b. Login Page



The login page features a dark blue header banner with the text "BOSE EINSTEIN CONDENSATION" on the left and the University of Glasgow logo on the right. Below the banner, there are navigation links: "LOGIN", "VIEW JOBS", "ADD JOB", and "VIEW STATUS". The main login form includes:

- A "Username:" label and a text input field.
- A "Password:" label and a text input field.
- A "Login" button.

Figure A.1: a. Login dialog when trying to access any of the BEC pages. b. Login page (login.php) which can be used to login to the server.

## A.3 Logging In

When you first visit the server, you will be required to log in. This is to ensure only authorised users can see or edit jobs and to keep track of who created which simulations. The login will be of one of the forms shown in Figure A.1, both require the same username and password. Once you're logged in, you should be able to see all the simulations which you have access to on the server<sup>1</sup> as well as the ability to create new simulations and edit old ones.

Logging in from multiple locations is permitted, however you should not share your details with anyone else. If someone else wishes to use the system, contact the administrator to create a new user account for them.

## A.4 Working with jobs

When you define a new simulation to the server, it stores all the relevant details as a "job". This job contains the details of the simulation including all the steps and the options for each one. When the client connects to the server, the server sends the job to the client which stores it as a JobDef file (often named "JobDef.xml" in the directory associated with the job). This file is a standard XML file which can be edited or created by hand if the server is not required.

To help organise the list of jobs, it is possible to store them in a hierarchy. To do this each one can contain any number of "child jobs" and thus act in

---

<sup>1</sup>It has been observed that if you have trouble accessing information after logging in using the password box shown in Figure A.1a you may need to goto the login option to enable you to see it.

---

a similar way to the folders on a computer drive. A job which has child jobs still acts as a normal, as their behavior is independent of the location in the hierarchy. A job can be referenced either by its JobID (a unique reference number) or by its name. When referencing a job by its name, each level in the hierarchy is separated by a back slash character “\”.

### A.4.1 Viewing jobs

To view the list of jobs currently stored on the computer, click on “list jobs” in the heading of any of the server pages<sup>2</sup>. You should be presented with a list similar to that shown in Figure A.2. The list contains several pieces of important information:

1. the name of the job - this name is relative to its parent job and so will not have any \ characters,
2. the number of child jobs - in brackets is the number of child jobs the job has. The first number is the number of children directly under the listed job while the second number is the total number of jobs (not including the listed one) in the hierarchy under this job,
3. commands related to the job - if you have permission to, it is possible to copy or edit the job or create a new child job under the current one directly from this page,
4. priority - the priority of a job is used when the server decides which

---

<sup>2</sup>“List Jobs” links to [listjobs.php](#) by default. If no link is present, try entering the URL directly.



### View Currently stored Jobs

Here you can see the jobs which are currently stored by the server as well as monitor their progress.

By Clicking on a job you can see details of it's setup and any progress log entries.

It is also possible to edit the setup of any stopped Jobs by clicking on their names, but if a Job has started and you wish to modify it, you will have to create a new copy and edit that job.

[\[Show all jobs\]](#)

Job Name ▼	▲Priority ▼	▲Status ▼	▲Time Added ▼	▲Time Finsihed ▼
<a href="#">3D Light (5/32)</a> <a href="#">[Copy]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a> <a href="#">[Stop]</a>	Fri 4 Jun 04 10:31:57	
<a href="#">BEC Focussing (2/24)</a> <a href="#">[Copy]</a> <a href="#">[Edit]</a> <a href="#">[Child]</a>	[ -10   - ] 1 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a>	Mon 20 Sep 04 15:21:36	
<a href="#">Delme</a> <a href="#">[Copy]</a> <a href="#">[Edit]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a>	Mon 10 Jul 06 16:59:46	
<a href="#">FT By Waiting (2/13)</a> <a href="#">[Copy]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Pause]</a> <a href="#">[Stop]</a>	Wed 20 Apr 05 16:11:17	
<a href="#">New Job</a> <a href="#">[Copy]</a> <a href="#">[Edit]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a>	Tue 14 Nov 06 12:14:21	
<a href="#">Test AddJob</a> <a href="#">[Copy]</a> <a href="#">[Edit]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a>	Fri 1 Oct 04 11:24:18	
<a href="#">Test and Debug (4/22)</a> <a href="#">[Copy]</a> <a href="#">[Edit]</a> <a href="#">[Child]</a>	[ -10   - ] 50 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Resume]</a>	Thu 27 May 04 11:12:18	
<a href="#">Test Job (7/8)</a> <a href="#">[Copy]</a> <a href="#">[Child]</a>	[ -10   - ] 52 <a href="#">[+1]</a> <a href="#">[+10]</a>	<a href="#">[Restart]</a>	Mon 20 Sep 04 10:47:16	Fri 18 Mar 05 15:38:15

Figure A.2: Screen captures of the list of the list of available jobs and their status.

---

job should be processed first. The higher the priority the sooner it will be processed,

5. status - The status of the job. It is also possible to change this status from the listing page using the commands listed,
6. the time it was added to the list - when the user entered all the details of the job,
7. the time when a client finished computing it (if already processed) - if it has been processed already, this records the time when it was finished.

When you click on the name of a job, you will be taken to the details of that job which are listed on a page similar to that shown in Figure A.4. The important details are:

1. the job name - the full name of the job which includes its parents,
2. the job creator - who originally created the job,
3. priority - the priority of the job. A job with a higher priority will in general be sent to a client before a lower one,
4. status - the current status of the job and options to change the status,
5. description - the description of the job which was entered when it was created,
6. comments - comments about the job. These can be entered at any time by anyone who can view the job. This allows for additional information to be added without disrupting the original specifications. Comments can not be removed or edited,

- 
7. simulation setup - the list of steps which the simulation must take,
  8. parameters - the settings for each of the steps,
  9. edit button - if the job is in an editable state (see section A.4.3) you can change the settings and update them,
  10. add components - if the job is in an editable state, it is possible to insert new components at any point,
  11. child jobs - a list of any child jobs presented in a similar way to the main list jobs.

### A.4.2 Creating jobs

Creating a new job is a simple six step process which uses the “add job wizard” to help guide you through the process of creating a new job. This process is illustrated in Figure A.3

1. Select the parent job (or no parent if appropriate<sup>3</sup>),
2. Click either Add new job or Child - this brings up the new first screen of the new job wizard,
3. Enter the name, description, priority and how many steps you'd like to add (zero is an acceptable value and is often used to create a “folder” job to organise child jobs,

---

<sup>3</sup>If no parent is required, navigate to list page and use the add job option in the page banner.



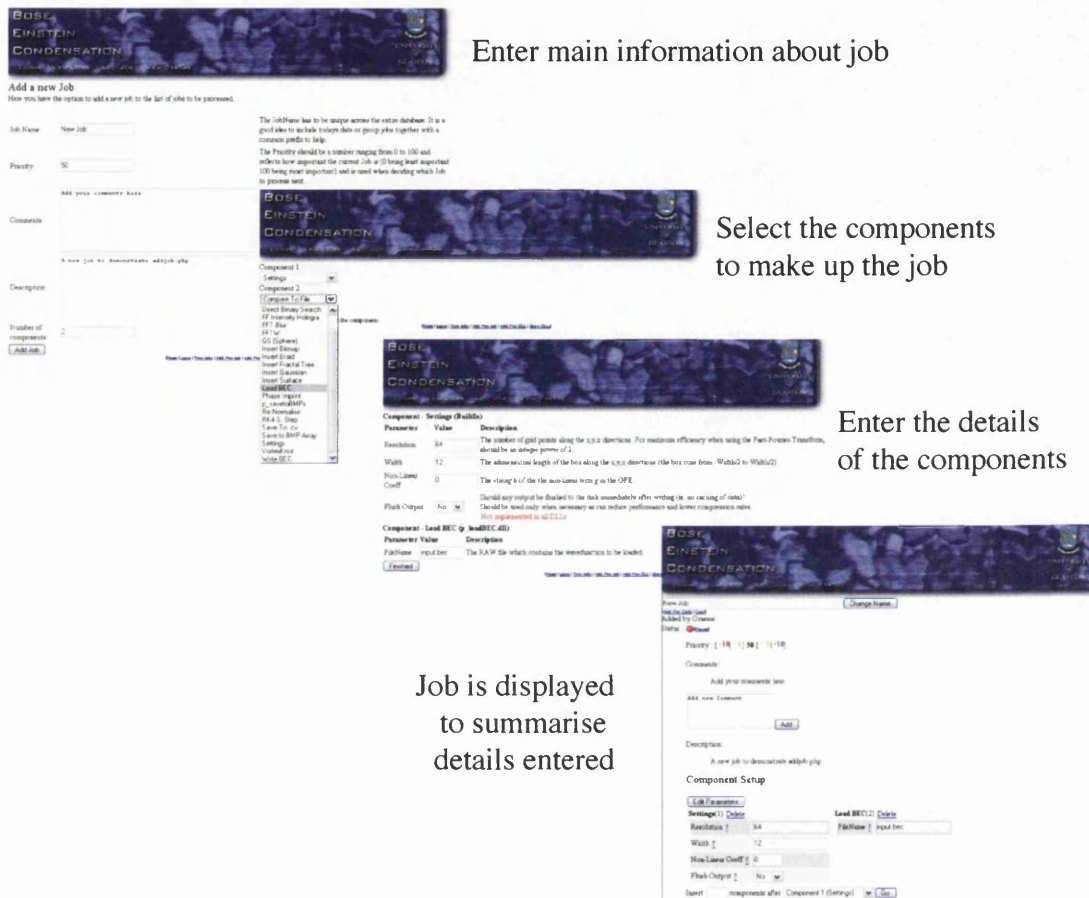


Figure A.3: Screen captures of the new job wizard which helps you create a new job. The wizard has three main steps: Main details of the job, the required components of the job and the parameters of each of those components. Once the wizard is completed the resulting job is displayed.

4. In the component selection page, select the steps which the simulation requires (in the order they should be evaluated) from the list of available steps,
5. The final page of the wizard requests the values for every parameter of each of the steps. If the required value is not the default, change it here. All these values can be changed later,
6. Once all the details have been entered you should be taken to the new job's page where the details can be edited if required and the job can be added to the list of jobs to be processed.

It is important to remember that **jobs are not added to the list of jobs to be executed immediately**<sup>4</sup> (see section A.4.4 for details).

If there is an existing job which is closely related to the job you wish to create, you can copy the existing job and then edit the details which need to be updates. For the majority of jobs, you can create a copy either at the same hierarchical level, or as a child of the original job. Whichever way you choose to copy the job, it is possible to move it to anywhere within the hierarchy (see section A.4.3 for details). If you are able to copy an existing job, the "Copy" or "Copy as Child" options will be presented in either the list view of the available jobs, or in the job details page of the original job.

**Note.** Editing a previously copied job has no effect on the copied version. The copied version is copied as a snapshot of the original at the moment of copy and no relationship between the two are maintained.

---

<sup>4</sup>They are set to "Stopped" by default to allow last-minute edits and a final check of the details.

### A.4.3 Editing jobs

If you wish to change the details of a job, you need to go to the details page for that job and **ensure the job status is stopped** (Otherwise you can only change the priority and add comments). When the job is stopped, the details page should look similar to Figure A.4.

#### Editing the name

The large text edit box at the top of the page should contain the full name of the job including any parent jobs separated by backslash (\) characters. If you wish to change the name of the job without affecting its position in the hierarchy, edit only the part of the name after the last backslash and click the button to update the name.

It is also possible to move the job around in the hierarchy using the rename option. By changing the full name of the job, that job can be moved. Listed below are several examples to show how to effectively rename a job.

#### Rename a job without moving

`\Parent1\Parent2\Job` → `\Parent1\Parent2\New_Name` or  
`\Parent1\Parent2\Job` → `New_Name`

#### Move a job to become a child of its grandparent

`\Parent1\Parent2\Job` → `\Parent1\Job` or  
`\Parent1\Parent2\Job` → `...\Job`

#### Move a job to become one of the main jobs

`\Parent1\Parent2\Job` → `\Job`



Figure A.4: Comparison of the view of a job which is ready to be processed (top) and so the job can not be edited and a stopped job (bottom) which can be edited.

### Move a job into another position in the hierarchy

`\Parent1\Parent2\Job → \Parent1\New_Parent\Job` or

`\Parent1\Parent2\Job → ..\New_Parent\Job` or

When moving the job around within the hierarchy, the parents which you move the job into don't have to already exist, empty jobs are created to ensure the hierarchy ends up being of the form of the new name<sup>5</sup>.

### Editing the setup

It is possible to alter the steps which the simulation is going to take by deleting any of the steps by clicking on the delete option next to that step. Deleting a step is permanent and irrecoverable and inserting the same step in the same place does not restore any non-default settings which were entered.

You can also add a new step into any position within the simulation by using the insert options at the bottom of the page<sup>6</sup>.

Unfortunately steps currently can not be moved around within the simulation without deleting and re-inserting either that step, or the steps around it.

---

<sup>5</sup>This also means that if a parent is misspelt when moving a job, many additional empty parent jobs can be created. Care must be taken to ensure that the name of existing jobs are entered correctly.

<sup>6</sup>The built-in step settings should not be present more than once within a simulation. If it is, only the instance closest to the beginning is used for the entire simulation.

---

## Changing the parameters

When a job is in a stopped state, all the available parameters become editable (either text boxes or drop down menus depending on the parameter). When changing the parameters, it is important to enter the type of data the server is expecting, i.e. not entering the word "high" into a parameter which is expecting a number. There are three types of data which can be entered using a text box:

**String** any text, is generally used for descriptions and filenames. If used for a filename, care needs to be taken that characters which can not be used in files are avoided ( '\*', '?', '\', '/', ':')

**Float** any number. Numbers can be entered in scientific notation (1E-6 is allowed)

**Int** integer numbers.

There is also an extension to the numeric types (threeintpoint and threepointt) which are used to describe three numbers in one parameter (used for defining three-dimensional parameters). There are three possible methods of defining a threeintpoint or threepoint number:

**"12"** If only one number is present, all three numbers are assumed to be the same and equal to the entered number.

**"12,6"** If two numbers are present, separated by a comma, the first two numbers (often  $x$  and  $y$ ) are assumed to be the same and equal to the first number entered while the third number of the threeintpoint or threepoint is assumed to be the second number entered.

“12,6,3” If three numbers have been entered, they are used as the three numbers of the threeintpoint or threepoint.

To check which type is expected for any of the parameters, click on the question mark next to the parameter which will pop-up a small window explaining the details of the parameter including its type and description.

#### A.4.4 Starting jobs

When a new job is created it is not added to the list of jobs to be processed automatically; it is placed into a state named stopped.

##### States of a job

**Stopped** Default for a new job. Allows all the details to be edited and changed before starting

**Ready** The job is in the list of jobs to be processed. Changes (apart from priority changes and comments) are locked-out.

**Paused** The job is ready to be processed, but is temporarily removed from the list of to be processed.

**Started** The job is being processed by a client computer.

**Finished** The job has been processed by a computer and the results have been reported.

To add a job to the list of jobs to be processed, the state of the job has to be changed to “**Ready**”. This can be done by clicking on “**Start**” which is next to the status indicator for the job (this can be found either on the

---

details page for the job, or in the listing of the jobs parent). If the current state isn't Stopped, the “**Start**” option is not available, see below for the required tasks to start the job:

**Stopped** : Click **Start**

**Paused** : Click **Resume**

**Started** : Click **Force Restart** → **Start**

**Finished** : Click **Restart** → **Start**

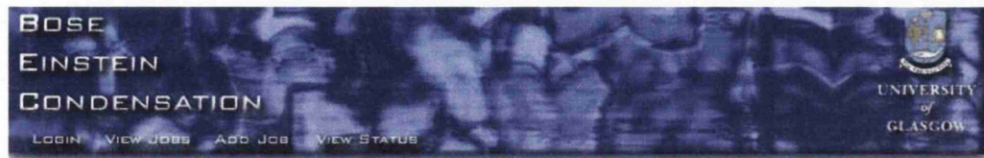
It is important to note that the job will not start processing immediately, a free client needs to request a new job and so it is not unusual for a job to remain at “**Ready**” for a time.

#### A.4.5 Stopping jobs

A job status of “Stopped” means the job will not be downloaded by a client and all the parameters of the job can be edited. Stopping a job results in it being removed for the list of jobs to be run. Stopping a currently running job does not stop the client running that job. The server does not inform the client that it has been stopped and the client will continue running the job until it is finished.

All jobs can be stopped at any time by clicking “Stop” next to the status of a job.





Displays a list of the available components and their details.

ShortName	FileName	Long Name	Description	No. Of Parameters
Compare To File	p_comparetofile.dll	Compare BEC Wavefunction to Wavefunction file	Compares the current BEC wavefunction to a previously stored wavefunction. It is possible to compare the complex wavefunction, probability density pattern or phase. The Comparison is a sum of the differences over the range of the wavefunction.	2
FF Intensity Hologra	p_int_hologram.dll	Fourier Filtering Intensity Hologram	Calculates an intensity hologram which would produce the current wavefunction after a Fourier transform	2
FFT Blur	p_fftblur.dll	Fourier Gaussian Blur	Blurs the BEC using Fourier blurring	5

Figure A.5: Screen capture showing a section of the list of available components and a description detailing their function. This can be accessed from almost every page by clicking on “List DLLs”.

## A.5 Components

Each individual step which can be taken in a simulation is performed by a “component”. To make using components easier, the server maintains a list of the components and the parameters which they require.

A list of the currently installed components can be found by selecting “List DLLs”<sup>7</sup> from the links at the bottom of each page. This should result in a page similar to the one shown in Figure A.5. If you are unsure of the function of a particular component, this page can be useful in identifying the function of a component as the description listed here often gives useful detail as to what the component does.

### A.5.1 Parameters

As mentioned, each component has parameters which it requires for successful operation. These can range from simple yes/no questions through numeric parameters and filenames to mathematical functions. Unfortunately because the server cannot know exactly what the DLL requires in its parameters it is limited to only a handful of datatypes:

**string** A number of ASCII text characters. This should only be used to represent alphanumeric characters. If binary data is required, it should be stored in a file and uploaded.

**float** A floating point number. This can be represented as a decimal number or in scientific notation (for example, -1E-15 is a valid float number).

If a non numeric entry is detected, the server returns an error

---

<sup>7</sup>This links to listdlls.php and if no link is present, the URL can be entered manually.

---

**int** An integer number. If a non integer number or non numeric entry is detected, the server returns an error

**file** A user defined file. This requires the user to upload a file from their own computer which is stored on the server and given to the client when required. If a file is not uploaded, no error is given. Check with the documentation of any specific component as to the file requirements and consequences.

**threepoint** Three floating point numbers. The threepoint datatype is used to represent three numbers in one parameter (often  $x,y$  and  $z$  data). There are three ways to enter a threepoint: a single number (e.g. "12.5") which results in all three numbers being equal to the number given, two numbers separated by a comma (e.g. "12.6, 6.3") which results in the first two numbers of the threepoint being equal to the first given number and the last threepoint being equal to the second number, and finally three numbers separated by commas, which results in each of the threepoint numbers being equal to the three given numbers.

**threeintpoint** Three integer numbers. This has the same behavior as the threepoint datatype, except each of the numbers must be an integer.

## A.5.2 Adding new components

If you have a component which is not represented by the server, you can add the details to the list of available components. To begin the add component wizard click on "Add DLL" link at the bottom of any page.

**Warning.** It is very important to ensure the datatypes and parameter names you enter here are correct and can lead to undefined behavior and loss of data if they are incorrectly represented. If you are unsure get help from the developer of the component.

The first step in adding a new component is to give it a name and description to help identify it and give other users an idea of what it should be used for. All components are visible and usable by other users, so please ensure that the name and description are meaningful and if the component is very specific or in testing please mention that.

Once the name and description have been entered, find the DLL file which is associated with the component and upload it to the server<sup>8</sup>. You will also need to enter the number of parameters required by the component.

In the next page enter the details of the individual parameters of the component. The order in which you enter them here defines the order in which they will be displayed to the user. If there are many parameters, try to group them together in terms of function and place the most important or relevant to the top.

It is often useful to select good default values to reduce the number of edits the user will have to perform. Remember to ensure the default value is of the correct form. Defaults are required for all the datatypes except the "file" datatype.

After you have entered all the details, click "check parameters" to verify all the details entered are correct and then click "add parameters" which will

---

<sup>8</sup>Check with the administrator if DLL syncing is enabled on the server to find out if this step is required

---

update the component to include the parameters. The component can now be used in a simulation.

## A.6 Client software

The client software is the software which performs the calculations of a simulation. The client software can be seen as a set of two units: the DLL files which perform the individual steps of a simulation and a shell program which coordinates the individual DLLs and communicates with the server to get new jobs.

### A.6.1 Installing

Once you have obtained the software package, it can be installed by unzipping the package and copying the folder to its desired location. No other installation sets are required. The other method of using the client software is to use a centralised version shared across a network. Check with the administrator if such a setup is enabled as it has several advantages over local installation onto a computer.

Once the software is installed (or you have access to a shared installation) ask an administrator to ensure the settings are correct.

### A.6.2 Running

To run the application, double click on its icon. If no custom options are detected, it will use its built-in defaults. Once it has started, you should see

---

all of the windows<sup>9</sup>. The client does not automatically start processing any jobs on startup (see section A.6.5).

### A.6.3 Options

Selecting “Parameters” from the “File” menu brings up the options dialog as shown in Figure A.6. The first seven of these options are the various URLs which are used to communicate with the server and retrieve jobs and files. These should only be changed if you are sure of the new values. Below the URL options is the username and password the client should use to connect to the server. By default this is a “Machine” user. If however you would like the client to use your own username or a client specific username, enter the details here.

The directory options allow you to change where the client looks for the DLLs associated with each component and where it should store any results. Altering the DLL location can allow you to alter the DLLs and run specific jobs using the altered versions without affecting the other clients using the common DLL pool.

The final options relate the program as a whole. The process thread priority allows you to alter the priority of the processing parts of the program. This allows the processing of jobs not to affect other processes by only using spare computing time (if set to “Idle”). By increasing the priority, the client program can be made to run at the expense of other programs<sup>10</sup>.

---

<sup>9</sup>The windows may appear on top of each other and to see them all you may have to move them.

<sup>10</sup>This is not recommended as the computer can become unresponsive if the client uses too much computing time.

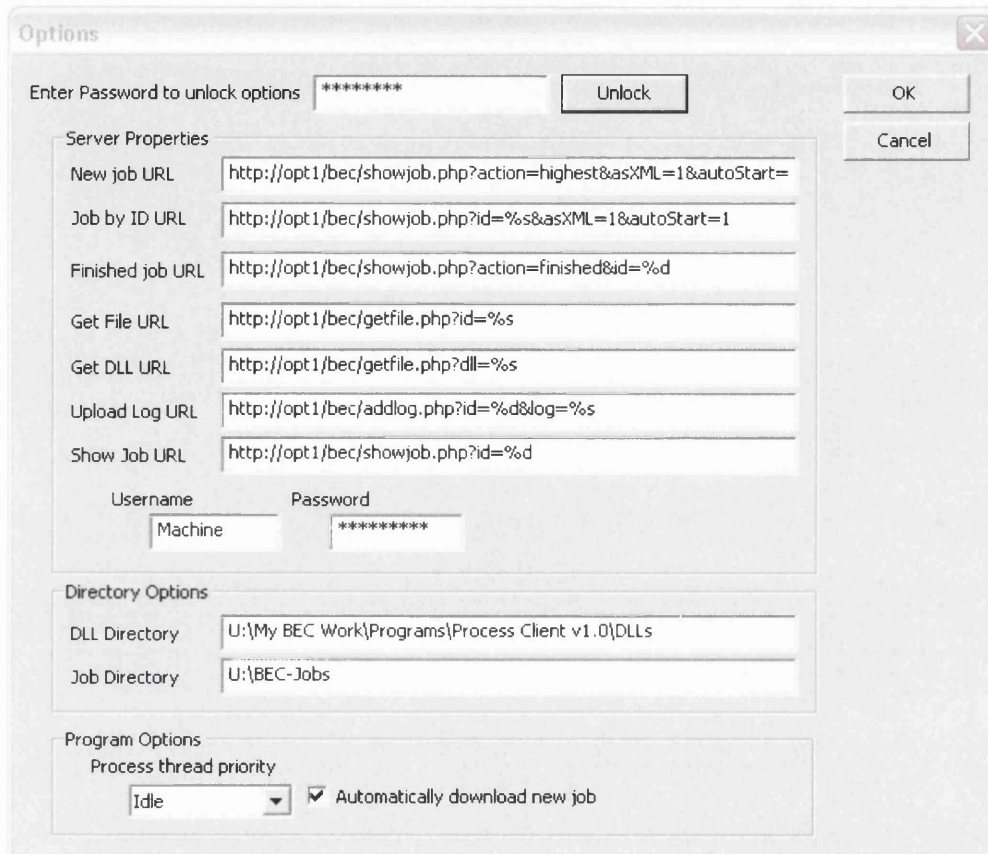


Figure A.6: The options window which allows you to alter the parameters which the client uses. Many of the parameters are the URLs which are used to communicate with the server and any changes to these can lead to unexpected results. Other options include the locations of the DLLs and the storage location of any results as well as the priority the system should run at on the local computer.

The “automatically download new job” option enables the client to try and download a new job once its finished with any current job. enabling this option results in the client running unattended and will continue to process jobs until there are no new jobs available on the server.

#### A.6.4 Windows

The client software is made up of 4 windows:

**Main Window** The main window contains a representation of the current simulation which includes all the steps and parameters and gives the user an indication of the progress of the simulation.

**Job Queue** The job queue window holds the list of jobs which are to be run by this client. This may include specified jobs from a server, jobs from a local file as well as the most appropriate job from the server.

**Component Log** The component log keeps track of reports from the individual components and any errors which they may have.

**Application Log** The application log tracks events and errors given by the main application.

#### Main Window

The main window shows a representation of the current (and previously run) jobs. The main portion of the window is made up of a list of the jobs which have been run and the currently running job. Each of these jobs can be expanded by clicking on the plus symbol next to their name which reveals



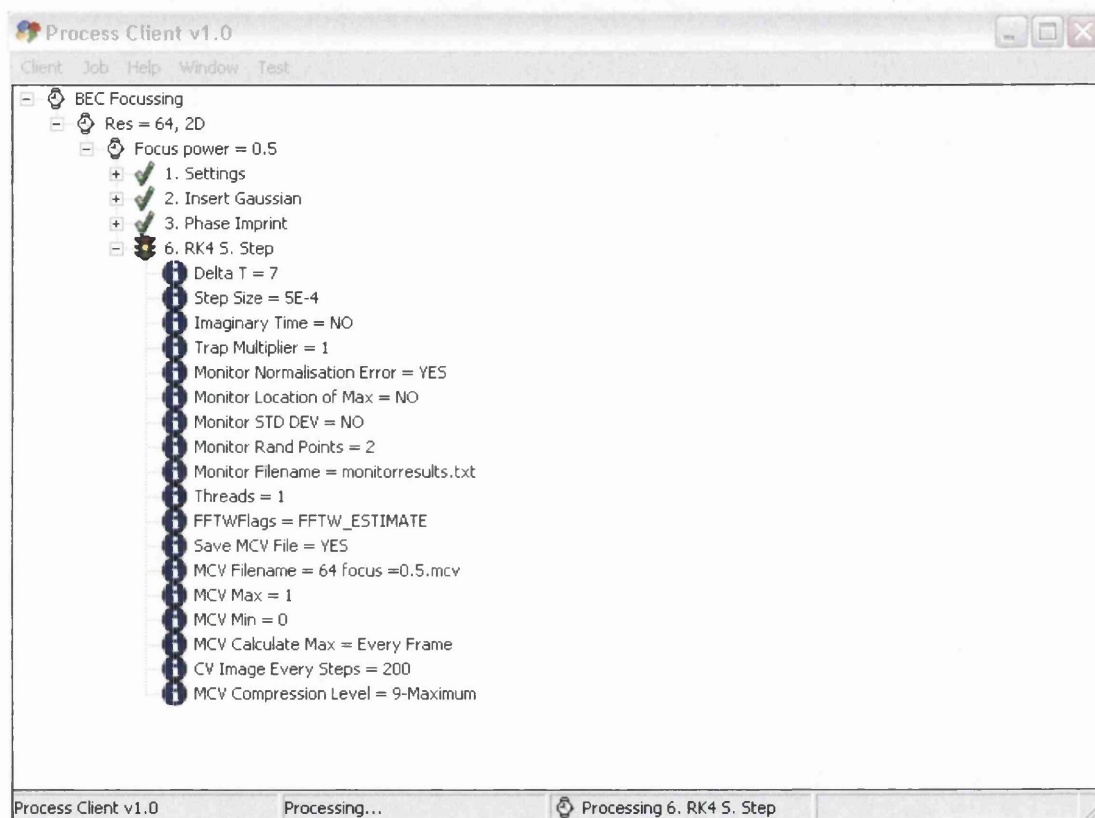


Figure A.7: The main window of the client user interface. A large portion of the window is a list of the jobs which the client is processing. This list can be expanded to show the steps of the job and each of the steps can be expanded to show the parameters for that step.

the individual steps used by the job. Each of these can also be expanded to show the individual parameters for that step. As the steps are performed, the symbol next to the step changes to show they are being processed and once they are completed, the symbol changes once again to reflect this.

Along the bottom of the main window shows the current status of the job which includes a progress bar to indicate the progress of the current component.

### **Job Queue Window**

The Job Queue window contains a list of the URLs of jobs which the client should process. This list can also include local files<sup>11</sup> and keeps track of jobs which the client has processed since it began running. The controls in this window allow you to manipulate this list by adding new jobs, removing jobs and manipulating the order in which they are processed.

When the client has finished processing a job, it scans through the list from top to bottom to find the next job which is waiting to be processed. By manipulating the order of the list you can specify which jobs should be processed first. Because this list is only checked when the job is finished, moving a waiting job to a position above the currently running job has no effect on the current job. Equally, rearranging the previously processed jobs alters nothing about those jobs.

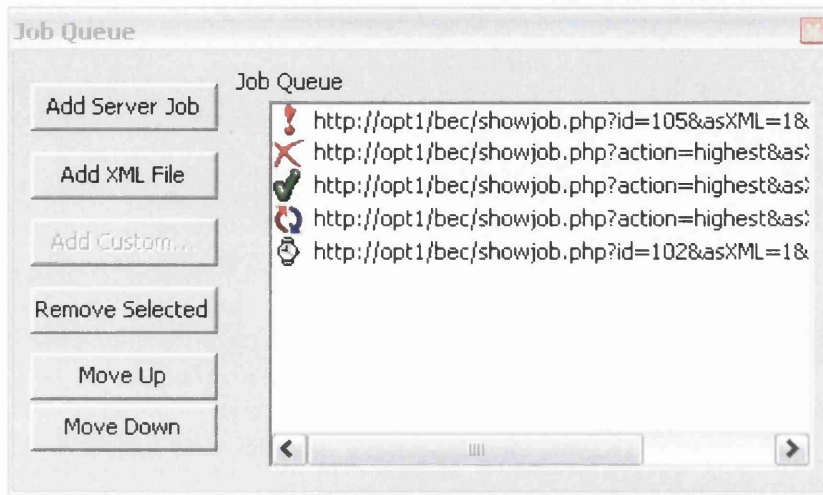
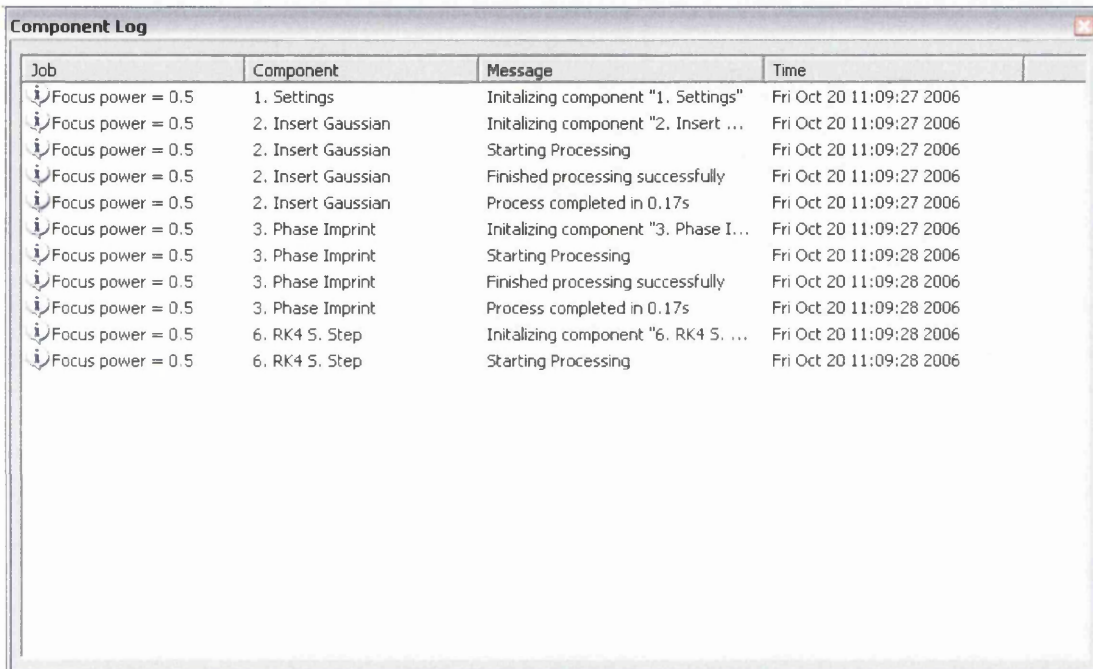


Figure A.8: The job queue window which contains the list of jobs which are to be processed and those which have already been processed. The symbols next to the URL for the job illustrates the current status of the job (from top to bottom : warning - a non-fatal error occurred while processing the job, error - a fatal error occurred which stopped the simulation from continuing, completed - the job finished processing successfully, current processing, waiting)



The screenshot shows a window titled "Component Log" with a table containing the following data:

Job	Component	Message	Time
Focus power = 0.5	1. Settings	Initializing component "1. Settings"	Fri Oct 20 11:09:27 2006
Focus power = 0.5	2. Insert Gaussian	Initializing component "2. Insert ..."	Fri Oct 20 11:09:27 2006
Focus power = 0.5	2. Insert Gaussian	Starting Processing	Fri Oct 20 11:09:27 2006
Focus power = 0.5	2. Insert Gaussian	Finished processing successfully	Fri Oct 20 11:09:27 2006
Focus power = 0.5	2. Insert Gaussian	Process completed in 0.17s	Fri Oct 20 11:09:27 2006
Focus power = 0.5	3. Phase Imprint	Initializing component "3. Phase I..."	Fri Oct 20 11:09:27 2006
Focus power = 0.5	3. Phase Imprint	Starting Processing	Fri Oct 20 11:09:28 2006
Focus power = 0.5	3. Phase Imprint	Finished processing successfully	Fri Oct 20 11:09:28 2006
Focus power = 0.5	3. Phase Imprint	Process completed in 0.17s	Fri Oct 20 11:09:28 2006
Focus power = 0.5	6. RK4 S. Step	Initializing component "6. RK4 S. ..."	Fri Oct 20 11:09:28 2006
Focus power = 0.5	6. RK4 S. Step	Starting Processing	Fri Oct 20 11:09:28 2006

Figure A.9: The component log keeps track of events related to the individual components within a simulation, including start and end times, errors, warnings and results returned from the component.

## Component Log

The component log (shown in Figure A.9) keeps a log of all the component related events, including initialisation, starting and finishing processing as well as any component specific events raised during processing. If a component does not behave as expected, the component log can give useful insight into the progression of a component. The component log also keeps track of the processing time of each component which allows you to profile a given component.

## Application Log

The application log (shown in Figure A.10) records all the non-component related events. This can include the application starting up, communications with the server, failures out-with components. If the client is not able to collect a new job or if it fails before processing anything, the application log can give insight into the behavior of the application before it failed.

### A.6.5 Running jobs

Running jobs on the client software involves adding a job which you'd like to be run into the job queue. This queue can be seen in the job queue window (Window → Job Queue) which can be seen in Figure A.8. The queue contains the list of jobs which have been run since the software was started, the currently running job and any jobs to be ran in the future. The symbol next to the job describes the status of it and can change from waiting

---

<sup>11</sup>Local files are designated with a file:// prefix.

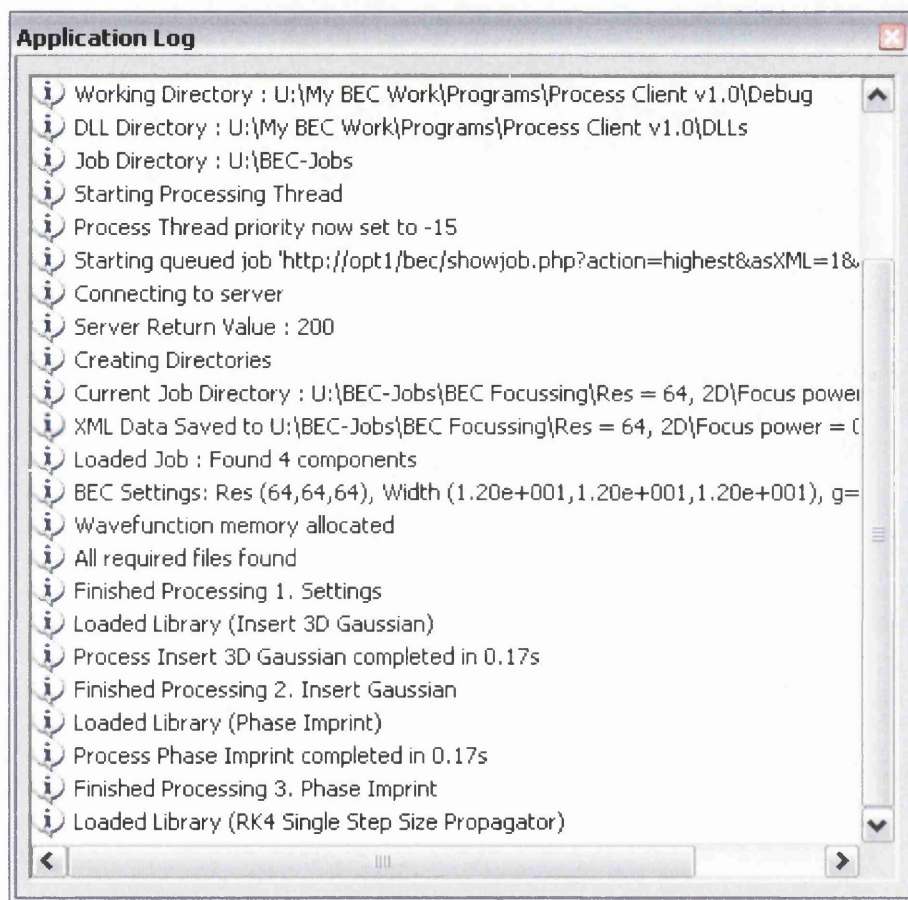


Figure A.10: The application log keeps track of events within the client which are not linked to components (for example, communication with the server, loading of user settings, initialisation of the client).

---

to processing to a representation of the result of the processing. To the left of the list there are a number of buttons which allow you to edit the queue. The most useful of these is the Add Server Job and Add XML File.

The other available buttons control the items within the list to order them appropriately. The list is run from the top to the bottom and jobs to be run can be arranged above jobs which have already completed<sup>12</sup>.

### **From a web-server**

To add a job which is stored on the server, click the Add Server Job button which brings up an additional dialog which allows you to choose either the numeric ID of the specific job you wish to run<sup>13</sup>, or an option to get the most appropriate job. If you know the JobID of a specific job you'd like to run, uncheck the get highest priority option and enter the JobID in the text box. Otherwise click OK and let the server choose which job should be processed next. When you select to get the highest priority job the server enters a loop to retrieve jobs periodically. This ensures that if there are multiple jobs which need to be processed, the client will work through them. This behavior can be changed through the preferences (see section A.6.3).

### **From a locally stored job**

If you have created an XML file to describe the job you wish to run, you can import this directly using the Add XML File. It is also possible to re-run

---

<sup>12</sup>By placing a waiting job above the currently processing job does not alter the status of the currently running job, which will complete its processing before beginning the waiting job.

<sup>13</sup>Is available from the server details page of the job of interest

---

a simulation by adding the JobDef.xml file associated with the simulation directly into the job queue, bypassing the server altogether.

### A.6.6 Results

When a component is saving files, it places them into the directory associated with the job. This directory is located in the Jobs Directory which can be specified by an administrator (see the user preferences to find out where this file is stored<sup>14</sup>) and the directory structure is similar to the hierarchy of the jobs stored on the server. In the folder of each job, the JobDef.xml file which holds the details of the job is stored (which allows you to rerun the job at any time using the locally stored copy) along with any required files which have been downloaded from the server.

When a component saves a file, by default this file should be stored in the same directory. By placing this directory on a centralised network drive, all the clients can access the same directory and you can access the resulting files from anywhere within the network. For maximum compatibility, the shared network drive should be mapped to a drive name (as opposed to a UNC name)<sup>15</sup>.

### A.6.7 Running unattended

The client has been designed to run unattended once it has the highest priority job in the queue (see above). If the highest priority job has been run and the queue has been completed, the client will poll the server for new

---

<sup>14</sup>alternatively click GoTo → Directory.

<sup>15</sup>If you are not sure about doing this ask an administrator to help.



jobs periodically. If the server has no new jobs, the client will wait longer each time upto a maximum wait time of half an hour in order to reduce load on the server. Once a job has been collected, the wait timer is reset to 15 seconds and the process repeats itself.

# Bibliography

- [1] W. G. Kaenders, F. Lison, I. Müller, A. Richter, R. Wynands, and D. Meschede, “Refractive components for magnetic atom optics,” *Phys. Rev. A* **54**, 5067–5075 (1996).
- [2] S. R. Muniz, S. D. Jenkins, T. A. B. Kennedy, D. S. Naik, and C. Raman, “Axicon lens for coherent matter waves,” *Opt. Express* **14**, 8947 (2006).
- [3] M. Kozuma, L. Deng, E. W. Hagley, J. Wen, R. Lutwak, K. Helmerson, S. L. Rolston, and W. D. Phillips, “Coherent Splitting of Bose-Einstein Condensed Atoms with Optically Induced Bragg Diffraction,” *Phys. Rev. Lett.* **82**(5), 871–875 (1999).
- [4] M. Morinaga, M. Yasuda, T. Kishimoto, F. Shimizu, J.-i. Fujita, and S. Matsui, “Holographic Manipulation of a Cold Atomic Beam,” *Phys. Rev. Lett.* **77**(5), 802–805 (1996).
- [5] D.-I. Choi and Q. Niu, “Bose-Einstein Condensates in an Optical Lattice,” *Phys. Rev. Lett.* **82**(10), 2022–2025 (1999).
- [6] J. Hecker Denschlag, J. E. Simsarian, H. Häffner, C. McKenzie, A. Browaeys, D. Cho, K. Helmerson, S. L. Rolston, and W. D. Phillips,

- 
- “A Bose-Einstein condensate in an optical lattice ,” *J. Phys. B* **35**, 3095–3110 (2002). [cond-mat/0206063](#).
- [7] K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle, “Bose-Einstein condensation in a gas of sodium atoms,” *Phys. Rev. Lett.* **75**, 3969–3973 (1995).
- [8] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell, “Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor,” *Science* **269**, 198–201 (1995).
- [9] B. S. Kademani, G. Surwase, S. Anil, and K. Vijai, “Mapping of literature on Bose Einstein condensation,” <http://eprints.rclis.org/archive/00007492/> (2006).
- [10] J. Courtial and M. J. Padgett, “Performance of a cylindrical lens mode converter for producing Laguerre-Gaussian laser modes,” *Opt. Commun.* **159**, 13–18 (1999).
- [11] J. Courtial, “Fractal multiple reduction imaging lasers,” *Opt. Commun.* **174**, 235–241 (2000).
- [12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, chap. 16 (Cambridge University Press, Cambridge, United Kingdom, 1992).
- [13] B. M. Caradoc-Davies, “Vortex Dynamics in Bose-Einstein Condensates,” Ph.D. thesis, University of Otago, Dunedin, New Zealand (2000).

- 
- [14] <http://www.ni.com/labview/>.
- [15] <http://msdn.microsoft.com/vstudio/>.
- [16] F. Dalfovo, C. Minniti, S. Stringari, and L. Pitaevskii, “Nonlinear dynamics of a Bose condensed gas,” *Phys. Lett. A* **227**, 259–264 (1997).  
`cond-mat/9612175`.
- [17] A. Griffin, “Conserving and gapless approximations for an inhomogeneous Bose gas at finite temperatures,” *Phys. Rev. B* **53**(14), 9341–9347 (1996).
- [18] J. Weiner, V. S. Bagnato, S. Zilio, and P. S. Julienne, “Experiments and theory in cold and ultracold collisions,” *Rev. Mod. Physics* **71**, 1–85 (1999).
- [19] P. A. Ruprecht, M. J. Holland, K. Burnett, and M. Edwards, “Time-dependent solution of the nonlinear Schrödinger equation for Bose-condensed trapped neutral atoms,” *Phys. Rev. A* **51**, 4704–4711 (1995).
- [20] A. Görlitz, J. M. Vogels, A. E. Leanhardt, C. Raman, T. L. Gustavson, J. R. Abo-Shaeer, A. P. Chikkatur, S. Gupta, S. Inouye, T. Rosenband, and W. Ketterle, “Realization of Bose-Einstein Condensates in Lower Dimensions,” *Phys. Rev. Lett.* **87**(13), 130,402 (2001).
- [21] E. A. Sziklas and A. E. Siegman, “Mode calculations in unstable resonators with flowing saturable gain. 2: Fast Fourier transform method,” *Appl. Opt.* **14**, 1874–1889 (1975).

- 
- [22] C. C. Davis, *Lasers and Electro-Optics*, chap. 16 (Cambridge University Press, 1996).
- [23] G. Whyte, P. Öhberg, and J. Courtial, “Transverse laser modes in Bose-Einstein condensates,” *Phys. Rev. A* **69**, 53,610 (2004).
- [24] L. Allen, M. W. Beijersbergen, R. J. C. Spreeuw, and J. P. Woerdman, “Orbital angular momentum of light and the transformation of Laguerre-Gaussian modes,” *Phys. Rev. A* **45**, 8185–8189 (1992).
- [25] M. W. Beijersbergen, L. Allen, H. E. L. O. van der Veen, and J. P. Woerdman, “Astigmatic laser mode converters and transfer of orbital angular momentum,” *Opt. Commun.* **96**, 123–132 (1993).
- [26] A. E. Siegman, *Lasers*, chap. 14–23 (University Science Books, Mill Valley, California, 1986).
- [27] G. Whyte, P. Öhberg, and J. Courtial, “Fourier transforming a trapped BoseEinstein condensate by waiting a quarter of the trap period: simulation and applications,” *New J. Phys.* **8**, 196 (2006).
- [28] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform: with Applications in Optics and Signal Processing*, chap. 4 (John Wiley, New York, 2001).
- [29] A. C. McBride and F. H. Kerr, “On Namias’s Fractional Fourier Transforms,” *IMA J Appl Math* **39**(2), 159–175 (1987).

- 
- [30] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform: with Applications in Optics and Signal Processing*, chap. 9 (John Wiley, New York, 2001).
- [31] B. C. Platt and R. Shack, “History and principles of Shack-Hartmann wavefront sensing,” *J. Refract. Surg.* **17**, 573–577 (2001).
- [32] S. Stringari, “Collective excitations of a trapped Bose-condensed gas,” *Phys. Rev. Lett.* **77**, 2360–2363 (1996).
- [33] P. Öhberg, E. L. Surkov, I. Tittonen, S. Stenholm, M. Wilkens, and G. V. Shlyapnikov, “Low-energy elementary excitations of a trapped Bose-condensed gas,” *Phys. Rev. A* **56**, R3346–R3349 (1997).
- [34] M. Fliesser, A. Csordás, P. Szépfalusy, and R. Graham, “Hydrodynamic excitations of Bose condensates in anisotropic traps,” *Phys. Rev. A* **56**, R2533–R2536 (1997).
- [35] <http://graphics.stanford.edu/software/volpack/>.
- [36] P. Lacroute and M. Levoy, “Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation,” in *Proc. SIGGRAPH '94*, pp. 451–458 (1994).
- [37] D. L. Feder, C. W. Clark, and B. I. Schneider, “Vortex Stability of Interacting Bose-Einstein Condensates Confined in Anisotropic Harmonic Traps,” *Phys. Rev. Lett.* **82**, 4956–4959 (1999). [cond-mat/9904269](https://arxiv.org/abs/cond-mat/9904269).

- 
- [38] J. Leach, M. R. Dennis, J. Courtial, and M. J. Padgett, “Vortex knots in light,” *New J. Phys.* **7**, 55 (2005).
- [39] L. Dobrek, M. Gajda, M. Lewenstein, K. Sengstock, G. Birkl, and W. Ertmer, “Optical generation of vortices in trapped Bose-Einstein condensates,” *Phys. Rev. A* **60**(5), R3381–R3384 (1999).
- [40] J. Ruostekoski and Z. Dutton, “Engineering vortex rings and systems for controlled studies of vortex interactions in Bose-Einstein condensates,” *Phys. Rev. A* **72**(6), 063,626 (2005). [cond-mat/0507032](#).
- [41] J. Leach, M. J. Padgett, S. M. Barnett, S. Franke-Arnold, and J. Courtial, “Measuring the orbital angular momentum of a single photon,” *Phys. Rev. Lett.* **88**, 257,901 (2002).
- [42] K. W. Madison, F. Chevy, W. Wohlleben, and J. Dalibard, “Vortex Formation in a Stirred Bose-Einstein Condensate,” *Phys. Rev. Lett.* **84**, 806–809 (2000).
- [43] J. R. Abo-Shaer, C. Raman, J. M. Vogels, and W. Ketterle, “Observation of Vortex Lattices in Bose-Einstein Condensates,” *Science* **292**, 476–479 (2001).
- [44] D. R. Tilley and J. Tilley, *Superfluidity and Superconductivity* (Adam Hilger, Bristol, 1990).
- [45] F. Chevy, K. W. Madison, V. Bretin, and J. Dalibard, “Interferometric detection of a single vortex in a dilute Bose-Einstein condensate,” *Phys. Rev. A* **64**, 31,601 (2001).

- 
- [46] G. Molina-Terriza, J. P. Torres, and L. Torner, “Management of the Angular Momentum of Light: Preparation of Photons in Multidimensional Vector States of Angular Momentum,” *Phys. Rev. Lett.* **88**, 013,601 (2002).
- [47] E. Lundh, “Multiply quantized vortices in trapped Bose-Einstein condensates,” *Phys. Rev. A* **65**, 043,604 (2002).
- [48] T. P. Simula, S. M. M. Virtanen, and M. M. Salomaa, “Stability of multi-quantum vortices in dilute Bose-Einstein condensates,” *Phys. Rev. A* **65**, 033,614 (2002).
- [49] M. D. Lukin, “Colloquium: Trapping and manipulating photon states in atomic ensembles,” *Rev. Mod. Phys.* **75**, 457–472 (2003).
- [50] L. Allen, S. M. Barnett, and M. J. Padgett, eds., *Orbital Angular Momentum* (Institute of Physics Publishing, London, 2003).
- [51] A. L. Fetter and A. A. Svidzinsky, “TOPICAL REVIEW: Vortices in a trapped dilute Bose-Einstein condensate,” *Journal of Physics: Condensed Matter* **13**, 135 (2001).
- [52] K. O’Holleran, M. J. Padgett, and M. R. Dennis, “Topology of optical vortex lines formed by the interference of three, four, and five plane waves,” *Opt. Ex.* **14**, 3039–3044 (2006).
- [53] C. Tamm and C. O. Weiss, “Bistability and optical switching of spatial patterns in a laser,” *J. Opt. Soc. Am. B* **7**, 1034–1038 (1990).



- 
- [54] T. Isoshima and K. Machida, “Vortex stabilization in Bose-Einstein condensate of alkali-metal atom gas,” *Phys. Rev. A* **59**, 2203–2212 (1999). [cond-mat/9807250](#).
- [55] H. Wei, X. Xue, J. Leach, M. J. Padgett, S. M. Barnett, S. Franke-Arnold, E. Yao, and J. Courtial, “Simplified measurement of the orbital angular momentum of photons,” *Opt. Commun.* **223**, 117–122 (2003).
- [56] J. Javanainen and M. Y. Ivanov, “Splitting a trap containing a Bose-Einstein condensate: Atom number fluctuations,” *Phys. Rev. A* **60**, 2351–2359 (1999).
- [57] Y. Torii, Y. Suzuki, M. Kozuma, T. Sugiura, T. Kuga, L. Deng, and E. Hagley, “Mach-Zehnder Bragg interferometer for a Bose-Einstein condensate,” *Phys. Rev. A* **61**, 041,602 (2000).
- [58] Y. Kawaguchi and T. Ohmi, “Splitting instability of a multiply charged vortex in a Bose-Einstein condensate,” *Phys. Rev. A*. **70**(4), 043,610 (2004). [cond-mat/0402553](#).
- [59] D. M. Giltner, R. W. McGowan, and S. A. Lee, “Atom interferometer based on Bragg scattering from standing light waves,” *Phys. Rev. Lett.* **75**, 2638 (1995).
- [60] U. V. Poulsen and K. Mølmer, “Quantum beam splitter for atoms,” *Phys. Rev. A* **65**, 33,613 (2002).
- [61] K.-P. Marzlin and W. Zhang, “Vortex Coupler for Atomic Bose-Einstein Condensates,” *Phys. Rev. Lett.* **79**, 4728–4731 (1997).

- 
- [62] K.-P. Marzlin and W. Zhang, “Laser-induced rotation of a trapped Bose-Einstein Condensate,” *Phys. Rev. A* **57**, 3801 (1998).
- [63] K.-P. Marzlin and W. Zhang, “Quantized circular motion of a trapped Bose-Einstein condensate: Coherent rotation and vortices,” *Phys. Rev. A* **57**, 4761 (1998).
- [64] J. Denschlag, J. E. Simsarian, D. L. Feder, C. W. Clark, L. A. Collins, J. Cubizolles, L. Deng, E. W. Hagley, K. Helmerson, W. P. Reinhardt, S. L. Rolston, B. I. Schneider, and W. D. Phillips, “Generating Solitons by Phase Engineering of a Bose-Einstein Condensate,” *Science* **287**, 97–101 (2000).
- [65] S. Inouye, M. R. Andrews, J. Stenger, H.-J. Miesner, D. M. Stamper-Kurn, and W. Ketterle, “Observation of Feshbach resonances in a Bose-Einstein condensate,” *Nature* **392**, 151–154 (1998).
- [66] J. E. Molloy and M. J. Padgett, “Lights, action: optical tweezers,” *Contemp. Phys.* **43**, 241–258 (2002).
- [67] M. Reicherter, T. Haist, E. U. Wagemann, and H. J. Tiziani, “Optical particle trapping with computer-generated holograms written on a liquid-crystal display,” *Opt. Lett.* **24**, 608–610 (1999).
- [68] J. Liesener, M. Reicherter, T. Haist, and H. J. Tiziani, “Multi-functional optical tweezers using computer-generated holograms,” *Opt. Commun.* **185**, 77–82 (2000).

- 
- [69] R. W. Gerchberg and W. O. Saxton, “A practical algorithm for the determination of the phase from image and diffraction plane pictures,” *Optik* **35**, 237–246 (1972).
- [70] V. Soifer, V. Kotlyar, and L. Doskolovich, *Iterative Methods for Diffractive Optical Elements Computation* (Taylor & Francis Ltd, London, 1997).
- [71] J. E. Curtis, B. A. Koss, and D. G. Grier, “Dynamic holographic optical tweezers,” *Opt. Commun.* **207**, 169–175 (2002).
- [72] G. Shabtay, “Three-dimensional beam forming and Ewald’s surfaces,” *Opt. Commun.* **226**, 33–37 (2003).
- [73] M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney, “Synthesis of digital holograms by direct binary search,” *Appl. Opt.* **26**, 2788–2798 (1987).
- [74] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, chap. 10 (Cambridge University Press, Cambridge, United Kingdom, 1992). URL <http://library.lanl.gov/numerical/bookcpdf.html>.
- [75] P. C. Hohenberg, “Existence of Long-Range Order in One and Two Dimensions,” *Phys. Rev.* **158**, 383–386 (1967).
- [76] G. Whyte and J. Courtial, “Experimental demonstration of holographic three-dimensional light shaping using a Gerchberg-Saxton algorithm,” *New J. Phys.* **7**, 117 (2005).

- 
- [77] Z. Bouchal and J. Kyvalský, “Controllable 3D spatial localization of light fields synthesized by non-diffracting modes,” *J. Mod. Opt.* **51**, 157–176 (2004).
- [78] E. Abramochkin and V. Volostnikov, “Generation of spiral-type laser beams,” *Opt. Commun.* **141**, 59–64 (1997).
- [79] V. Soifer, V. Kotlyar, and L. Doskolovich, *Iterative Methods for Diffractive Optical Elements Computation*, p. 14 (Taylor & Francis Ltd, London, 1997).
- [80] R. L. Higdon, “Absorbing boundary condition for difference approximations to the multi-dimensional wave equation,” *Math. Comput.* **47**, 437–459 (1986).
- [81] J. C. Vaughan, T. Feurer, and K. A. Nelson, “Automated spatiotemporal diffraction of ultrashort laser pulses,” *Opt. Lett.* **28**, 2408–2410 (2003).
- [82] “LC-R 2500,” HOLOEYE Photonics AG, [www.holoeye.com](http://www.holoeye.com) (2004).
- [83] G. Gibson, L. Barron, F. Bec, G. Whyte, and M. Padgett, “Optically controlled grippers for manipulating micron-sized particles,” submitted to *Opt. Ex. Sep.* 2006.
- [84] A. Ashkin, “Acceleration and Trapping of Particles by Radiation Pressure,” *Phys. Rev. Lett.* **24**, 156–159 (1970).
- [85] A. Ashkin and J. M. Dziedzic, “Optical Levitation by Radiation Pressure,” *App. Phys. Lett.* **19**(8), 283–285 (1971).

- 
- [86] A. Ashkin and J. M. Dziedzic, "Observation of light scattering from nonspherical particles using optical levitation," *Appl. Opt.* **19**, 660–668 (1980).
- [87] P. Jordan, J. Leach, M. Padgett, P. Blackburn, N. Isaacs, M. Goksor, D. Hanstorp, A. Wright, J. Girkin, and J. Cooper, "Creating permanent 3D arrangements of isolated cells using holographic optical tweezers," *Lab on a Chip* **5**(11), 1224–1228 (2005). URL <http://dx.doi.org/10.1039/b509218c>.
- [88] M. E. J. Friese, H. Rubinsztein-Dunlop, J. Gold, P. Hagberg, and D. Hanstorp, "Optically driven micromachine elements," *App. Phys. Lett.* **78**, 547–549 (2001).
- [89] K. Ladavac and D. G. Grier, "Microoptomechanical pumps assembled and driven by holographic optical vortex arrays," *Optics Express* **12**, 1144–1149 (2004).
- [90] J. Leach, H. Mushfique, R. di Leonardo, M. Padgett, and J. Cooper, "An optically driven pump for microfluidics," *Lab Chip* **6**, 735–739 (2006).
- [91] K. Schutze, A. Clement-Sengewald, and A. Ashkin, "Zona drilling and sperm insertion with combined laser microbeam and optical tweezers," *Fertil. Steril.* **61**(4), 783–786 (1994).
- [92] S. Suresh, J. Spatz, J. P. Millis, A. Micoulet, M. Dao, C. T. Lim, M. Beil, and T. Seufferlein, "Connections between single-cell biome-

- chanics and human disease states: gastrointestinal cancer and malaria,” *Acta Biomater* **1**(1), 15–30 (2005).
- [93] M. Wang, H. Yin, R. Landick, J. Gelles, and S. Block, “Stretching DNA with optical tweezers,” *Biophys. J.* **72**(3), 1335–1346 (1997).
- [94] V. Bingelyte, J. Leach, J. Courtial, and M. J. Padgett, “Optically controlled three-dimensional rotation of microscopic objects,” *Appl. Phys. Lett.* **82**, 829–831 (2003).
- [95] V. L. Y. Loke, T. A. Nieminen, A. M. Branczyk, N. R. Heckenberg, and H. Rubinsztein-Dunlop, “Modelling optical micro-machines,” *ArXiv Physics e-prints* (2006). [physics/0607286](https://arxiv.org/abs/physics/0607286).
- [96] M. Pelton, M. Liu, H. Y. Kim, G. Smith, P. Guyot-Sionnest, and N. F. Scherer, “Optical trapping and alignment of single gold nanorods by using plasmon resonances,” *Opt. Lett.* **31**(13), 2075–2077 (2006).
- [97] S. Sato, Y. Harada, and Y. Waseda, “Optical trapping of microscopic metal particles,” *Opt. Lett.* **19**, 1807–1809 (1994).
- [98] A. T. O’Neil and M. J. Padgett, “Three-dimensional optical confinement of micron-sized metal particles and the de-coupling of the spin and orbital angular momentum within an optical spanner,” *Opt. Commun.* **185**, 139–143 (2000).
- [99] K. Sasaki, M. Koshioka, H. Misawa, N. Kitamura, and H. Masuhara, “Optical trapping of a metal particle and a water droplet by a scanning laser beam,” *App. Phys. Lett.* **60**, 807–809 (1992).

- 
- [100] J. Leach, K. Wulff, G. Sinclair, P. Jordan, J. Courtial, L. Thomson, G. Gibson, K. Karunwi, J. Cooper, Z. J. Laczik, and M. Padgett, “Interactive approach to optical tweezers control,” *Appl. Opt.* **45**, 897–903 (2006).
- [101] I.-Y. Park, S.-U. Hwang, J.-H. Song, and Y.-G. Lee, “Grasping microscopic objects by an optical trapping system controlled by five finger tips,” in *Optical Trapping and Optical Micromanipulation II. Proceedings of the SPIE*, K. Dholakia and G. C. Spalding, eds., vol. 5930, pp. 510–515 (2005).
- [102] J. A. Dharmadhikari, S. Roy, A. K. Dharmadhikari, S. Sharma, and D. Mathur, “Torque-generating malaria-infected red blood cells in an optical trap,” *Optics Express* **12**, 1179 (2004).

