

# LightPipes for Mathcad

Beam Propagation Toolbox Manual  
version 1.3

Toolbox Routines:

Gleb Vdovin,  
Flexible Optical B.V.  
Rontgenweg 1,  
2624 BD Delft  
The Netherlands  
Phone: +31-15-2851547  
Fax: +31-51-2851548  
e-mail: oko@xs4all.nl  
web site: [www.okotech.com](http://www.okotech.com)  
©1993--1996, Gleb Vdovin

Mathcad dynamic link library:

Fred van Goor,  
University of Twente  
Department of Applied Physics  
Laser Physics group  
P.O. Box 217  
7500 AE Enschede  
The Netherlands  
web site: [edu.tnw.utwente.nl/inlopt/lpmcad](http://edu.tnw.utwente.nl/inlopt/lpmcad)  
©2006, Fred van goor.



<b>1. INTRODUCTION .....</b>	<b>1-5</b>
<b>2. WARRANTY .....</b>	<b>2-7</b>
<b>3. AVAILABILITY .....</b>	<b>3-9</b>
<b>4. INSTALLATION OF LIGHTPIPES FOR MATHCAD .....</b>	<b>4-11</b>
4.1 System requirements.....	4-11
4.2 Installation .....	4-13
<b>5. LIGHTPIPES FOR MATHCAD DESCRIPTION .....</b>	<b>5-15</b>
5.1 First steps .....	5-15
5.1.1 Help .....	5-15
5.1.2 Starting the calculations.....	5-15
5.1.3 The dimensions of structures .....	5-16
5.1.4 Apertures and screens .....	5-16
5.1.5 Graphing and visualisation. ....	5-18
5.2 Free space propagation.....	5-19
5.2.1 FFT propagation (spectral method). ....	5-19
5.2.2 Direct integration as a convolution: FFT approach.....	5-22
5.2.3 Direct integration.....	5-23
5.2.4 Finite difference method. ....	5-23
5.2.5 Splitting and mixing beams .....	5-26
5.2.6 Interpolation.....	5-29
5.2.7 Phase and intensity filters .....	5-30
5.2.8 Zernike polynomials.....	5-31
5.2.9 Spherical co-ordinates .....	5-32
5.2.10 User defined phase and intensity filters.....	5-34
5.2.11 Random filters.....	5-35
5.2.12 FFT and spatial filters.....	5-35
5.2.13 Laser amplifier .....	5-36
5.2.14 Diagnostics: Strehl ratio, beam power.....	5-37
5.2.15 Polarization. ....	5-38
5.2.16 Reflection from and transmission through multilayer coatings. ....	5-40
<b>6. EXAMPLE MODELS .....</b>	<b>6-45</b>
6.1 Shearing interferometer .....	6-45
6.2 Rotational shearing interferometer .....	6-47
6.3 Radial shear interferometer .....	6-49
6.4 Michelson Interferometer.....	6-51
6.5 Twyman-Green interferometer.....	6-55

1-4

6.6	Unstable laser resonator .....	6-57
6.6.1	Calculation of the far field .....	6-60
6.7	Propagation in a lens-like/ absorptive medium. ....	6-63
6.8	Inverse problem: reconstructing the phase from measured intensities.....	6-71
6.9	Optical information processing.....	6-75
6.10	Generation and reconstruction of interferograms. ....	6-77
7.	COMMAND REFERENCE .....	7-81
8.	REFERENCES.....	8-87

## 1. Introduction

LightPipes for Mathcad is a set of functions written in C available to Mathcad. It is designed to model coherent optical devices when the diffraction is essential. The toolbox consists of a number of functions. Each function represents an optical element or a step in the light propagation. There are apertures, intensity filters, beam-splitters, lenses and models of free space diffraction in LightPipes. There are also more advanced tools for manipulating the phase and amplitude of the light. The program operates on a large data structure, containing square two-dimensional arrays of complex amplitudes of the optical field of the propagating light beam.

The LightPipes for Mathcad routines are modifications of the LightPipes C routines written by Gleb Vdovin for Unix, Linux, DOS and OS2 workstations or PC. The Mathcad version of LightPipes has a number of advantages:

1. Enhanced readability of the document with text added to the commands.
2. The graphics-, animation- and other features of Mathcad can be combined with the LightPipes commands.
3. You can use variable arguments in the function calls and handle complex data structures in a very simple way.
4. Enhanced flexibility and fast execution.

Most of the commands of LightPipes for Mathcad are the same as the Unix/DOS version. Only the commands handling the in- and output of the results to disk and the plot commands have been skipped as one can use the build-in commands of Mathcad for disk communications and graphics with more advantage. The names of the commands are preceded with the letters "LP" in order to keep the commands together in the Mathcad's function list and to prevent the confusion with existing Mathcad functions. The arguments are tested during execution of the command and error messages appear in the case of bad arguments. (Mathcad's "red" error boxes).



## **2. Warranty**

**There is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.**



### 3. Availability

The LightPipes for Mathcad package for beam propagation is copyright ©(1993--1996) of Gleb Vdovin and ©(1998) of Fred van Goor. This document (also © of Gleb Vdovin and Fred van Goor) may be freely distributed together with the demo executables. No part of this document can be reproduced without written permission of the authors. A demo version of LightPipes for Mathcad with a limited functionality of 64x64 grid dimension and this manual are freely available from WWW: <http://www.okotech.com>  
LightPipes for Mathcad with unlimited grid dimension (limited by your computer memory) is available from:

Flexible Optical B.V.  
Rontgenweg 1,  
2624 BD Delft  
The Netherlands  
Phone: +31-15-2851547  
Fax: +31-51-2851548  
e-mail: [oko@xs4all.nl](mailto:oko@xs4all.nl)  
web site: [www.okotech.com](http://www.okotech.com)

Additional information and more examples can be found on:  
<http://edu.tnw.utwente.nl/inlopt/lpmcad>



## 4. Installation of LightPipes for Mathcad

### 4.1 System requirements.

LightPipes for Mathcad must be installed on a system with the following hardware and software:

Hardware:

1. An 80386-, 80486 or Pentium based IBM or compatible computer.
2. At least 4 megabytes of memory.
3. A hard disk with at least 2 megabytes of free space.

Because the calculations require large arrays of data we recommend 16 megabytes of memory and a fast processor like a 66MHz 486 or a 100MHz Pentium or better.

Software:

1. Microsoft Windows.
2. Mathcad PLUS 6.0 or higher. ([www.mathsoft.com](http://www.mathsoft.com))

The authors tested LightPipes for Mathcad with Mathcad PLUS 5.0. All the routines work with this version. However, the help and the example programs, installed on your hard disk, have been written for the 6.0 version and higher and do not work under Mathcad PLUS 5.0. You have to type over the examples in the manual your self. Of course, the examples using specific features of version 6.0 and higher (Bitmaps, programming, some graphics, etc.) cannot be written in Mathcad PLUS 5.0 and have to be modified.



## 4.2 Installation

### Mathcad versions older than Mathcad 2000 Professional:

LightPipes for Mathcad can be installed on your system by copying the self extracting file **LPxx.EXE** (for version x.x), the **setup.bat** batch file and the **readme.txt** file to a temporal directory on your hard disk or to floppy disk. You should read the file **readme.txt** because this file contains recent information which is not in the manual. Then open a DOS window and type 'setup' or double click on this file name in a file manager like Windows Explorer. You can also use the Start/Run facility of Windows'95. 'Setup' creates a temporary directory, 'lptemp', on your hard disk and copies LPxx.EXE to this directory. Then it unpacks LPxx.EXE and starts the installation. The installation program will ask you for the directory where Mathcad was installed (default: 'c:\winmcad') and copies the DLL file containing the LightPipes functions to your '....\winmcad\userefi' directory. It also copies help-files and some examples to two new created directories: '...\winmcad\lphelp' and '...\winmcad\lpexamp' respectively, in your Mathcad directory. Finally 'Setup' deletes the temporary directory 'lptemp'. The LightPipes for Mathcad functions are available to you the next time you start Mathcad and should now be listed under the Math/Choose Function command of Mathcad. We advise you to start the help program: '....\winmcad\lphelp\lphelp.mcd' first. This program introduces you to the commands. Double-click on the bold, underlined words to jump to commands and examples. You can remove LightPipes for Mathcad from your hard disk by deleting the lphelp and lpexamp directories together with their contents, as well as the file LPMcad.dll in the '....\winmcad\userefi' directory. Note that LightPipes for Mathcad works only with the PLUS or Professional versions of Mathcad.

### Mathcad 2000 Professional and higher:

For Mathcad 2000 Professional and higher we have made new installers for each Mathcad version. They can be found on the installation CD ROM or obtained from Flexible Optical by download. The installer will copy LightPipes for Mathcad to your Mathcad directory, usually 'c:\Program Files\Mathsoft'. It will add 'LPMcad.xml' or 'LPMcad\_EN.xml' to the '...\doc\funcdoc' directory, 'LightPipes for Mathcad Manual.pdf' to '....\doc', the 'LPMcad.dll' to '....\userefi' and it will install a new Handbook with help for LightPipes in the '....\Handbook' directory. Now Help will be available when you insert a LightPipes function or when you open the LightPipes for Mathcad Handbook with help for each command and some examples.

### Mathcad 2001:

Mathcad version 2001 can not be used with LightPipes for Mathcad because this version has bugs and will generate errors when dealing with large arrays in a user dll as is the case with LightPipes for Mathcad. Mathsoft solved this by releasing a new version soon after version 2001 called Mathcad 2001i (I = improved?!). LightPipes for Mathcad works without problems with version 2001i.

### Mathcad 12 and higher.

From version 12 Mathcad is not 100% compatible with older Mathcad versions and the LightPipes dll had to be rewritten. Also one must make the arguments of the functions

dimensionless. This can be done by dividing by the unit (the meter in most cases). Another elegant method is to redefine the meter in a LightPipes for Mathcad document: type:  $m \equiv 1$  in your document and redefine  $nm \equiv 10^{-9} m$ ,  $\mu m \equiv 10^{-6} m$ , etc. You can put a hidden area in each LightPipes for Mathcad document with this as shown in the examples.

## 5. LightPipes for Mathcad description

### 5.1 First steps

#### 5.1.1 Help

Help can be obtained by clicking on the Math/Choose Function of Mathcad. A list of all the Mathcad functions and the LightPipes for Mathcad commands (Beginning with the letters ‘LP’) with a short description will appear.

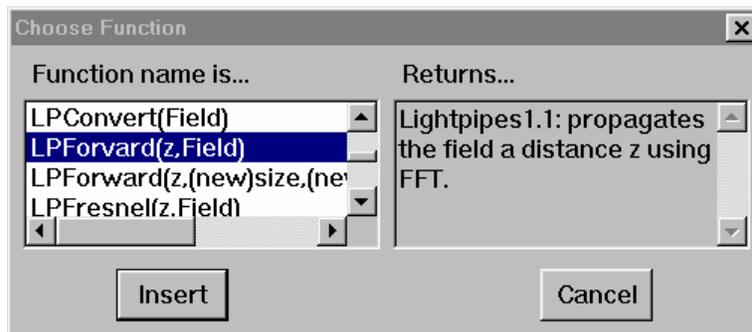


Figure 1 The Math/Choose Function command of Mathcad shows a short description of the function.

You can also obtain help by running the help program: ‘lphelp.mcd’ in the ‘...\\winmcad\\lphelp’ directory. Double-click on the bold, underlined words to jump to the commands and examples. If an argument of a function cannot be accepted by the routine a “red-box” error message appears giving information about the error. In version Mathcad 8 and 2000 Professional the “red-box” has been removed. Now the command causing the error will be colored red. If you installed **LightPipes for Mathcad 2000 or higher** you can obtain help by opening the LightPipes for Mathcad Handbook. Choose Help, Handbooks, LightPipes for Mathcad Help to open this book. The insertion of functions has been organized in a different way: Choose Insert, Function (Ctrl+E) and go to the LightPipes for Mathcad category. A short help will be given about the LightPipes command selected.

#### 5.1.2 Starting the calculations

All the calculations must start with the LPBegin function. This function defines the size of the square grid, the grid dimension and the wave length of the field. It is very convenient to use units. The arguments must be dimensionless, however. This can be done by dividing the variables by their units. When you use **Mathcad 8 or higher (up to version 11)** it is not necessary to past dimensionless arguments to the LightPipes functions anymore. Just type the arguments without dividing them through the units! You can still do that however, without altering the results. This has been done in order to be compatible with documents made with older versions. For Mathcad versions 12 and 13, however, Mathsoft decided to delete this option (!?) and you have to make the arguments dimensionless. A work around to this is to make your own units of length by redefinition of the meter: simply define the meter 1: type  $m \equiv 1$ ,  $nm \equiv 10^{-9}m$ , etc at the top of your document. In the help handbook you find a (closed) area (see Mathcad help how to manage areas) with this in each help topic and example.

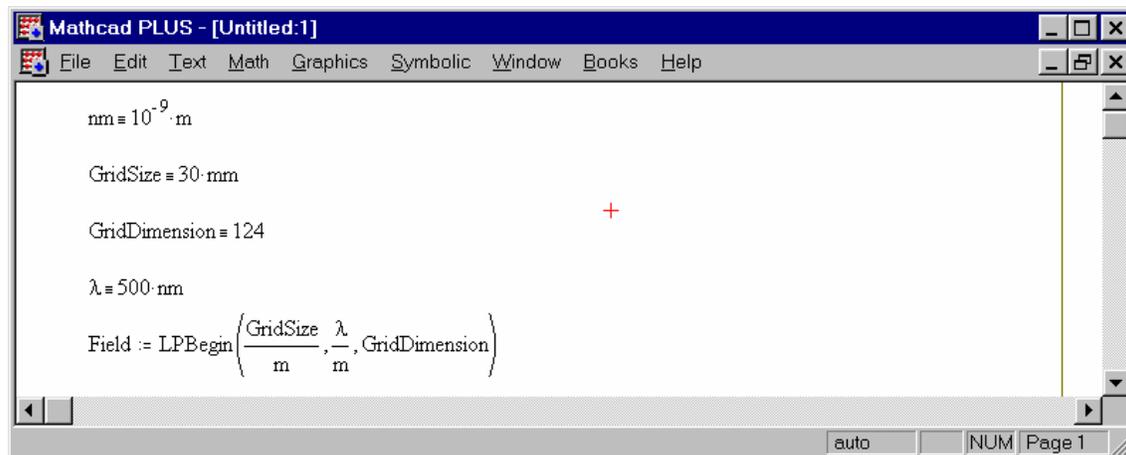


Figure 2 The use of LPBegin to define a uniform field with intensity 1 and phase 0.

A two dimensional array will be defined containing the complex numbers with the real and imaginary parts equal to one and zero respectively. The minimal dimension of the grid must be 8x8 and the maximum will be determined by your computer memory (or 64x64 if you have the demo version of LightPipes for Mathcad). The grid dimension must be an even number. An extra row is added to the field array to carry information such as the grid dimension, grid size and wave length with the field.

### 5.1.3 The dimensions of structures

The field structures in LightPipes for Mathcad are two dimensional arrays of complex numbers. For example a grid with 256x256 points asks about 1Mb of memory. 512x512 points ask more then 4Mb and 1024x1024 16Mb. Some commands, however need more memory because internal arrays are necessary. LightPipes for Mathcad works fast and without disk-swapping with arrays up to 256x256 points for a Pentium PC with 32Mb RAM memory and running under Windows'95. A function called 'Time.dll' is available from the author to measure the execution time of the programs.

### 5.1.4 Apertures and screens

The simplest component to model is an aperture. There are three different types of apertures:

1. The circular aperture: LPCircAperture(R,  $x_s$ ,  $y_s$ , Field)
2. The rectangular aperture: LPRectAperture( $w_x$ ,  $w_y$ ,  $x_s$ ,  $y_s$ ,  $\phi$ , Field)
3. The Gaussian diaphragm: LPGaussAperture(R,  $x_s$ ,  $y_s$ , T, Field)

Where R=radius,  $x_s$  and  $y_s$  are the shift in x and y direction respectively,  $\phi$  is a rotation and T is the centre transmission of the Gaussian aperture. In addition, there are three commands describing screens: LPCircScreen (inversion of the circular aperture), LPRectScreen, (inversion of the rectangular aperture) and LPGaussScreen (inversion of the gauss aperture). Figure 3 shows an example of the usage of a circular aperture. All kinds of combinations of circular, rectangular and Gaussian apertures and screens can be made as illustrated in Figure 4.

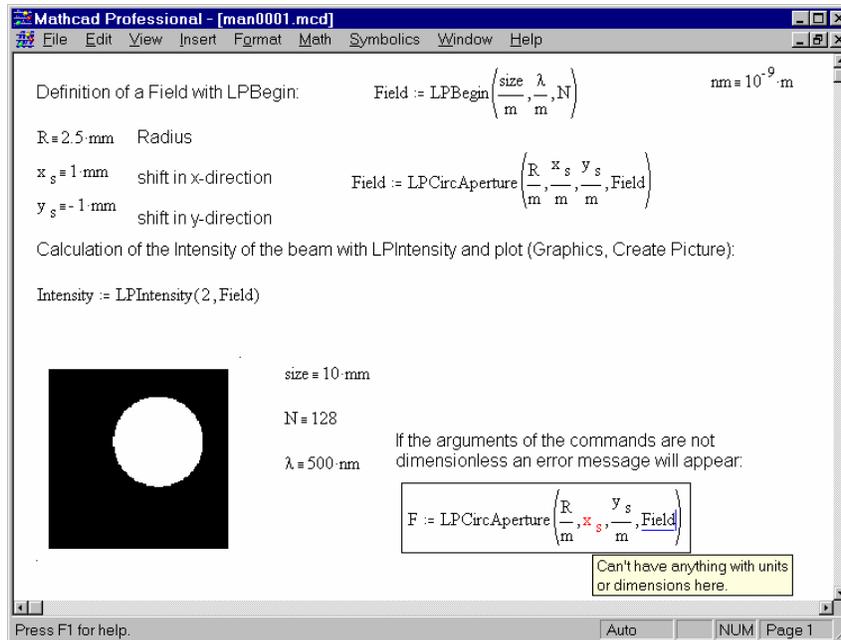


Figure 3 Example of the application of a circular aperture. Demonstration of an error message.

In Figure 3 and Figure 4 the field (called 'Field') was defined with the LPBegin command with a uniform intensity and phase distribution, a wavelength, a grid size and a grid dimension. In Figure 3 we applied units to the variables. The variables in the commands must be made dimensionless, however, which has been done by dividing them by their units. If this is not done, an error message appears as shown in Figure 3.

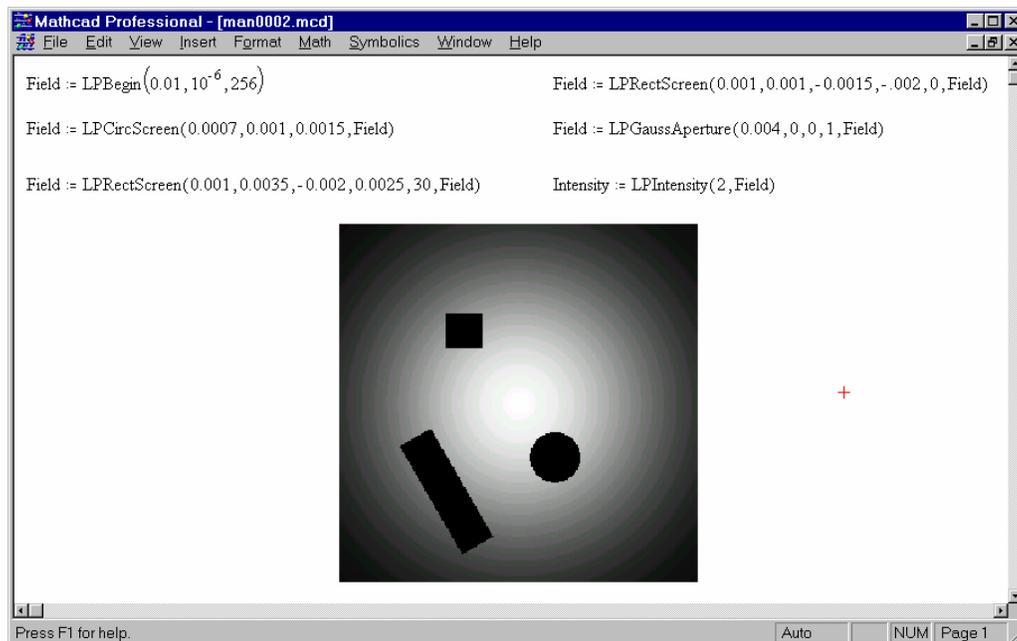


Figure 4 The use of screens and apertures.

The use of units in the simulations is recommended because it is very convenient and it reduces the chance on errors.

### 5.1.5 Graphing and visualisation.

In Figure 3 and Figure 4 the intensity of the field is calculated after aperturing 'Field' with the LPIntensity command using option 2, which means that a bitmap with gray values is produced which can be displayed with the 'Create Picture' command of Mathcad. This is the fastest way to display the results of the LightPipes calculations. All the other powerful graphic features of Mathcad can be applied for displaying the intensity or phase distributions including coloured contour plots, surface plots, etc., but these are rather time consuming. With the command LPInterpolate, however, you can reduce the grid dimension of the field speeding up subsequent plotting considerably. This is illustrated in Figure 5. The use of interpolation will be explained in more detail in paragraph 5.2.6. The results of the calculations can be written to file using Mathcad's WRITE or WRITEPRN commands. You can also type the numerical data on your screen, select them and copy them to the clip board. This is a very convenient method to import your data into your favourite graphics program or what ever you want to do with it. See your Mathcad user's guide or the help facility for details. In Figure 5 we also plotted the cross section of the beam in a two dimensional XY plot. For this we have to define an integer,  $i$ , ranging from 0 to the grid dimension minus one. This integer must be used to define the element of the (square) array,  $I$ , which is used for the vertical axis of the XY plot. You can also use it to define a vector  $x$  for the horizontal axis. You can hide the array elements and define your own text along the axis to produce a nicer plot. These things are explained in the Mathcad user's guide.

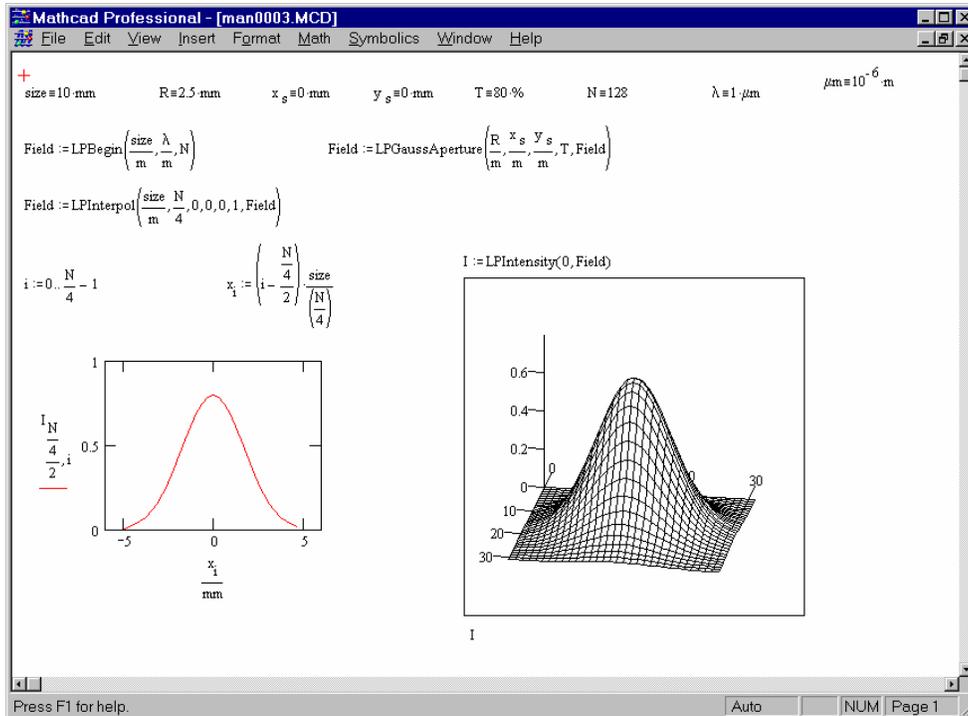


Figure 5 The use of XY-plots and surface plots to present your simulation results. Note the usage of LPInterpol to reduce the grid dimension for a faster and nicer surface plot.

## 5.2 Free space propagation

There are five different possibilities for modelling the light propagation in LightPipes for Mathcad.

### 5.2.1 FFT propagation (spectral method).

Let us consider the wave function  $U$  in two planes:  $U(x,y,0)$  and  $U(x,y,z)$ . Suppose then that  $U(x,y,z)$  is the result of propagation of  $U(x,y,0)$  to the distance  $z$ , with the Fourier transforms of these two (initial and propagated) wave functions given by  $A(\alpha,\beta,0)$  and  $A(\alpha,\beta,z)$  correspondingly. In the Fresnel approximation, the Fourier transform of the diffracted wave function is related to the Fourier transform of the initial function via the frequency transfer characteristic of the free space  $H(\alpha,\beta,z)$ , given by [1,2]:

$$H = \frac{A(\alpha,\beta,z)}{A(\alpha,\beta,0)} = \exp\{-ikz(1 - \alpha^2 - \beta^2)^{1/2}\} \quad (5.1)$$

where:

$$A(\alpha,\beta,0) = \iint_{-\infty}^{\infty} U(x,y,0) \exp\{-ik(\alpha x + \beta y)\} dx dy \quad (5.2)$$

$$A(\alpha,\beta,z) = \iint_{-\infty}^{\infty} U(x,y,z) \exp\{-ik(\alpha x + \beta y)\} dx dy \quad (5.3)$$

Expressions (5.1, 5.2, 5.3) provide a symmetrical relation between the initial and diffracted wave functions in the Fresnel approximation. Applied in the order  $\rightarrow$  (5.1)  $\rightarrow$  (5.3) they result in the diffracted wave function, while being applied in the reversed order they allow for reconstruction of the initial wave function from the result of diffraction. We shall denote the forward and the reversed propagation operations defined by expressions (5.1, 5.2 and 5.3) with operators  $L^+$  and  $L^-$  respectively.

The described algorithm can be implemented numerically using Fast Fourier Transform (FFT) [2, 3] on a finite rectangular grid with periodic border conditions. It results in a model of beam propagation inside a rectangular wave guide with reflective walls. To approximate a free-space propagation, wide empty guard bands have to be formed around the wave function defined on a grid. To eliminate the influence of the finite rectangular data window, Gaussian amplitude windowing in the frequency domain should be applied—see [2, 3] for extensive analysis of these computational aspects.

The simplest and fastest LightPipes command for propagation is LPForward. (The ‘v’ is a type error made on purpose!) It implements the spectral method described by (5.1, 5.2, 5.3). The syntax is simple, for example if you want to filter your field through a 1cm aperture and then propagate the beam 1m forward, you type the commands listed in Figure 6:

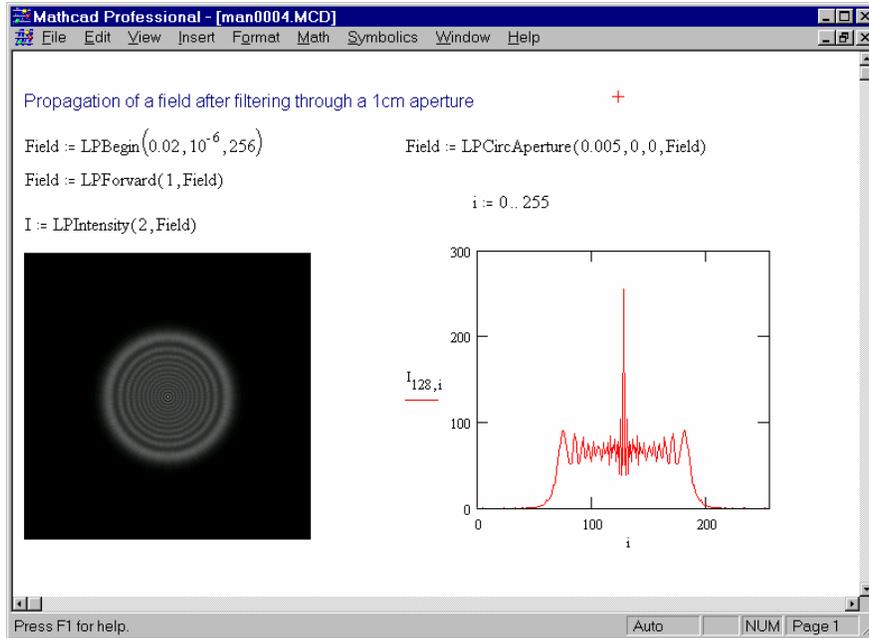


Figure 6 The result of the propagation: density and cross section intensity plots.

We see the diffraction effects, the intensity distribution is not uniform anymore. The algorithm is very fast in comparison with direct calculation of diffraction integrals. Features to be taken into account:

- The algorithm realises a model of light beam propagation inside a square wave guide with reflecting walls positioned at the grid edges. To approximate a free space propagation, the intensity near the walls must be negligible small. Thus the grid edges must be far enough from the propagating beam. Neglecting these conditions will cause interference of the propagating beam with waves reflected from the wave guide walls.
- As a consequence of the previous feature, we must be extremely careful propagating the plane wave to a distance comparable with  $D^2/\lambda$  where  $D$  is the diameter (or a characteristic size) of the beam, and  $\lambda$  is the wavelength. To propagate the beam to the far field (or just far enough) we have to choose the size of our grid much larger than the beam itself, in other words we define the field in a grid filled mainly with zeros. The grid must be even larger when the beam is aberrated because the divergent beams reach the region border sooner. Due to these two reasons the commands:

```
Field:=LPBegin(0.02,10-6,256)
Field:=LPRectAperture(0.02,0.02,0,0,0,Field)
Field:=LPForward(1,Field)
I:=LPIntensity(2,Field)
```

make no sense (zero intensity). The cross section of the beam (argument of LPRectAperture) equals to the section of the grid (the first argument of LPBegin), so we have a model of light propagation in a wave guide but not in a free space. One has to put:

```
Field:=LPBegin(0.04,10-6,256)
Field:=LPRectAperture(0.02,0.02,0,0,0,Field)
```

```
Field:=LPForward(1,Field)
I:=LPIntensity(2,Field)
```

for propagation in the near field, and may be:

```
Field:=LPBegin(0.2,10-6,512)
Field:=LPRectAperture(0.02,0.02,0,0,0,Field)
Field:=LPForward(400,Field)
I:=LPIntensity(2,Field)
```

for far field propagation.

If we compare the result of the previous example with the result of:

```
Field:=LPBegin(0.06,10-6,512)
Field:=LPRectAperture(0.02,0.02,0,0,0,Field)
Field:=LPForward(400,Field)
I:=LPIntensity(2,Field)
```

we'll see the difference.

We have discussed briefly the drawbacks of the FFT algorithm. The good thing is that it is very fast, works pretty well if properly used, is simple in implementation and does not require the allocation of extra memory. In LightPipes.1.1 and later a negative argument may be supplied to LPForward. It means that the program will perform "propagation back" or in other words it will reconstruct the initial field from the one diffracted. For example:

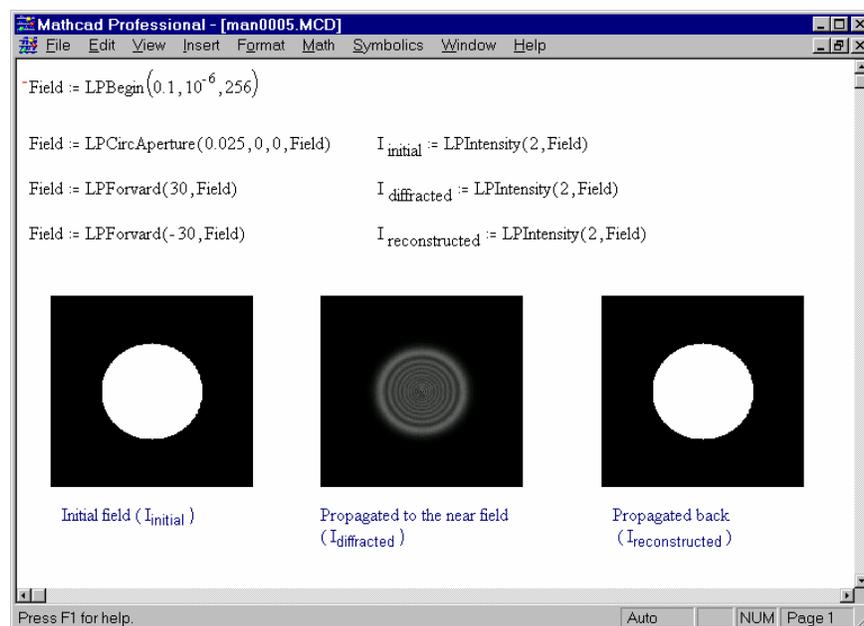


Figure 7 The initial field, the field propagated to the near field and the field propagated back (from left to right)

### 5.2.2 Direct integration as a convolution: FFT approach

Another possibility of a fast computer implementation of the operator  $L^+$  is free from many of the drawbacks of the described spectral algorithm. The operator  $L^+$  may be numerically implemented with direct summation of the Fresnel-Kirchoff diffraction integral:

$$U(x_1, y_1, z) = \frac{k}{2\pi i z} \iint U(x, y, 0) \exp\left\{ik \frac{(x-x_1)^2 + (y-y_1)^2}{2z}\right\} dx dy \quad (5.4)$$

with functions  $U(x, y, 0)$  and  $U(x, y, z)$  defined on rectangular grids. This integral may be converted into a convolution form which can be efficiently computed using FFT [4,5]. This method is free from many drawbacks of the spectral method given by the sequence (5.3)→(5.2)→(5.4) although it is still very fast due to its use of FFT for computing of the integral sums.

We'll explain this using a two-dimensional example, following [4], p.100. Let the integral be defined in a finite interval  $-L/2 \dots L/2$ :

$$U(x_1, z) = \sqrt{\frac{k}{2\pi i z}} \int_{-L/2}^{L/2} U(x, 0) \exp\left\{ik \frac{(x-x_1)^2}{2z}\right\} dx \quad (5.5)$$

Replacing the functions  $U(x)$  and  $U(x_1)$  with step functions  $U_j$  and  $U_m$ , defined in the sampling points of the grid with  $j=0 \dots N$ , and  $m=0 \dots N$  we convert the integral (5.5) to the form:

$$U_m = \sqrt{\frac{k}{2\pi i z}} \left[ \sum_{j=1}^{N-1} U_j \int_{x_j-0.5}^{x_j+0.5} \exp\left\{ik \frac{(x_m-x)^2}{2z}\right\} dx + \right. \\ \left. + U_0 \int_{x_0}^{x_{0.5}} \exp\left\{ik \frac{(x_m-x)^2}{2z}\right\} dx + U_N \int_{x_N-0.5}^{x_N} \exp\left\{ik \frac{(x_m-x)^2}{2z}\right\} dx \right] \quad (5.6)$$

Taking the integrals in (5.6) we obtain:

$$U_m = \sum_{j=1}^{N-1} U_j K_{mj} + U_0 K_{m0} + U_N K_{mN} \quad (5.7)$$

where:  $K_{m0}$ ,  $K_{mj}$  and  $K_{mN}$  are analytical expressed with the help of Fresnel integrals, depending only onto the difference of indices. The summations  $\sum_{j=1}^{N-1} U_j K_{mj}$  can easily be calculated for all indices  $m$  as one convolution with the help of FFT.

The command LPFresnel, defined starting from version LightPipes.1.1, implements this algorithm using the trapezoidal rule. It is almost as fast as LPForward (from 2 to 5 times slower), it uses 8 times more memory than LPForward and it allows for "more honest" calculation of near and far-field diffraction. As it does not require any protection bands at the edges of the region, the model may be built in a smaller grid, therefore the resources consumed and time of execution are comparable or even better than that of LPForward. LPFresnel does not accept a negative propagation distance. When possible LPFresnel has to be used as the main computational engine within LightPipes for Mathcad.

**Warning:** LPFresnel does not produce valid results if the distance of propagation is comparable with (or less than) the characteristic size of the aperture at which the field is diffracted. In this case LPForward or LPSteps should be used.

### 5.2.3 Direct integration

Direct calculation of the Fresnel-Kirchoff integrals is very inefficient in two-dimensional grids. The number of operations is proportional to  $N^4$ , where  $N$  is the grid sampling. With direct integration we do not have any reflection at the grid boundary, so the size of the grid can just match the cross section of field distribution. LightPipes include a program LPForward realising direct integration. LPForward has the following features:

- arbitrary sampling and size of square grid at the input plane
- arbitrary sampling and size of square grid at the output plane, it means we can propagate the field from a grid containing for example 52x52 points corresponding to 4.9x4.9cm to a grid containing 42x42 points and corresponding let's say 8.75x8.75 cm.

### 5.2.4 Finite difference method.

It can be shown that the propagation of the field  $U$  in a medium with complex refractive coefficient  $A$ , is described by the differential equation:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + 2ik \frac{dU}{dz} + A(x, y, z)U = 0 \quad (5.8)$$

To solve this equation, we re-write it as a system of finite difference equations:

$$\frac{U_{i+1,j}^{k+1} - 2U_{i,j}^{k+1} + U_{i-1,j}^{k+1}}{\Delta x^2} + \frac{U_{i,j+1}^k - 2U_{i,j}^k + U_{i,j-1}^k}{\Delta y^2} + 2ik \frac{U_{i,j}^{k+1} - 2U_{i,j}^k}{\Delta z} + A_{i,j}^{k+1}U_{i,j}^{k+1} = 0 \quad (5.9)$$

Collecting terms we obtain the standard three-diagonal system of linear equations, the solution of which describes the complex amplitude of the light field in the layer  $z + \Delta z$  as a function of the field defined in the layer  $z$ :

$$-a_i U_{i-1,j}^{k+1} + c_i U_{i,j}^{k+1} - b_i U_{i+1,j}^{k+1} = f_i \quad (5.10)$$

where (we put  $\Delta x = \Delta y = \Delta$ )

$$a_i = b_i = -\frac{1}{\Delta^2} \quad (5.11)$$

$$c_i = A_{i,j}^{k+1} - \frac{2}{\Delta^2} + \frac{2ik}{\Delta z} \quad (5.12)$$

$$f_i = \frac{2ik}{\Delta z} U_{i,j}^k - \frac{U_{i,j+1}^k - 2U_{i,j}^k + U_{i,j-1}^k}{\Delta^2} \quad (5.13)$$

The three-diagonal system of linear equations (5.10) is solved by the standard elimination (double sweep) method, described for example in [ 6 ]. This scheme is absolutely stable (this variant is explicit with respect to the index  $i$  and implicit with respect to the index  $j$ ). One step of propagation is divided into two sub-steps: the first sub-step applies the described procedure to all rows of the matrix, the second sub-step changes the direction of elimination and the procedure is applied to all columns of the matrix.

The main advantage of this approach is the possibility to take into account uniformly diffraction, absorption (amplification) and refraction. For example, the model of a waveguide with complex three-dimensional distribution of refraction index and absorption coefficient (both are

defined as real and imaginary components of the (three-dimensional in general) matrix  $A_{i,j}^k$  ) can be built easily.

It works also much faster than all described previously algorithms on one step of propagation, though to obtain a good result at a considerable distance, many steps should be done. As the scheme is absolutely stable (at least for free-space propagation), there is no stability limitation on the step size in the direction  $Z$ . Large steps cause high-frequency errors, therefore the number of steps should be determined by trial (increase the number of steps in a probe model till the result stabilizes), especially for strong variations of refraction and absorption inside the propagation path.

Zero amplitude boundary conditions are commonly used for the described system. This, again, creates the problem of the wave reflection at the grid boundary. The influence of these reflections in many cases can be reduced by introducing an additional absorbing layer in the proximity of the boundary, with the absorption smoothly (to reduce the reflection at the absorption gradient) increasing towards the boundary.

In LightPipes for Mathcad version 1.2 the refraction term is not included into the propagation formulas, instead the phase of the field is modified at each step according to the distribution of the refractive coefficient. This "zero-order" approximation happened to be much more stable numerically than the direct inclusion of refraction terms into propagation formulas. It does not take into account the change of the wavelength in the medium, it does not model backscattering and reflections back on interfaces between different media. Perhaps there are other details to be mentioned. The described algorithm is implemented in a filter LPSteps.

Mathcad PLUS - [FocLens.mcd]

File Edit Text Math Graphics Symbolic Window Books Help

## Calculation of the intensity distribution in the focus of a positive lens.

$nm = 10^{-9} \cdot m$

*Fred A. van Goor*  
 Department of Applied Physics  
 University of Twente  
 P.O. Box 217  
 7500 AE Enschede  
 The Netherlands  
 F.A.vanGoor@utwente.nl

In this example of LightPipes for Mathcad, version 1.2, we demonstrate the use of the LPSteps command to calculate the intensity distribution in the focus of a positive lens.

Definition of the grid:    size = 4 mm     $\lambda = 632.8 \text{ nm}$     N = 100

Definition of the refractive index:     $i := 0..N - 1$      $j := 0..N - 1$      $n_{i,j} := 1$

$I := \left[ \begin{array}{l} K \leftarrow \text{LPBegin} \left( \frac{\text{size}}{m}, \frac{\lambda}{m}, N \right) \\ K \leftarrow \text{LPCircAperture} \left( \frac{R}{m}, 0, 0, K \right) \\ K \leftarrow \text{LPLens} \left( \frac{f}{m}, 0, 0, K \right) \\ \text{for } k \in 0..K_{\text{max}} \\ \quad \left[ \begin{array}{l} K \leftarrow \text{LPSteps} \left( \frac{dz}{m}, N_{\text{steps}}, n, K \right) \\ \text{Int} \leftarrow \text{LPIntensity}(0, K) \\ I_k \leftarrow \text{Int} \end{array} \right. \\ I \end{array} \right.$	<p>Define the initial field</p> <p>Transmission through an aperture</p> <p>and a lens</p> <p>Propagation of the field through the medium with index, n using a distance, Nsteps*dz.</p> <p>Calculation of the intensity and storage</p>
---	---

auto    NUM    Page 1

Figure 8. The use of LPSteps to calculate the intensity distribution in the focus of a lens.

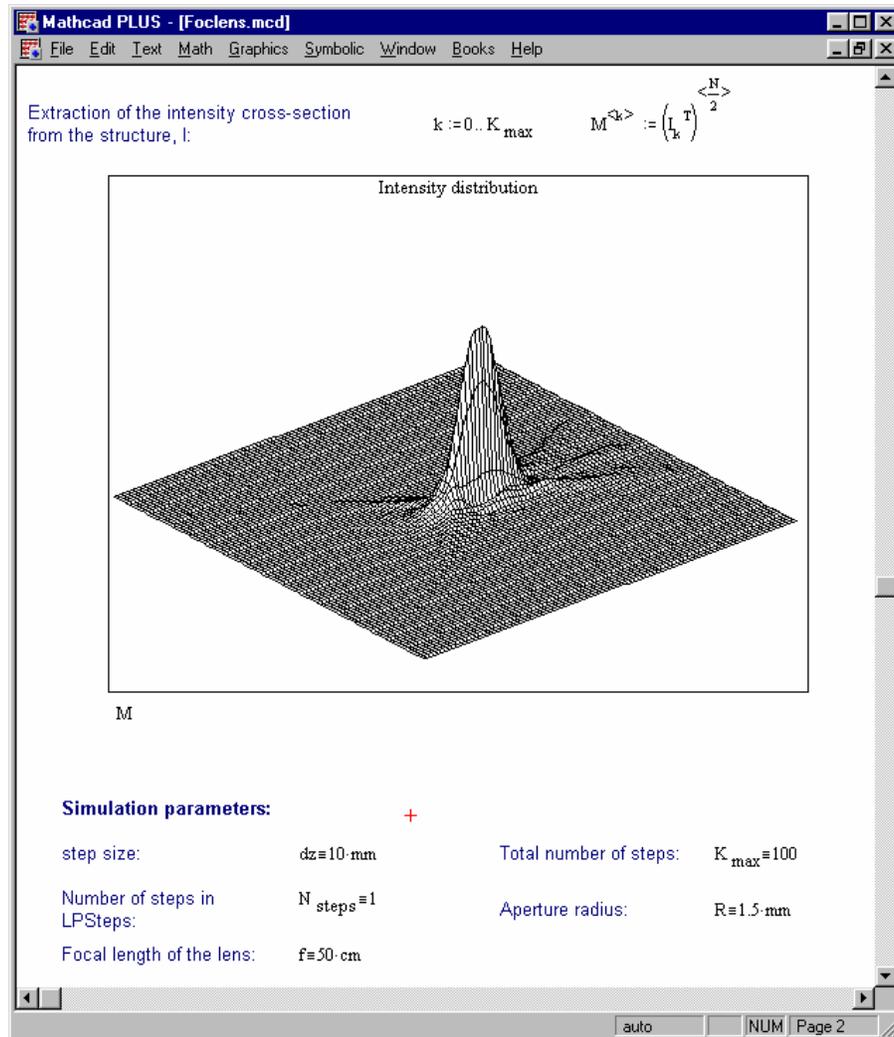


Figure 8. The use of LPSteps to calculate the intensity distribution in the focus of a lens. (cont.)

LPSteps has a built-in absorption layer along the grid boundaries (to prevent reflections), occupying 10% of grid from each side. LPSteps is the only filter in LightPipes for Mathcad allowing for modeling of (three-dimensional) waveguide devices.

Like LPForward, LPSteps can inversely propagate the field, for example the sequence ...LPSteps( 0.1, 1, n, Field) LPSteps(-0.1, 1, n, Field)... doesn't change anything in the field distribution. The author has tested this reversibility also for propagation in absorptive/refractive media, examples will follow.

LPSteps implements scalar approximation, it is not applicable for modeling of waveguide devices in the vector approximation, where two components of the field should be taken into account.

### 5.2.5 Splitting and mixing beams

There are two commands in LightPipes which are useful for modelling of interferometers. With LPIntAttenuator we can split the field structure (amplitude division) - The two obtained fields could be processed separately and then mixed again with the routine LPBeamMix. In this script we

have formed two beams each containing one “shifted” hole. After mixing these two beams we have a screen with two holes: a Young’s interferometer.

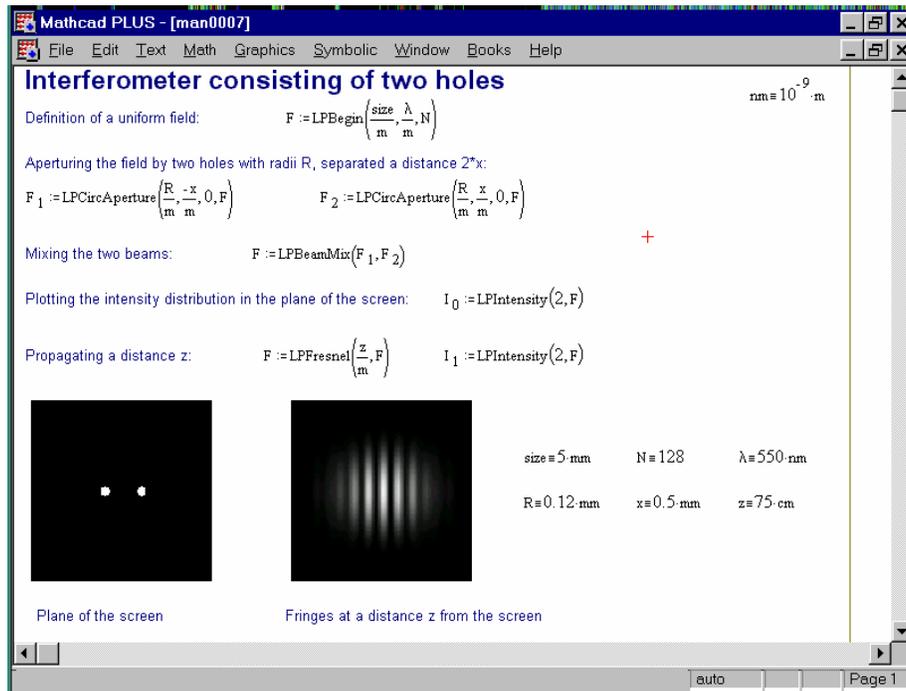


Figure 9 Young’s interferometer.

Having the model of the interferometer we can “play” with it, moving the pinholes and changing their sizes. The following models the result of the interference of a plane wave diffracted at three round apertures:

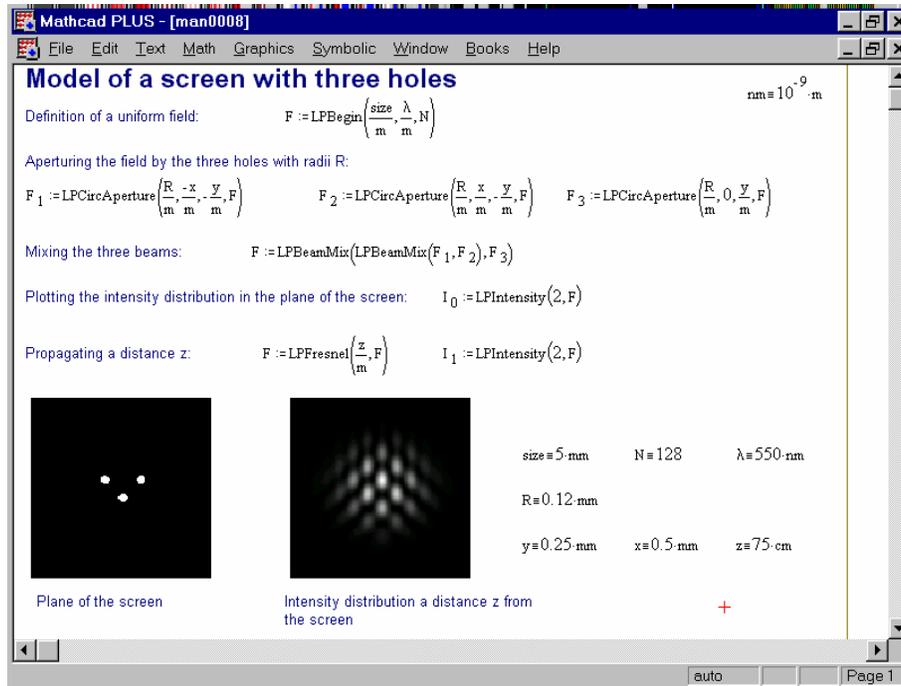


Figure 10 Intensity distributions in the plane of the screen and 75cm behind the screen.

The next interferometer is more interesting:

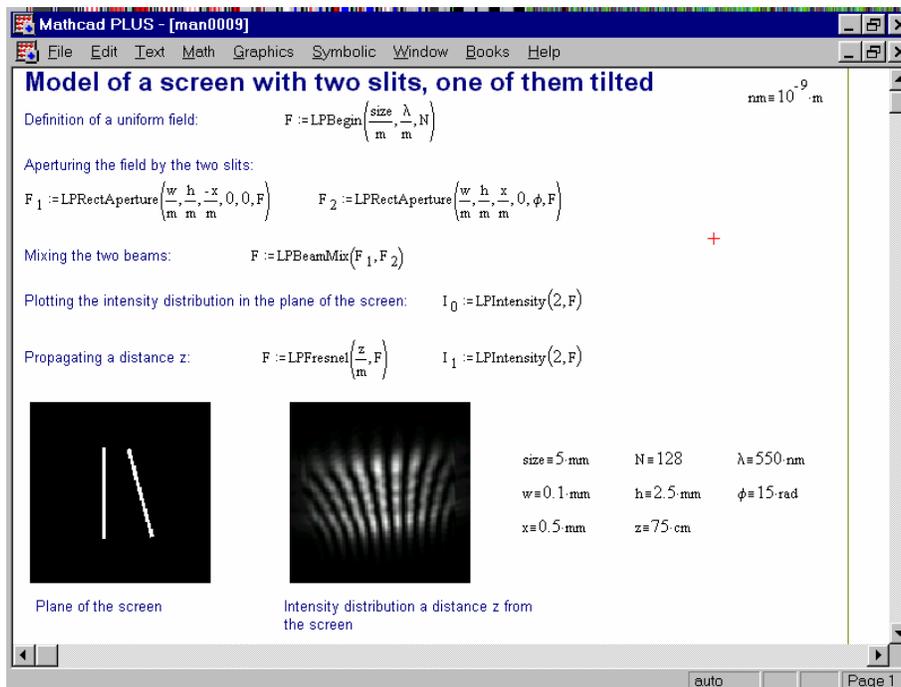


Figure 11 Intensity distributions in the plane of the screen and 75cm behind the screen.

In the last example the intensity distribution is modulated by the wave, reflected from the grid edge, nevertheless it gives a good impression about the general character of the interference pattern. To obtain a better result, the calculations should be conducted in a larger grid or other

numerical method should be used. The following example uses a direct integration algorithm (the input and output are in different scales and have different samplings):

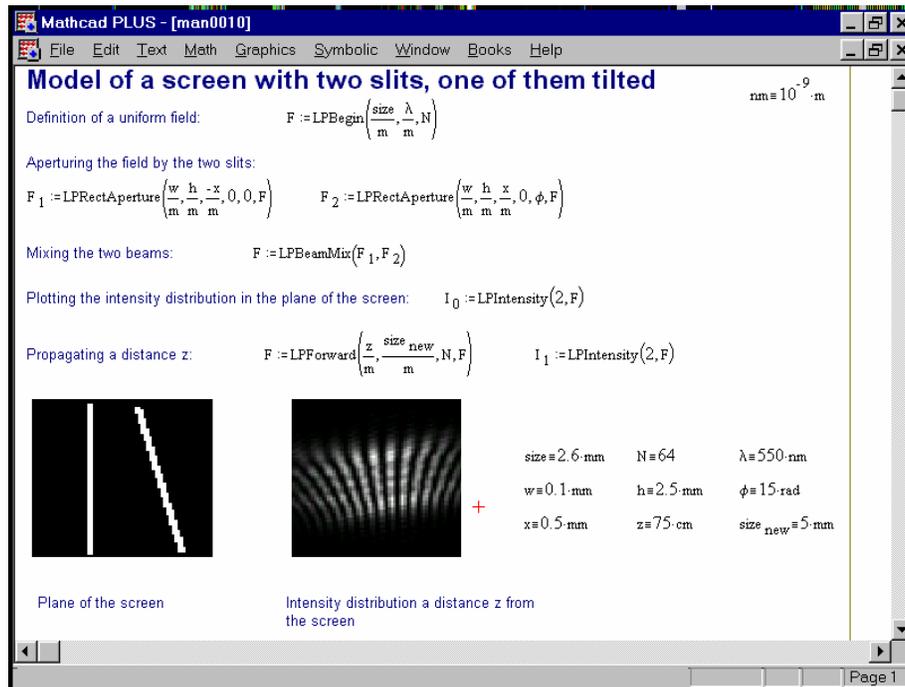


Figure 12 Intensity distributions in plane of the screen and 75 cm after the screen, note that input and output have different scales, input grid is 2.5x2.5mm, output is 5x5mm

This example uses approximately 25 times less memory than the previous FFT. The calculation may take from minutes to tens of minutes, depending on the speed of the computer.

### 5.2.6 Interpolation

The program LPInterpol is the tool for manipulating the size and the dimension of the grid and for changing the shift, rotation and the scale of the field distribution. It accepts six command line arguments, the first is the new size of the grid. The second argument gives the new number of points, the third gives the value of transverse shift in the X direction, the fourth gives the shift in the Y direction, the fifth gives the field rotation (first shift and then rotation). The last sixth argument determines the magnification, its action is equivalent to passing the beam through a focal system with magnification M (without diffraction, but preserving the integral intensity). For example if the field was propagated with FFT algorithm LPForward, then the grid contains empty borders, which is not necessary if we want to propagate the field further with LPForward. Other way around, after LPForward we have to add some empty borders to continue with LPForward. LPInterpol is useful for interpolating into a grid with different size and number of points. Of course it is not too wise to interpolate from a grid of 512x512 points into a grid of 8x8, and then back because all information about the field will be lost. The same is true for interpolating the grid of 1x1m to 1x1mm and back. When interpolating into a grid with larger size, for example from 1x1 to 2x2, the program puts zeros into the new added regions. Figure 13 illustrates the usage of LPInterpol for the transition from a fine grid used by LPForward (near field) to a coarse grid used by LPForward (far field).

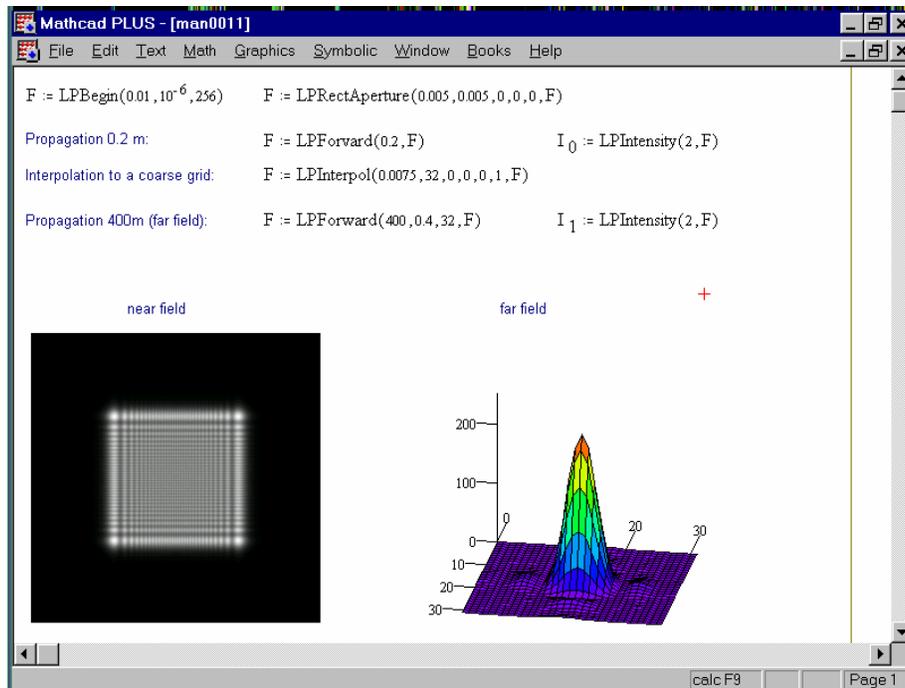


Figure 13 Illustration of the usage of LPInterpol for the transition from a fine grid used by LPForward (near field) to a coarse grid used by LPForward (far field).

### 5.2.7 Phase and intensity filters

There are four kinds of phase filters available in LightPipes -wave front tilt, the quadratic phase corrector called lens, a general aberration in the form of a Zernike polynomial, and a user defined filter. To illustrate the usage of these filters let's consider the following examples:

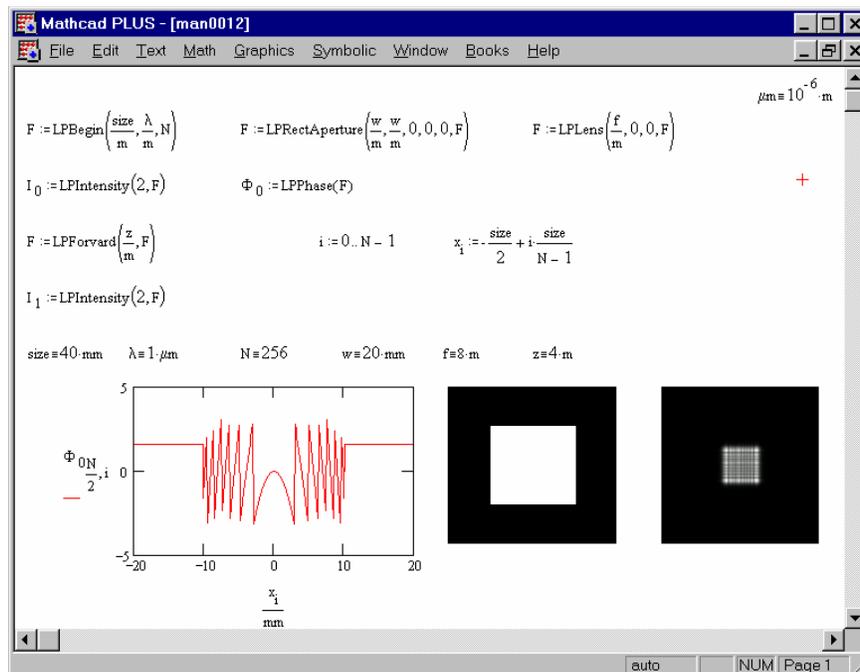


Figure 14 The phase distribution after passing the lens, intensity in the plane of the lens and at a distance equal to the half of the focal distance. (from left to right)

The first sequence of operators forms the initial structure, filters the field through the rectangular aperture and then filters the field through a positive lens with optical power of 0.125D (the focal distance of  $1/0.125=8\text{m}$ ). With the second command we propagate the field 4m forward. As 4m is exactly the half of the focal distance, the cross section of the beam must be reduces twice.

We have to be very careful propagating the field to the distance which is close to the focal distance of a positive lens- the near—focal intensity and phase distributions are localised in the central region of the grid occupying only a few grid points. This leads to the major loss of information about the field distribution. The problem is solved by applying the co-ordinate system which is tied to the divergent or convergent light beam, the tools to do this will be described later.

The lens may be decentered, `LPLens(8, 0.01, 0.01, Field)` produces the lens with a focal length of  $1/0.125$  shifted by 0.01 in X and Y directions. Note, when the lens is shifted, the aperture of the lens is not shifted, the light beam is not shifted also, only the phase mask correspondents to the lens is shifted.

The wave front tilt is illustrated by following examples:

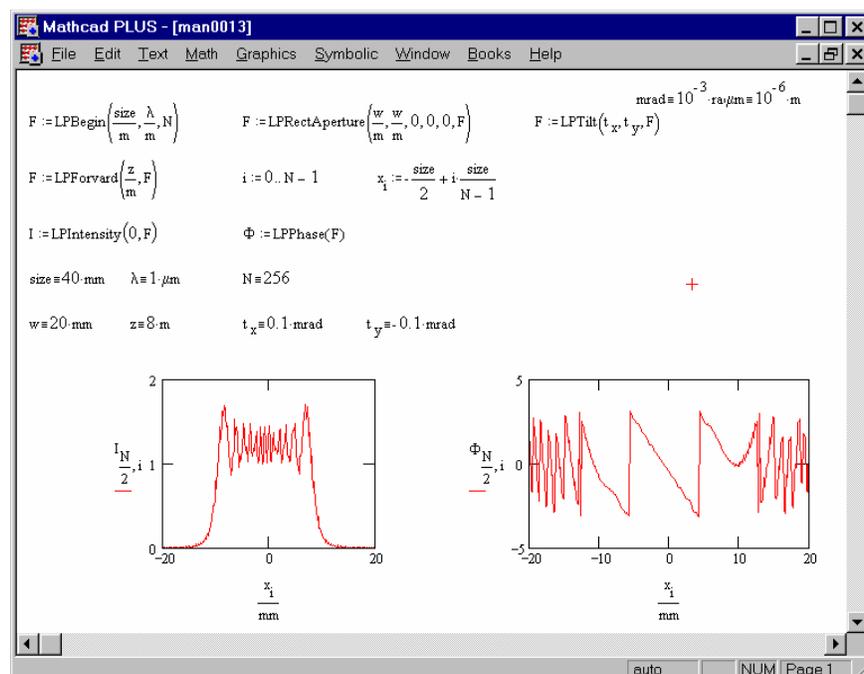


Figure 15 The intensity and the phase after tilting the wave front by 0.1 mrad and propagating it 8 m forward. Note the transversal shift of the intensity distribution and the phase tilt.

In this example the wave front was tilted by  $\alpha=0.0001$  rad in X and Y directions, then propagated it to the distance  $Z=8\text{m}$ , so in the output distribution we observe the transversal shift of the whole intensity distribution by  $\alpha Z=0.8\text{mm}$ .

### 5.2.8 Zernike polynomials

Any aberration in a circle can be decomposed over a sum of Zernike polynomials. Formulas given in [7] have been directly implemented in LightPipes. The program is called LPZernike and accepts four command line arguments:

1. The radial order  $n$  (first column in Table 13.2 [8] p. 465).

2. The azimuthal order  $m$ ,  $|m| \leq n$ , polynomials with negative  $n$  are rotated  $90^\circ$  relative to the polynomials with positive  $n$ . For example  $\text{LPZernike}(5,3,1,1,\text{Field})$  gives the same aberration as  $\text{LPZernike}(5,-3,1,1,\text{Field})$ , but the last is rotated  $90^\circ$ . This index corresponds to the  $n-2m$  given in the third column of Table 13.2 in [7], p. 465.
3. The radius,  $R$
4. The amplitude of aberration in radians at  $R$

We can uniformly introduce  $\text{LPLens}$  and  $\text{LPTilt}$  with  $\text{LPZernike}$ , the difference is that we pass the amplitude of the aberration to  $\text{LPZernike}$ .  $\text{LPLens}$  and  $\text{LPTilt}$  accept conventional meters and radians, which are widely in use for the description of optical setups, while  $\text{LPZernike}$  uses the amplitude of the aberration, which frequently has to be derived from the technical description.

A cylindrical lens can be modelled as a combination of two  $\text{LPZernike}$  commands:

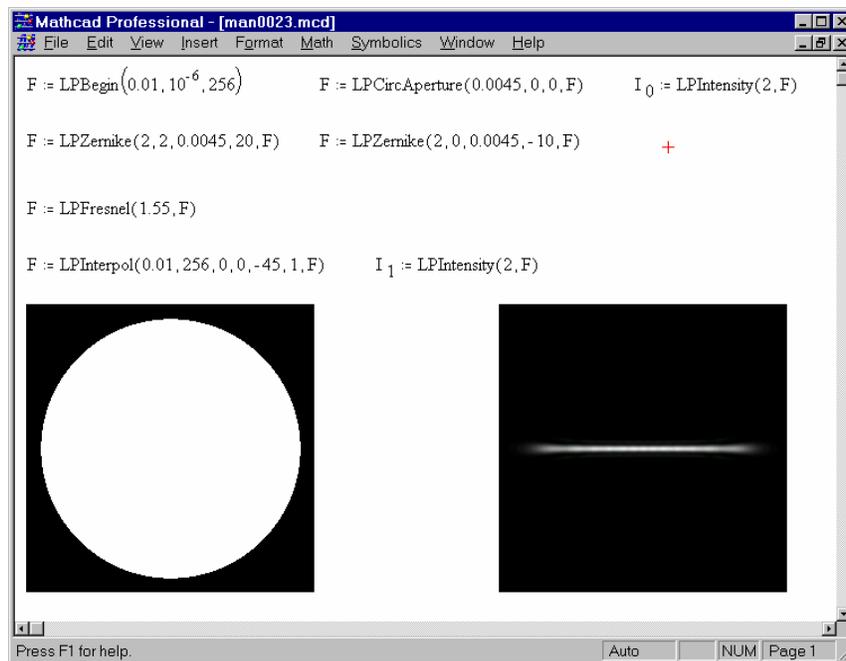


Figure 16 The intensity in the input plane and after propagation through a cylindrical system modelled as a combination of Zernike polynomials and free space propagation.

### 5.2.9 Spherical co-ordinates

The principle of beam propagation in the “floating” co-ordinate system (for the case of a lens wave guide) is shown in Figure 17

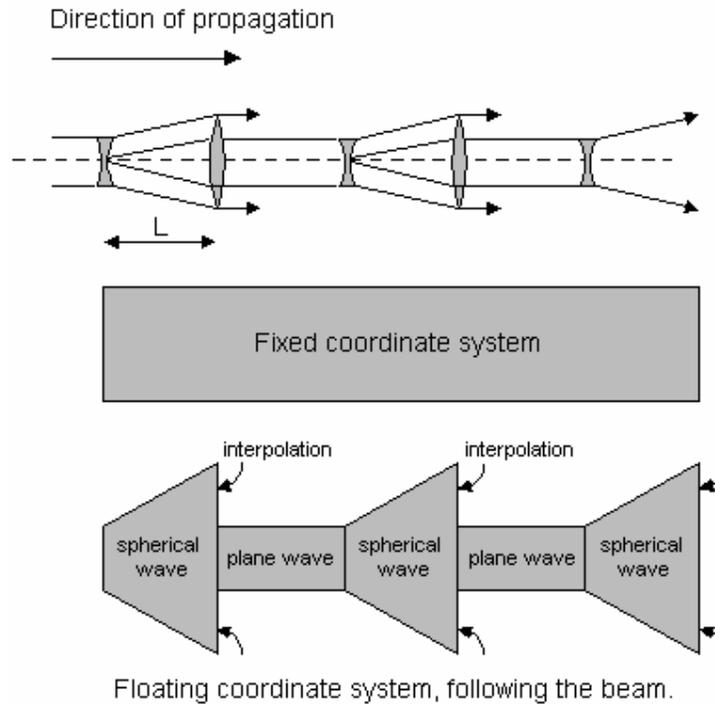


Figure 17 Illustration for the light propagation in lens wave guides with fixed and floating co-ordinate systems.

The spherical co-ordinates follow the geometrical section of the divergent or convergent light beam. Propagation in spherical co-ordinates is implemented with programs `LPensForward` and `LPensFresnel`. Both filters accept two parameters: the focal distance of the lens, and the distance of propagation. When `LPensForward` or `LPensFresnel` is called, it “bends” the coordinate system so, that it follows the divergent or convergent spherical wave front, and then propagates the field to the distance  $z$  in the transformed co-ordinates. The filter `LPConvert` should be used to convert the field back to the rectangular co-ordinate system. Some `LightPipes` filters can not be applied to the field in spherical co-ordinates. As the co-ordinates follow the geometrical section of the light beam, operator `LPensForward(10,10,Field)` will produce floating exception because the calculations can not be conducted in a grid with zero size (that is so in the geometrical approximation of a focal point). On the other hand, diffraction to the focus is equivalent to the diffraction to the far field (infinity), thus the FFT convolution algorithm will not work properly anyway. To model the diffraction into the focal point, a more complicated trick should be used:

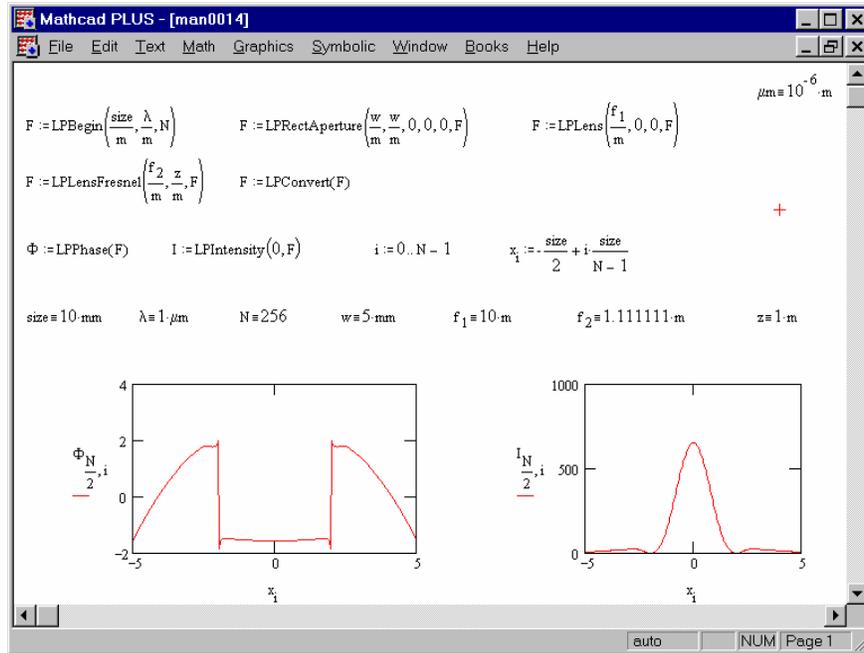


Figure 18 Diffraction to the focus of a lens using spherical co-ordinates.

In Figure 18 we calculate the diffraction to the focus of a lens with a focal distance of 1m. It is represented as a combination of a weak phase mask  $\text{LPLens}(f_1, F)$  and a “strong” geometrical co-ordinate transform  $\text{LPLensFresnel}(f_2, z, F)$ . The grid after propagation is 10 times narrower than in the input plane. The focal intensity is 650 times higher than the input intensity and the wave front is plain as expected.

#### 5.2.10 User defined phase and intensity filters

The phase and intensity of the light beam can be manipulated in several ways. The phase and intensity distributions may be produced within Mathcad as shown in the next examples:

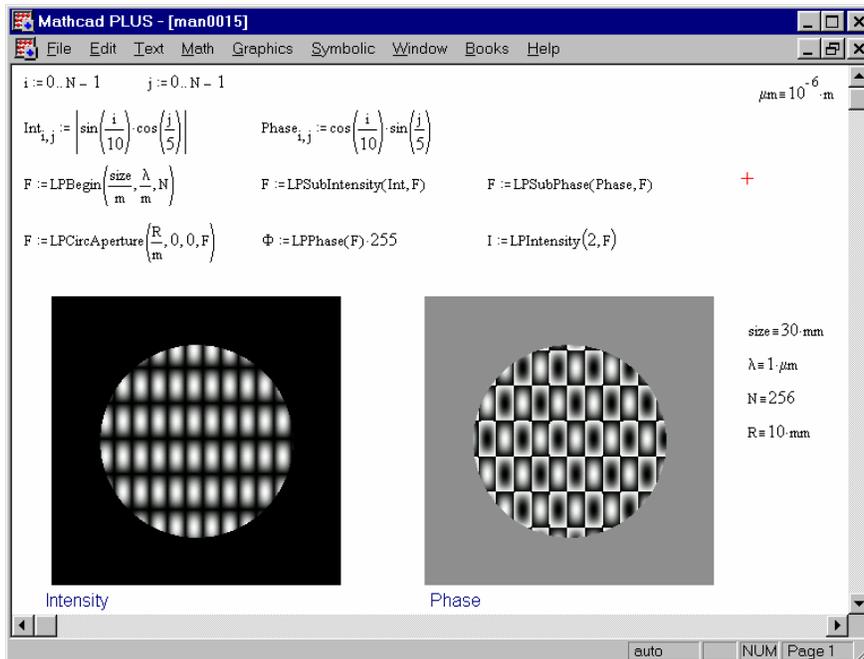


Figure 19 An arbitrary intensity- and phase distribution.

You can also create your own mask with a program like Microsoft Windows95 Paint. In the next example we made an arrow using Paint and stored it as a 200 pixels width x 200 pixels height, black-and-white, monochrome bitmap file (for example: arrow.bmp). This file can be read into Mathcad using the READBMP command. Next the arrow can be used as a filter:

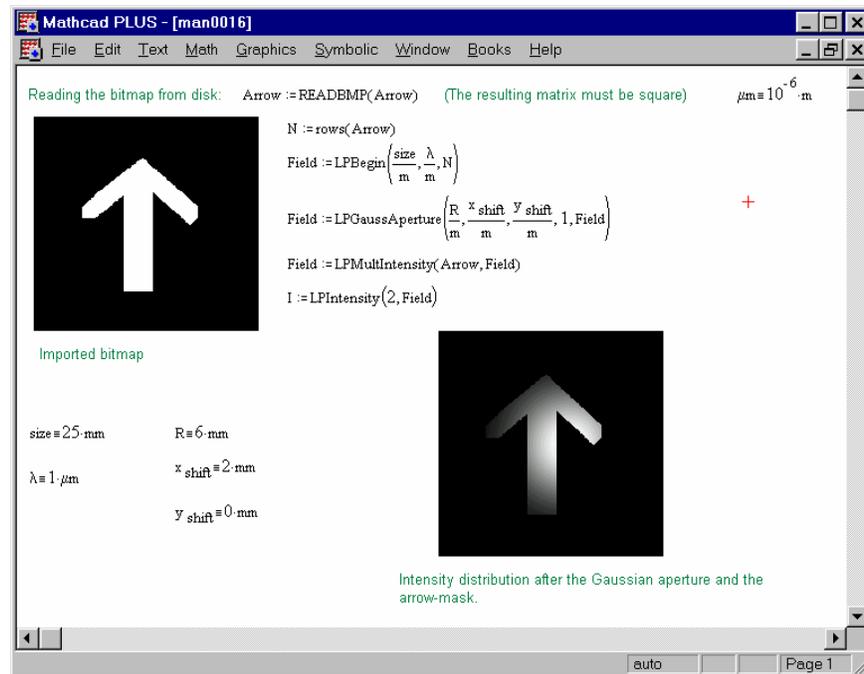


Figure 20 Illustration of importing a bit map from disk.

### 5.2.11 Random filters

There are two random filters to introduce a random intensity or a random phase distribution in the field. The commands LPRandomIntensity and LPRandomPhase need a seed to initiate the random number generator. Use has been made of the standard C function rand. The LPRandomPhase command needs a maximum phase value (in radians). The LPRandomIntensity command yields a normalised random intensity distribution. The LPRandomIntensity and the LPRandomPhase commands leave the phase and the intensity unchanged respectively.

### 5.2.12 FFT and spatial filters

LightPipes For Mathcad 1.1 and later versions provide a possibility to perform arbitrary filtering in the Fourier space. There is an operator, performing the Fourier transform of the whole data structure: LPPipFFT.

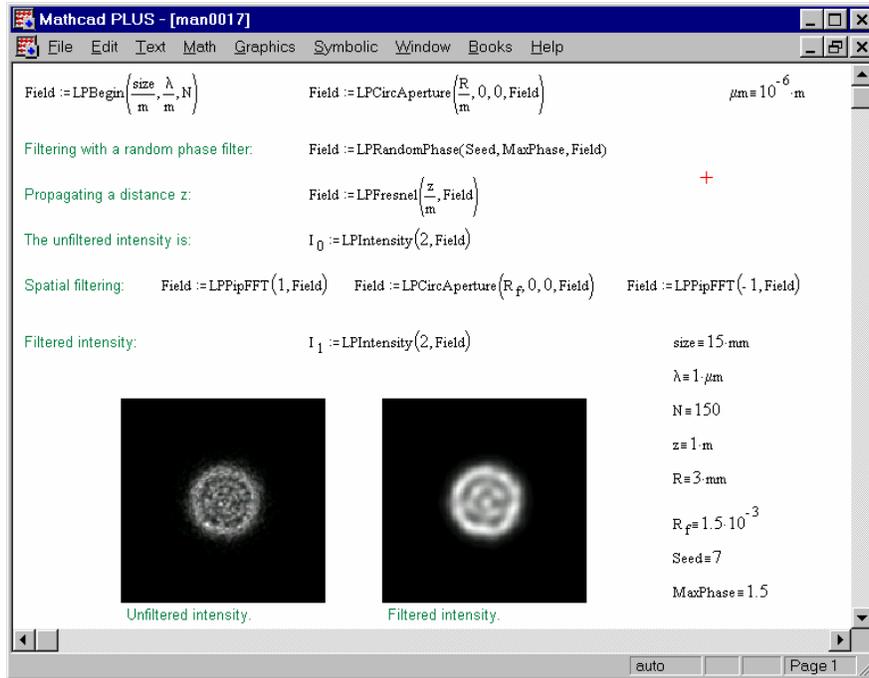


Figure 21 Intensity distributions before and after applying a spatial filter

Note, we still can apply all the intensity and phase filters in the Fourier-space. The whole grid size in the angular frequency domain corresponds to  $2\pi\lambda/\Delta x$ , where  $\Delta x$  is the grid step. One step in the frequency domain corresponds to  $2\pi\lambda/x$ , where  $x$  is the total size of the grid. LightPipes for Mathcad filters do not know about all these transformations, so the user should take care about setting the proper size (using the relations mentioned) of the filter (still in linear units) in the frequency domain.

### 5.2.13 Laser amplifier

The LPGain command introduces a simple single-layer model of a laser amplifier. The output field is given by:

$$F_{out}(x, y) = \sqrt{\exp\left[\frac{G_0 L}{1 + \frac{I(x, y)}{I_{sat}}}\right]} F_{in}(x, y) \quad (5.14)$$

where  $G_0$  is the small signal gain,  $L$  is the length of the gain medium,  $I(x, y)$  is the intensity and  $I_{sat}$  is the saturation intensity.

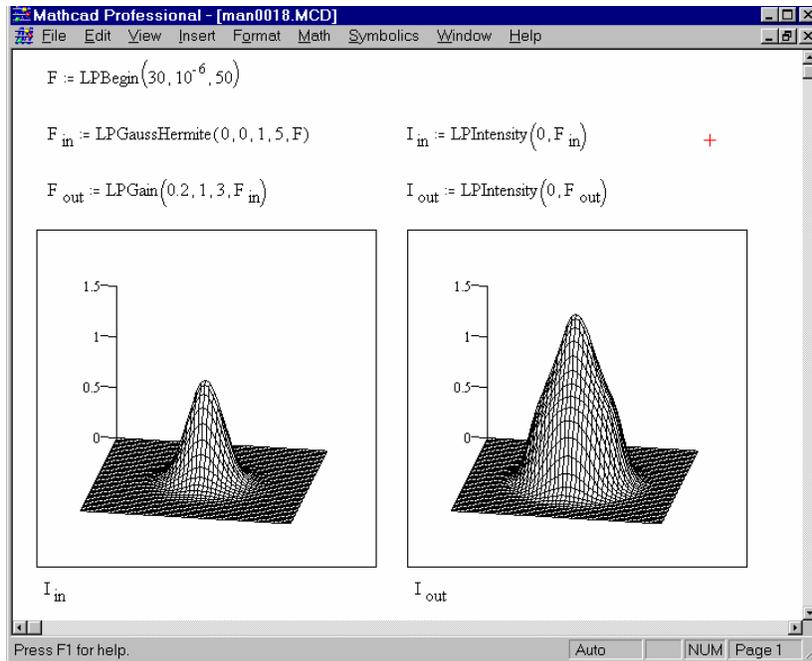


Figure 22 The effect of a saturable laser gain-section on the field.

#### 5.2.14 Diagnostics: Strehl ratio, beam power

The LPStrehl command calculates the Strehl ratio of the field, defined as:

$$\text{StrehlRatio} = \frac{\left( \iint \text{Re}(F_{in}(x, y)) dx dy \right)^2 + \left( \iint \text{Im}(F_{in}(x, y)) dx dy \right)^2}{\left( \iint |F_{in}(x, y)| dx dy \right)^2} \quad (5.15)$$

The next example calculates the Strehl ratio of a field with increasing random phase fluctuations:

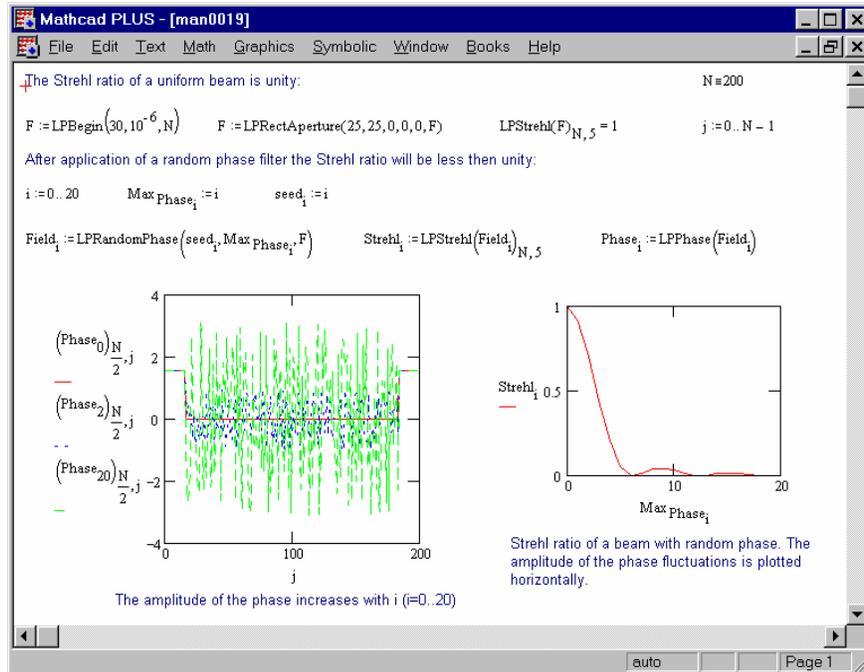


Figure 23 Demonstration of the use of the LPStrehl function to calculate the Strehl ratio (beam quality).

The LPNormal command normalises the field according to:

$$F_{out} = \frac{F_{in}(x, y)}{\sqrt{P}} \quad (5.16)$$

$$P = \iint (|F_{in}(x, y)|)^2 dx dy$$

where P is the total beam power.

### 5.2.15 Polarization.

In LightPipes for Mathcad version 1.3 polarization has been introduced simply by introducing a second field. Three new commands are introduced in this version: LPPolarizer, LPReflectMultiLayer and LPTransmitMultiLayer. LPPolarizer outputs a linearly polarized beam calculated from two input fields for the s- and the p-components. LPReflectMultiLayer and LPTransmitMultiLayer describe reflection from and transmission through a stack of thin layers on a substrate respectively. The polarization state, the thickness and the (complex) refractive index of the layers can be given to calculate more or less complicated coatings and study their properties.

In Figure 24 we define a linear polarized beam polarized at an angle of  $45^\circ$  because the amplitudes and phases of the s- and p- components are chosen equal. In Figure 25 the polarization of the beam is analyzed with the LPPolarization command to simulate an analyzer. In Figure 26 we simulate a phase retarding plate.

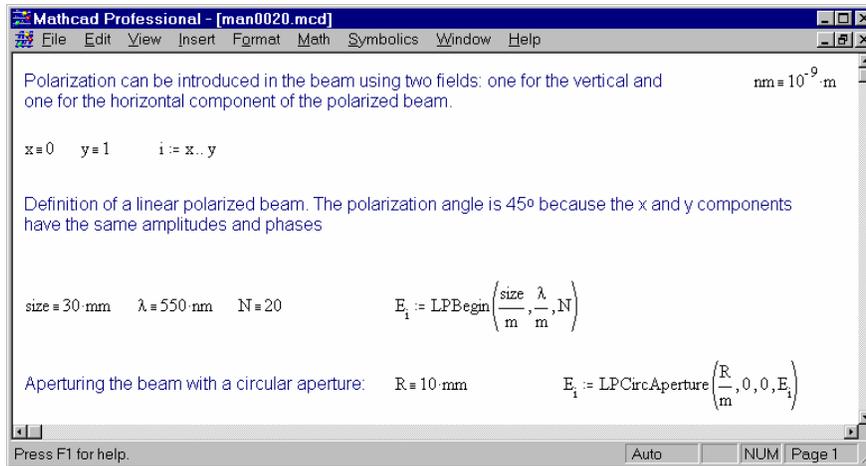


Figure 24 Polarized beams in LightPipes for Mathcad.

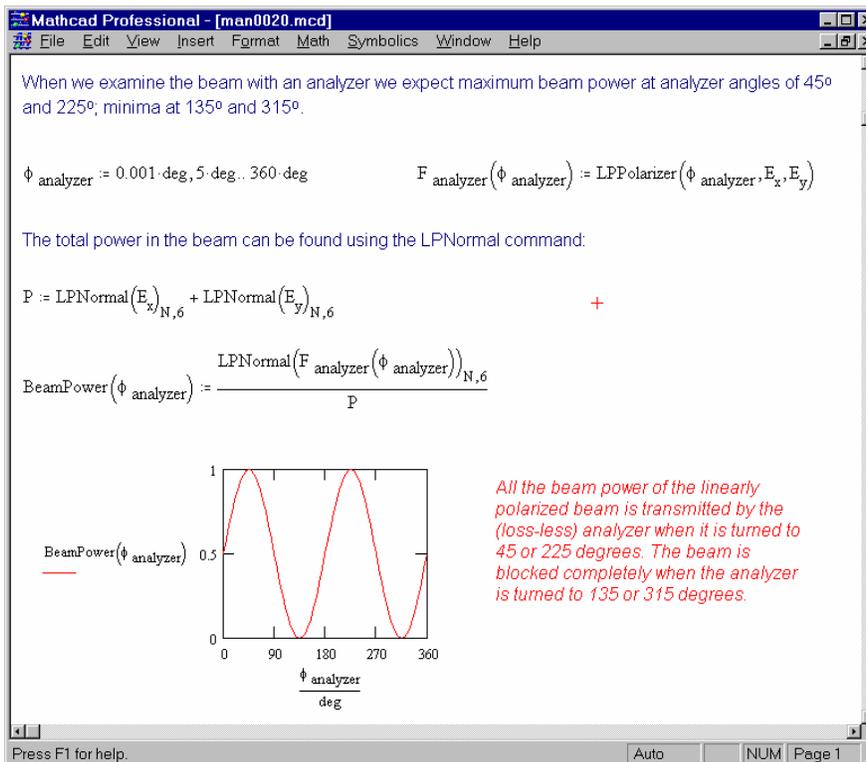


Figure 25 Use of the LPPolarizer command to simulate an analyzer.

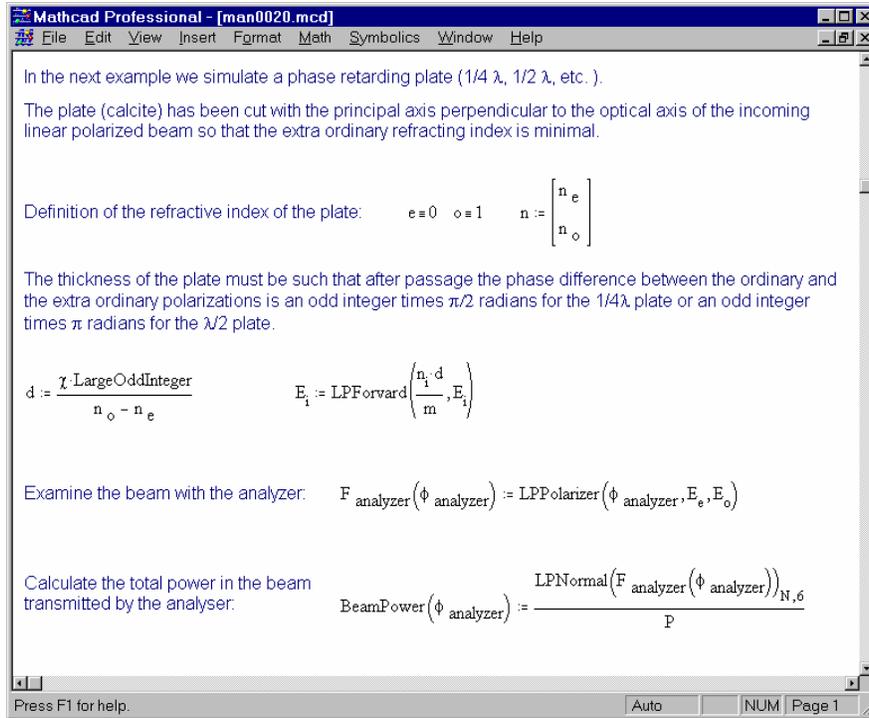


Figure 26 Simulation of a phase retarding plate.

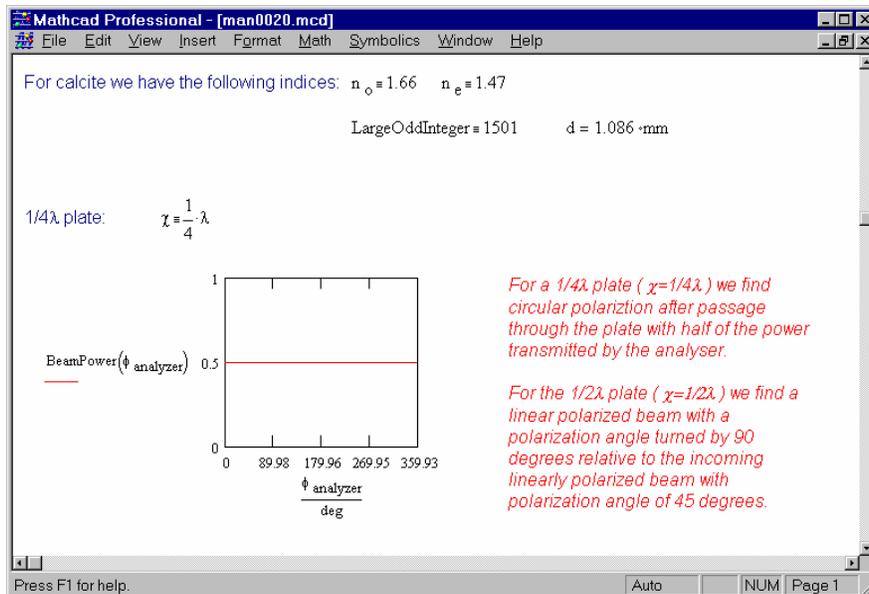


Figure 26 Simulation of a phase retarding plate. (cont.)

### 5.2.16 Reflection from and transmission through multilayer coatings.

The LPTransmitMultiLayer and LPReflectMultiLayer commands describe transmission through and reflection from a multi-layer film for arbitrary angles of incidence and for p- or s-polarization. The layers of the film must be input to the commands as vectors for the thickness and refractive index. The refractive index of the layers can be complex in order to introduce absorption.

Mathcad Professional - [man0021.mcd]

File Edit View Insert Format Math Symbolics Window Help

$\mu\text{m} = 10^{-6} \cdot \text{m}$      $\text{nm} = 10^{-9} \cdot \text{m}$

In this example we demonstrate the reflection from and transmission through a coated substrate denoted by: glass[AB(2C)BA]<sup>10</sup>air

See Optics Handbook part I, OSA, p. 42.55, fig. 55(b)

Index air:  $N_0 = 1$       Index substrate:  $N_{\text{sub}} = 1.5$       Angle of incidence:  $\theta_0 = 0 \cdot \text{deg}$

size = 30 mm     $n_{\text{grid}} = 8$      $s = 0$      $p = 1$

We use the smallest grid-size to reduce execution time.

Building the index and thickness vectors of one unit: a stack of materials A, B and C organized as ABCCBA:

Design wavelength:  $\lambda_0 = 500 \cdot \text{nm}$

Refractive Indices:     $n_A = 1.38$        $n_B = 1.781$        $n_C = 2.3$

Thicknesses:       $d_A := \frac{\lambda_0}{12} \cdot \frac{1}{n_A}$        $d_B := \frac{\lambda_0}{12} \cdot \frac{1}{n_B}$        $d_C := \frac{\lambda_0}{12} \cdot \frac{1}{n_C}$

Vectors for the index and thickness:     $N_e := \begin{bmatrix} n_A \\ n_B \\ n_C \\ n_C \\ n_B \\ n_A \end{bmatrix}$        $d_e := \begin{bmatrix} d_A \\ d_B \\ d_C \\ d_C \\ d_B \\ d_A \end{bmatrix}$       +

Press F1 for help.      Auto      NUM      Page 1

Figure 27 Simulation of a multilayer coating using the LPTransmitMultiLayer and LPReflectMultiLayer commands.

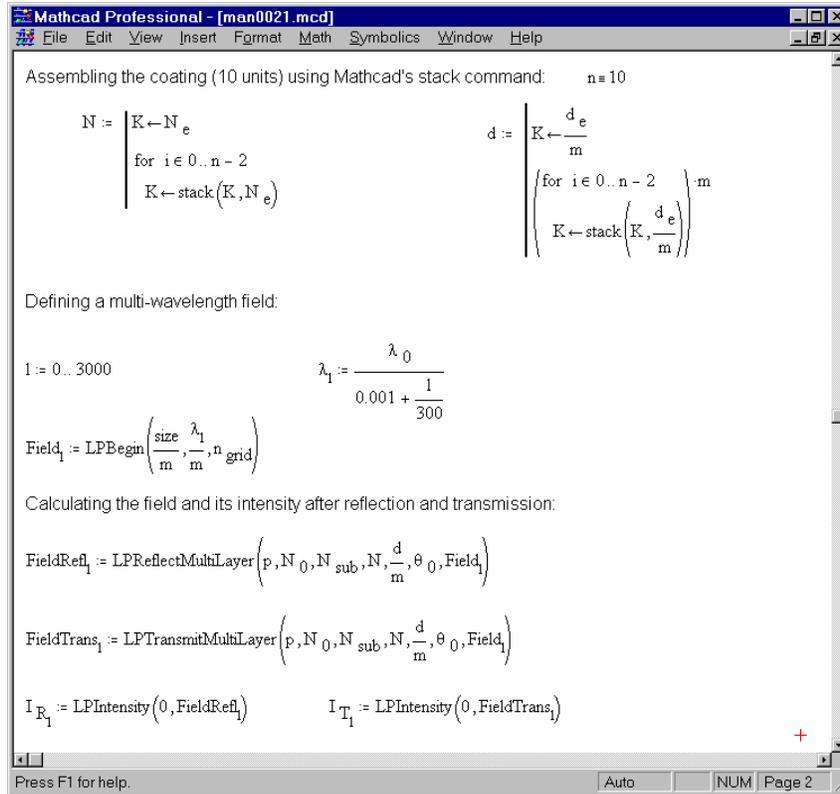


Figure 27 Simulation of a multilayer coating using the LPTransmitMultiLayer and LPReflectMultiLayer commands. (cont.)

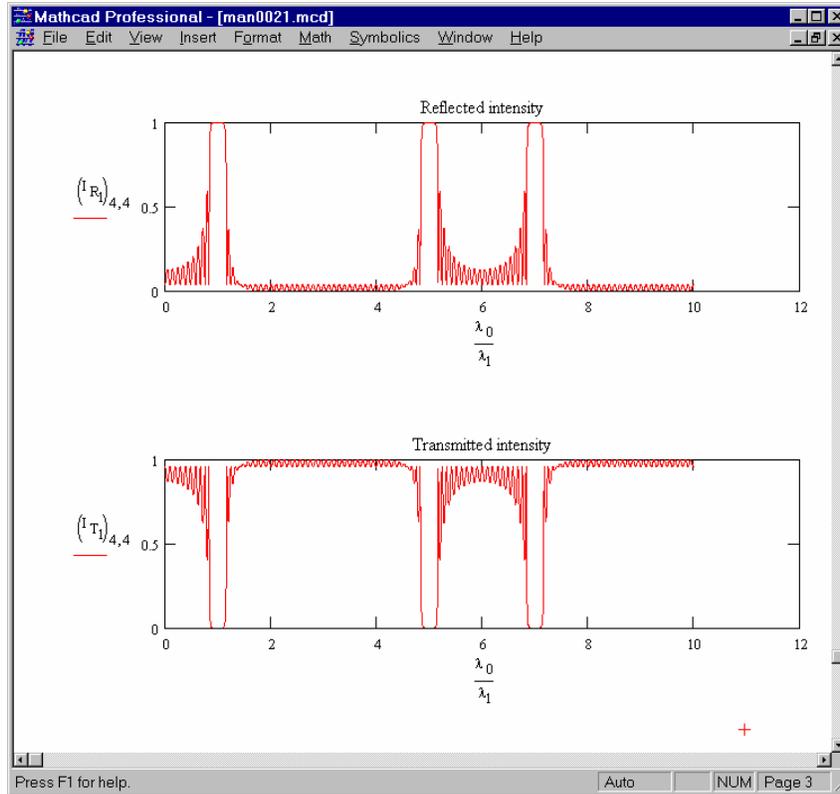


Figure 27 Simulation of a multilayer coating using the LPTransmitMultiLayer and LPReflectMultiLayer commands. (cont.)



## 6. Example models

### 6.1 Shearing interferometer

We can easily model a shearing interferometer using the operators introduced in the previous sections. Suppose we have to model a device shown in Figure 28.

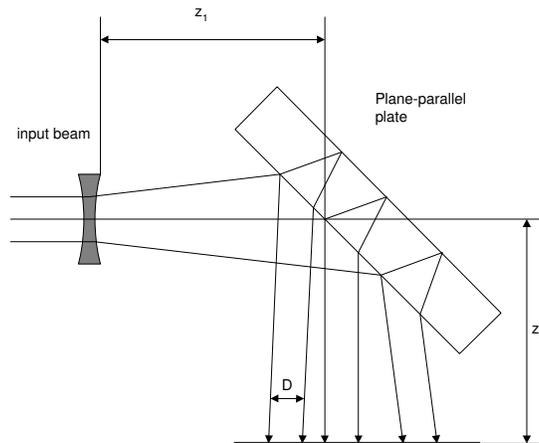


Figure 28 Shearing interferometer

Let  $z_1 = 0.5\text{m}$ ,  $z_2 = 0.5\text{m}$ ,  $\lambda = 500\text{ nm}$ , the focal length of the lens is  $-20\text{m}$ , the beam is divided 1:1 and one of the two beams is shifted by  $D = 3\text{mm}$  in the  $x$ -direction and  $D_1 = 1\text{mm}$  in the  $Y$ -direction after reflection on the back surface of the plane-parallel plate. Figure 29 shows the calculation in LightPipes for Mathcad.

**Shearing interferometer.** nm =  $10^{-9}$  · m

The shearing interferometer consists of a weak negative lens followed by a plane-parallel plate with reflectivity,  $R_{\text{plate}}$  that splits the beam in two beams and shifts one of the beams. An interference pattern can be observed on a screen.

Definition of the light source:  $\text{Field} := \text{LPBegin}\left(\frac{\text{size}}{\text{m}}, \frac{\lambda}{\text{m}}, N\right)$        $\text{Field} := \text{LPCircAperture}\left(\frac{R}{\text{m}}, 0, 0, \text{Field}\right)$

Transmission of the field through the negative lens and propagation to the plane-parallel plate:

$\text{Field} := \text{LPLens}\left(\frac{f}{\text{m}}, 0, 0, \text{Field}\right)$        $\text{Field} := \text{LPForward}\left(\frac{z_1}{\text{m}}, \text{Field}\right)$

Reflection of the beam by the front surface of the plate:  $\text{Field}_1 := \text{LPIntAttenuator}(R_{\text{plate}}, \text{Field})$

Reflection of the beam by the rear surface and shifting the beam  $D$  in the  $x$ -direction and  $D_1$  in the  $y$ -direction:

$\text{Field}_2 := \text{LPIntAttenuator}(1 - R_{\text{plate}}, \text{Field})$        $\text{Field}_2 := \text{LPInterpol}\left(\frac{\text{size}}{\text{m}}, N, \frac{D}{\text{m}}, \frac{D_1}{\text{m}}, 0, 1, \text{Field}_2\right)$

Combining the two beams by addition of the fields:  $\text{Field} := \text{LPBeamMix}(\text{Field}_1, \text{Field}_2)$

Propagation of the beam to the screen at a distance  $z_2$  from the plate:  $\text{Field} := \text{LPForward}\left(\frac{z_2}{\text{m}}, \text{Field}\right)$

$I := \text{LPIntensity}(2, \text{Field})$

$i := 0..N - 1$        $x_i := -\frac{\text{size}}{2} + i \frac{\text{size}}{N - 1}$

auto      Page 1

Figure 29 Shearing interferometer program.

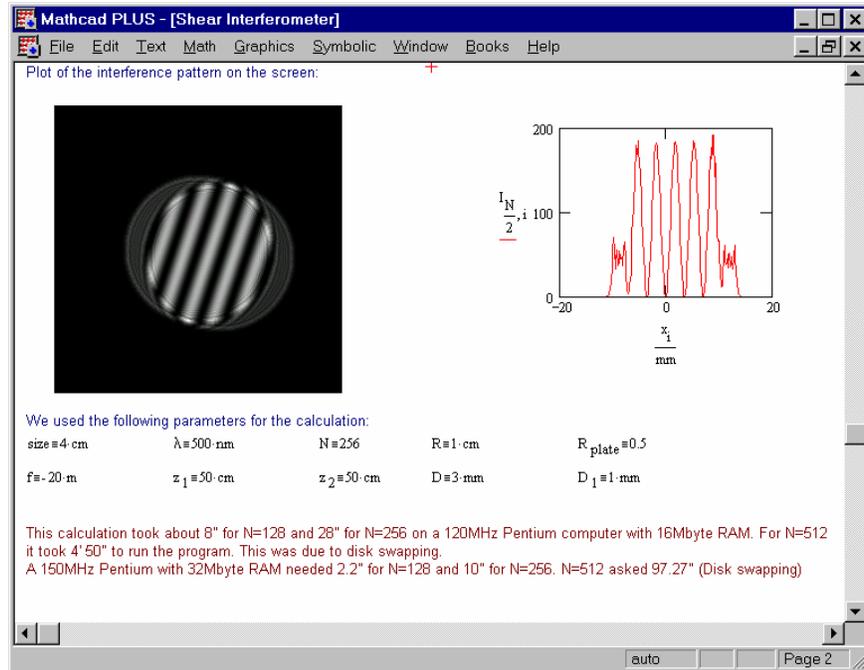


Figure 29 Shearing interferometer program. (cont.)

We can put a source of arbitrary aberration on the place of the lens. Changing the amplitude and the order of aberration we can obtain all the shearing interferograms shown in the chapter about shearing interferometers of [8]. Figure 30 shows the use of the LPZernike command to obtain these aberrations.

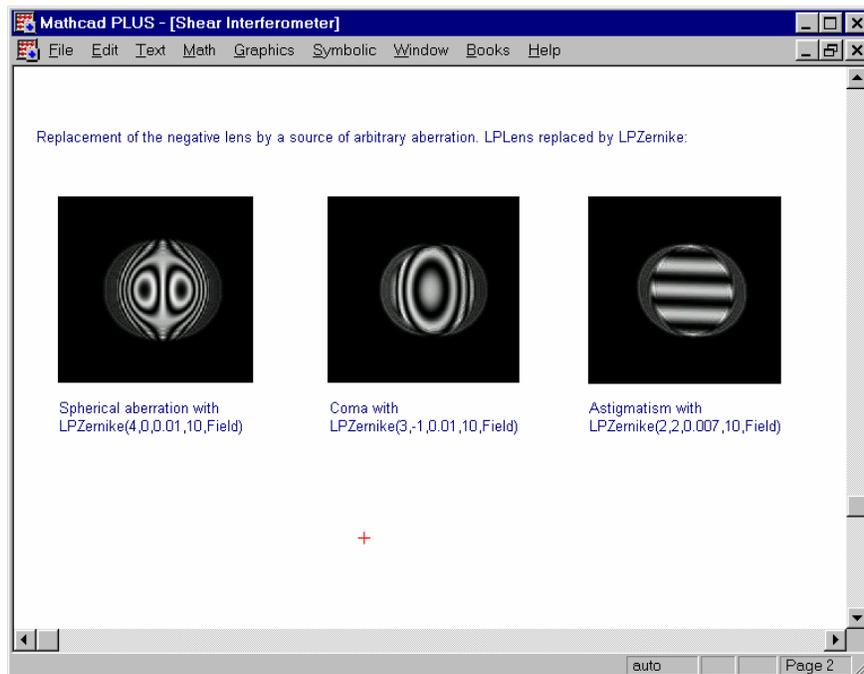


Figure 30 Shearing interferograms of spherical aberration, coma and astigmatism.

## 6.2 Rotational shearing interferometer

In the rotational shear interferometer the beam interferes with a copy of itself rotated by an angle  $\alpha$  around the optical axis. This interferometer is useful for detecting asymmetrical aberrations. We shall only consider a bare-bone (no propagation and diffraction) model of such an interferometer. The results of aberration calculations for coma, astigmatism and high order Zernike aberration are shown in Figure 32.

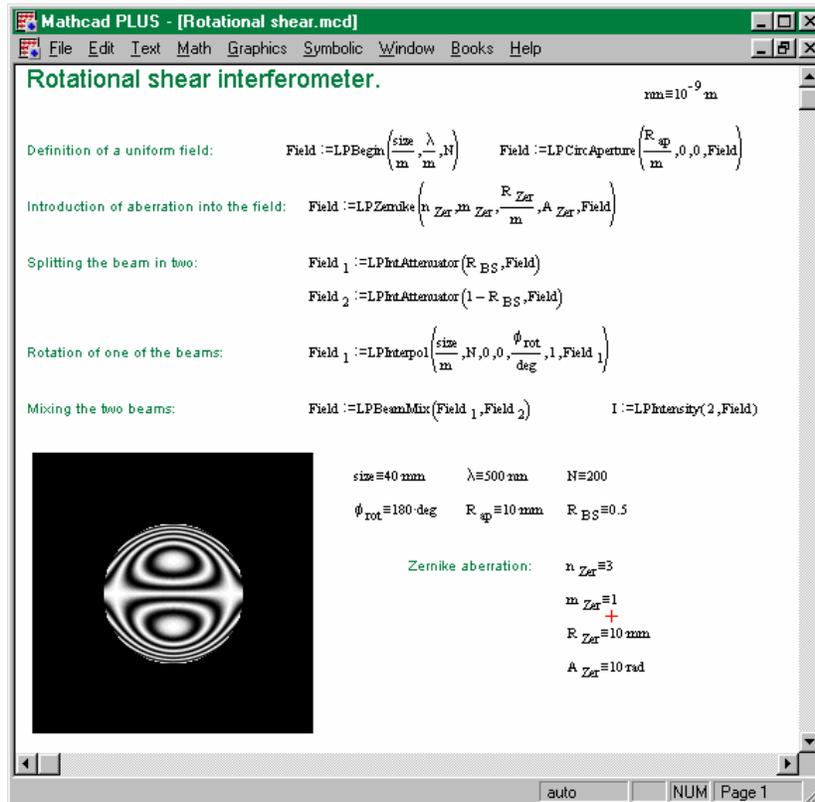


Figure 31 Rotational shearing interferometer.

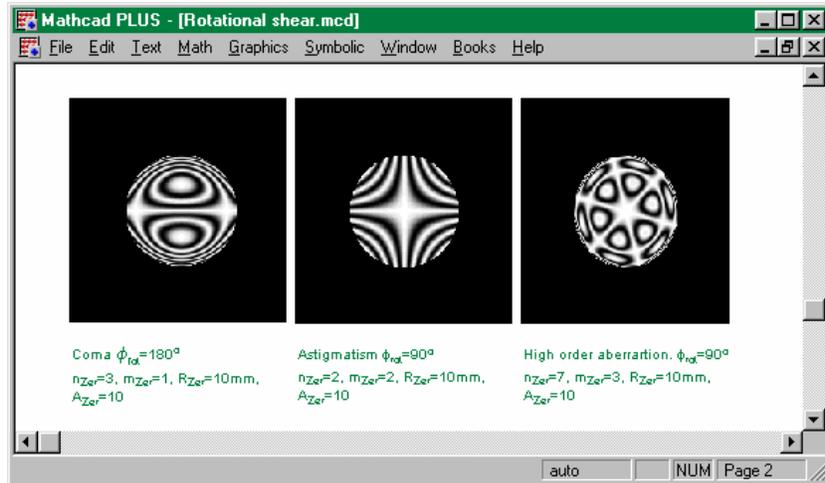


Figure 32 Rotational interferograms of coma  $180^\circ$ , astigmatism  $90^\circ$  and high order Zernike aberration  $90^\circ$

### 6.3 Radial shear interferometer

In the radial shearing interferometer the beam interferes with a copy of itself magnified by a factor  $M$ . This interferometer is useful for detecting axisymmetrical aberrations. As for the previous case, we shall consider only a bare-bone (no propagation and diffraction) model of such an interferometer.

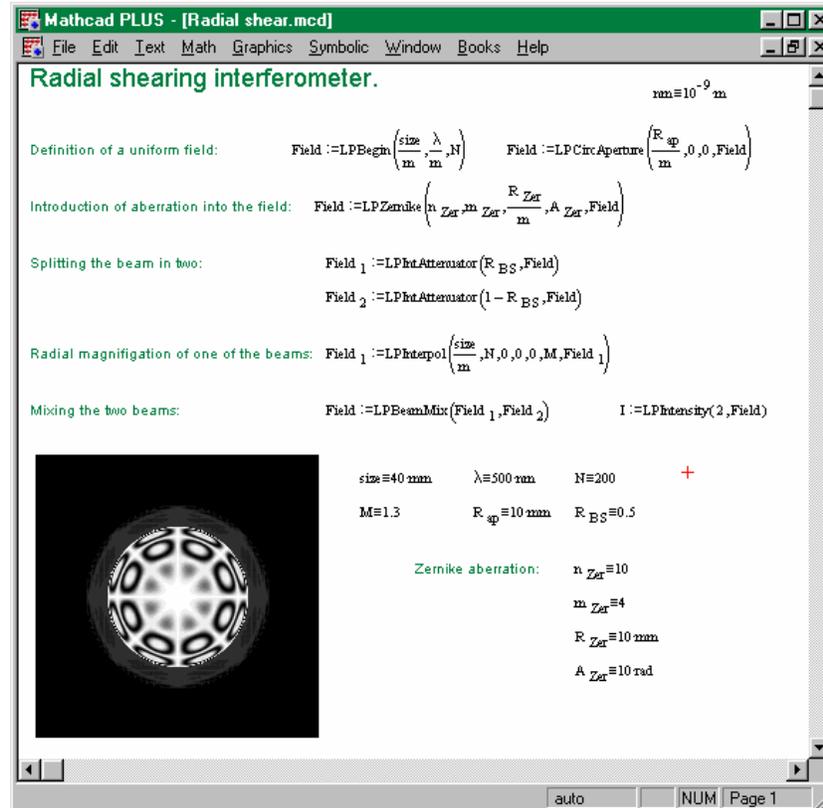


Figure 33 Radial shearing interferometer.

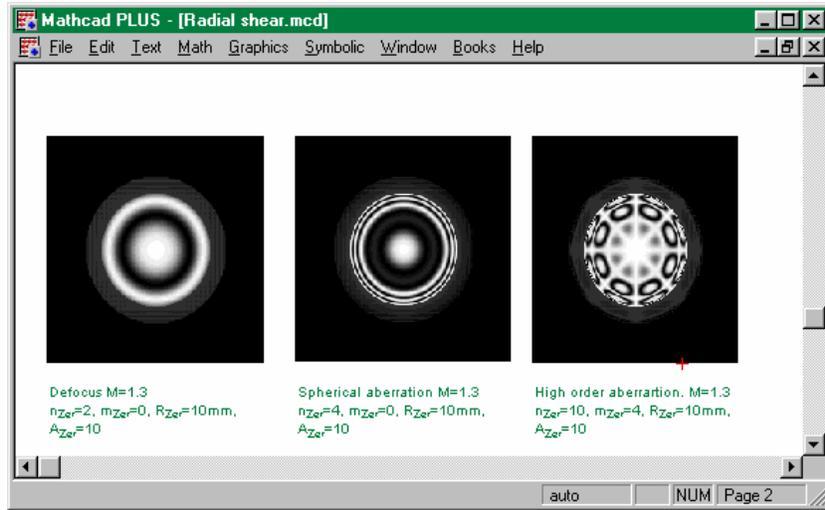


Figure 33 Radial shearing interferometer. (cont.) Radial shearing interferograms of defocus, spherical aberration and higher-order aberration.

## 6.4 Michelson Interferometer.

In this example we simulate a Michelson interferometer without and with a mis-aligned mirror to demonstrate circular and localized fringes.

**The Michelson interferometer.**  $nm = 10^{-9} \text{ m}$   $mrad = 10^{-3}$

The Michelson interferometer has a beam splitter to split the beam in two parts. The two beams are reflected from two mirrors of which one of them is movable. After the beams pass the beamsplitter again, they are recombined and fringes can be observed on a screen. One of the mirrors can be tilted to change the form of the fringes.

Definition of the beam:  $F := LPBegin\left(\frac{\text{size}}{m}, \frac{\lambda}{m}, N\right)$   $F := LPCircAperture\left(\frac{R}{m}, 0, 0, F\right)$

Weak converging lens:  $F := LPLens\left(\frac{f}{m}, 0, 0, F\right)$

Propagation the beam a distance  $z_1$  towards the beamsplitter:  $F := LPForward\left(\frac{z_1}{m}, F\right)$

Splitting the beam in two:  $F_1 := LPIntAttenuator(R_{BS}, F)$   
 $F_2 := LPIntAttenuator(1 - R_{BS}, F)$

Propagation of beam #1 to the mirror at  $z_2$  from the beamsplitter and back to the beamsplitter:  
 $F_1 := LPForward\left(\frac{z_2}{m}, F_1\right)$

Propagation of beam #2 to the second mirror at  $z_3$  from the beamsplitter, tilting the mirror and propagation back to the beamsplitter:  
 $F_2 := LPForward\left(\frac{z_3}{m}, F_2\right)$   $F_2 := LPTilt(t_x, t_y, F_2)$   $F_2 := LPForward\left(\frac{z_3}{m}, F_2\right)$

Passing beam #1 through the beam splitter and reflecting beam #2 at the beam splitter:  
 $F_1 := LPIntAttenuator(1 - R_{BS}, F_1)$   $F_2 := LPIntAttenuator(R_{BS}, F_2)$

Figure 34 Simulation of a Michelson interferometer.

Combining the two beams:  $F := LPBeamMix(F_1, F_2)$

Propagation of the beam to the screen at  $z_4$  from the beam splitter:  $F := LPForward\left(\frac{z_4}{m}, F\right)$

Calculation of the intensity at the screen:  $I := LPIntensity(2, F)$

Figure 34 Simulation of a Michelson interferometer. (cont.)

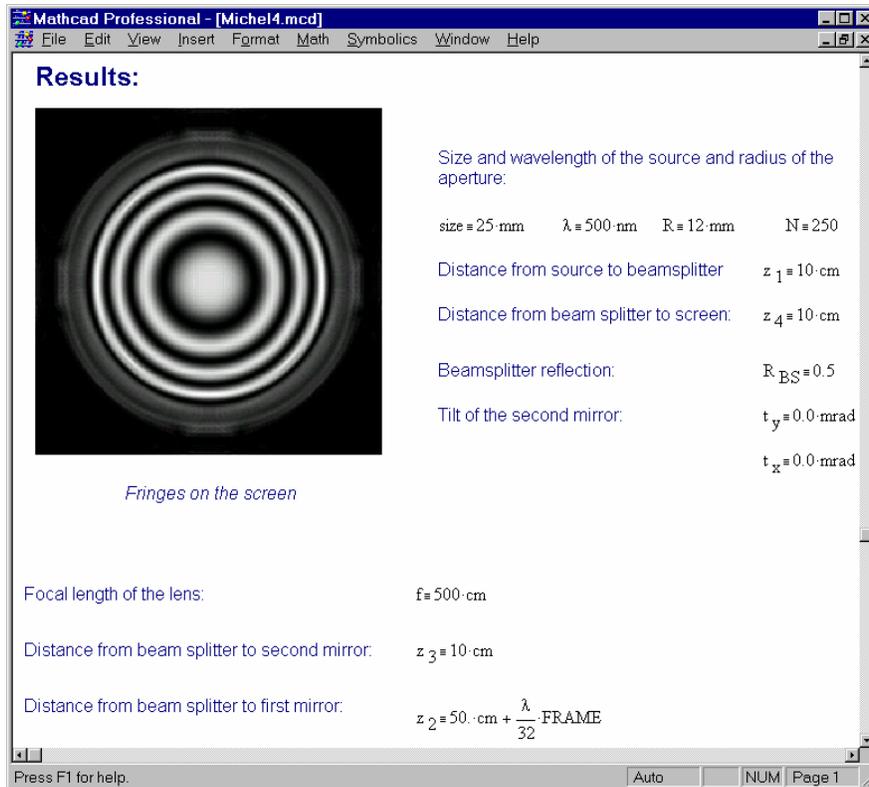


Figure 34 Simulation of a Michelson interferometer.(cont.) Circular fringes with aligned mirrors.

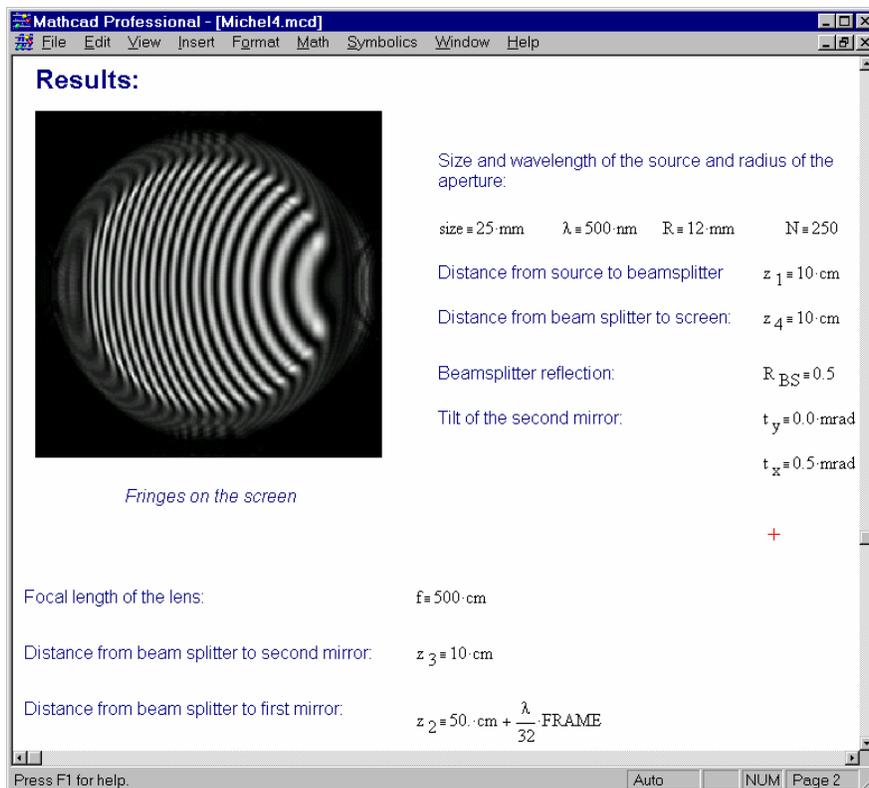


Figure 35. As Figure 34 with one of the mirrors mis-aligned to demonstrate localized fringes.

Using the animation feature of Mathcad it is illustrative to make a movie of the fringes while a mirror is moving. To do this let the variable 'FRAME' (See the Mathcad documentation) run from 0 to 31 in the examples shown in Figure 34 and Figure 35.



## 6.5 Twyman-Green interferometer

An example of the Twyman-Green interferometer is shown in Figure 36. Mirror 1 is plane and mirror 2 is aberrated (coma) with an aberration amplitude of  $1\mu\text{m}$ . The beam splitter is ideal and divides the beam 3:7.

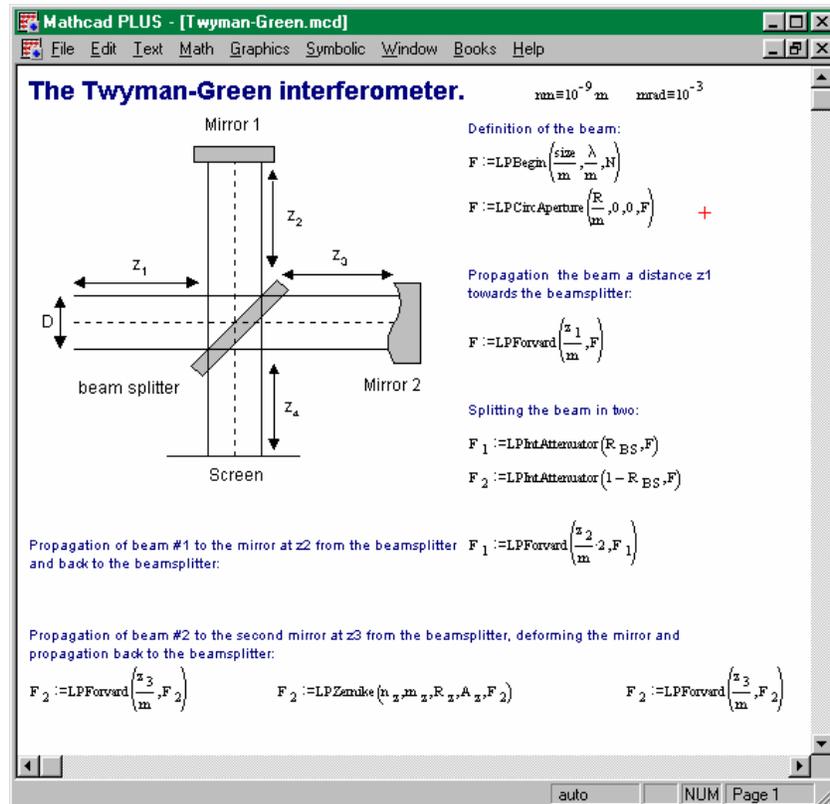


Figure 36 Twyman-Green interferometer.

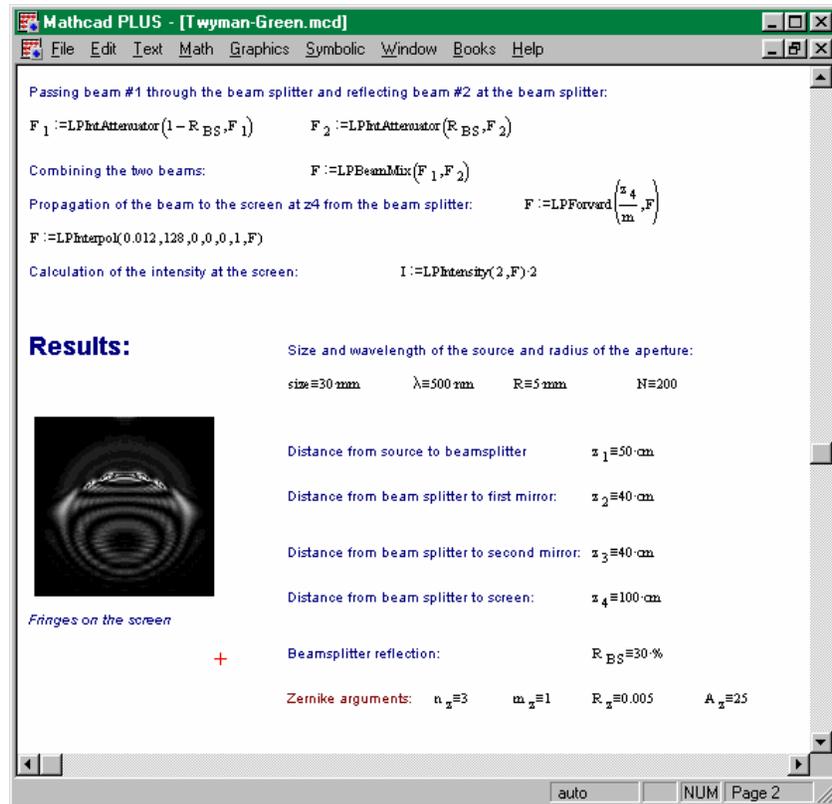


Figure 36 Twyman-Green interferometer. (cont.)

## 6.6 Unstable laser resonator

The unstable resonator can be simulated starting with an arbitrary field that circulates between the mirrors towards a steady state solution: the eigen-mode of the resonator. In this example we simulate a positive branch, confocal unstable resonator. In stead of mirrors, we use a lens-guide of alternating positive and negative lenses, separated a distance,  $L$ , the resonator length. The arguments of the commands have been made dimensionless by dividing them by their units.

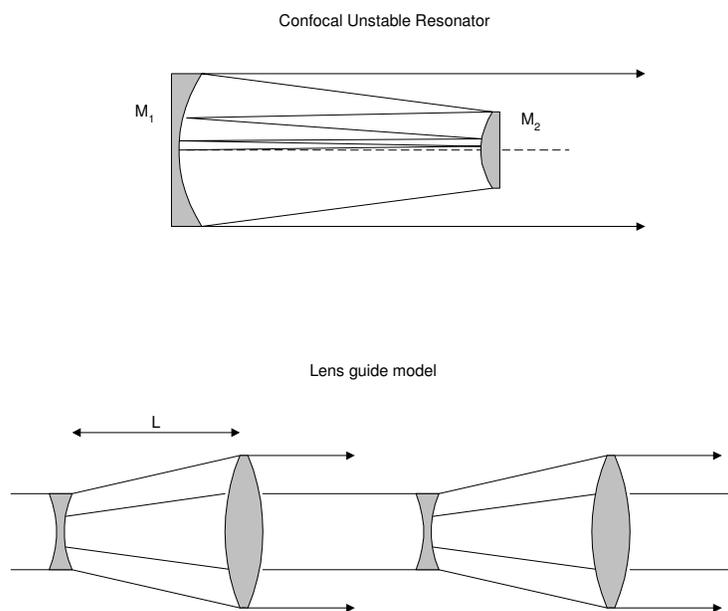


Figure 37 An unstable confocal resonator and its equivalent lens guide.

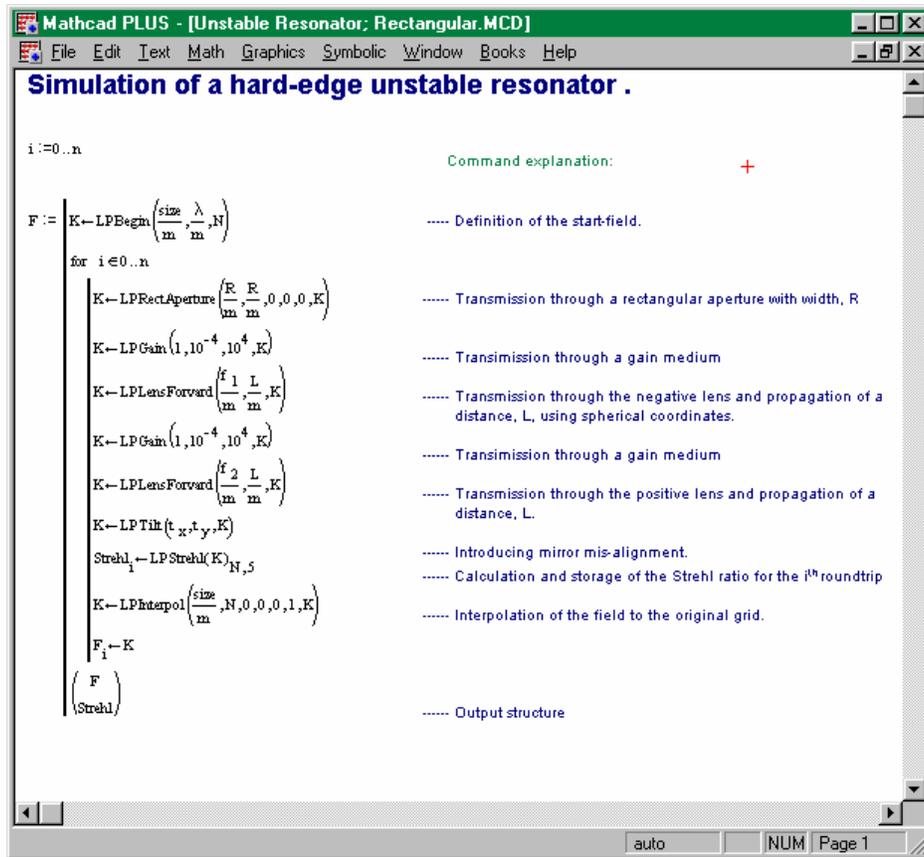


Figure 38 Simulation of a hard-edge confocal unstable resonator.

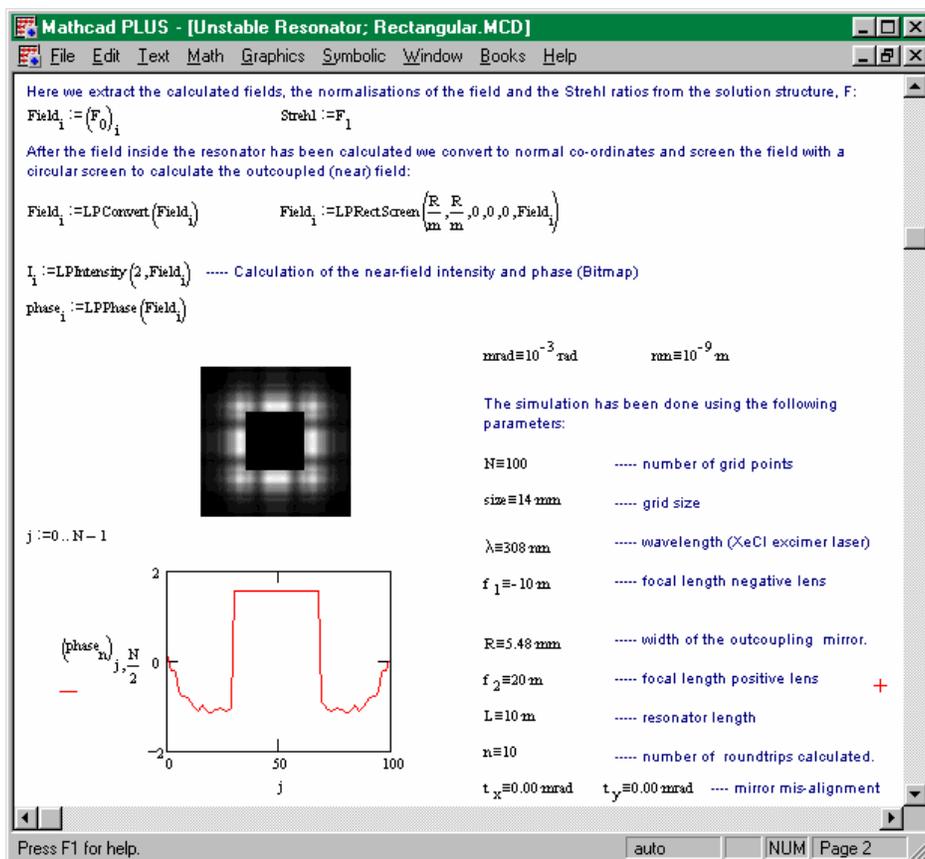


Figure 38 Simulation of a hard-edge confocal unstable resonator. (cont.)

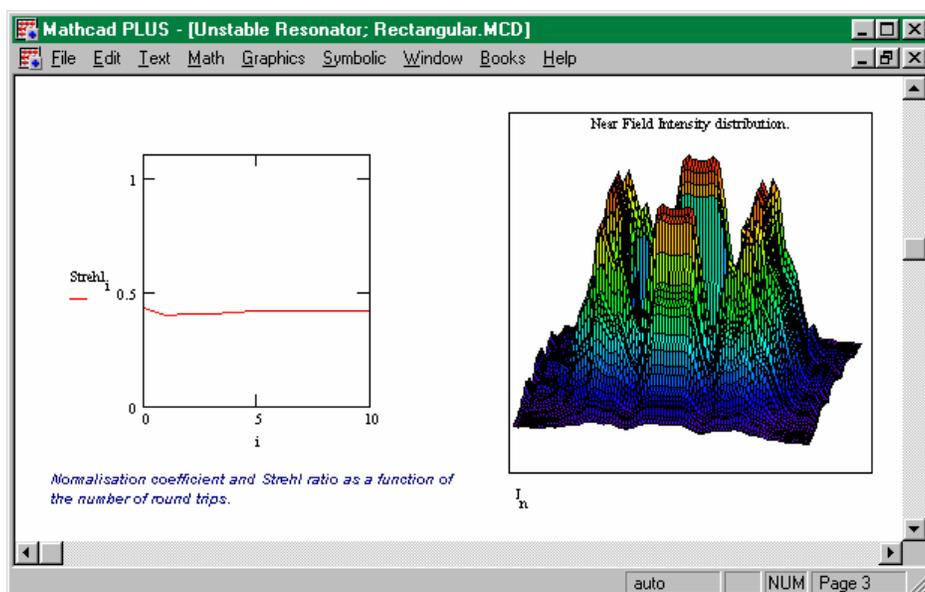


Figure 38 Simulation of a hard-edge confocal unstable resonator. (cont.)

### 6.6.1 Calculation of the far field

The far-field can be calculated by propagation of the field to the focus of a lens. Because the density of the grid soon becomes too low for a correct calculation we use spherical co-ordinates using the LPLensFresnel command. The LPLensFresnel(z,z) command, however, cannot be used for propagation to the focus, because the program tries to define a zero grid dimension and hence will generate a floating point error. To solve this problem we can use an extra weak positive lens with focal length  $f$  and propagate the field to a distance  $z$  using spherical co-ordinates with the LPLensFresnel( $f_f$ ,z) command.  $z$  is the focal length of the combined lenses:

$$\frac{1}{z} = \frac{1}{f_f} + \frac{1}{f}$$

or:

$$f_f = \frac{z f}{f - z}$$

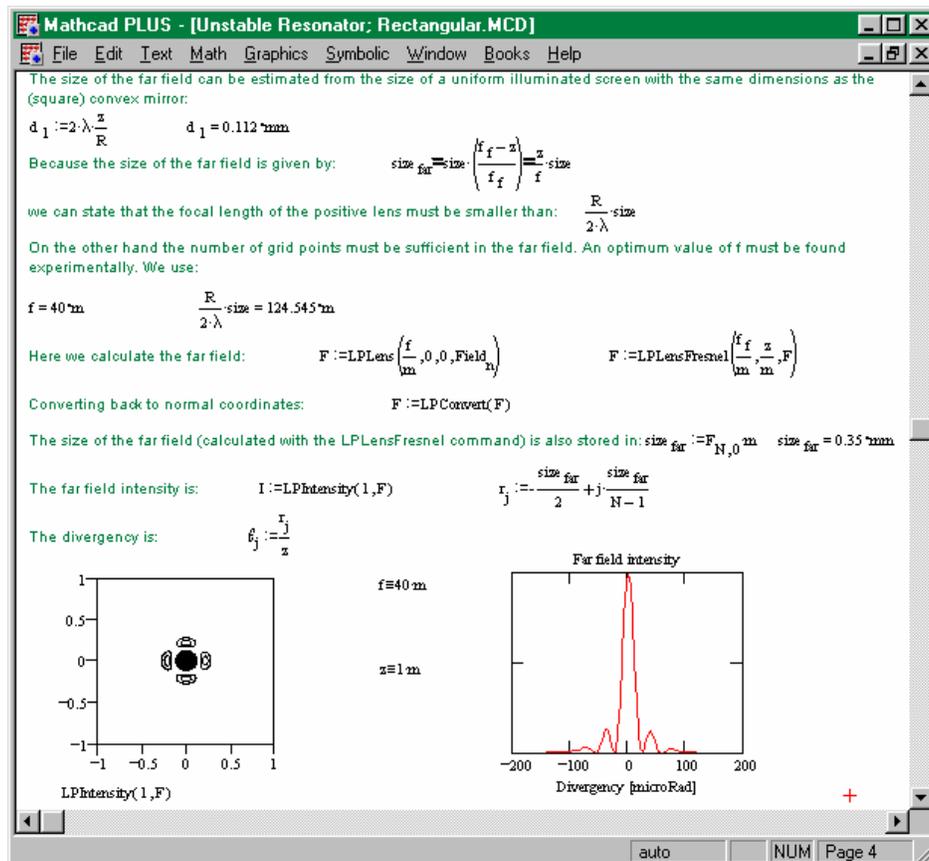


Figure 38 Simulation of a hard-edge confocal unstable resonator. (cont.)

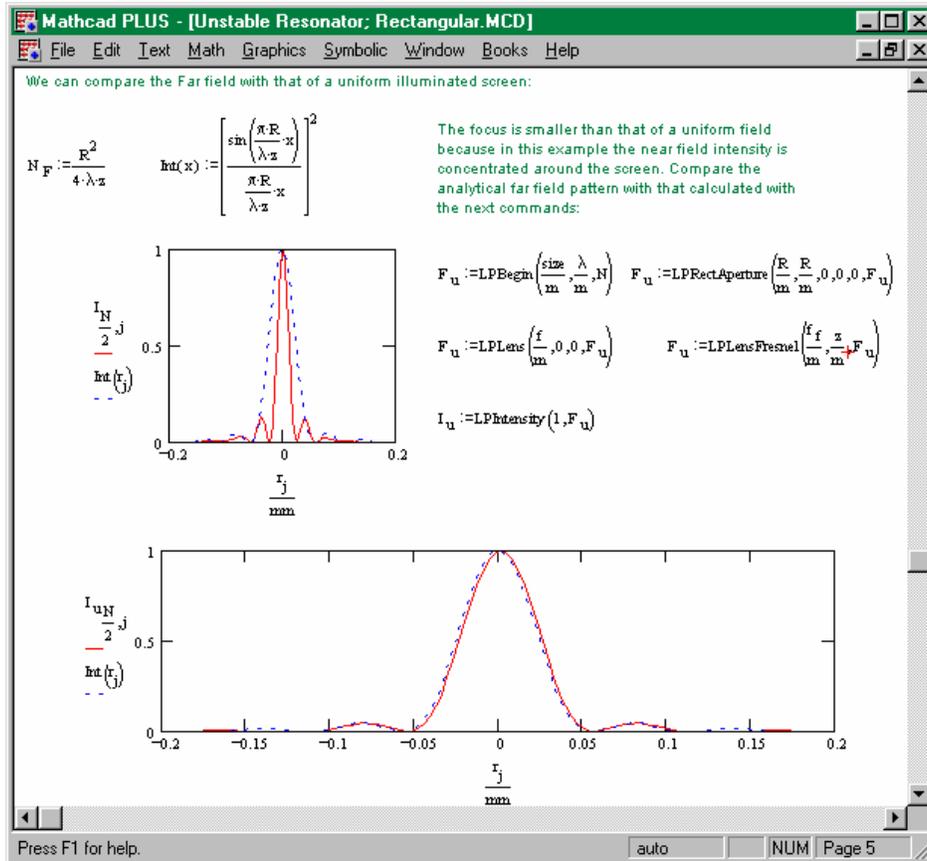


Figure 38 Simulation of a hard-edge confocal unstable resonator. (cont.)

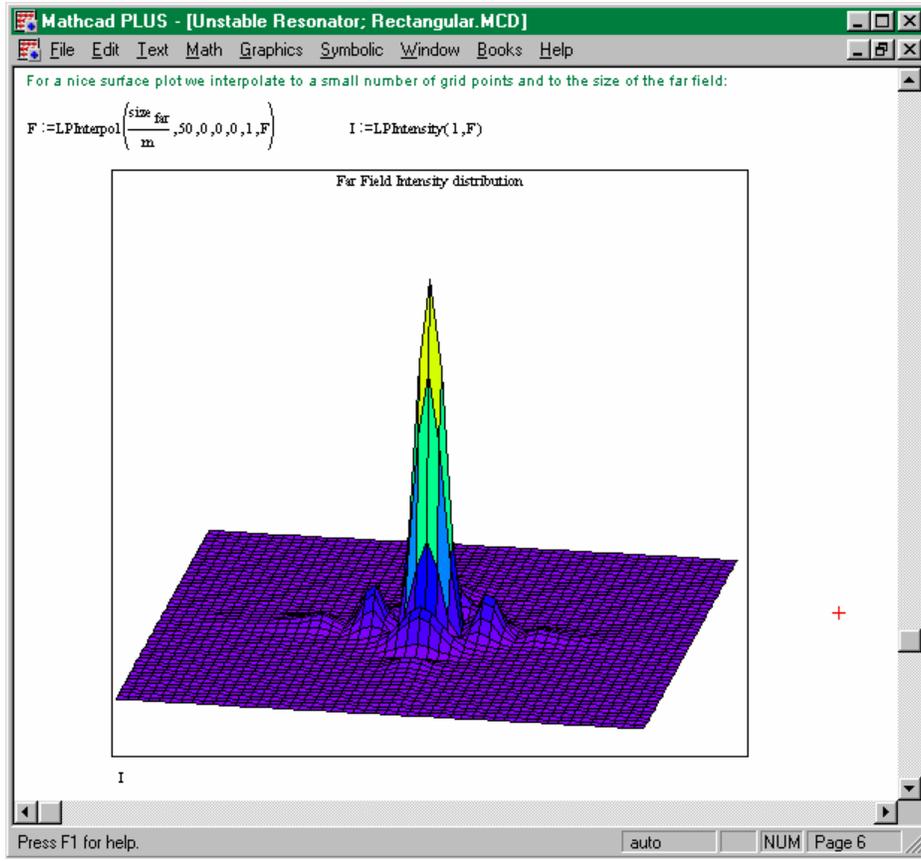


Figure 38 Simulation of a hard-edge confocal unstable resonator. (cont.)

## 6.7 Propagation in a lens-like/ absorptive medium.

In this example we model the propagation of a Gaussian beam in a lens-like waveguide. The profile of the refractive index is chosen such, that the beam preserves approximately its diameter in the waveguide (we use the fundamental mode). We'll consider the propagation of an axial mode, tilted with respect to the waveguide axis and a non-axial mode. In Figure 39 the quadratic index profile of the absorptive lens-like medium is made.

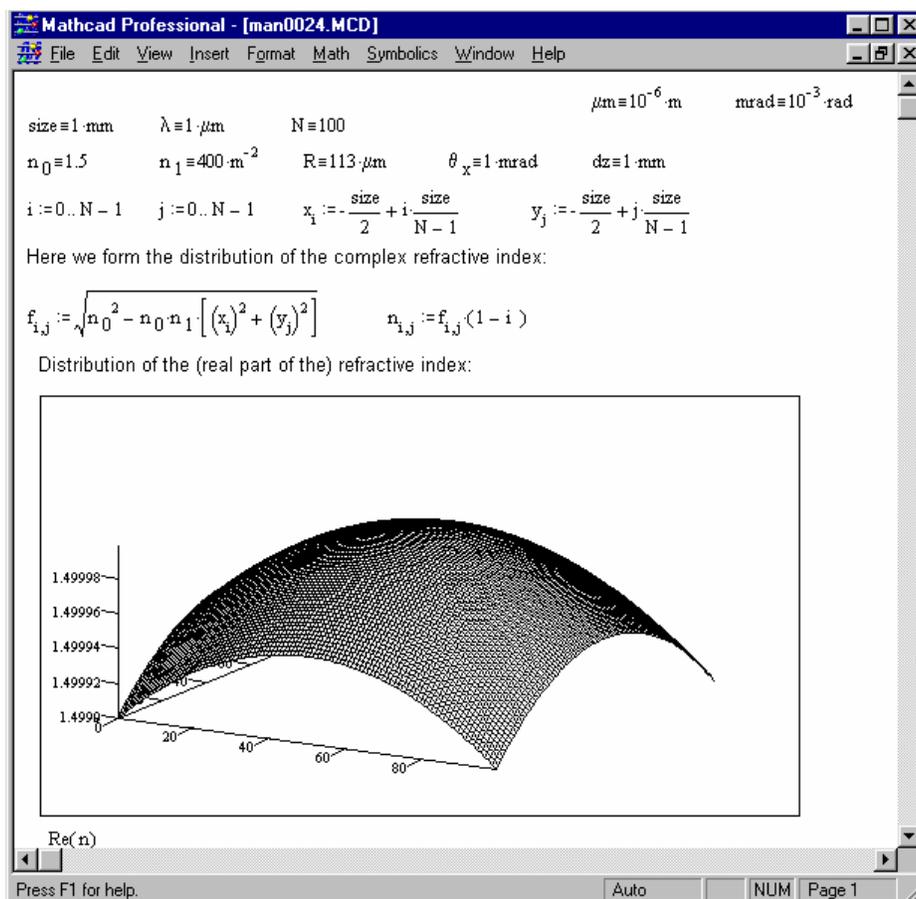


Figure 39 Construction of a quadratic refractive index profile.

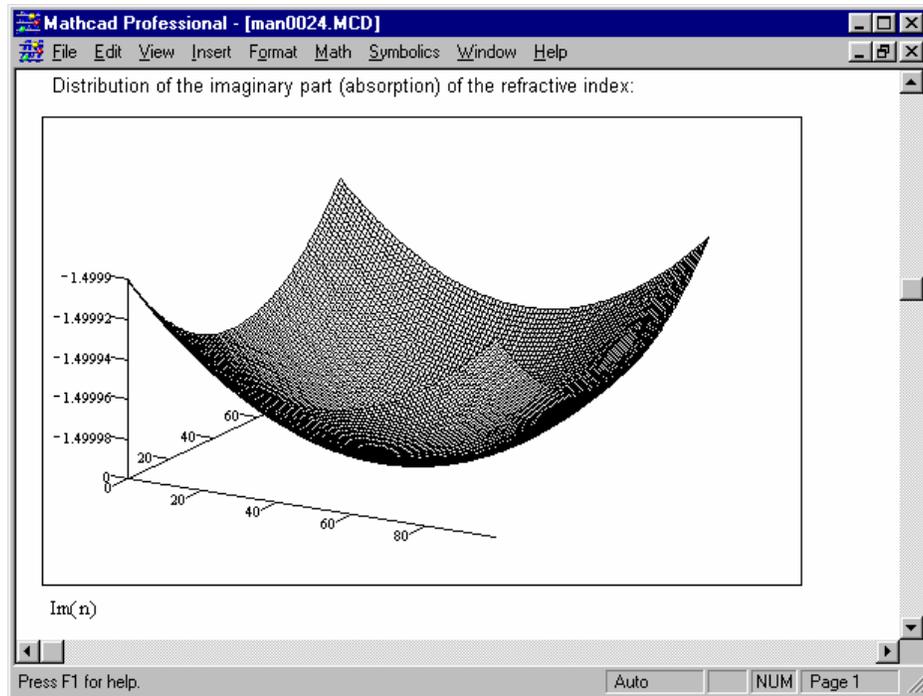


Figure 39 Construction of a quadratic refractive index profile. (cont.)

In Figure 39 we use the approximation for the profile of the refractive coefficient in the form:  $n^2 = n_0^2 - n_0 n_1 r^2$ . It is a well-known fact [9] that the half-width of the fundamental

Gaussian mode of a lens-like waveguide is defined as:  $w_0^2 = \frac{2}{K(n_0 n_1)^{\frac{1}{2}}}$ . For a waveguide

of 1x1mm,  $n_0=1.5$ ,  $n_1=400$  and  $\lambda=1\mu\text{m}$ , the Gaussian mode has a diameter of 226 $\mu\text{m}$ . In Figure 40 we demonstrate the propagation of a tilted Gaussian beam through the waveguide.

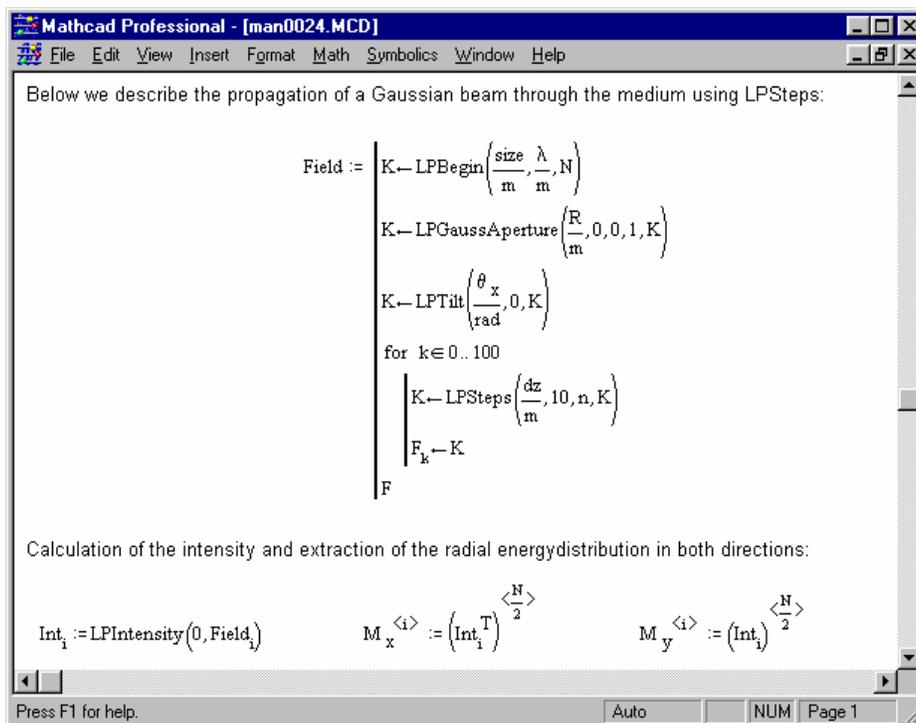


Figure 40 Propagation of a tilted Gaussian beam in a lens-like absorptive medium.

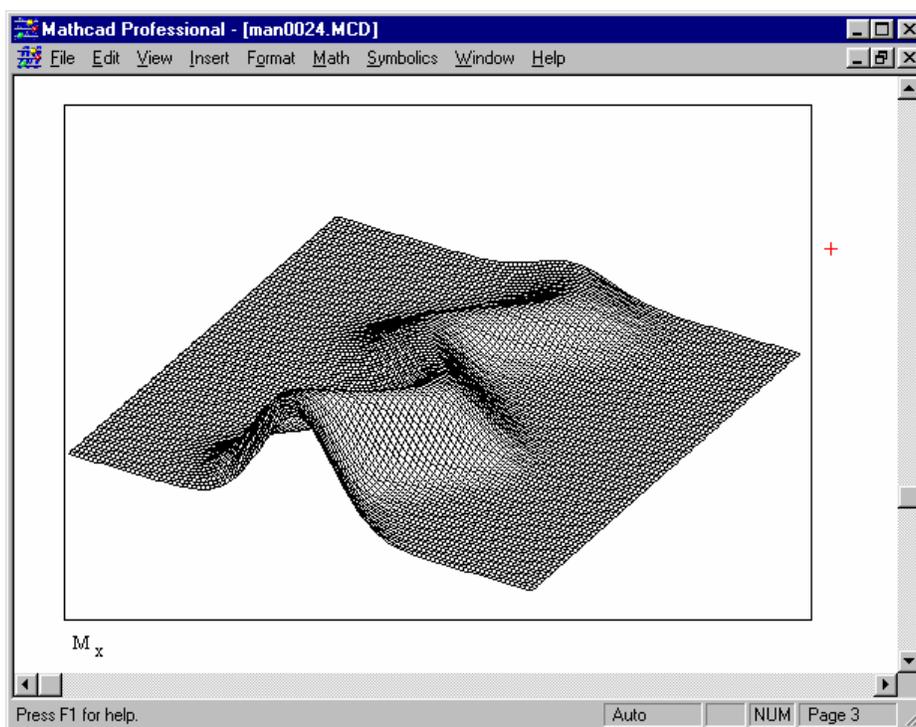


Figure 40 Propagation of a tilted Gaussian beam in a lens-like absorptive medium. (cont.) Radial energy distribution in the x-direction.

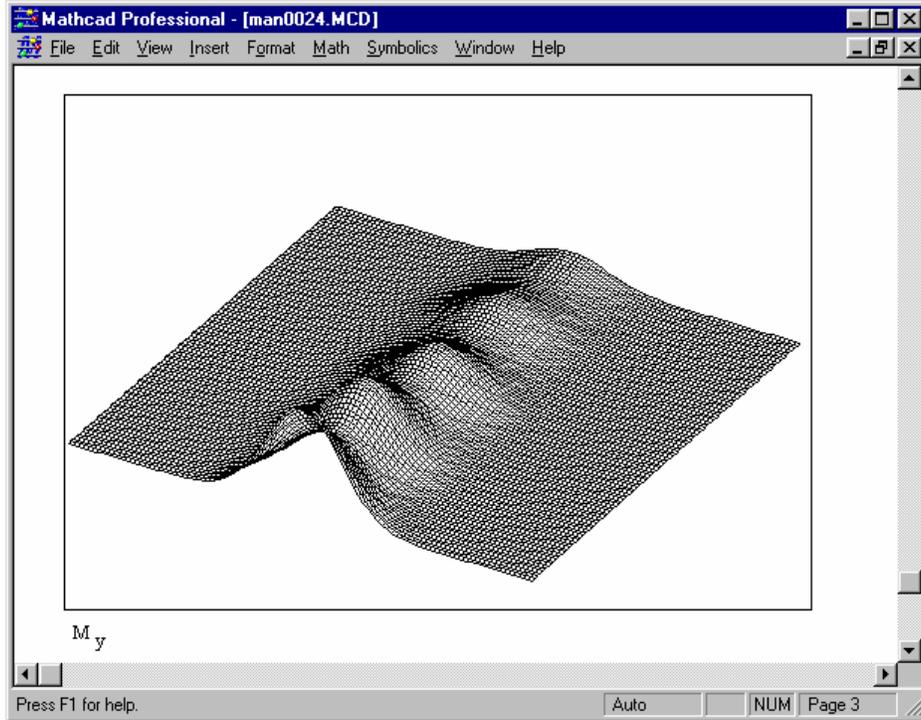


Figure 40 Propagation of a tilted Gaussian beam in a lens-like absorptive medium. (cont.) Radial energy distribution in the y-direction.

It can be shown that a reversed propagation of the beam using LPSteps yields the same input beam. The result of the propagation of a shifted Gaussian beam, incident parallel to the axis is shown in Figure 41. The result of the propagation of two shifted Gaussian beams, both incident parallel to the axis of the waveguide is shown in Figure 42. LPSteps is a "rich" filter. Whole classes of waveguide devices can be simulated. The simple examples in this manual illustrate only the very basic capabilities.

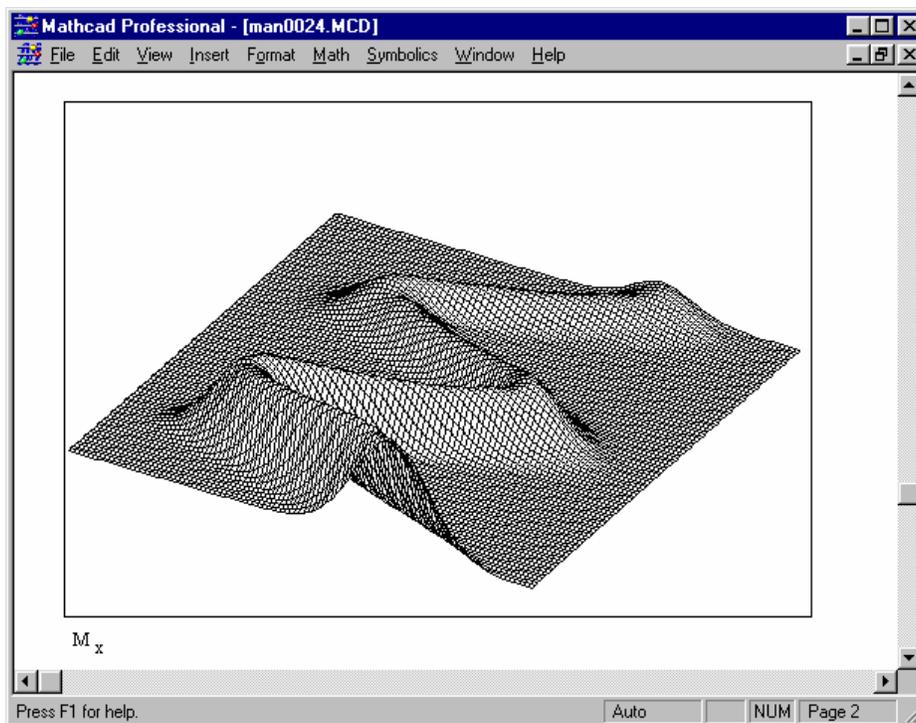


Figure 41 Propagation of a shifted Gaussian beam incident parallel to the waveguide axis.

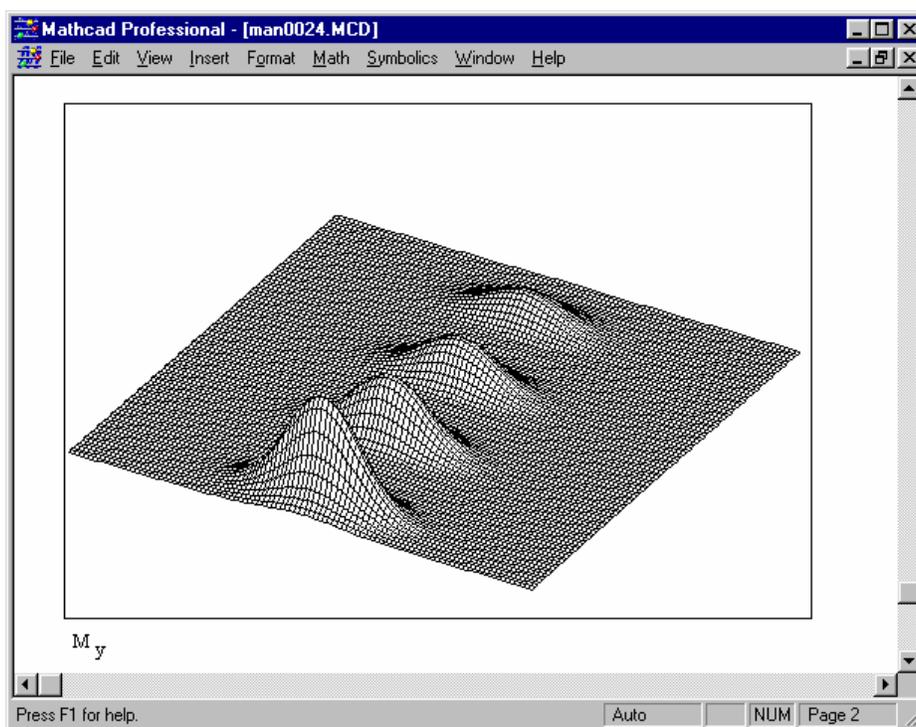


Figure 41 Propagation of a shifted Gaussian beam incident parallel to the waveguide axis. (cont.)  
energy distribution in the y-direction.

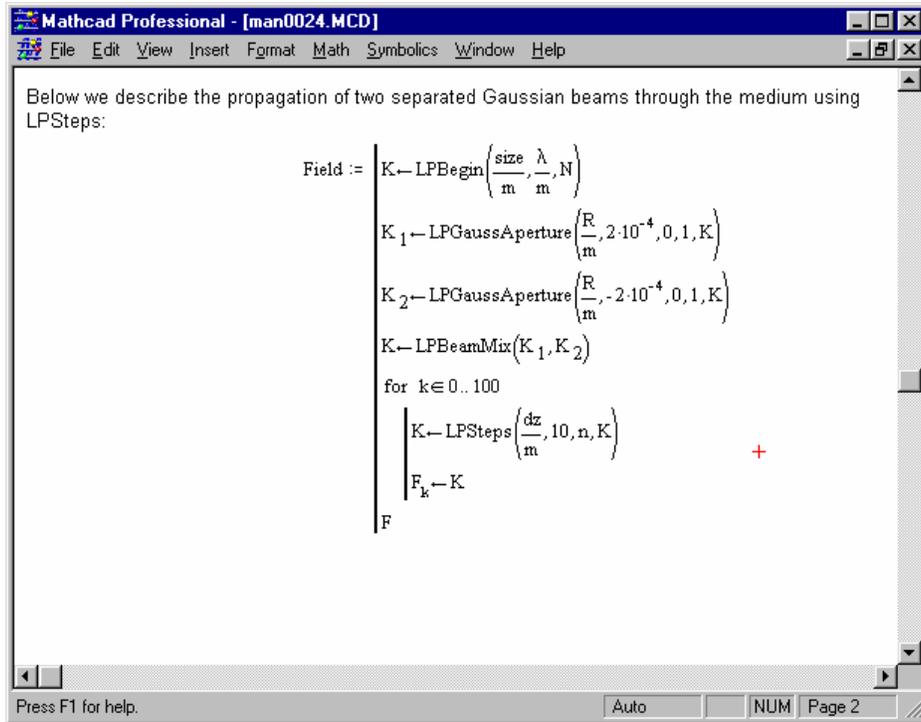


Figure 42 Propagation of two Gaussian beams incident parallel to the waveguide axis.

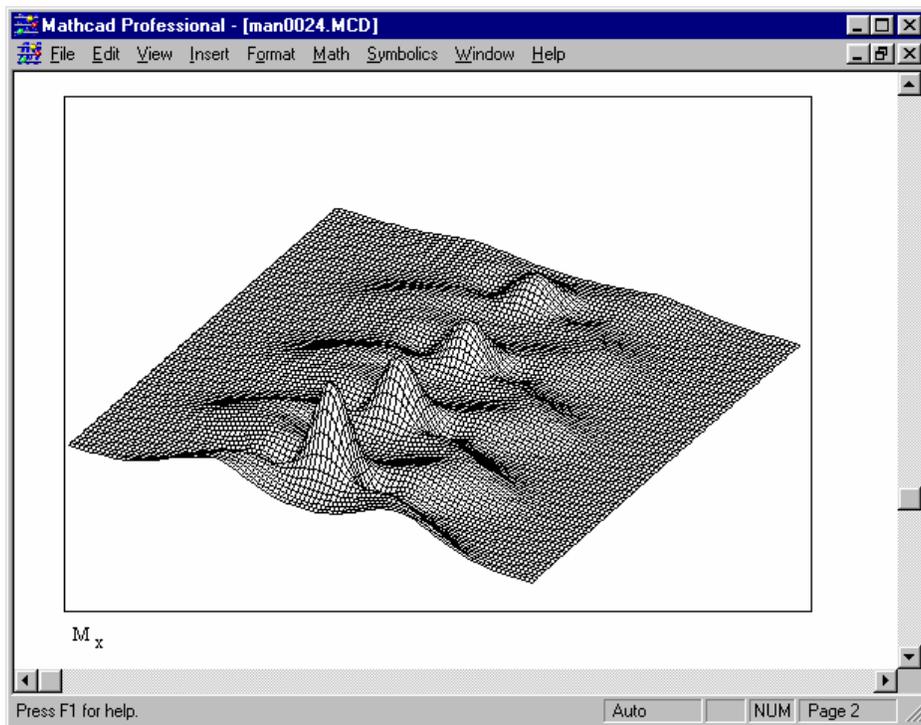


Figure 42 Propagation of two Gaussian beams incident parallel to the waveguide axis. (cont.)  
Radial energy distribution in the x-direction.

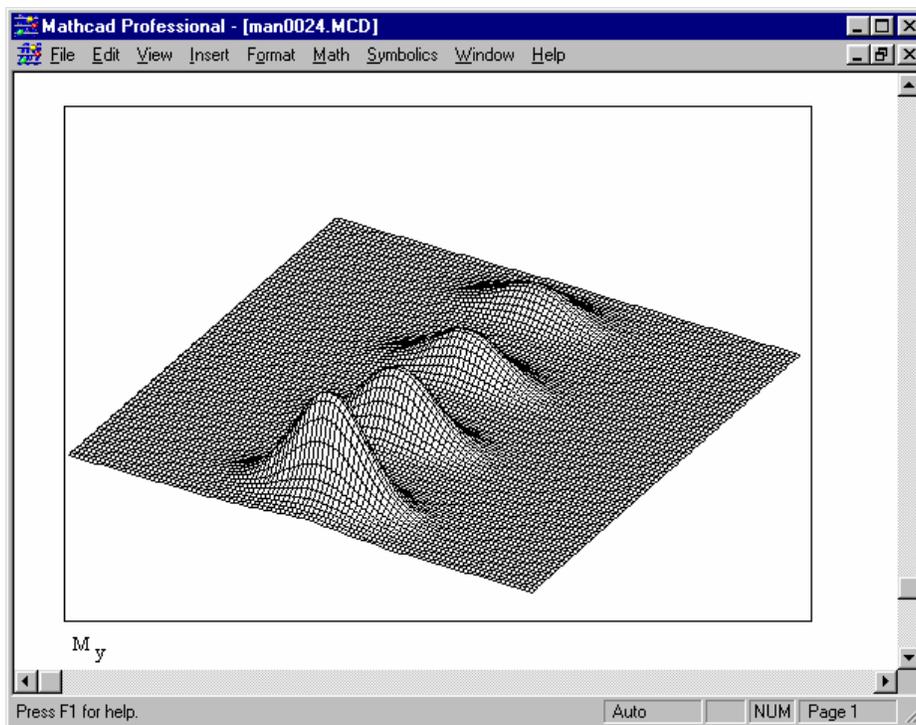


Figure 42 Propagation of two Gaussian beams incident parallel to the waveguide axis. (cont.)  
Radial energy distribution in the y-direction.



## 6.8 Inverse problem: reconstructing the phase from measured intensities.

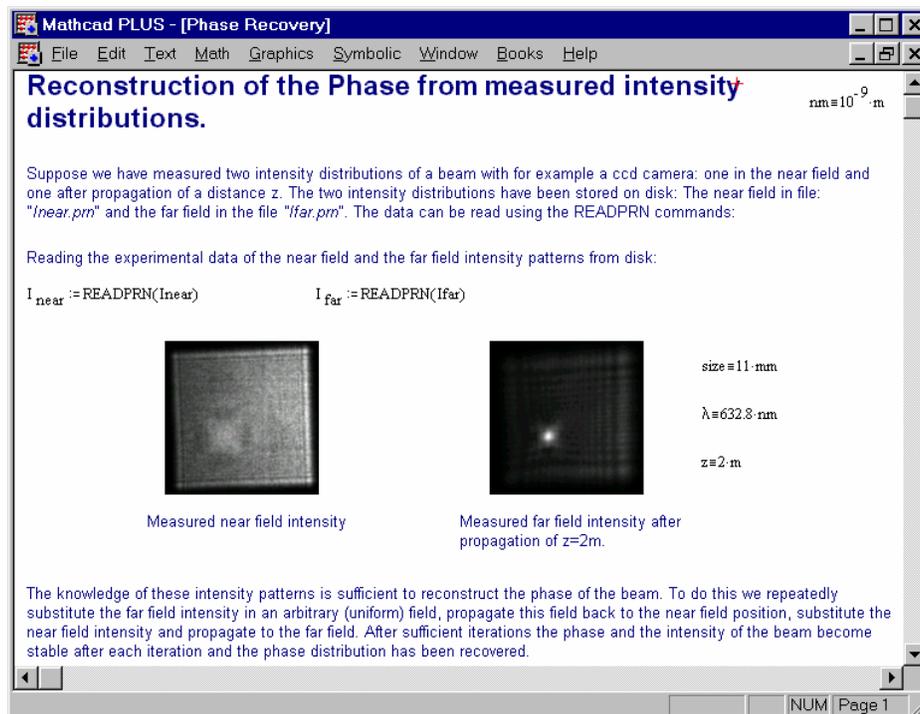


Figure 43 Reconstruction of the phase of a beam from two measured intensity patterns.

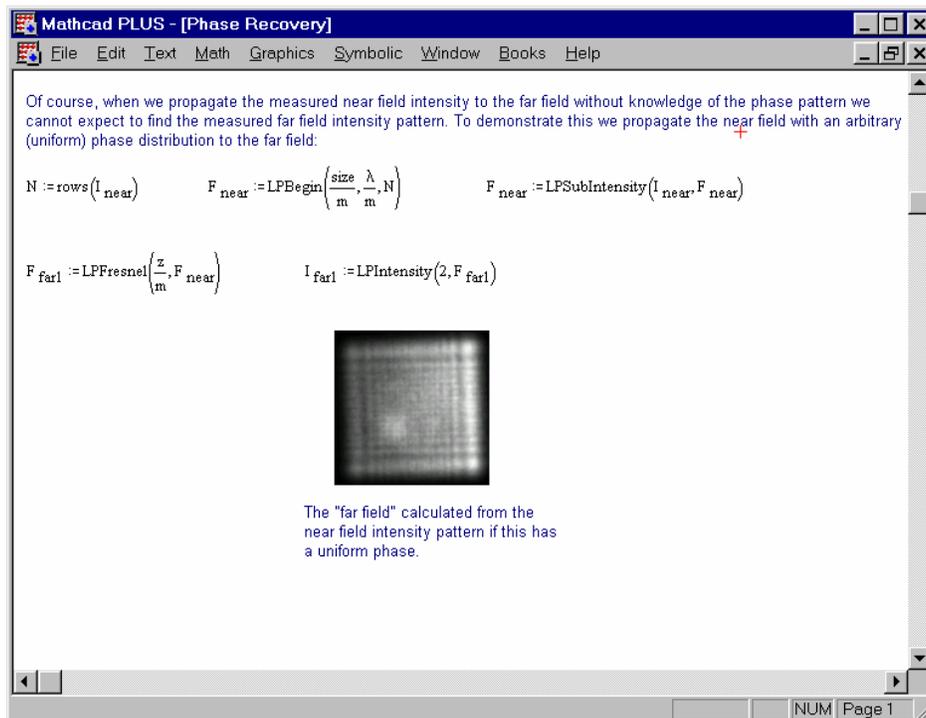


Figure 43 Reconstruction of the phase of a beam from two measured intensity patterns. (cont.)

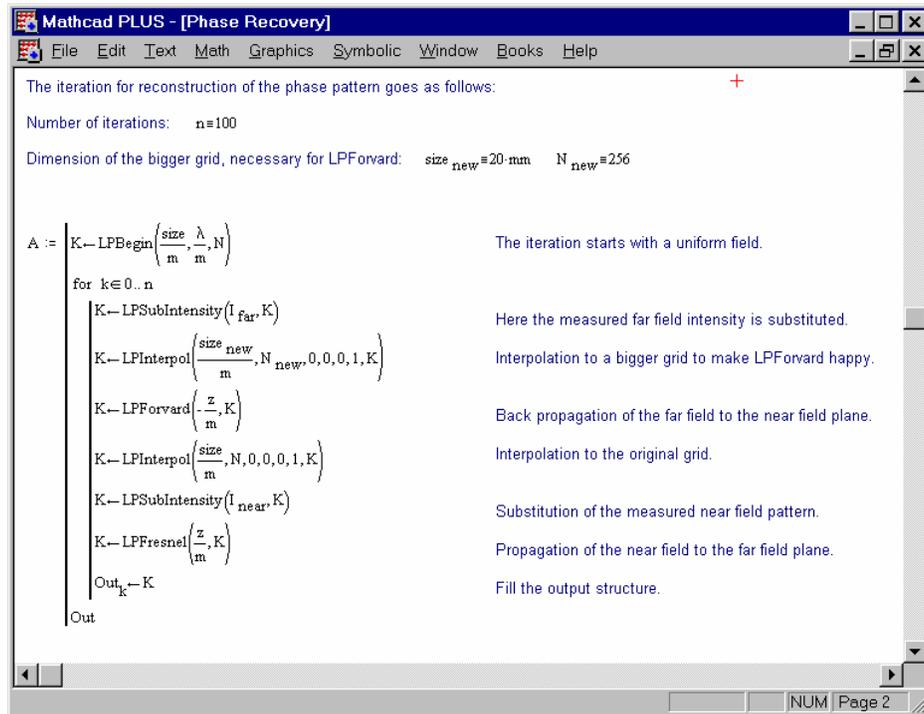


Figure 43 Reconstruction of the phase of a beam from two measured intensity patterns. (cont.)

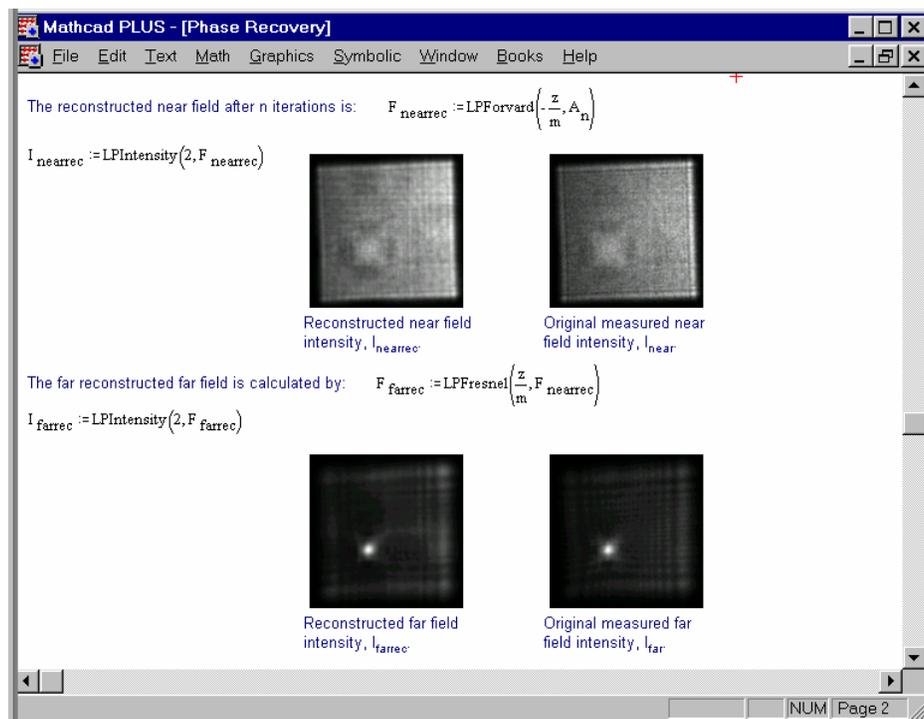


Figure 43 Reconstruction of the phase of a beam from two measured intensity patterns. (cont.)

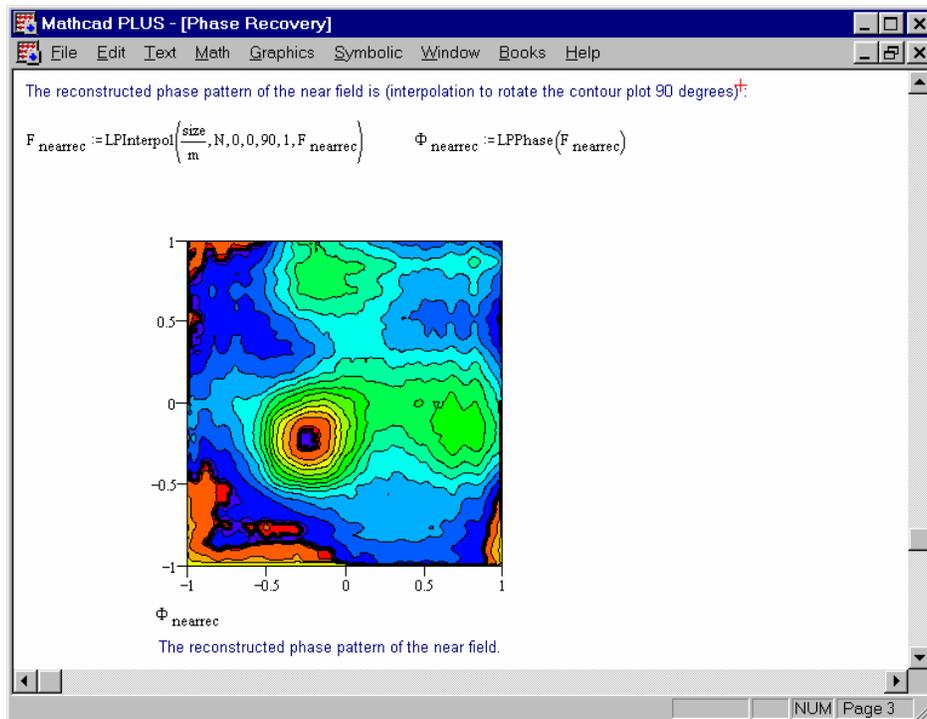


Figure 43 Reconstruction of the phase of a beam from two measured intensity patterns. (cont.)



## 6.9 Optical information processing.

LightPipes for Mathcad can be used to model classical setups of Fourier-optics information processing. For example Figure 44 models an optical computer performing Fourier-plane image filtering.

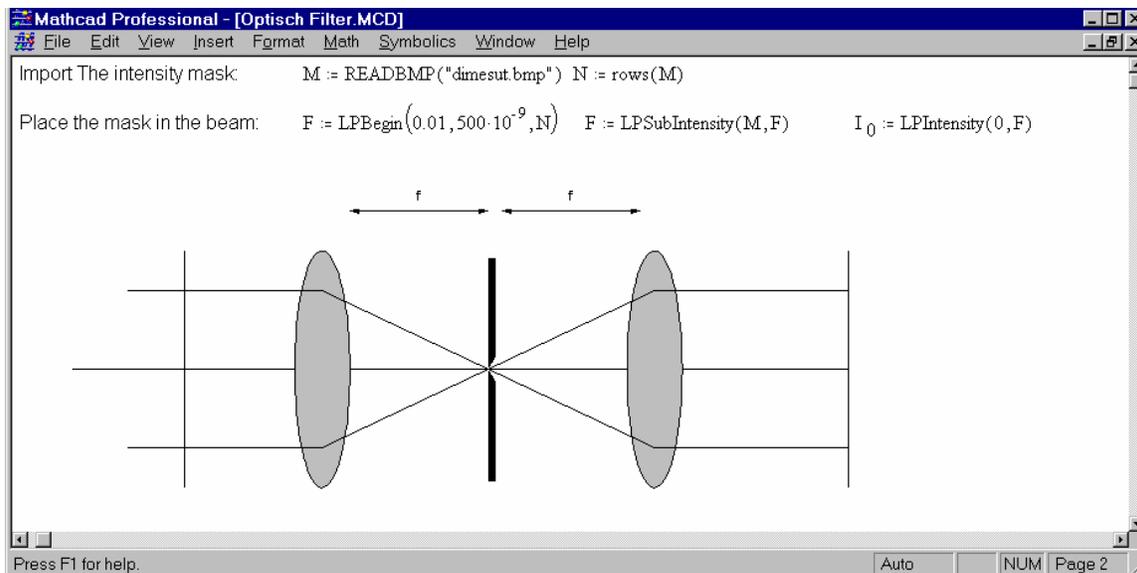


Figure 44 Optical information processing (Fourier optics).

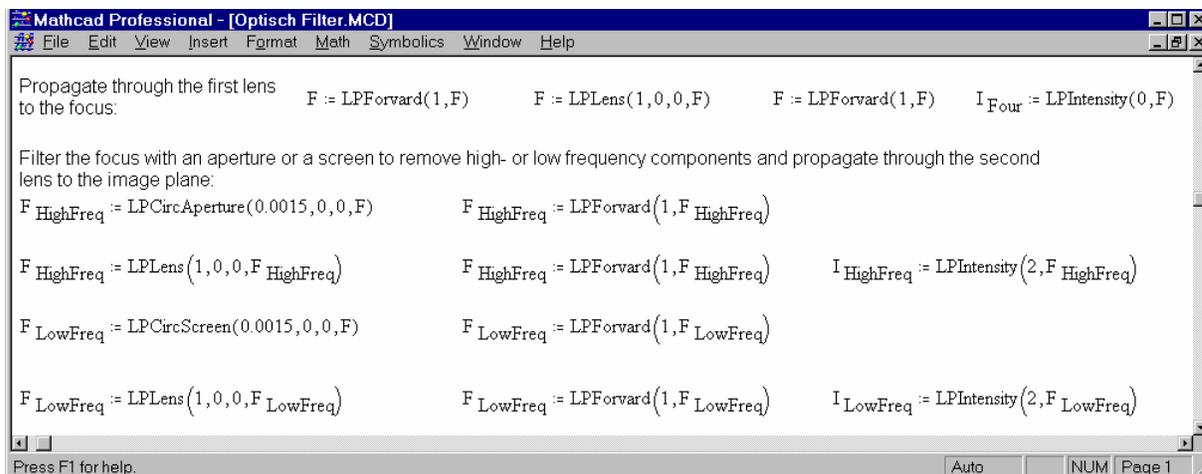


Figure 44 Optical information processing (Fourier optics). (cont.)

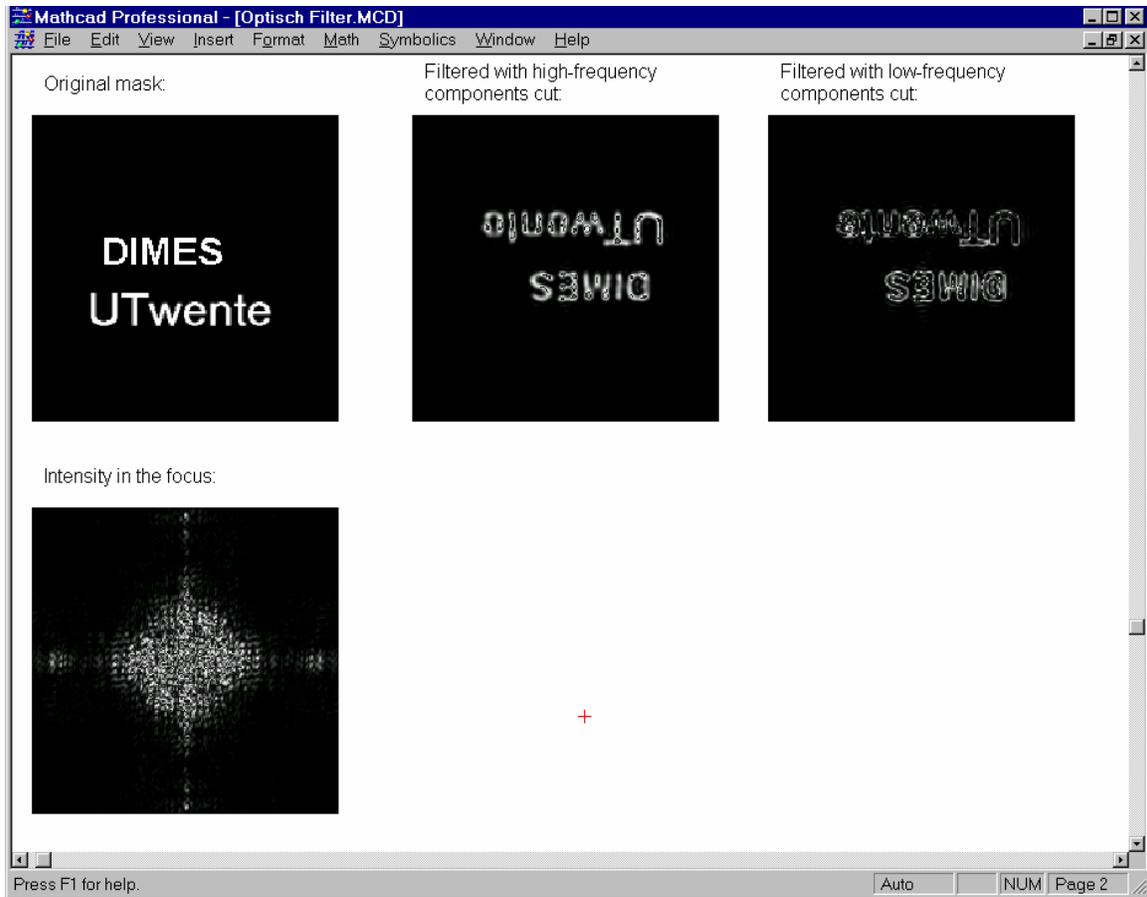


Figure 44 Optical information processing (Fourier optics). (cont.)

The initial distribution and two filtered images are shown in Figure 44. Also shown is the intensity distribution in the focal plane of the two lenses.

## 6.10 Generation and reconstruction of interferograms.

In the simplest case the interferogram (hologram) is generated by mixing of an aberrated beam with a plane wave:

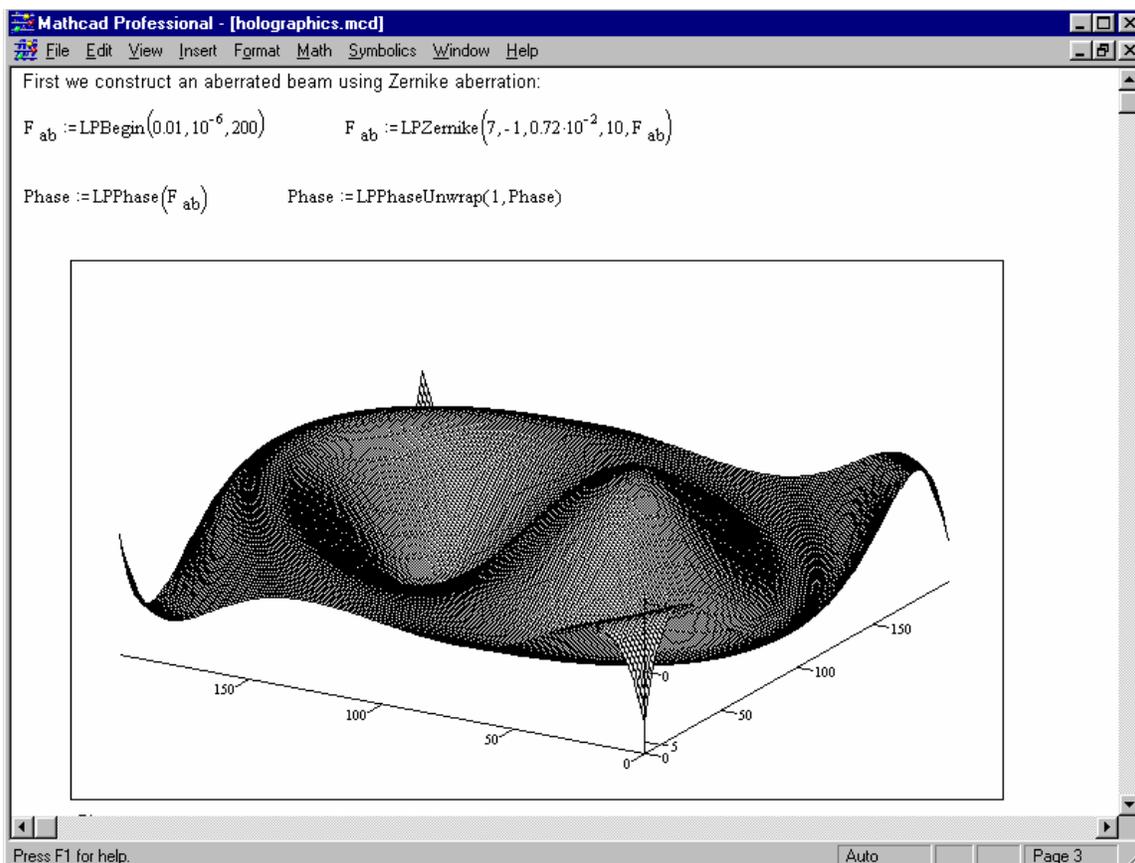


Figure 45 The phase distribution of the aberrated beam generated using the LPZernike command.

In Figure 45 the phase distribution of an aberrated beam is shown. The beam has been made using the LPZernike command. Mixing this beam with an ideal plane wave yields an interferogram as shown in Figure 46.

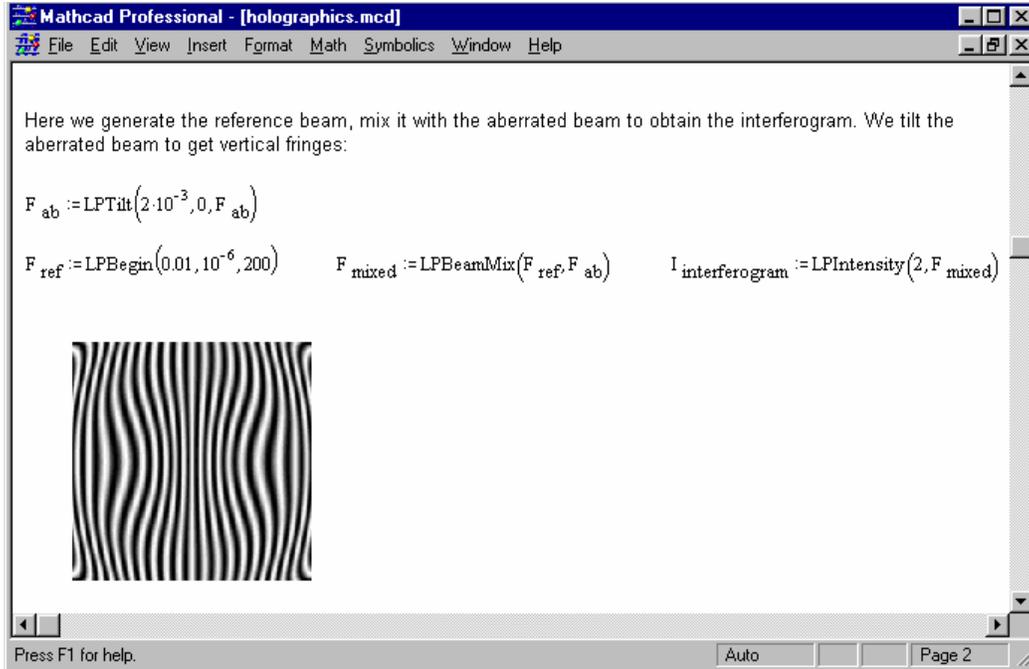


Figure 46 Intensity distribution of the interference pattern obtained by mixing the aberrated beam with a plane wave.

To reconstruct the phase from the obtained interferogram we shall build a model of the algorithm of phase reconstruction, using shift and filtering in the Fourier domain. Let the fringe pattern has the form:

$$g(x, y) = c(x, y) \exp(2\pi i f_0 x) + c^*(x, y) \exp(-2\pi i f_0 x) \quad (6.1)$$

where:

$$c(x, y) = \frac{1}{2} \exp(i\psi(x, y)) \quad (6.2)$$

where the phase  $\psi(x, y)$  contains information about the mirror shape and the term  $f_0 x$  describes the wavefront tilt. Fourier transform of expression (6.1) gives:

$$G(f, y) = A(f, y) + C(f + f_0, y) + C^*(f - f_0, y) \quad (6.3)$$

where the capitals  $A$  and  $C$  denote the Fourier spectra and  $f$  is the spatial frequency. We may take one of the two side-lobe spectra  $C(f + f_0, y)$  or  $C^*(f - f_0, y)$  and translate it to the origin with zero spatial frequency. Now we can take the inverse Fourier transform of the translated spectrum to obtain  $c(x, y)$  defined in the expressions (6.1) and (6.2).

Calculating the complex logarithm of the expression (6.2) we obtain the phase  $\psi(x, y)$ :

$$i\psi(x, y) = \log[c(x, y)] \quad (6.4)$$

The phase  $\psi(x, y)$  is obtained indeterminate to a factor  $2\pi$  with its principal value localized in the range of  $-\pi \dots \pi$ . To obtain the continuous phase map a special unwrapping algorithm, removing discontinuities with an amplitude close to  $2\pi$  is applied to the reconstructed phase map. The algorithm implementation includes the following steps shown in Figure 47:

After the substitution of the interferogram (hologram) into the field the spectrum of the

initial intensity distribution is calculated with LPPipFFT. It consists of a central lobe and two side-lobes containing information about the phase.

1. Shift of the obtained spectrum to put one of the side lobes into the origin with the LPInterpol command.
2. Filtering, leaving only the side lobe containing the phase information. This operation is performed as a product with a rectangular spectral window LPRectAperture.
3. Reversed Fourier transform using LPPipFFT.
4. Calculation of the phase with LPPhase and unwrapping it with LPPhaseUnwrap.

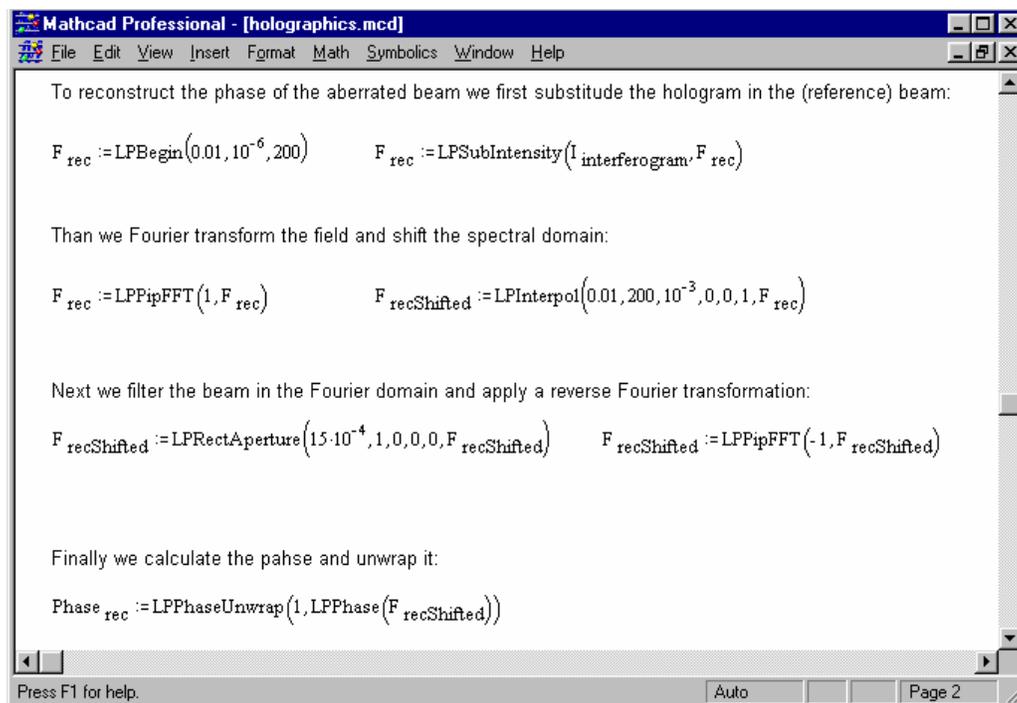


Figure 47 Algorithm to reconstruct the phase from the interferogram.

The result is shown in Figure 48.

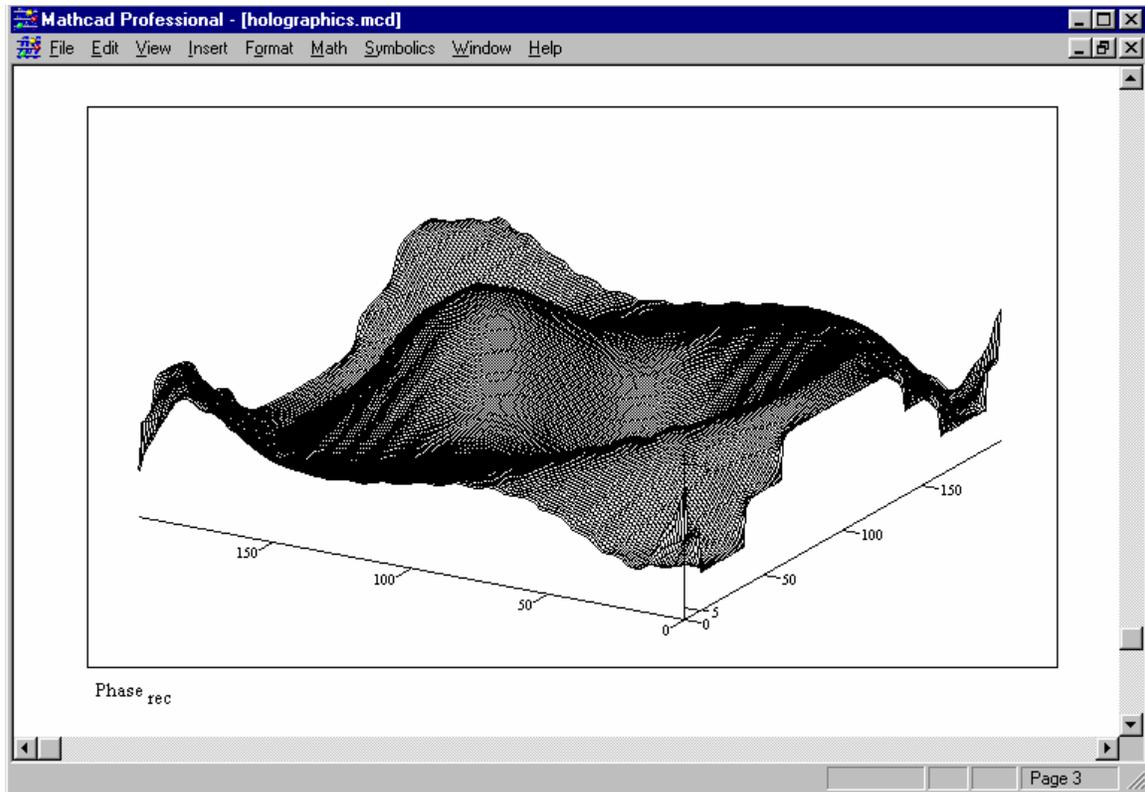


Figure 48 Reconstructed phase.

The initial interferogram is even (symmetric) with respect to the vertical axis, but the phase is odd, thus the phase is reconstructed with inverted sign (this depends on the direction of the shift in the frequency domain). The amount of shift in the frequency domain can be calculated using relations, given in section 5.2.12 containing the description of the LPPipFFT command, it also can be determined experimentally by visualizing the spectrum and shifting it to position one of its side lobes at the origin. This can be done with the LPInterpol command. Visualization of the spectrum side lobes requires a high gamma of the (bitmap) picture (of the order of 10).

## 7. Command reference

The following commands are available in LightPipes for Mathcad. Please refer to the lphelp.mcd document in the ....winmcd\lphelp directory for more details.

### **LP Axicon(phi,n1,x shift,y shift,Field)**

propagates the field through an axicon.

phi: top angle in radians;

n1: refractive index (must be real);

x\_shift, y\_shift: shift from the centre.

### **LP BeamMix(Field1,Field2)**

Addition of Field1 and Field2. If the wavelengths are different, a new, average wavelength, will be calculated.

### **LP Begin(grid size,lambda,grid dimension)**

initialises a uniform field with unity amplitude and zero phase.

grid size: the size of the grid in units you choose,

lambda: the wavelength of the field with the same units as the grid size,

grid dimension: the dimension of the grid. This must be an even number larger than 8.

### **LP CircAperture(R,x shift,y shift,Field)**

circular aperture.

R=radius,

x\_shift, y\_shift = shift from centre.

### **LP CircScreen(R,x shift,y shift,Field)**

circular screen.

R=radius,

x\_shift, y\_shift = shift from centre.

### **LP Convert(Field)**

converts the field from a spherical grid to a normal grid.

### **LP Forward(z,Field)**

propagates the field a distance z using FFT.

### **LP Forward(z,new size,new number,Field)**

propagates the field a distance z, using direct integration of the Kirchoff-Fresnel integral.

### **LP Fresnel(z,Field)**

propagates the field a distance z. A convolution method has been used to calculate the diffraction integral

### **LP Gain(Isat,gain,length,Field)**

Laser saturable gain sheet.

Isat=saturation intensity,  
 gain= small signal gain,  
 length=length of gain medium

**LPGaussAperture(R,x shift,y shift,T,Field)**

Gauss aperture.

R=radius,

x\_shift, y\_shift = shift from centre;

T= centre transmission

**LPGaussHermite(n,m,A,w0,Field)**

substitutes a TEM<sub>m,n</sub> mode with waist w<sub>0</sub> and amplitude A into the field.

**LPGaussScreen(R,x shift,y shift,T,Field)**

Gauss screen.

R=radius,

x\_shift, y\_shift = shift from centre;

T= centre transmission

**LPIntAttenuator(Att,Field)**

Intensity Attenuator. Attenuates the field-intensity by a factor Att

**LPIntensity(flag,Field)**

calculates the intensity of the field.

flag=0: no scaling;

flag=1: normalisation of the intensity;

flag=2: bitmap with gray scales

**LPInterpol(new size,new n,xs,ys,phi,magn,Field)**

interpolates the field to a new grid.

new\_size: new grid size;

new\_n: new grid dimension;

xs, ys : the shifts of the field in the new grid;

phi: angle of rotation (after the shifts);

magn: magnification (last applied)

**LPLens(f,x shift,y shift,Field)**

propagates the field through a lens.

f: focal length;

x\_shift, y\_shift: transverse shifts of the lens optical axis

**LPLensForward(f,z,Field)**

propagates the field a distance z using a variable coordinate system.

f: focal length of the input lens;

z: distance to propagate

**LPFresnel(f,z,Field)**

propagates the field a distance z using a variable coordinate system.

f: focal length of the input lens;

z: distance to propagate

**LPMultIntensity(Intensity,Field)**

multiplies the field with an intensity profile stored in the array: Intensity

**LPMultPhase(Phase,Field)**

multiplies the field with a phase profile stored in the array: Phase

**LPNormal(Field)**

normalizes the field. The normalization coefficient is stored in Field(n\_grid,6)

**LPPhase(Field)**

calculates the phase of the field

**LPPhaseUnwrap(Phase)**

unwraps phase (remove multiples of  $\pi$ )

**LPPipFFT(Direction,Field)**

Performs a Fourier transform to the Field.

Direction = 1: Forward transform;

Direction = -1: Inverse transform

**LPolarizer(Phi,Fx,Fy)**

Polarizer. Outputs a linear polarized field.

Phi=polarizer angle,

Fx, Fy=input fields of horizontal and vertical components.

**LPRandomIntensity(seed)**

Random intensity mask.

'seed' is an arbitrary number to initiate the random number generator

**LPRandomPhase(seed,max)**

Random phase mask.

'seed' is an arbitrary number to initiate the random number generator;

'max' is the maximum value of the phase.

**LPRectAperture(sx,sy,x\_shift,y\_shift,phi,Field)**

rectangular aperture.

sx, sy: dimensions of the aperture;

x\_shift, y\_shift: shift from centre;

phi: rotation angle.

**LPRectScreen(sx,sy,x\_shift,y\_shift,phi,Field)**

rectangular screen.

sx, sy: dimensions of the screen;

x\_shift, y\_shift: shift from centre;

phi: rotation angle.

**LPReflectMultiLayer(P,N0,Ns,N,d,Th,F)**

MultiLayer Reflector. Reflection of the field by a multi-layer mirror.

P=0: s-Polarization; P=1: p-Polarization;

N0=refractive index of incident medium (must be real);

Ns=refractive index of substrate;

N= array with (complex) index of the layers;

d= array with the geometrical thicknesses of the layers;

Th= angle of incidence at first layer;

F= the input field

**LPSteps(z,N\_steps,Refract[N,N],Field)**

Propagates 'Field' a distance z in 'N\_steps' steps in a medium with a complex refractive index stored in the NxN matrix 'Refract'. 'N' is the grid dimension.

**LPStrehl(Field)**

calculates the Strehl ratio. The Strehl ratio is stored in: Field(n\_grid,5).

**LPSubIntensity(intensity,Field)**

substitutes an intensity profile into the field. The profile must be stored in the array:

Intens

**LPSubPhase(phase,Field)**

substitutes a phase profile into the field. The phase profile must be stored in the array:

phase.

**LPTilt(tx,ty,Field)**

introduces tilt into the field distribution. tx, ty = tilt components in radians.

**LPTransmitMultiLayer(P,N0,Ns,N,d,Th,F)**

MultiLayer Transmission. Transmission of the field through a multi-layer coated substrate.

P=0: s-Polarization; P=1: p-Polarization;

N0=refractive index of incident medium (must be real);

Ns=refractive index of substrate;

N= array with (complex) index of the layers;

d= array with the geometrical thicknesses of the layers;

Th= angle of incidence at first layer;

F= the input field

**LPZernike(n,m,R,A,Field)**

Introduces arbitrary Zernike aberration into the field.

$n$  and  $m$  are the integer orders, (See Born and Wolf, 6th edition p.465, Pergamon 1993).  
 $R$  is the radius at which the phase amplitude reaches  $A$  (in radians)



## 8. References

---

- 1 J.W. Goodman, Introduction to Fourier Optics, 49-56, McGraw-Hill (1968).
- 2 W.H. Southwell, J. Opt. Soc. Am., **71**, 7-14 (1981).
- 3 A.E. Siegman, E.A. Sziklas, Mode calculations in unstable resonators with flowing saturable gain: 2. Fast Fourier transform method, Applied Optics 14, 1874-1889 (1975).
- 4 N.N. Elkin, A.P. Napartovich, in "Applied optics of Lasers", Moscow TsniAtomInform, 66 (1989) (in Russian).
- 5 R. Bacarat, in "The Computer in Optical Research", ed. B.R. Frieden, Springer Verlag, 72-75 (1980).
- 6 A.A. Samarskii, E.S. Nikolaev, Numerical Methods for Grid Equations, V.1, Direct methods, pp. 61-65, Birkhäuser Verlag 1989.
- 7 M. Born, E. Wolf, Principles of Optics, Pergamon, 464, 767-772 (1993)
- 8 D. Malacara, Optical shop testing, J. Wiley & Sons, (1992)
- 9 D. Marcuse, Light Transmission Optics, Van Nostrand Reinhold, 267-280, 1972.