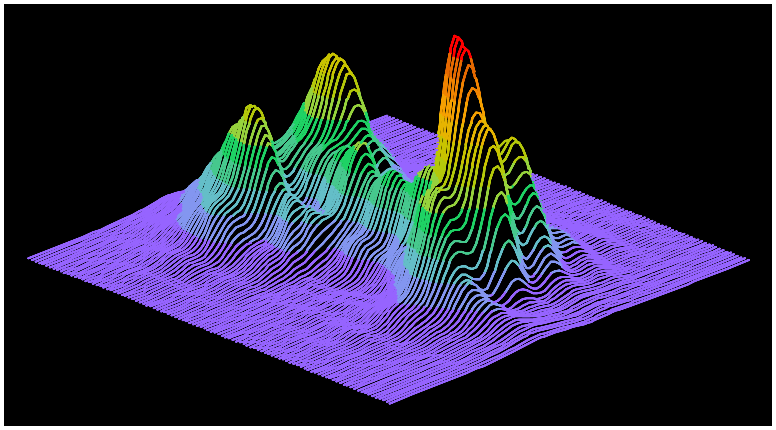
GLAD Course

Applied Optics Research



Applied Optics Research
1087 Lewis River Road #217
Woodland, WA 98674
tel: 1 360 225 9718, fax: 1 360 225 0347
glad@aor.com, http://www.aor.com

Table of Contents

Table of Files

1. Introduction to Optical Modeling with GLAD

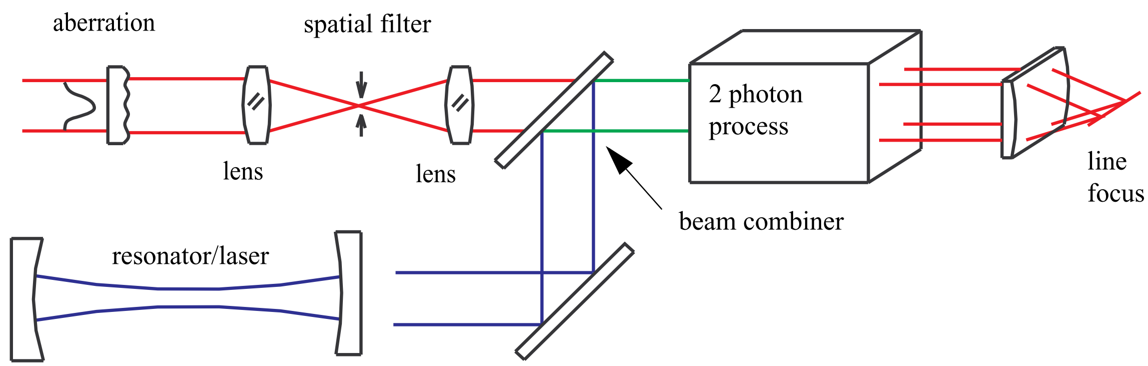
Principles and Practice of Optical Modeling with GLAD

Program: General Laser Analysis and Design (GLAD)

Author: Dr. George N. Lawrence, Applied Optics Research, glad@aor.com

Objective:

Gain intuitive understanding of diffraction, optical propagation, laser gain, waveguides, and selected nonlinear optics components.



What you will learn in this course

- Basic concepts of physical optics modeling
- GLAD command language, macro files, and math expressions
- Hands-on experience with modeling problems in GLAD
- Modeling selected topics
  - beam trains
  - lasers
  - waveguides
  - selected nonlinear optics
  - atmospheric and thermal effects (selected)

Course is not complete

- Not every GLAD feature covered

GLAD is the result of about 40 man-years of development

- Thousands of commands forms (counting commands and modifiers)
- More than 500 hundred examples

How to get help after the course

- AOR provides one year of warranty, technical support, and version upgrades
- Email is the best way to communicate
  - no extra typing or mistakes
  - AOR sees exact problem
  - AOR can email modified command files back to customer in ready-to-use form
- Download latest GLAD executable from demos/downloads: www.aor.com

Milestones in GLAD development

- All FFT's, no rays (1975)
- Physical optics code with user interface (1975)
- Memory management allows large array in small memory (1975)
- Atmospheric characterization by power spectrum in code (1980)
- Circular propagator, competitor with fast Hankle, worked on 48K, TRS 80 radio shack computer (1983)
- Zonal propagation with defined pixel control for multiple beams (1983)
- Path independent propagation control algorithm (1983)
- Integrating geometrical and physical optics (1985)
- Propagation with tilted and warped surfaces and thick elements (1986)
- Damped least squares optimization in physical optics (1988)
- Zonal model of adaptive optics (1988)
- Axicons included (1988)
- Thermal blooming (1988)
- Partial coherence (1989)
- Identification of problems with M^2 (1988)
- Incorporating rate equation gain (1991)
- Waveguides with free space propagation (1991)

- Rapid calculation of optical parametric oscillator (1991)
- Waveguide grating couplers, vector polarization effects (1991)
- Interferometery with moving elements (1990)
- Special resonator command provides stable numerical calculations (1990 to present)
- Rapid treatment of lens arrays (1990)
- High NA vector diffraction (1991)
- Transient Raman with complex medium polarization and growth from vacuum fluctuations (1991)
- Synthesis with phase retrieval incorporated (1991)
- Synthesis with simulated annealing incorporated (1992)
- Rate equation gain with Franz-Nodvik theory capable of transient and Q-switch modeling (1992)
- Finite element thermal blooming (1993)
- Reflecting wall waveguides by aliasing model (1994)
- Michelson interferometer with speckle plates and limited spatial and spectral coherence (1995)
- Nonlinear optics: Raman, doubling, limiting, sum-frequency generation, OPA, four-wave (1990 to present)
- Gain: four-level, three-level, semi-conductor (1988 to present)
- Frustrated total internal reflection (TIR) in polygon resonators (1994)

- Goos-Hanchen shift (1996)
- Thermally induced stress birefringence (1998)
- Excimer laser modeling (1999)
- Partial coherence in photolithography
- guide star from sodium layer in the upper atmosphere (2001)
- Laser diode array side pumping (2002)
- Command composer (2003)
- Variable monitor (2002)
- Partial coherence of a 3D object in broad band illumination
- Treatment of manifolds of upper and lower level in three and four level gain (2004)
- Sub-round-trip sampling (2006)
- Pulse compression in a grating rhomb (2007)
- Dynamic HTML output display using Javascript and SVG graphics (2007)
- Zigzag amplifier with exact 3D pixel matching (2008)
- External cavity mode competition (2009)
- Coherent treatment of gain for short pulse, longitudinal modes, etc. (2012)

Why physical optics?

- Beam propagation method (BPM) is the technique of choice for:
  - general diffraction: near-field, far-field and in between
  - vector diffraction
  - lasers
  - interferometry
  - diffractive elements
  - nonlinear optics
  - waveguides and optical fibers
  - most photonic applications
- Literature shows overwhelming preference for FFT-based BPM
- Systems may be modeled in modular fashion
- Geometrical optics may easily be incorporated within physical optics programs

Popular methods of determining laser system performance

Guessing

- simple
- no math or physics required
- ❑ often wrong for easy systems
- ❑ usually wrong for complex systems

Ray tracing

- very easy
- no math or physics required
- simply define configuration in a CAD form
- “a computer did is so it must be right”
- ❑ rays can not analyze laser systems or waveguides (to be discussed)
- ❑ ray distributions have little (if any) relationship to actual distributions
- ❑ only “flashlight” systems

Popular methods of determining laser system performance (cont'd)

Diffraction analysis in ray trace codes

- can see edge diffraction
- fully coherent light only
- can not model laser modes, speckle
- no gain or nonlinear effects
- no resonators
- no dynamic effects
- can only model systems that can be ray traced
- poor technical support from "ray benders"

M-squared

- better than guessing
- only theoretically ideal Hermite-Gaussian modes
- far-field second moment blows up with any aperture clipping, $M^2 \rightarrow \infty$
- excessive response to aberration
- not useful for system analysis

Evaluating optical modeling software

- Are there examples of all the things you need to do?
  (GLAD includes more than 500 examples)
- Documentation
  - derivations?
  - technical details?
  - limits of validity?
- Techical support
  - qualified to prove the models are correct?
  - willing to prove the models are correct?
  - call up and ask a technical question before buying

Why GLAD?

- GLAD contains the widest range of features of any code
  - full diffraction for all steps
  - rate equation laser gain
  - components and effects
  - atmospheric effects
  - selected nonlinear optics
  - programmable command language
- GLAD is the most widely used physical optics program

Structure of the course

- Lectures
- Hands-on practice

Some types of optical codes

- Lens design codes: Zemax, Code-V
  - Propagation: Ray tracing, Snell's Law, good 3D pictures
    - Some near-field diffraction with FFT's, no lasers or resonators
  - Applications: Complex geometrical optics systems: camera lenses, etc.
  - Form: Spread sheet, some programability

- LASCAD, highly specialized: diode pumped solid state laser (DPSSL)
  - Propagation: Hermite-Gaussian, stable cavity only. Low sampling.
  - Applications: laser welding for German automobile industry
  - Form: Specialized GUI interface, no programability

- GLAD, full diffraction for all laser and physical optics calculations
  - Propagation: full diffraction with FFT in all steps, full 3D propogation
    - some vector diffraction, waveguides, comlex resonators and beam trains
  - Applications: Sophisticated, high end systems, defense, laser fusion, excimers, complex lasers
  - Form: Script files, highly programable, math expressions

- Chinese market?

Table. 1.1. Comparison of GLAD and LASCAD

Feature	LASCAD	GLAD
Stable cavity resonators	■	■
Unstable cavity resonators		■
Stable-Unstable resonaors		■
Coupled resonators, external cavity diode lasers *A linear wavelength tuning configuration in mode-hop-free external cavity diode laser with all-dielectric thin film Fabry-Perot filter*", Xiao Xiao; CAS Shenzhen Inst of Advanced Technology and G. Lawrence, AOR		■
Finite element thermal dN/dT and dL/dT	■	■
Stress birefringence		■
4-level, rate equation gain	■	■
3-level, rate equation gain		■
semiconductor rate equation gain		■
Franz-Nodvik gain technique		■
Spontaneous emission		■
Speckle effects and distributions		■
Beam trains		■
Spatial filters		■
Complex lens groups		■
Ray trace analysis of lens groups		■
Complex aberrations		■
Complex apertures and obscurations		■

Table. 1.1. Comparison of GLAD and LASCAD (Continued)

Feature	LASCAD	GLAD
3D geometry and positioning		■
Full diffraction treatment with FFT's		■
Automatic control of diffraction algorithms		■
High Fresnel numbers		■
Extreme sampling sizes 16384 x 16384 and above		■
Some vector diffraction		■
Axicons		■
Atmospheric turbulence aberrations		■
Atmospheric thermal blooming		■
Waveguides: 3D and slab		■
Adaptive optics		■
Optimization: damped least squares		■
Optimization: phase retrieval (Gerchberg-Saxton)		■
Optimization: simulated annealing		■
Coherent injection		■
Interferometry		■
Nonlinear optics: Raman scattering		■
Nonlinear optics: optical parametric amplification		■
Nonlinear optics: optical limiting		■
Nonlinear optics: sum frequency generation		■
Nonlinear optics: four-wave mixing		■
Nonlinear optics: frequency doubling		■
Lens arrays		■
Laser diode arrays		■
Dynamic mode competition		■

Table. 1.1. Comparison of GLAD and LASCAD (Continued)

Feature	LASCAD	GLAD
Partial coherence		■
Excimer lasers		■
Phase gratings, binary optics, volume holograms		■
Programability and math expressions		■
Expandability		■
Examples: illustration and validation	?	500+
Documentation	61 pages	1897 pages
Technical support	?	■

Lawrence background

- Night vision devices (Dragon, 1969-1970)
- Optical system design of laser guided smart bombs: optical testing, lasers systems, laser diode banks, optics for TV systems, laboratory and field system testing (TOW, early 1970's)
- Optical design, aberration theory for Bob Shannon, Jim Wyant OSC, UoA
- Interferometry of large multiple mirror telescopes
- End-to-end, fully diffraction based system model for laser fusion program (for Los Alamos National Laboratory)
- Star wars systems, beam expanders, chemical lasers (LODE, ALPHA, for DARPA)
- Zonal model of adaptive optics (Air Force Weapons Lab)
- Raman system modeling
- Airborne laser laboratory testing (Air Force Weapons Lab)
- path-independent, general propagation method
- Integrating geometrical and physical optics
- Subaperture testing algorithm: applied to flats (DARPA), full spheres (LANL), annular zones, atmospheric layers

Lawrence background (cont'd)

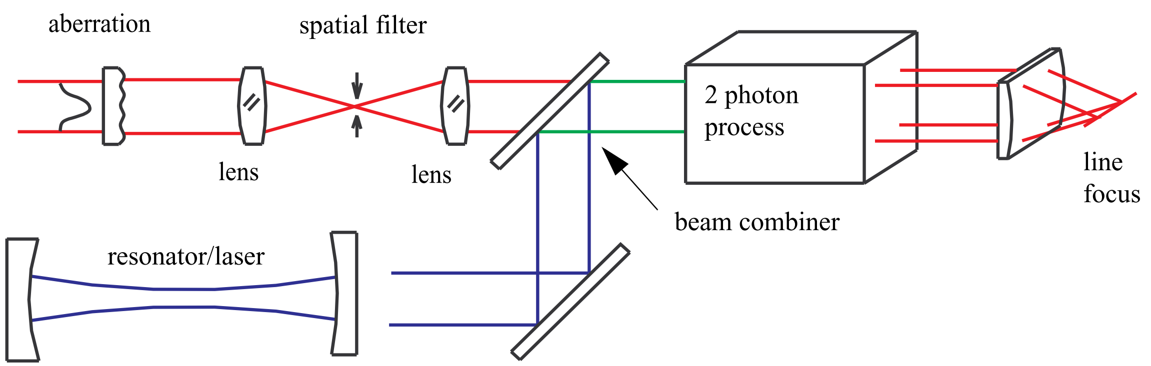
- Laser isotope separation (LANL, LLNL)
- Free electron laser modeling (LANL)
- Astronomy for planet detection (NASA)
- Optical data storage: focusing grating couplers, polarization (UoA)
- Optical design of binocular optics (Army)
- Reverse optimization of off-axis, three mirror system (DARPA)
- Teaching optical design, optical modeling, associate professor at Optical Sciences Center, UoA
- Hubble Space Telescope Review Panel (NASA)
- Rate equation gain modeling
- More laser fusion, lens arrays (U of Rochester) with Dr. Ying Lin
- Phase retrieval (Gerchberg-Saxton) design synthesis (U. of Rochester, Lin)
- Simulated annealing design synthesis (U. of Rochester, Lin)
- Transient Raman modeling, QED (U. of Rochester, Lin)
- GLAD: Franz-Nodvik rate equation method, stress birefringence, reflecting wall waveguides, non-fourier methods, zig-zag resonator, optical parameteric amplifier, excimer modeling, axicon systems,"resonator" command

General Laser Analysis and Design (GLAD) — Applied Optics Research

- Started 1975 for laser fusion program
- 1970's through 1980's: high energy programs for laser fusion and Star Wars programs
- Program-of-choice for national laboratories and major corporate research programs
- 1985 began commercial sales for PC's and workstations

GOALS:

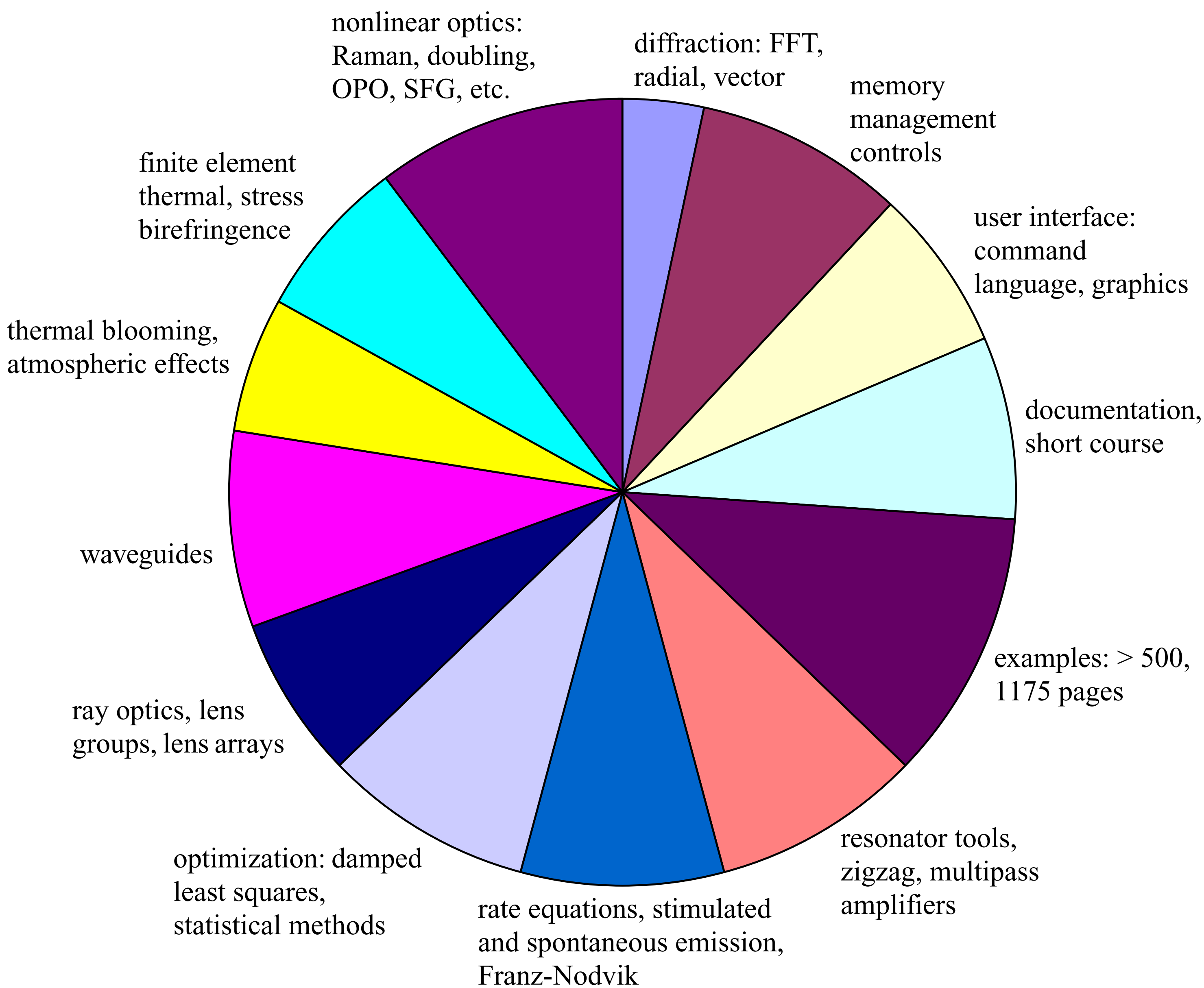
- Model every aspect of laser and physical optics systems
- Accurate, detailed analysis to match experiments precisely



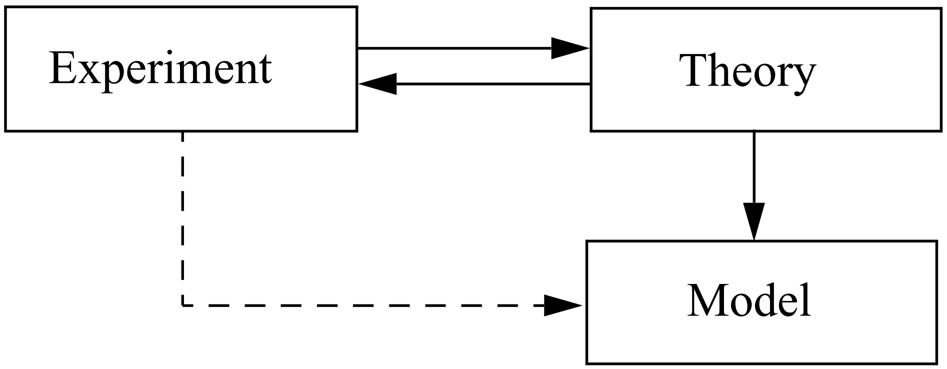
Structure of the program

- Computer calculations
  Physical optics and laser calculations frequently require computer calculations because of the difficulty and complexity of the calculations
- Complex amplitude
  The optical beam may be characterized by a two-dimensional array of complex amplitude points -- similar to intensity and phase
- Evolution of the beam
  GLAD determines the state of any laser system or beam train by calculating the evolution of one or more complex amplitude arrays.
- Text-based command language
  The enormous range of features is best supported by a text-based command language formulation. Facilitates in-line equations, branching, looping. Greatly facilitates technical support.

Proportions of development effort in GLAD



Anchoring the model to theory

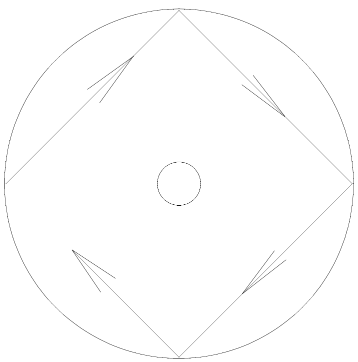


Modeling

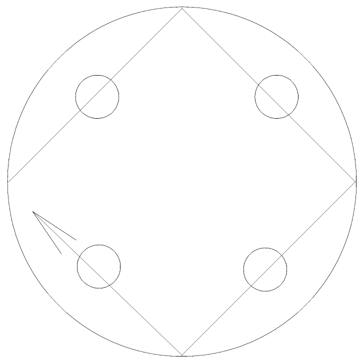
- Built on theory. (No theory. No model.)
- Anchored to theory.
- Model cannot be anchored to experiment.
- However, model can be invalidated by experiment.

Fiber optic laser with multiple cores

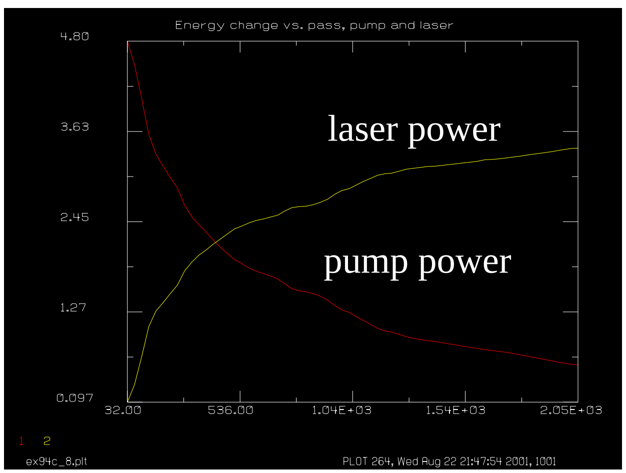
- Waveguide propagation in an optical fiber
- Four cores are doped to provide gain (rate equation theory)
- Mode structure varies with length

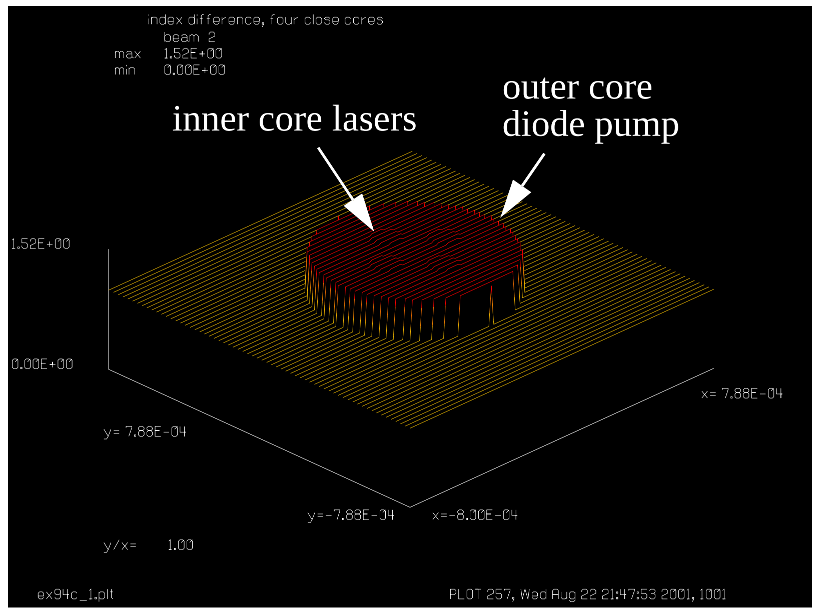


single core



multiple core

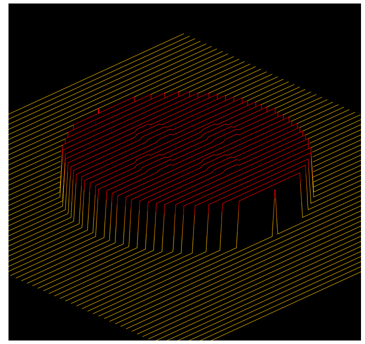


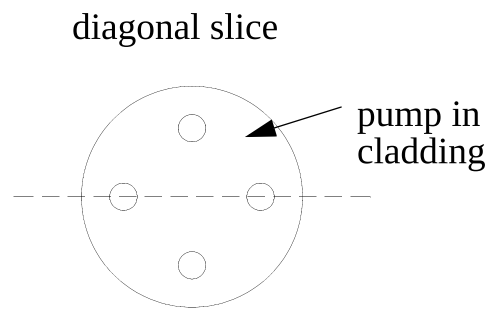


Outer core traps pump light.
Four inner cores lase.

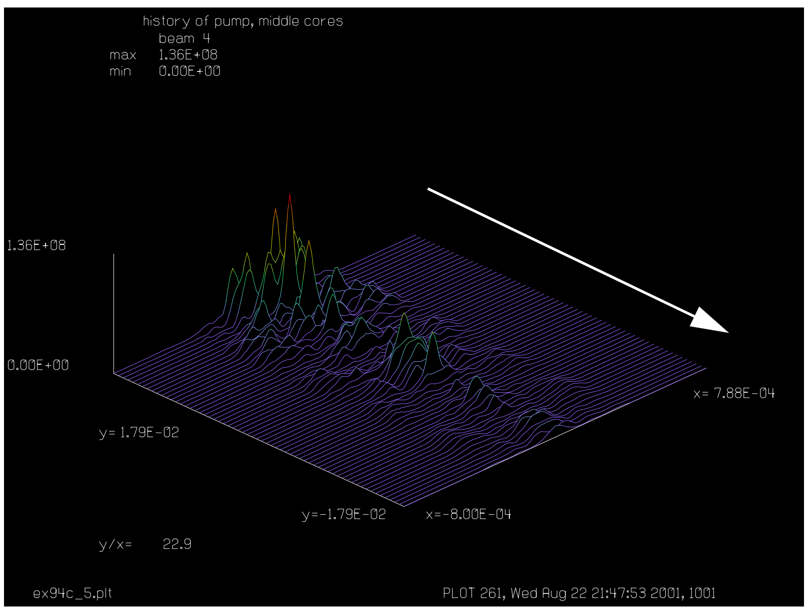
Pump depletion vs. length in four-core fiber

- Pump diffracts and reflects from walls in outer core
- Pump is depleted by four inner cores

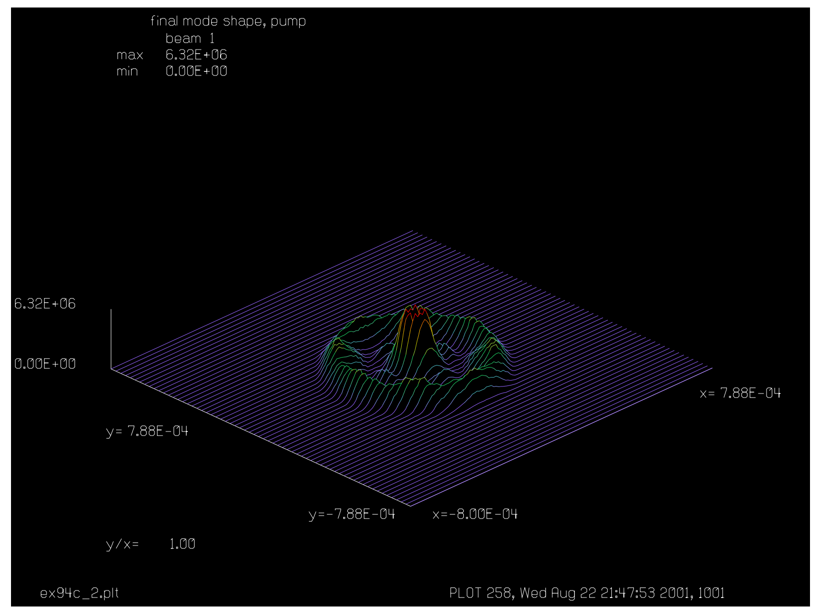




pump depletes vs distance

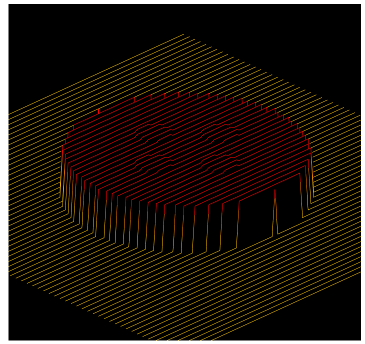


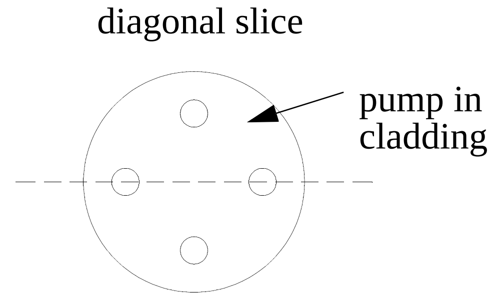
mode shape of depleted pump



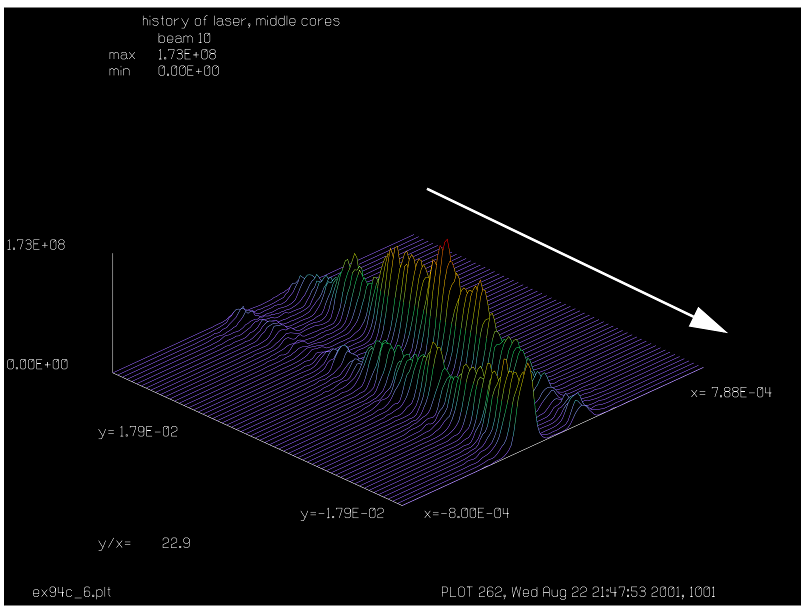
Laser mode vs. distance

- Cores couple by diffraction
- Mode beating between cores

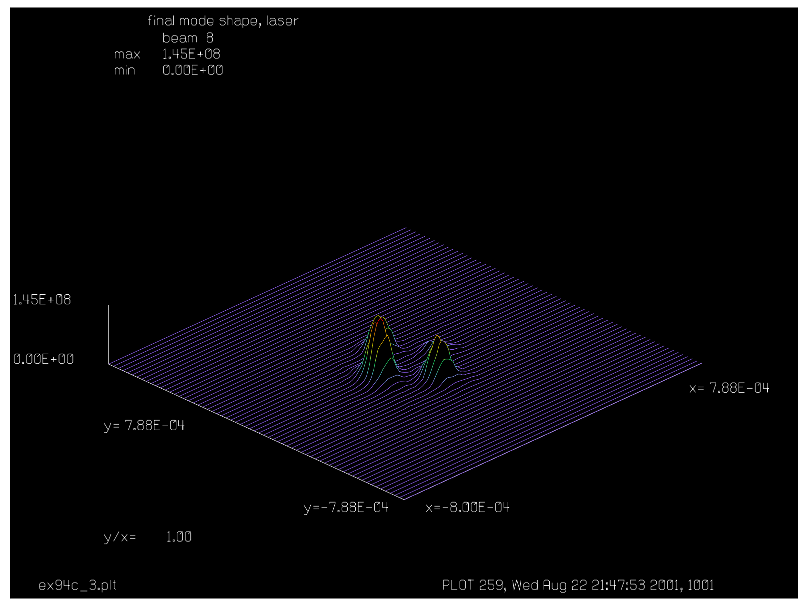




laser amplification vs distance

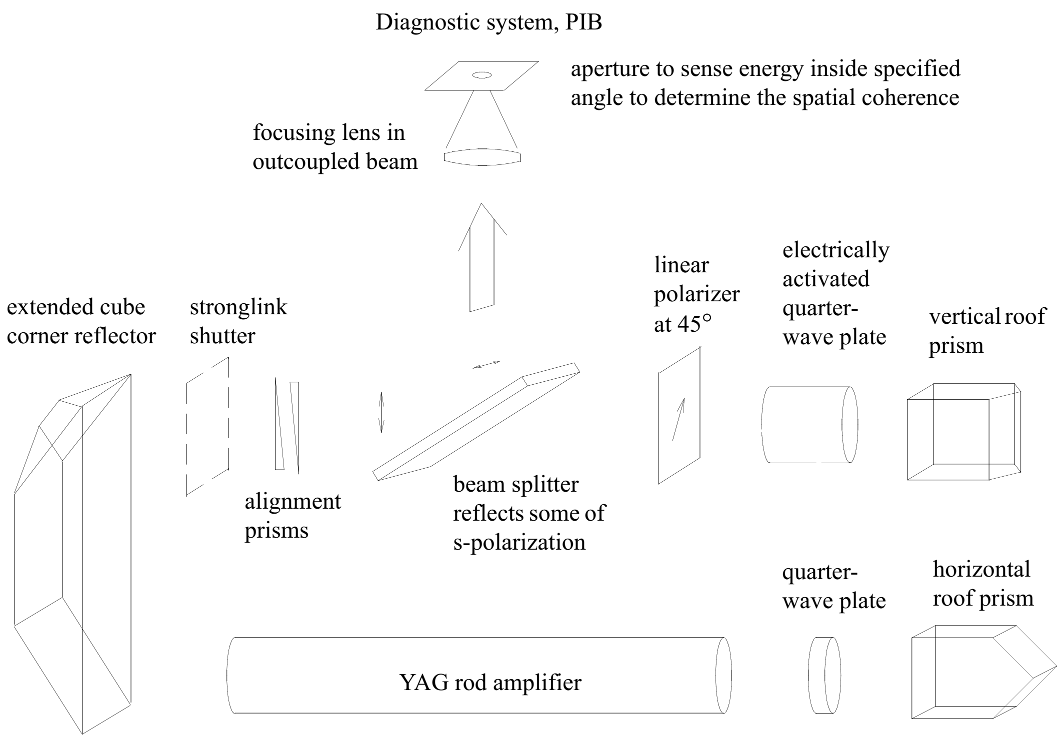


mode shape of laser

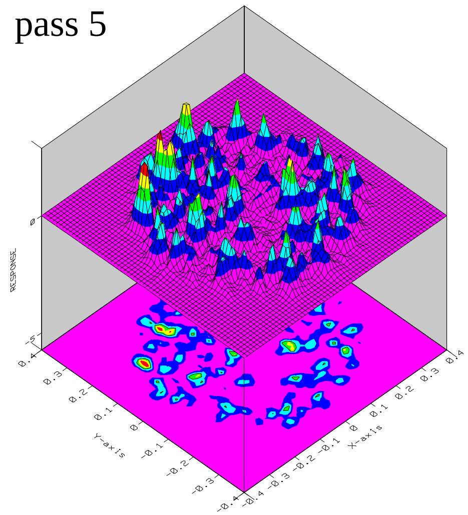


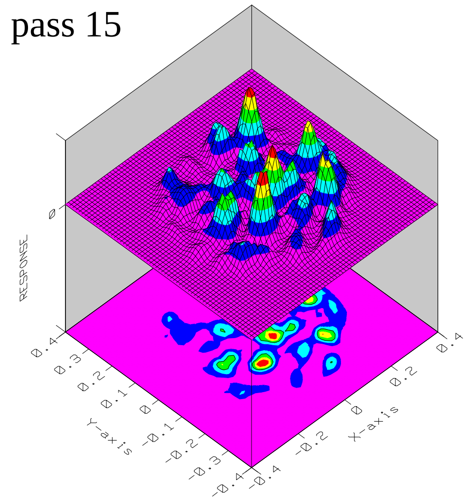
Q-switched YAG modeling

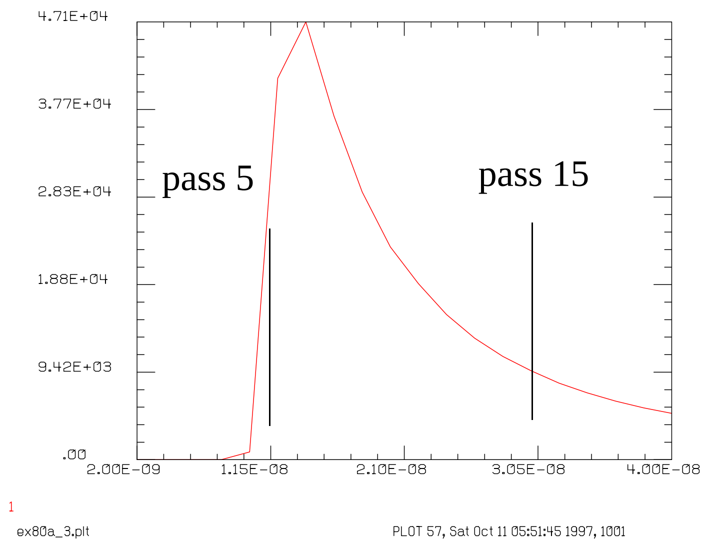
- Beam starts from spontaneous emission
- Complex amplitude propagates through the system
- Beam “cleans itself” during several passes as high spatial frequency light is removed by apertures
- Beam quality varies with time
- Simple diagnostic system:
  - focusing lens in outcoupled beam
  - aperture at focal point of lens forms simple power-in-bucket measurement



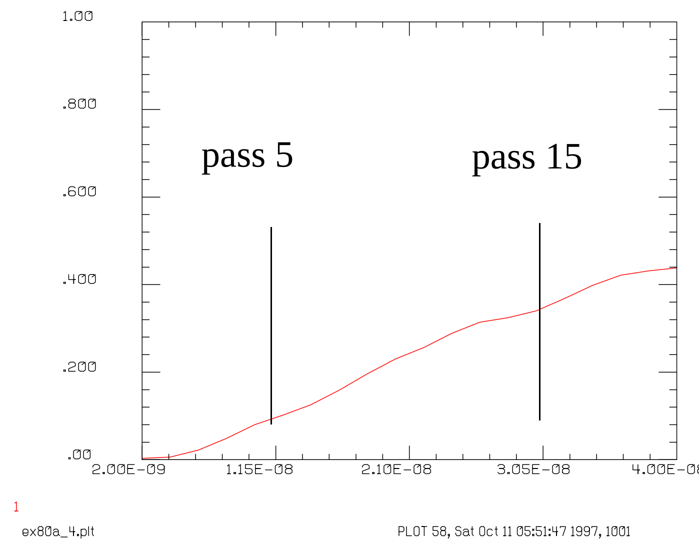
Beam quality varies with time for the Q-switch laser







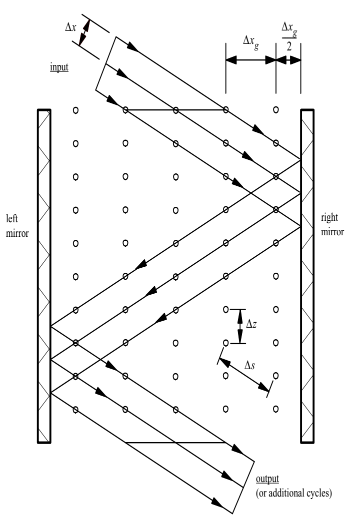
Power during pulse



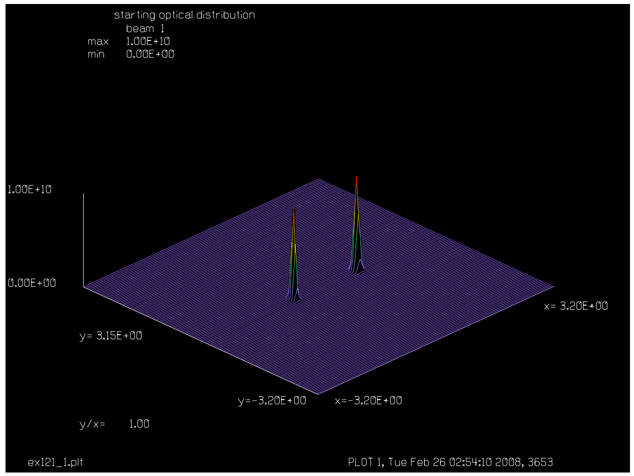
Relative power-in-bucket

Zigzag amplifier

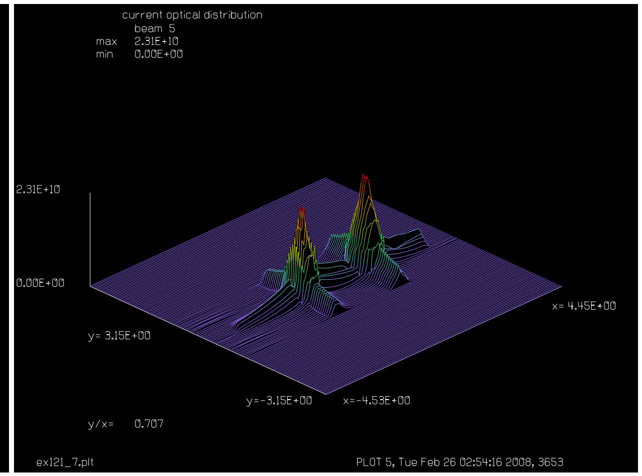
- Amplifier has about 10,000,000 points
- About 70,000,000 coupled differential equations solved
- Takes a few seconds on an ordinary PC.



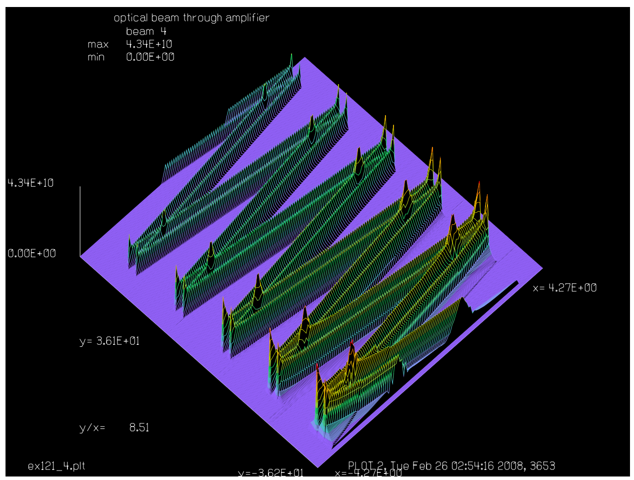
initial beam: double gaussian



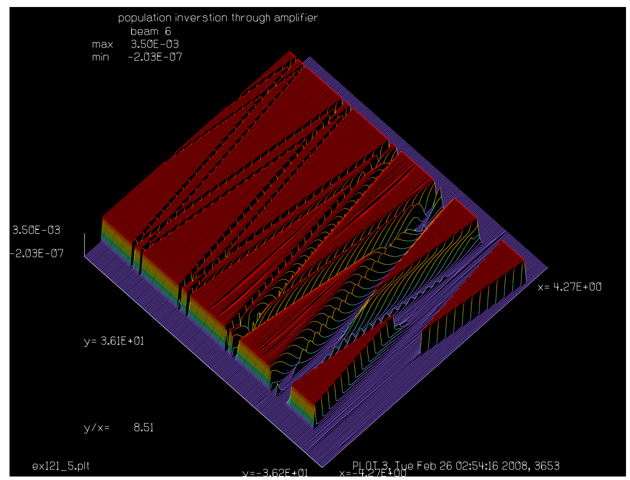
beams after amplification



growth of zigzagging double beam: 10 reflections

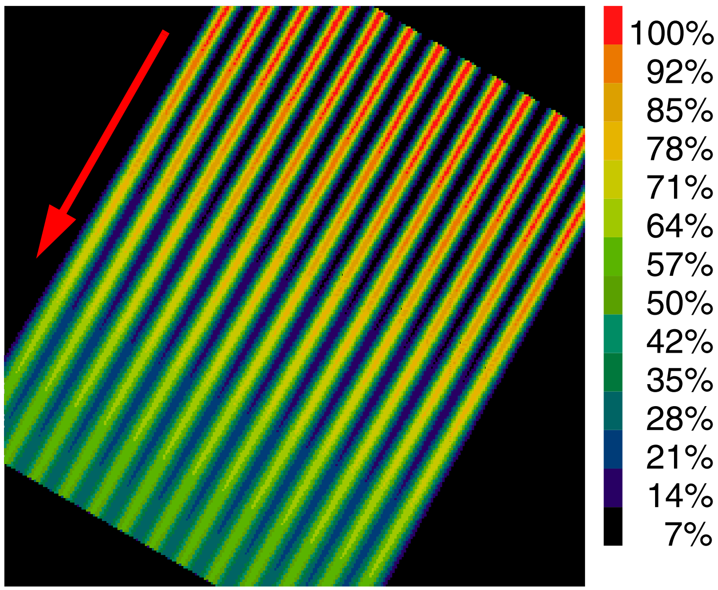


depletion of population inversion

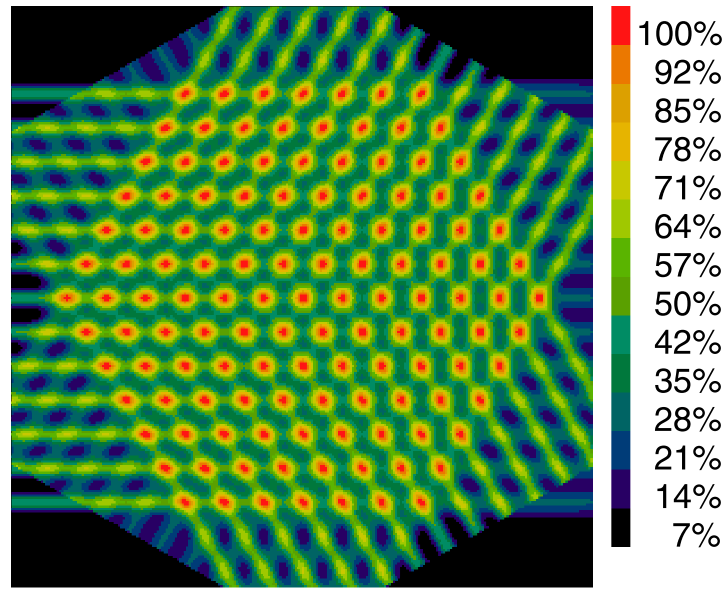


Side pumping by banks of laser diodes

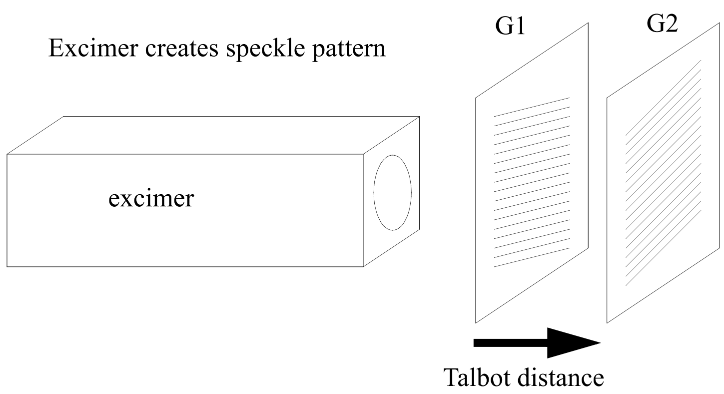
14 diode beams from three directions: +120 degees shown

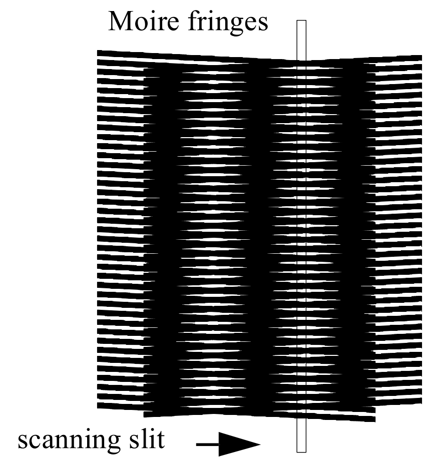


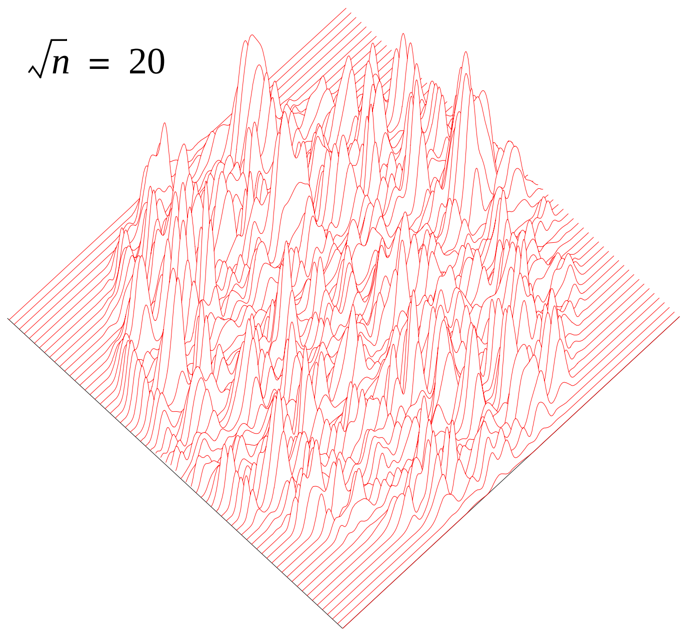
Cumulative pumpind distribution from three directions



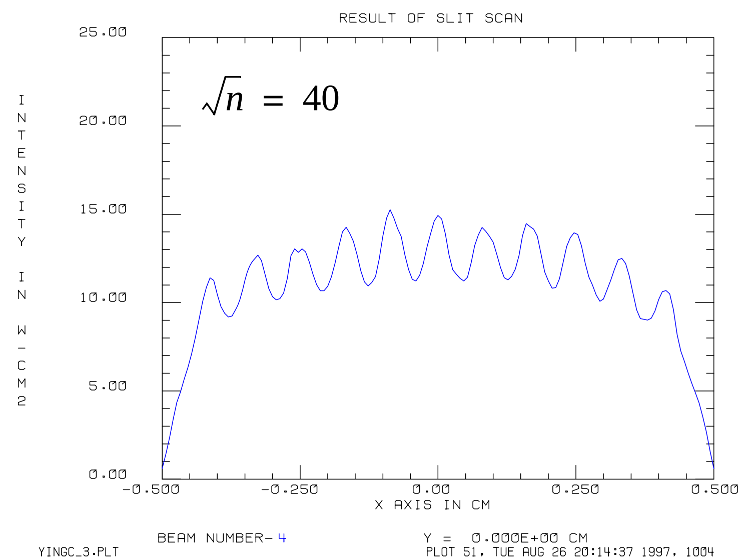
Measurement of excimer laser with Moire fringes







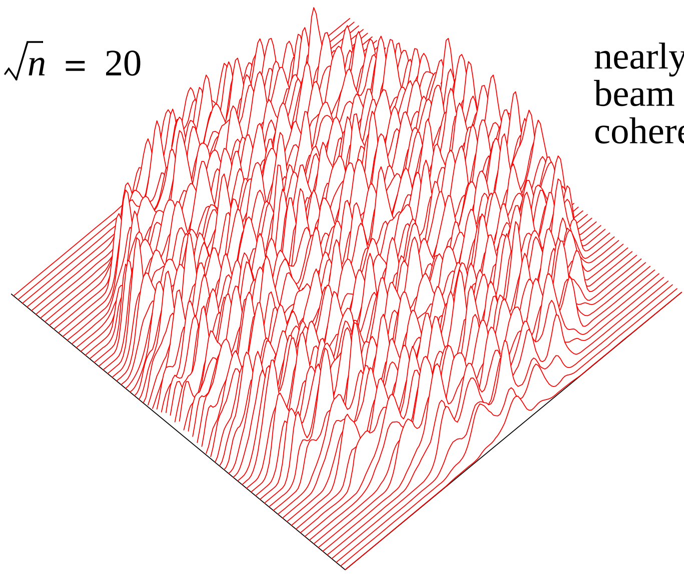
one instance (snap shot) of simulated excimer laser



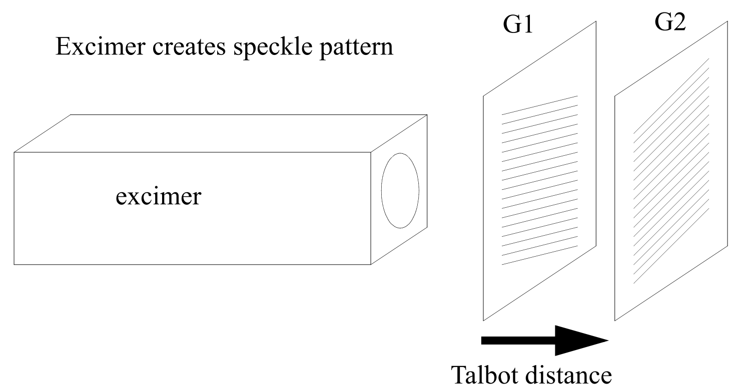
results of slit scan showing modulation

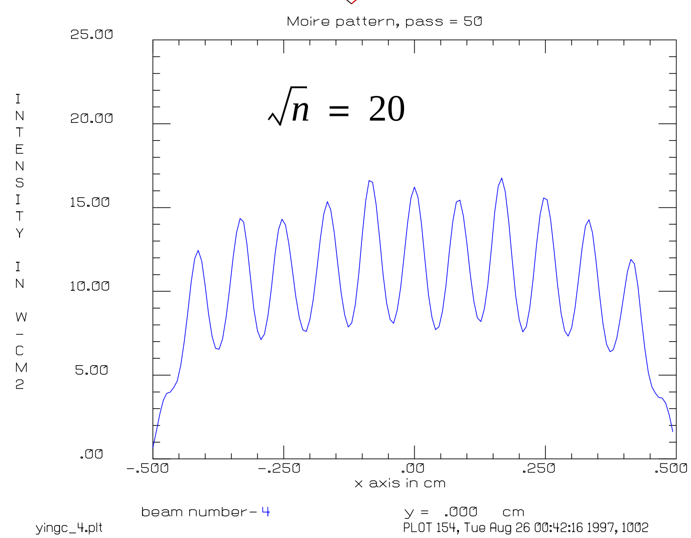
Time integrated slit scan of excimer beam

$\sqrt{n} = 20$



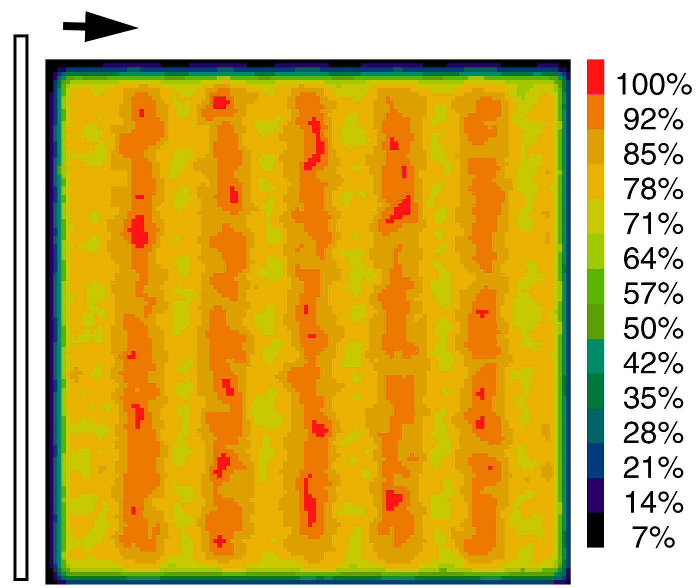
nearly uniform beam after 50 coherence times





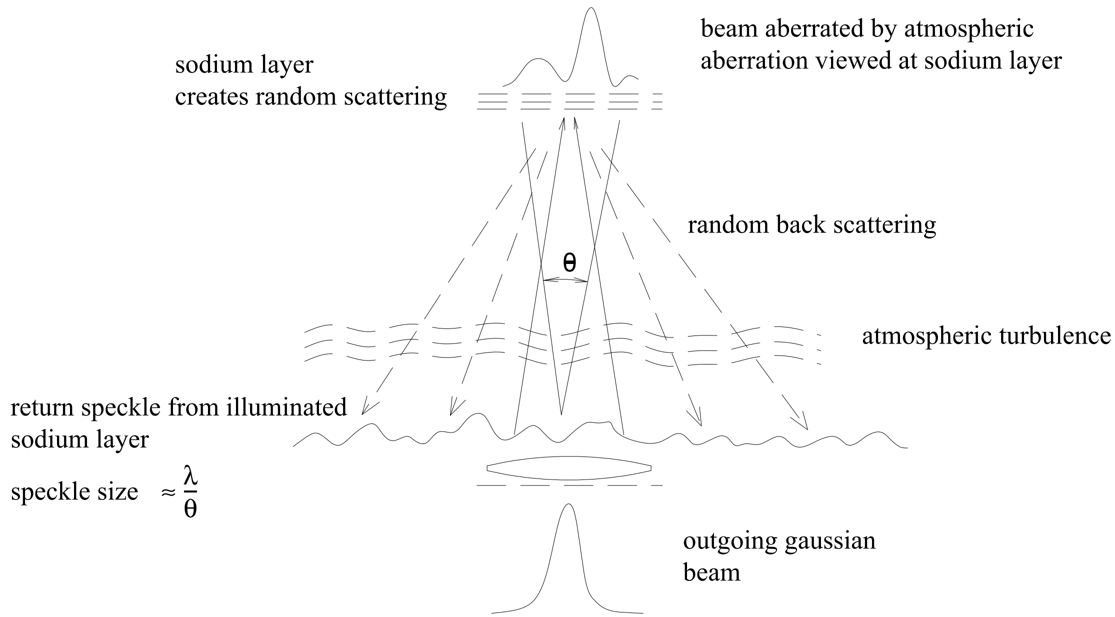
results of slit scan showing modulation

scanning slit



false color plot shows Moire pattern $\sqrt{n} = 20$

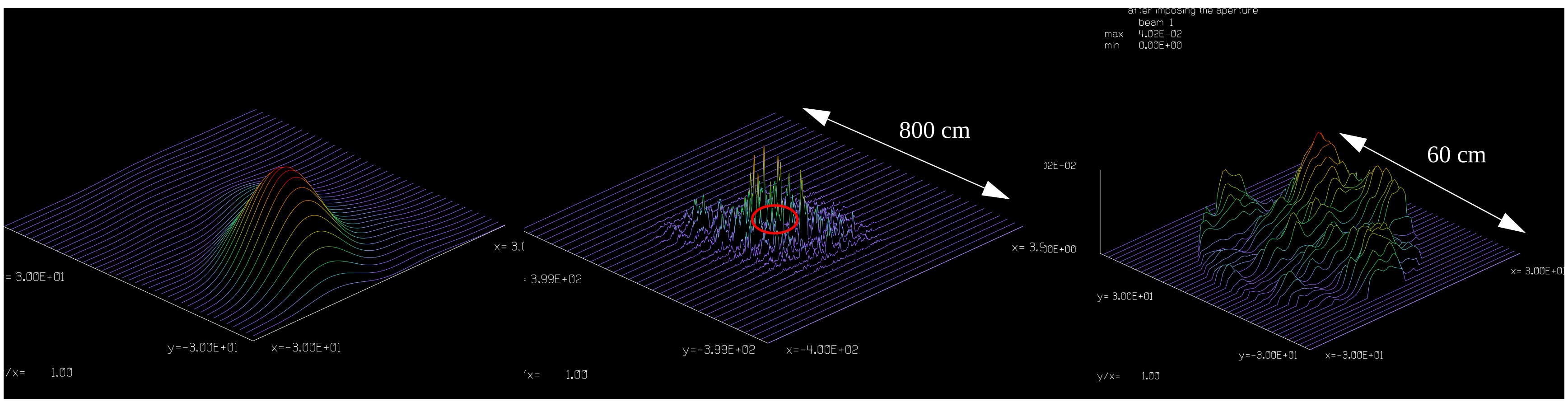
Use of atmospheric layer as a guide star



pulse on sodium layer distorted by atmosphere

speckle on ground illumination

speckle landing on receiver aperture



How to solve a problems

- Problems “solved” in the GLAD examples collection
  - Adapt one of the GLAD examples.
  - Very one variable at a time to move the command file to the configuration you need.
- Problems not solved in the GLAD examples collection
  - Try to visualize how your application works.
  - Devise a plan that breaks your problem into a series of steps.
  - Each successive step should add just <u>one</u> new issue to the model.
  - Keep copies of each step (you may need to backup).
  - Small “science” experiments may be useful to understand an important issue in isolation.
- Debugging
  - Try to stop the command file at the point where “well understood” behaviour changes to “strange” behavior.
  - Email the <u>bad</u> command file to AOR if you get stuck.

Guaranteed success!

2. GLAD Organization and Command Language

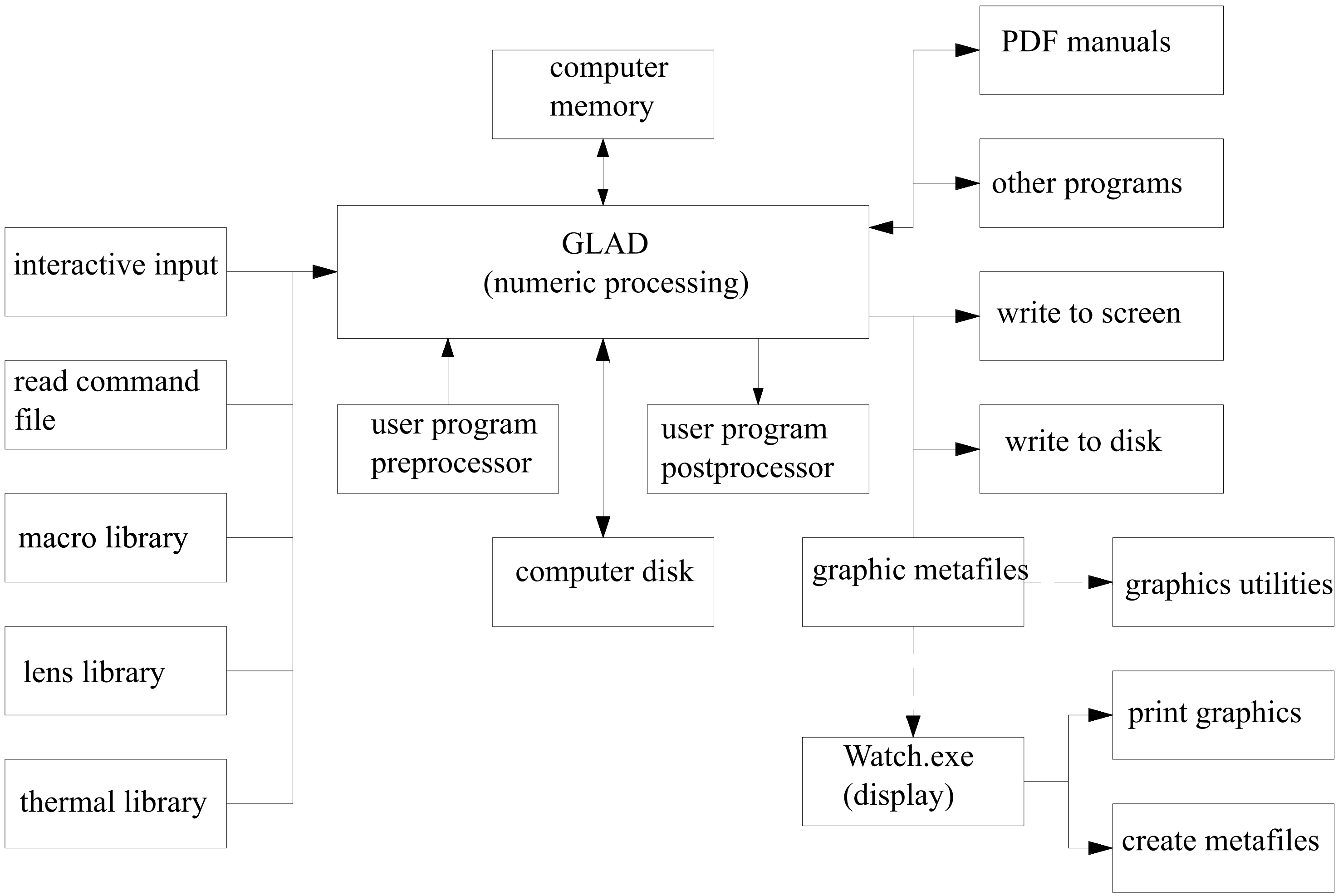
Infrastructure is the key to physical optics modeling

- Any size array $N \times M$, (tested for one-dimension to N and $M \leq 131072$)
- Built-in virtual memory
- Many interacting beams of variable size
- Separable diffraction allows high aspect beam sizes
- Automatic algorithm selection, path-invariant propagation (no near-field, far-field assumptions)
- Efficient, robust numerical methods
  - efficient, accurate methods: rate equations, nonlinear optics, finite element method
- Rich, scripted command language
- Huge repertoire of examples > 500
- Comprehensive documentation
- Expert technical support

GLAD executable programs

- Subdirectory (folder) for executables — c:\aor\glad58
- Subdirectory for initial examples — c:\aor\glad58\examples
- Number crunching — glad.exe, character-based console programming
- glad.exe performs all calculations
  - reads command files
  - interactive input
  - generates graphic metafiles: *.plt
  - maintains watch.dat which defines files to be displayed by watch.exe
  - import and export of beam data
- Graphic file displays — watch.exe, windows application
- Displays GLAD graphic metafiles, *.plt
- Gets file names from watch.dat
- Integrated design environment (IDE) — ide.exe, windows application calls glad.exe, watch.exe
- GLAD Comm server — utility to pass commands from IDE to GLAD and back from GLAD to IDE. Generally requires no user interaction
- Utilities such as, plt2ps.exe, keyread.exe

Structure of the program



GLAD is a programming language for physical optics

- Build a configuration file
- Run GLAD to test and evaluate file (batch processing)
  - graphics and text output
  - interrupt to return to interactive use
  - recontinue batch processing
- Correct and/or extend configuration file and rerun GLAD

Program solution is primarily batch processing. Interactive features are limited.

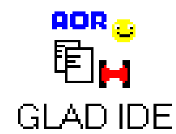
Batch processing advantages:

- Complex problems easily handled
- Accurate technical support possible by email (better than phone)

Running GLAD from the IDE

Select Start, Programs, GLAD 5.8, GLAD IDE

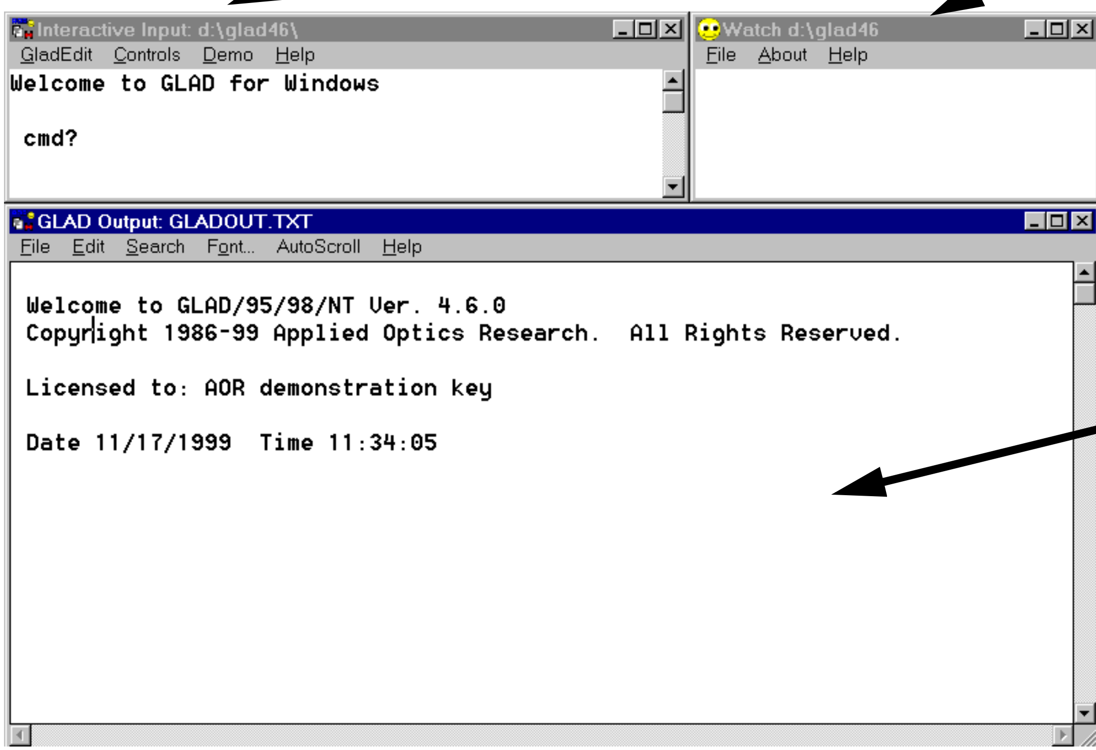
You will see several windows:



Input to GLAD	enter interactive commands here
GLAD Output	streaming output appears in this window
Watch	program to display graphics that GLAD creates

Interactive Input

Watch program



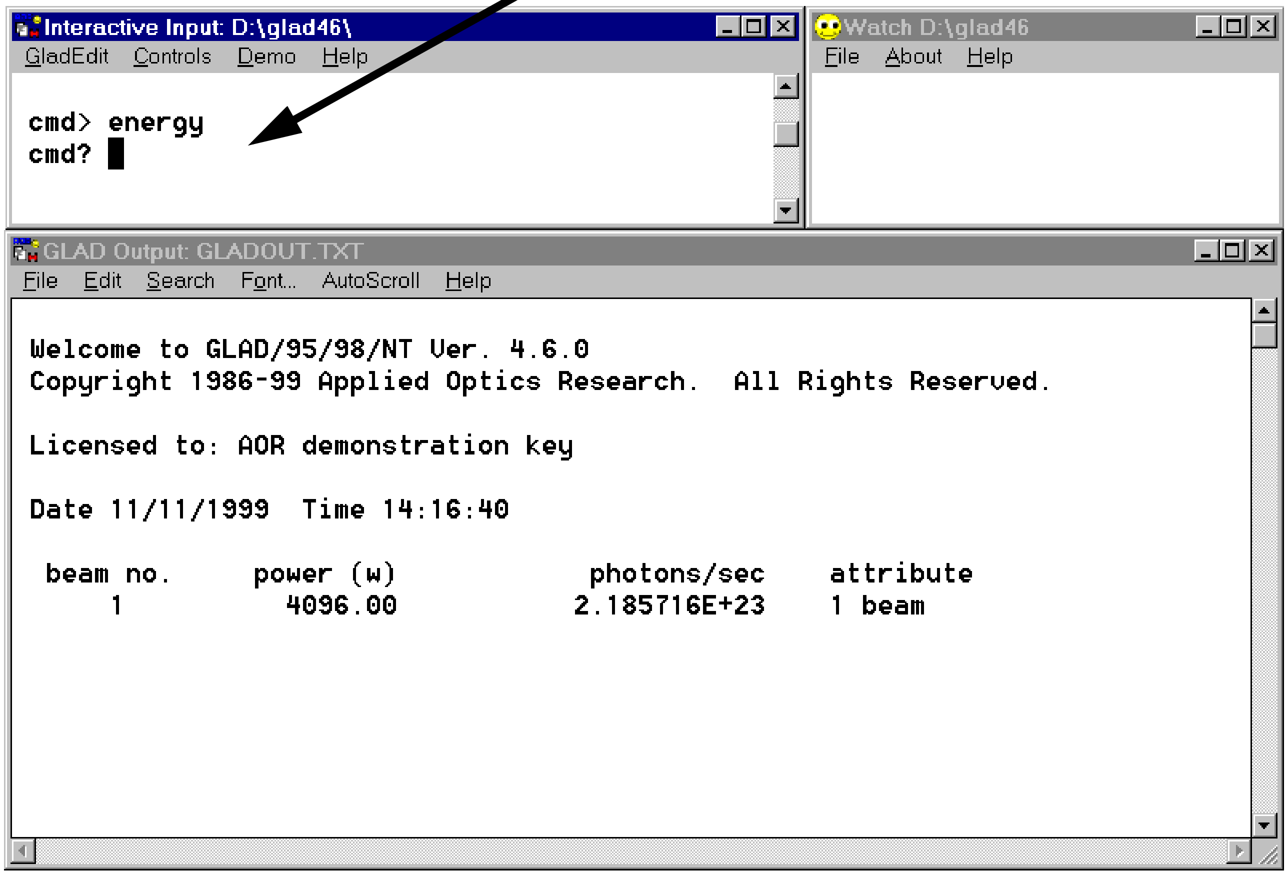
GLAD Out
text output

Interactive Input

commands are entered in Input to GLAD

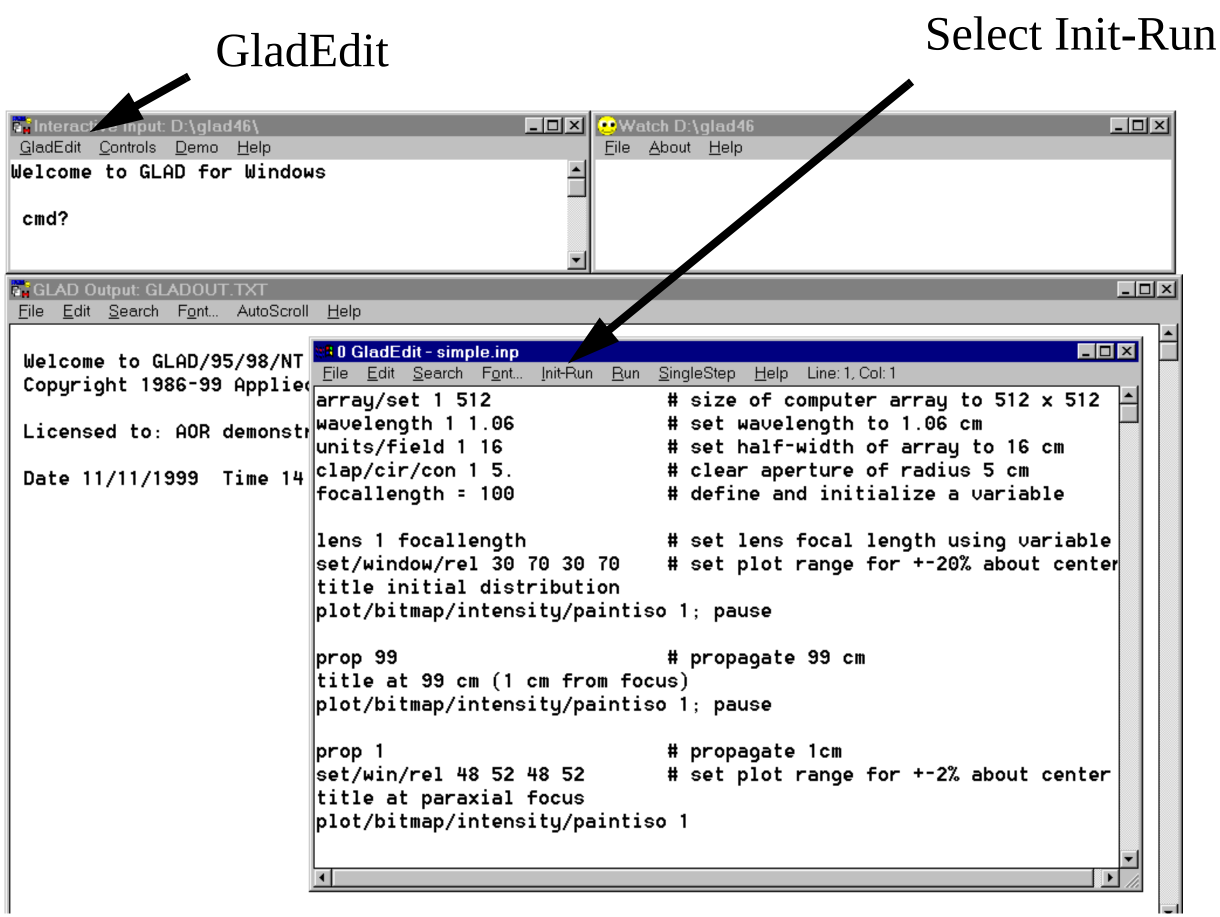
```
cmd> energy
```

Enter commands interactively

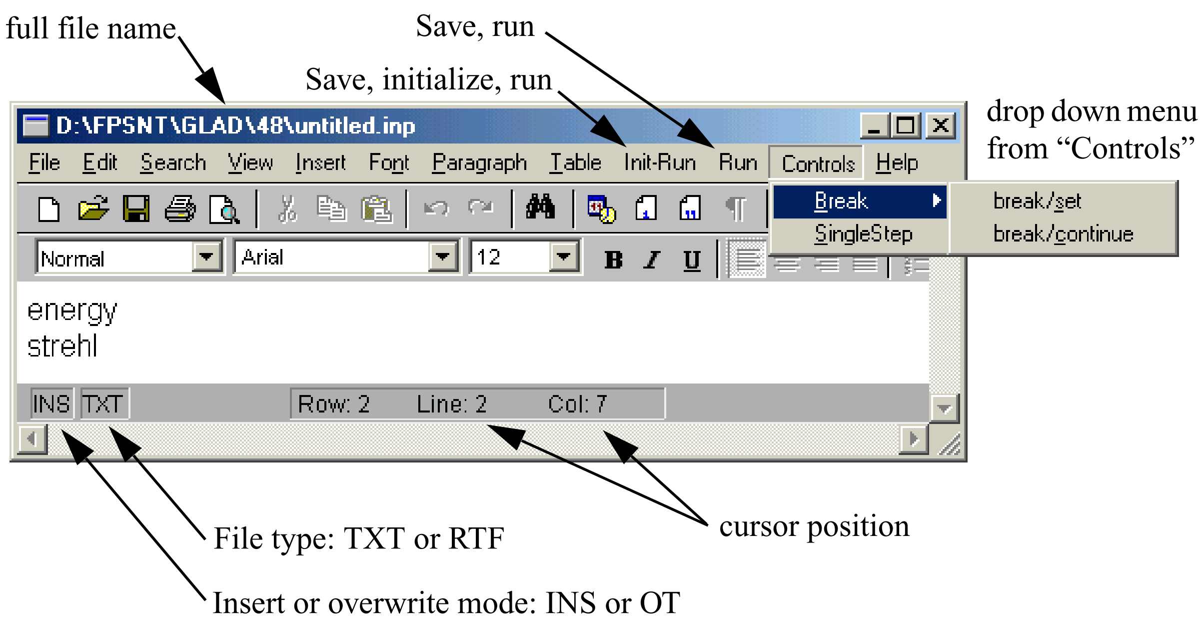


Starting GladEdit: simple.inp

- GladEdit is convenient for editing and running GLAD command files
- Click on GladEdit, Open simple.inp
  - Use Init-Run to reinitialize and run command file



GladEdit window

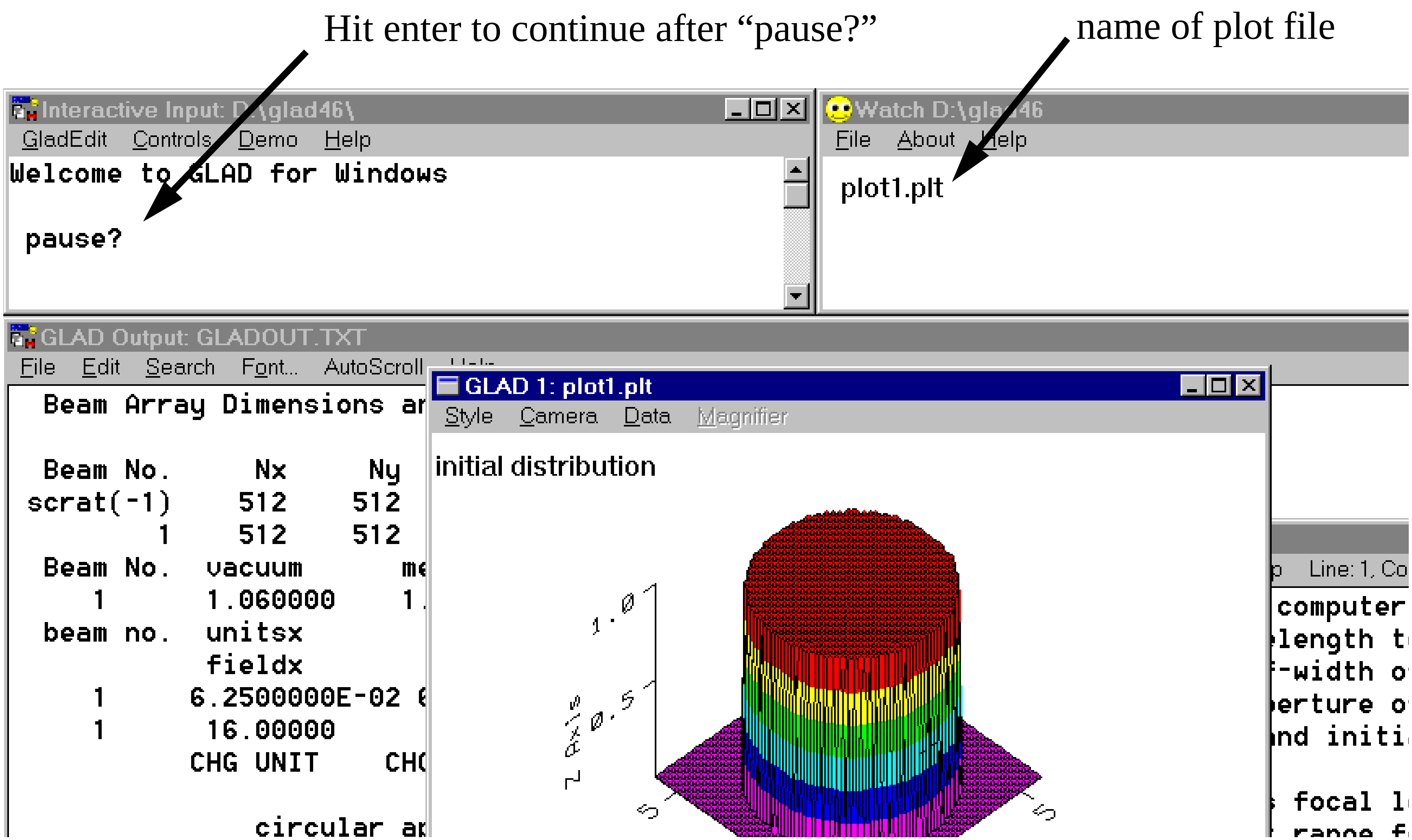


The title of the GladEdit window shows the full file name. “Init-Run” is the primary method of starting execution of a command file. “Run” starts without reinitializing GLAD. The “Controls” menu item gives access to the “break” controls and single step operation. “TXT” indicates the command file is in plain text format. “RTF” indicates rich text format is being used. The line and column number of the position of the cursor (more correctly called the caret).

Formatting is very similar to MS WordPad.

Running: simple.inp

- Running simple.inp with Init-Run
- Note “pause?” or (MessageBox pause) and graphic file display by Watch



Dynamic HTML output html6.inp

Polling ON (click to stop) | Poll | HTML line: 33 | Timer: 98 sec | Hide output

polling on/off control

tooltip definition

The index of refraction for the extraordinary ray.

First time notes
3 (1) [ ] html/location/off
4 (2) [ ] set/definitions/on
5 (3) [-] wavelength/list

Wavelength List

No.	Vacuum	Medium	Index	E-index
1	1.0600E+01	1.0600E+01	1.0000	1.0000

cursor on item

[+]Definitions
6 (4) [ ] macro/define/single test
7 (29) [ ] html/stop
8 (6) [ ] count = count + 1
9 (7) [ ] phase/random/sqr/clap 1 .1 5
10 ( ) [ ] html/start
11 (30) [-] plot/liso/wavefront 1

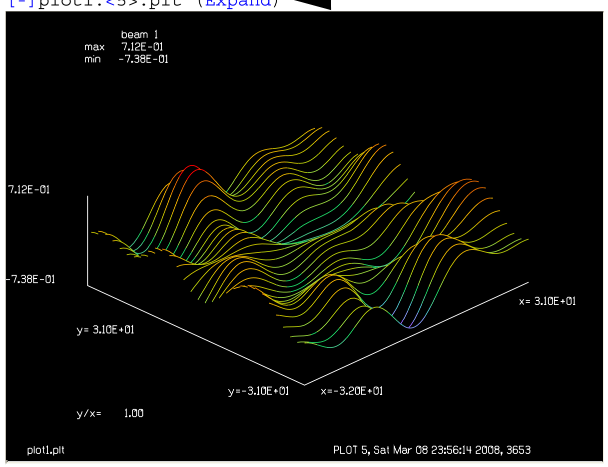
output display or hide controls [+] or [-]

angle bracket controls select between plot versions: 1, 2, 3, 4, 5

plot1.plt, ver. 5, PLOT 5, Wed Feb 27 01:51:28 2008, 3653

[-]plot1.<5>.plt (Expand)

select "expand" to double graphic size



high resolution SVG graphics made in-place below source line and automatically updated

12 (31) [ ] pause 2
13 (32) [ ] macro/end
14 (12) [ ] macro/run test/5
15 (33) [-] read/back/noclose # issued by GLAD for an end-of-file condition

source line number

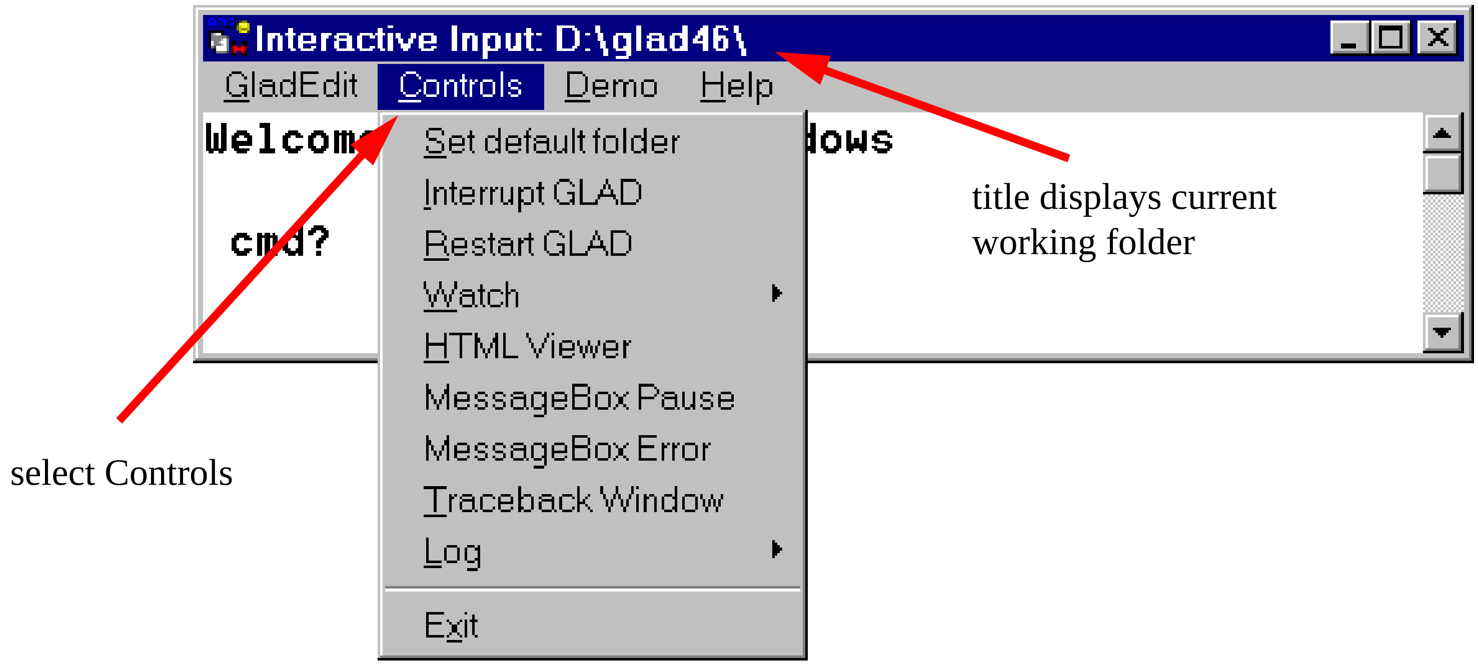
output line number. Red indicates current output line

read/back issued from open file: F:\fpsnt\glad\53\html5.inp

Interactive Input: reading from a command file

Command files may be entered directly in the Interactive Input window. `read/disk` will read simple.inp and execute the commands as they are read.

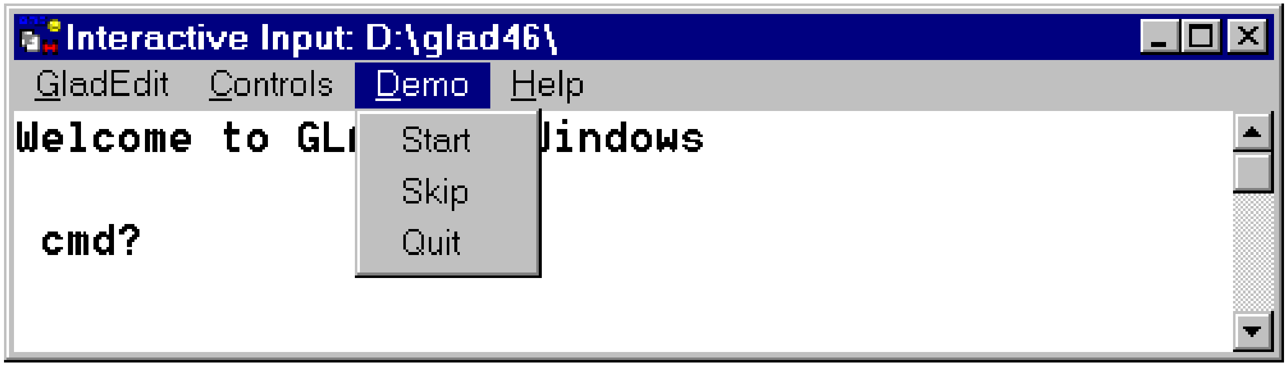
IDE: Controls



The Controls menu item allows selection of a number of operations. See the Help in Sectiion 1.2.8, GLAD Commands Manual for a detailed explanation. Use “Set default folder” to select the folder for GLAD to work from. The current working folder is displayed in the title of this window.

IDE: Demo

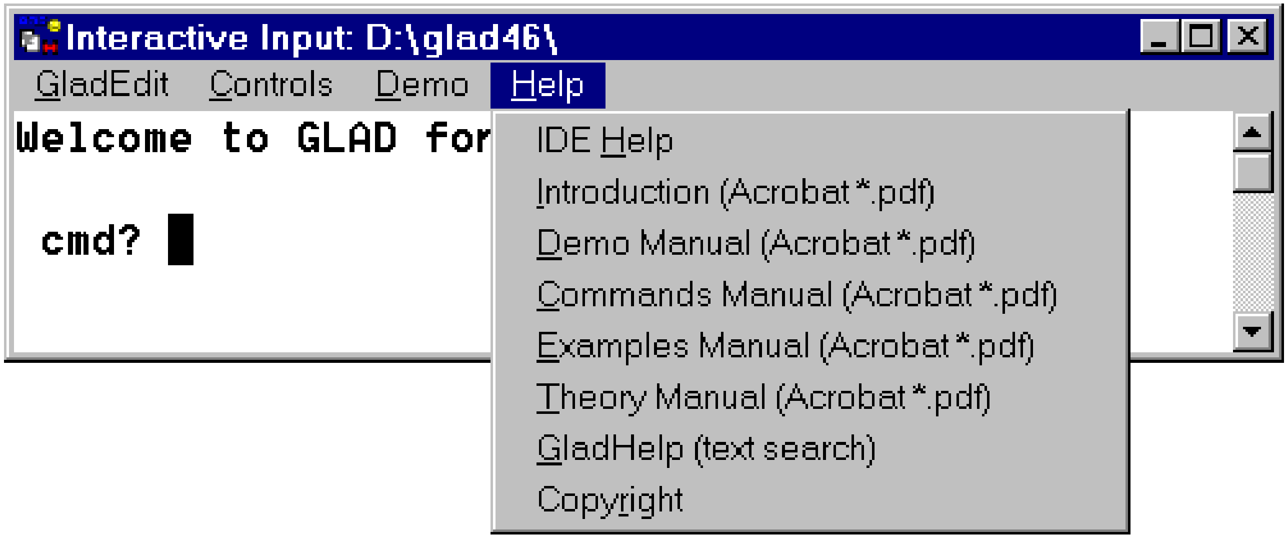
- Nine demo examples give a quick tour of GLAD
- Start demo, Skip an examples, or Quit



The Demo menu item runs preselected examples. Select Start to begin the demo, Skip to skip to the next example, and Quit to end the demonstration. See Demo.pdf in the installation folder for a description of the examples.

IDE: Help

- Windows Help: Ide Help
- Complete manuals, on-line form (Adobe Acrobat)



IDE Help gives specific information about operating GLAD IDE. Details about the commands, examples, and theory are in the respective PDF files, viewed with the Adobe Acrobat Reader.

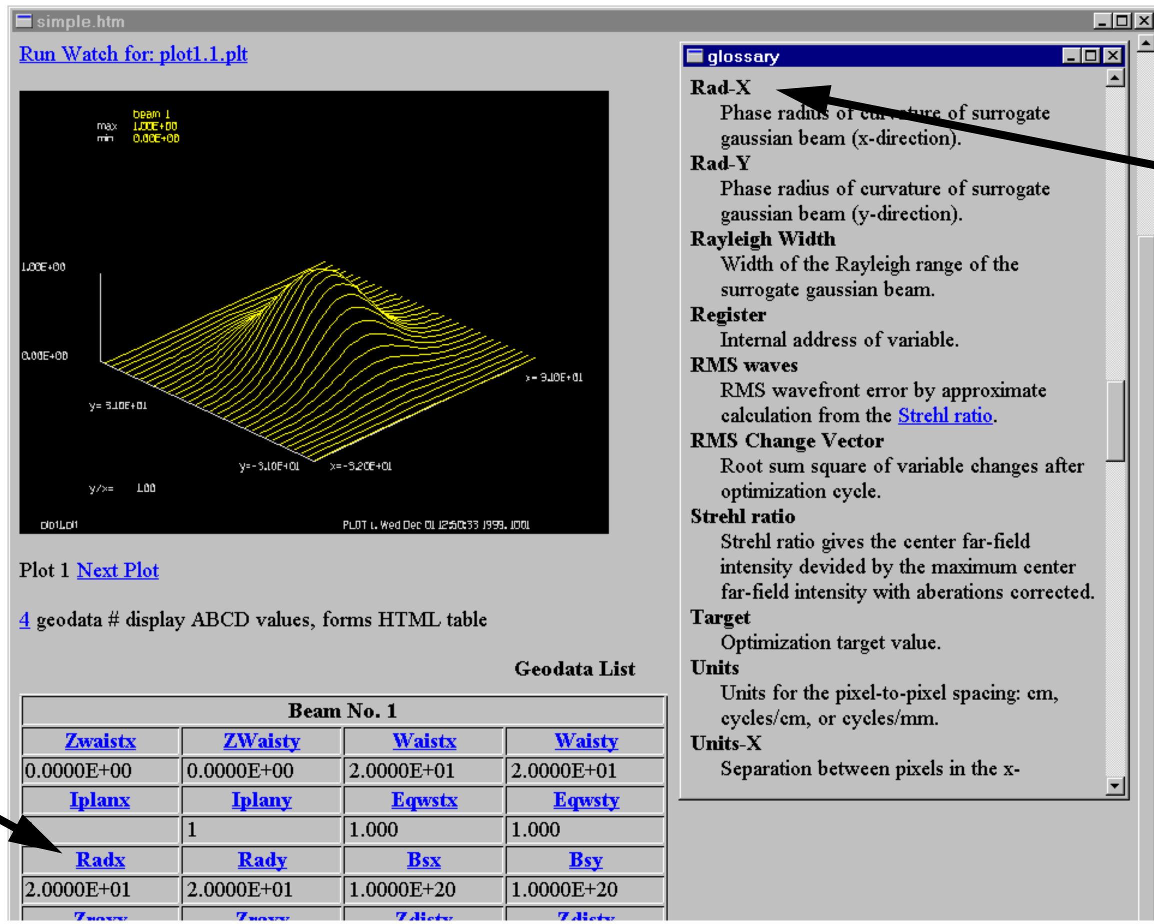
IDE: HTML: html.inp

```
gauss/c/c 1 1 20 # make a gaussian beam
html/write/on simple.htm # start writing to html file: simple.htm
html/wmf/on  # start output of plots as WMF files
html/viewer/start # start GLAD HTML viewer
plot/l  # make a plot geodata
geodata # display GEODATA values, forms a table
```

graphics are placed in sequence

table output

Item “Radx”



Radx defined in pop-up glossary

Running GLAD directly

- As console application:
  - Open DOS Command prompt window

```
cd \glad58
glad                                   (interactive)

glad comanndfile.inp                   (start from command file)
```

or

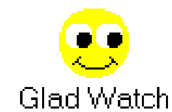
```
glad comanndfile.inp noconsole  (start from command file, do not open any GLAD windows)
```

Enter input file name as command line parameter.

- Can be used in DOS batch files (*.bat). Can be called from other programs.
  - Faster than running IDE

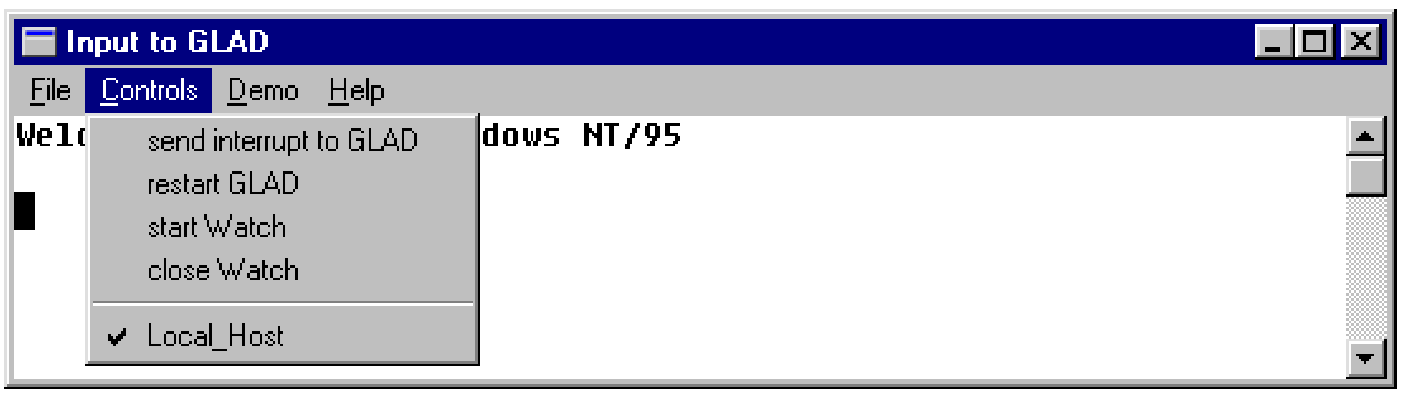
Running Watch

- Watch displays GLAD graphic metafiles as they are created
- files to be displayed are listed in watch.dat
- either IDE or GLAD will automatically start Watch.exe to display graphic metafiles.
- you can control Watch.exe from GLAD IDE
- you may also run Watch.exe independently by double clicking the icon:



  (useful for viewing the graphic files from the most recent GLAD run)
- you can control Watch.exe from GLAD (if started from GLAD) by

```
watch/close
watch/start
```

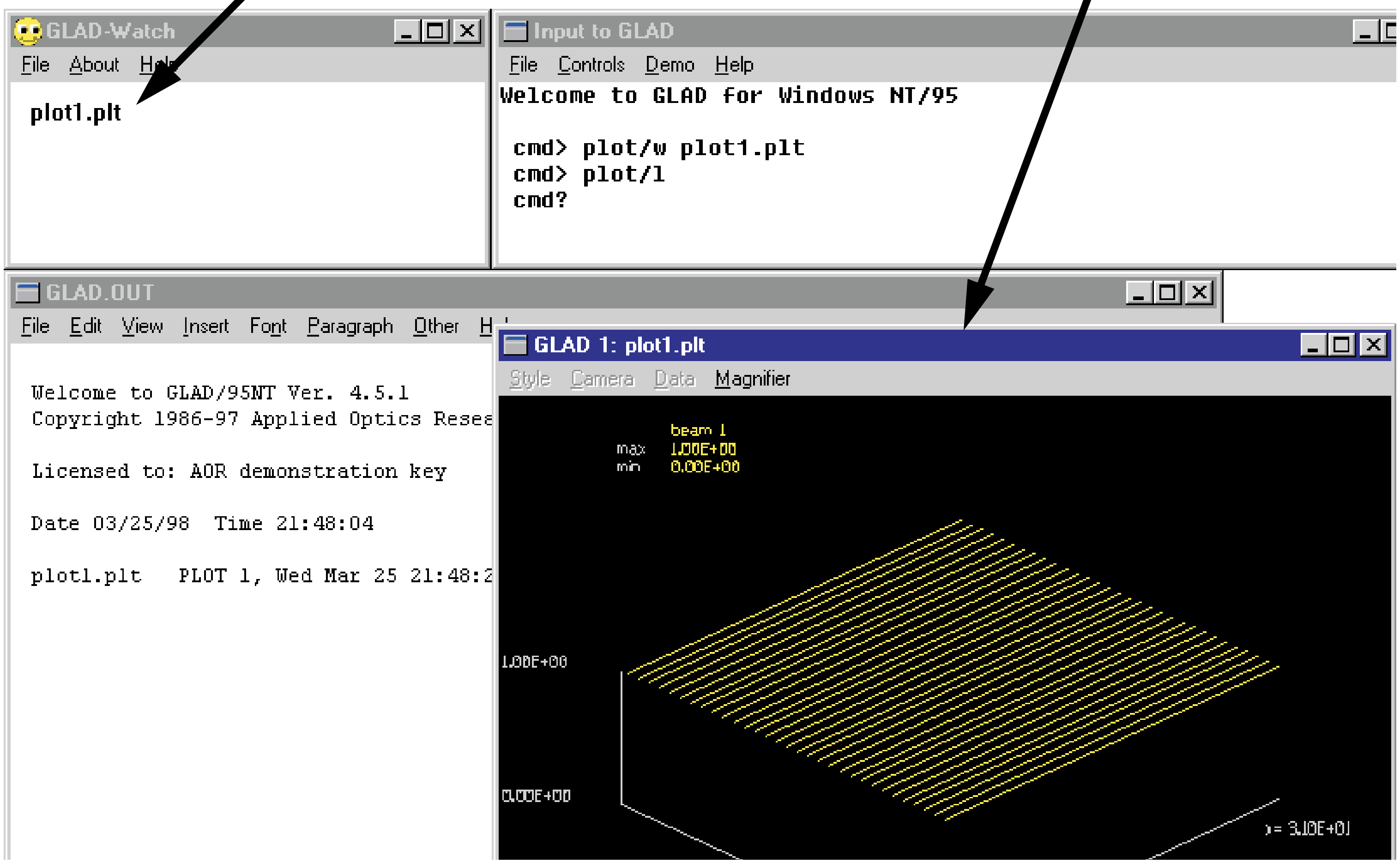


Watch

```
cmd> plot/watch plot1.plt # set name of GLAD meta file to write to
cmd> plot/l # make a simple plot
```

name of window displayed

Watch makes graphic window



Downloading latest code

From Demo/Download section of www.aor.com

glad58.zip	latest glad.exe
commands.pdf	Commands Manual (rather frequently updated)
theory.pdf	Theory Manual
examples.pdf	Examples Manual
watch.zip	latest version of Watch (may or may not be there)
ide.zip	latest version of IDE (may or may not be there)

Getting help

- online manuals, IDE Help, Watch Help
- use phone for philosophy, email for precise answers
- use email to send troublesome or curious files
  - send command file as attachments (using TXT suffix may help)
  - do not send output data or plot files, unless absolutely essential
  - do include some notes as to what is wrong
- avoid faxes if possible

GLAD files

For Ver. 5.8:

c:\aor\glad58	GLAD executable and utilities
c:\aor\glad58\examples	example files

Best to modify examples directoryc:\aor\glad58.

Types of files

*.exe	executable files: glad.exe, ide.exe, watch.exe
*.inp	command files (*.txt is also allowed)
	(*.inp is a registered file type. Starts GladEdit)
*.plt	GLAD graphic metafiles.
	(*.plt is a registered file type to start Watch)
*.bea	GLAD beam data files
*.dll	dynamic link libraries for various programs
*.cgm	computer graphic metafiles, from *.plt from Watch
*.wmf	windows metafile format, made from Watch
*.ps	Adobe Postscript, plt2ps. Use Acrobat Distiller to make
*.pdf files	Adobe compressed docment format.

Tools for files

- Built-in editor, GladEdit, from “Interactive Input” window, File, Open
- Making a report (such as GLAD manuals)
- convert GLAD metafiles (*.plt)
    - to *.wmf format from Watch
    - to *.cgm with plt2cgm.exe
    - to *.ps for further conversion to *.pdf with Acrobat Distiller
- Further conversion by Hijaak Pro, Corel Draw, Adobe Illustrator, etc. into other graphic formats as needed
- Making a movie:

ex122a.avi

    - Use special “capture” software on Watch window.

More about Watch

- GLAD puts the name (and optionally window position and size) into file: watch.dat
- Watch runs autonomously and displays all files in watch.dat
- Running from IDE, IDE starts and ends Watch
  - Can explicitly start and stop watch from Controls on Watch menu bar
- Running from GLAD, GLAD starts and ends Watch
  - can explicitly start and stop watch “watch/close” or “watch/start”
- Can run Watch independently of GLAD or IDE
  - run from Startup, Programs
  - run from DOS command line, watch.exe
- Running independently, Watch will display from watch.datleft from last GLAD run.
- File choice can be changed by editing watch.dat directly
- Also, can deselect “From watch.dat” and use “Add filename”
- Can print or make Windows metafiles from Watch
  - use placeable WMF files for including in MS Word
  - Can edit watch.dat directly

What can go wrong?

Multiple versions of Watch are allowed on the computer.

If GLAD crashes or is stopped in a nonstandard way, Watch may be left as a detached process.

- Watch is a detached process
  - You may see multiple copies of the graphic windows
  - slows down system operation
  - run task manager Ctl+Alt+Del
  - delete any extra copies of Watch you find running
- “Assess denied” under Windows NT
  - on occasion Windows NT will not clear the “busy” switch on files when the process using the file ends. File is then being used by a nonexistent process and Windows denies access to any other process
  - restart system
- *.bea files left after a crash -- wastes disk space

delete unwanted *.bea files

- macro library “maclib” or lens library “lenlib” becomes corrupted
  - delete the file, GLAD will rebuild

GLAD employs command language structure

- some form of command language is necessary for all programs to facilitate configuration saving — so we must have this in any case
- facilitates technical support by email
- least demanding of AOR programming resources
- most versatile structure
- can accommodate unlimited number of commands and subcommands
- command files of any length may be written and understood
- branching, looping, start and stop at any point
- macros (subroutines) and reading from different files easy
- inline equations easy
- requires frequent references to manual
- requires English proficiency
- intimidating to new user
- syntactically incorrect commands may be constructed(drag and drop icon method prevents syntactically incorrect choice)

Making command line format easier

- online manual using Adobe Acrobat format
- copious examples illustrate usage of commands
- technical support by email
- new command Composer will provide a command composition routine with drop down menus for writing syntactically correct commands.
- new 3D graphic layouts will aid in viewing configuration

Data input lines

No customer uses all the commands or even a majority of commands.
Do not attempt to learn all commands or understand all examples
— just learn the commands you need.

Data Input Lines

```
(line1) [CR]
(line2) [CR]
(line3) [CR]
```

(or)

```
line1; line2; line3 [CR]
```

```
clap/c/c 1 20;prop 100;plot/l
```

macro and `read` commands can not be followed by a semicolon

Input lines may be written on more than one physical line by using "&"

(first part of line) &[CR]

```
(second part of line) &[CR]
(third part of line) [CR]
```

Command format

The GLAD input line is one of four types:

Table. 2.1.

input line type	example
command line	`units/set 1 1`
conditional line	`if x=y status or if [x==y] status`
assignment	`x = x+3`
comment	`c some comment or # some comment`

```
command/mod1/mod2/mod3 string values parameters
```

Commands and their modifiers must be on the left, followed by strings, values, and parameters in that order if required. Fields are separated by one or more blanks.

```
c here are some representative lines
clap/cir/con 1 20 # command and two modifiers, two numeric values
read/disk myfile.inp # command, modifier, string
x = 3^2 + y list # "list" is a parameter
```

Command line components

Table. 2.2.

command line components	Definition
command	Defines function to be used.
mod1, mod2, mod3	Modifiers of the command line to direct operations of the function.
string	A string of characters to define a title, filename, macro name, lens name, system commands, etc. Strings have no preassigned values. File names should be enclosed in single quotes if they contain slashes, “/”, as is used in UNIX. Some strings have modifiers like commands.Variables may be included in strings if preceded by the '@' symbol.
values	Variables, numbers, mathematical expressions, or numerical assignments.
parameters	Parameters are similar to commands but are on the right end of the command line, after values. Parameters are checked against the list of preassigned values. Variables and expressions may be included in parameter names and their modifier names.

How GLAD parses a command

manual listing:

plot/xslice/intensity kbeam slice left right fmin fmax first last[parameter]

sample command: `plot/x/i 1 2 left=-10. right=2.*2.5 label`

- read command and modifiers from the left and transfer control to the appropriate routine, command: `plot`, modifier 1: `x` (xslice), modifier 2, `i` (intensity) transfer control to plot routine
- look for parameters on the left that match the parameter list for that command.parameter: `label`
- evaluate mathematical expressions in square brackets, right to leftmathematical expression: `2.*2.5 => 5.`
- string extraction for file names, etc., (no strings in this command)
- numeric values in order of occurrence, values: `first=1`, `last=2`.
- numeric assignment with an equal sign (lvalue = rvalue), `left=-10.`, `right=5.`

Numeric values and numeric assignments

Numerical values take the form of numbers, variables, mathematical expressions (which are evaluated to numbers and numerical assignments).

Table. 2.3.

numeric values	definition	example
numbers	Integers, floating point numbers, scientific notation, and complex.	`-1354, 1.23456, 1.23e+10, 2.5-4.3e-4i`
variables	Names consisting of no more than 20 characters which may be used in mathematical expressions. Allowed characters include A-Z, a-z, _, and $.	`x, y, peak, Pass_Counter, energy_$2`
mathematical expressions	For “IF” statement or if there are internal spaces, enclose in square brackets, [ ] (not needed for assignment lines).	`x = 2.*sin(2.*pi*y/period)` `pass = pass+min(3.4, x, z)`
numerical assignments	lvalue = rvalue form, where lvalue is one of the list of names for the numerical values the specific command.	`units ibeams=1 xunit=2.5 yunit=3.5`

Use of mathematical expressions

Consider the command line

```
x = sin(pi/2) list
```

(sets `x=1.0`, GLAD will identify "`x`" as a variable the first time it is used or we could use:

```
variable/declare/real x
x = sin(pi/2) list
units 1 x=y^2 y=[y=x+1.5]
```

command form: `units/set ibeams xunit yunit`

1) processing proceeds from right to left, `[y=x+1.5]` is processed first

2) `y` is recognized as a new variable name and assigned the value 2.5.

3) the lvalue `y` outside the brackets is recognized as an abbreviated form of the numerical assignment name yunit and the value 2.5 is assigned to it.

4) the second mathematical expression `y^2` uses the variable `y`, as just established, and the expression take the value 6.25.

5) The numerical assignment `x=y^2` is recognized as an abbreviated form of `xunit=y^2`. The lines above are equivalent to the command lines

```
x=1
y=6.25
units 1 6.25 2.5
```

Or

```
units ibeams=1 xunit=6.25 yunit=2.5
```

The variables `x` and `y` are available for use elsewhere in the program.

Use of mathematical expressions (cont’d)

Use of mathematical expressions (cont'd)

```
x = sin(pi/2) list # our funny expression
units 1 x=y^2 y=y=x+1.5 # set units
units 1 x=y^2 y=[y = x + 1.5] # spaces in math expression require brackets
variables # check the values of x and y
units # check the units
```

Try this!

More about variables

Variables may be used in strings or parameters and their modifiers by preceding them with the “@” symbol. For example,

```
variable/declare/int I # declare “i” as an integer
I=1
plot/watch plot@i.plt
I=I+1
title This is plot @i
plot/watch plot@i.plt
```

Establishes plot names plot1.plt and plot2.plt successively. If variables are included in titles, the current value is always used.

Variables and functions: function.inp

```
c## function
echo/on
x = step(2,1) list          # 1
pause
x = step(0,1) list          # 0
pause
x = ramp(3,1,3) list        # 6
pause
x = ramp(1,1,3) list        # 0
pause
x = ramp(0,1,3) list        # 0
pause
x = rect(3,1,2) list        # 1
pause
x = rect(4,1,1.5) list      # 0
pause
x = ramp(-1,1,1.5) list     # 0
pause
x = gauss(3,2,5,2) list     # .9984013
pause
x = 2*3/(4+5) list          # .666667
pause
x = 1||2 list               # 1
pause
x = 0||0 list               # 0
pause
x = 1&&2 list               # 1
pause
x = 0&&1 list               # 0
pause
x = !0&&1 list              # 1
pause
```

```
x = 1==1 list            # 1
pause
x = 1==2 list            # 0
pause
x = 1!=1 list            # 0
pause
x = 1!=2 list            # 1
pause
x = !1!=1 list           # 1
pause
x = !(1!=2) list         # 0
pause
x = 1>2 list             # 0
pause
x = 2>1 list             # 1
pause
x = 1<2 list             # 1
pause
x = 2<1 list             # 0
pause
x = 1<=1 list            # 1
pause
x = 1<=2 list            # 1
pause
x = 2<=1 list            # 0
pause
x = 1>=1 list            # 1
pause
x = 1>=2 list            # 0
pause
x = 2>=1 list            # 1
pause
x = 1>2||2>1 list        # 1
j = 0
m = 1
```

```
n = -1
x = 2.5
y = 0.
a  = j && m list                  # 0
pause
a = (j < m) && (n < m) list       # 1
pause
a = m + n || !j list              # 1
pause
a = x * 5 && m / n list           # 1
pause
a = j <= 10 && x >= 1 && m list   # 1
pause
a = !x || !n || m + n list        # 0
pause
a = x * y < j + m || n list       # 1
pause
a = (x > y) + !j || (n + 1) list  # 1
```

Command line prompting and help

Simple command prompting

The special parameter “?” causes GLAD to prompt for commands, modifiers, and values. For example,

```
clap/?/? ?
```

Conditional line or if Command

The `if` command allows conditional input line processing.

Single line form:

```
if arg1 (Relop) arg2 (input line)
```

BLOCKIF form:

```
if arg1 (Relop) arg2 then
 .
 .
else
 .
endif
```

where `arg1` and `arg2` are numbers, variables, or mathematical expressions enclosed in brackets. Relop is a relational operator (<, >, =, <=, =>, != or <>), number or mathematical expression in brackets.

In the `IF` commands, 0 = false and 1 = true

Examples of IF statements. Can you predict what will happen? (See: if.inp)

```
c## if
c  Example of various IF statements
c
echo/on
if 0 status          # false
if 1 status          # true
if 1=0 status        # false
if 1=1 status        # true
if [1==0] status     # false
if [1==1] status     # true
if 1 then            # true
   status
endif
if 0 then            # false
   status
endif
if 1 then            # true
   status            # selected
else
   color
endif
if 0 then            # false
   status
else
   color             # selected
endif
```

It is a really good idea to indent logical blocks of commands for readability.

```
if 0 then
   if 0 then
      color
   else
      units
   endif
   if 1 then
      status
   endif
else
   if 0 then
      status
   else
      color       # selected
   endif
endif
if 1 then
   if 1 then
      if 1 then
         if 0 then
            status
         else
            color # selected
         endif
      endif
   endif
endif
```

Macros: **count.inp**

Macros are similar to subroutines (Fortran) or functions (C). They constitute a collection of commands which may be repeated any number of times.

define the macro

execute the macro

For example, find the sum of the integers 1 to 100.

Simple counting macro: **count.inp**

```
macro/define sum/overwrite # start macro definition
 count = count + 1
 sum = sum + count
macro/end # end macro definition
variable/declare/int sum count
macro/run sum/100
sum=
```

Conditional exiting from Macro: precision.inp

Macro to calculate machine precision: precision.inp

```
c# find numerical precision
variab/dec/int count
epsilon=1.
macro/def step/o
 count = count + 1 list # count number of passes
 epsilon = epsilon/2. list # decrease epsilon each pass
 arg = (1.+epsilon) - 1. # subtract similar numbers
 if [!arg] macro/exit # exit when arg = 0
macro/end
macro step/100
```

Some simple coding

- `command/modifier1/modifier2 value1 value2` specify array size for beam 1

```
array/set 1 64              # set array size with a comment
arr/s 1 64                  # spell out enough to be unique
nbeam 2                     # increase number of beams to 2
                            # expand with same size as last beam
nbeam 3 128                 # specify beam size and expand
array/s 3 64 128            # respecify beam 3
array/list                  # list data for all arrays
array/list 2                # list data for specified beam
array                       # "list" is default modifier
```

- Try "array ?"
- Try "array/?"
- Use Command Composer
- Lookup "array" in online manual using Adobe Acrobat Reader.

More simple coding

Use variables for convenience and readability

```
variable/declare/integer Size
Size = 128# equation line needs no brackets
array/s 1 Size# use variable so array size can be changed easily
nbeam 2 Size
nbeam 3 Size*2# equation in command line requires bracket
```

Will this work as a complete program? Why?

```
array/s 1 Nline*2
Nline = Nline + 1
```

Will this work as a complete program? Why?

```
Nline = Nline + 1
array/s 1 Nline*2
```

Will this work as a complete program? Why?

```
array/s 1 Nline=2*64
```

```
variab/dec/rea x1 x2
if x1=x2 status# Why is this bad practice? What is better method?
```

memory and timing

- Timing a diffraction calculation

```
time/init;prop 1;time
```

- Set the memory to hold all arrays in memory if possibleIf not all arrays will fit, try to set memory large enough to hold largest array.

```
Bytes = 8*rows*columns*(polarization states)*(number of beams)
```

- Set memory to 2 MBytes (1 MByte = 1024 x 1024)

```
mem/set/b 2
```

- Calculate time to propagate a short step of 1024 x 1024 array for memory of .5 Mbytes, 2 MBytes, 8 MBytes
  - Observe disk IO activity

```
debug/add rdwr
```

- Rerun test of 1024 propagation at .5, 2, 8 Mbytes of memory

3. Initializing Arrays and Beams

Starting the calculation

- setting memory
  - Required memory per array is N × M × 8 bytes (× 2 if array is polarized e.g., 256 × 256 × 8 = .5 MByte)
  - GLAD runs <u>most</u> efficiently if all arrays can fit in memory at the same time
  - GLAD still runs efficiently if each array will fit in memory by itself
  - GLAD use built-in virtual memory if only part of array will fit in memoryat one timebuilt-in memory achieves maximum theoretical speed
  - GLAD is set to have 8 MByte of memory by default
  - memory may be changed by `memory/set/bytes MBytes` (in megabytes)
- memory may be checked by `memory/contents`

Time tests

- Time of execution may be checked by `time/init` followed by `time`

```
time/init # initialize time counter
prop 1 # propagate a short step
time # time
```

- Calculate the propagation time of a 512×512 array for memory allocations of 2 MBytes, 1 MBytes, and 0.5 MBytes.

Specifying the array size

- array/s kbeam=2 nlinx=128 nliny=64 ipol=1
  specifies array 2 to be of 128 × 64 with two polarization steps
- array reinitializes all beam data except wavelength
- use attribute `data` to make array impossible to propagate

```
array/s 1 64 64 data
```

Setting the number of beams

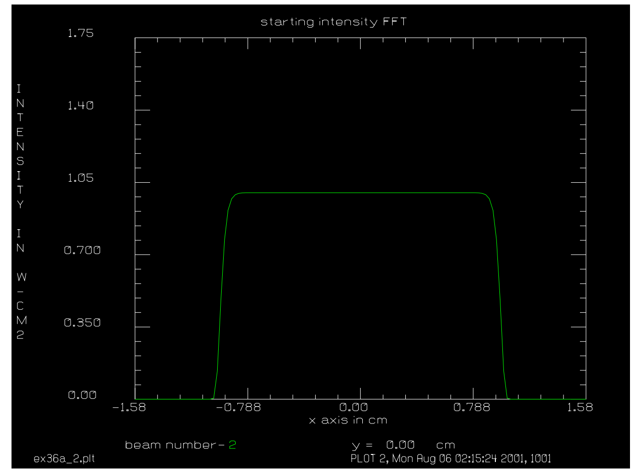
- up to 128 arrays (beams) may be specified for optical propagation beams, data arrays, etc.
- `nbeam n` expands the number of beams using the properties of the n-1 beam.
- `nbeam n nlinx nliny data` expands beam number using specified array size and specifying that the array has data attribute

Initialization of array values

- arrays are filled with 1's when defined

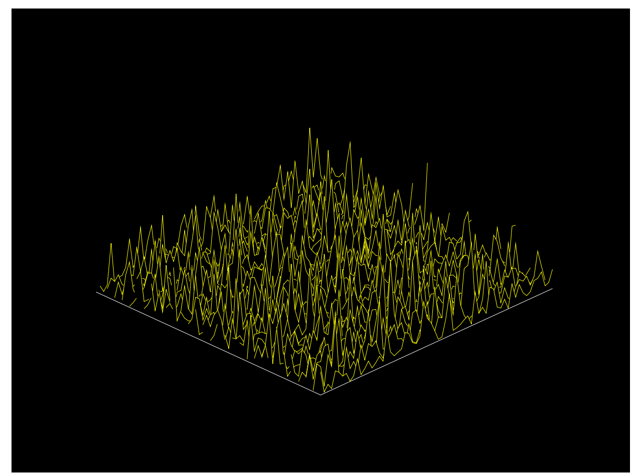
```
clear kbeam value # set all irradiance values to “value”
mult kbeam value # multiply all irradiance values to “value”
```

- `gaussian/cir kbeam pkflu r0 sgxp`
  makes supergaussian of peak irradiance “pkflu”,
  radius “r0”,
  supergaussian exponent “sgxp”



- start from random noise

```
clear 1 0 # zero array
noise 1 1 # random noise
```



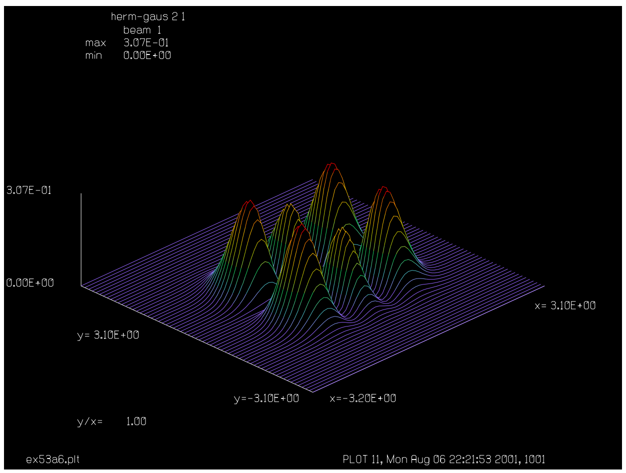
Hermite and Laguerre functions

- Hermite gaussian beams will develop naturally in stable resonators with no special effort required

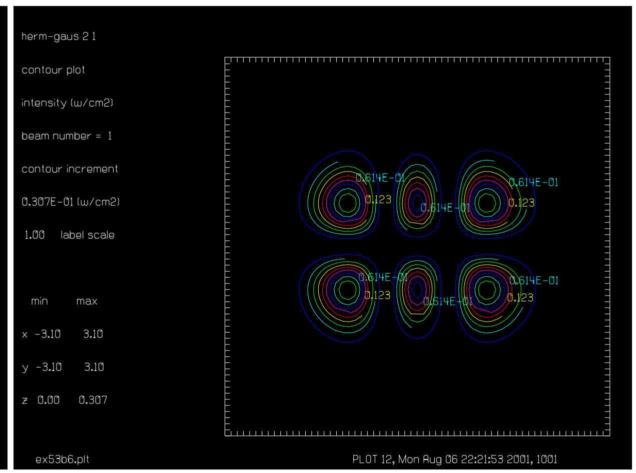
Hermite and Laguerre functions may be explicitly definedThe general polynomial form of the Hermite-gaussian functions is

$$u_n(x) = \left(\frac{2}{\pi}\right)^{1/4}\left(\frac{1}{2^n n! \omega_0}\right)^{1/2} H_n\left(\frac{\sqrt{2}x}{\omega(z)}\right) \exp\left[-\frac{x^2}{\omega(z)^2}\right] \tag{3.1}$$

where n is the order of the polynomial, w_0 is a waist radius parameter similar to the gaussian beam and $H_n(x)$ are the Hermite functions. The two-dimensional functions may be described by multiplying two one-dimensional functions. **x**



HG 2,1.



HG 2,1.

Donut mode consisting of two orthogonal Hermite modes: donut.inp

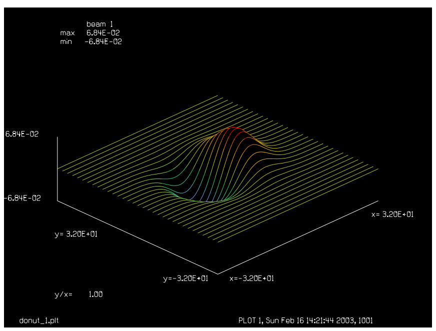
Donut mode: **donut.inp**

```
array/s 1 128 128 1                # make a polarized beam
nbeam 2                            # make another polarized beam
hermite/con 1 1 10 10 1 0
plot/w donut_1.plt
plot/l/r 1 xrad=32
pause
hermite/con 2 1 10 10 0 1
plot/w donut_2.plt
plot/l/r 2 xrad=32
pause
jones/set ar=0 br=0 cr=1 dr=0
jones/mult 2
set/density 16 16                  # 16 x 16 elements in plot
set/window/abs -32 32 -32 32       # set plot window for plot/ell
title horizontal mode
plot/w donut_3.plt
plot/ell 1
pause
title vertical mode
plot/w donut_4.plt
plot/ell 2
pause
add/coh/con 1 2                    # coherent addition
nbeam 1                            # back to just one beam
title composite mode
```

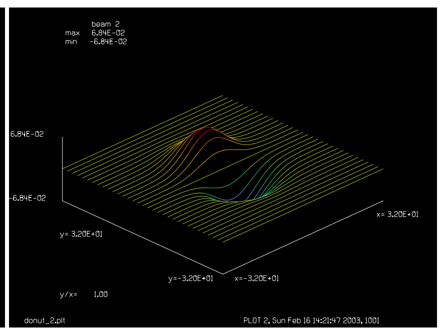
Donut mode (cont'd): donut.inp

Donut mode: **donut.inp** (cont'd)

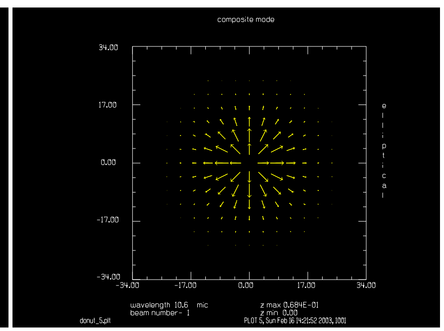
```
plot/w donut_5.plt
plot/ell 1
pause
plot/w donut_6.plt
plot/l xrad=32                  # plot/intensity
fitgeo 1                        # measure mean radius
zbound                        # Rayleigh width is 2.96e5 for 10.6 micron light
```



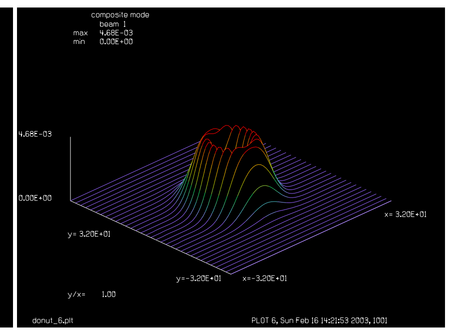
horizontal mode



vertical mode



donut polarization

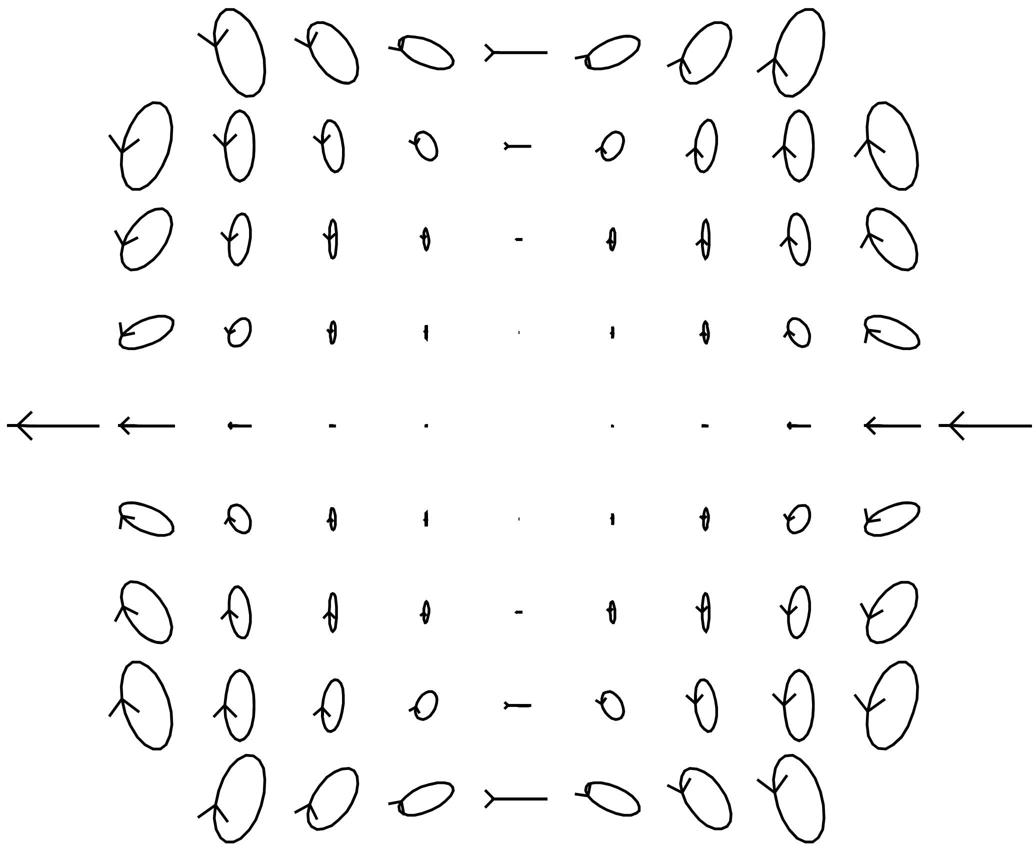


donut irradiance

Plots of elliptical polarization

- The polarization plot is found by the locus of points satisfying

$$\Delta x = \mathrm{Re}[E_x e^{-j2\pi\omega t}], \ \Delta y = \mathrm{Re}[E_y e^{-j2\pi\omega t}] \tag{3.2}$$



From Ex41, Effect of spatial filter on polarization—component orthogonal to output.

Review

- Does it matter whether we propagate the HG(1,0) and HG(0,1) modes separately before coherent addition?
- Write a program to measure the mean radius for these two cases
  - case 1: sum HG(1,0) and HG(0,1), then propagate 1e5 cm
  - case 2: propagate HG(1,0) and HG(0,1) separately, then coherent sum
  - use FITGEO to measure mean radius

Beam train analysis

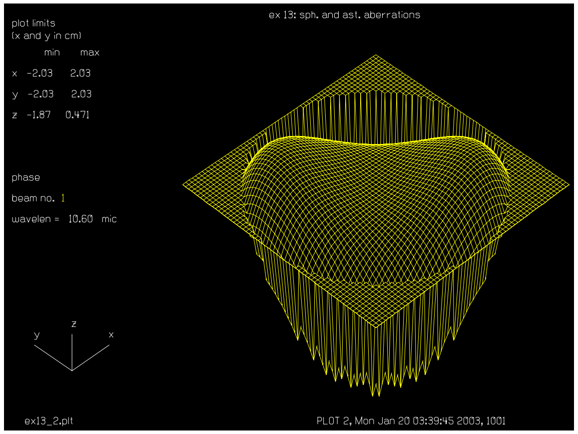
- beam trains consist of:
  - propagation
  - lens and mirrors
  - apertures
  - aberration
  - performance measures

Aberration

- aberration comes in many forms:
  - Seidel aberrations (common 3rd order aberrations)
  - Zernike aberrations (a complete, orthogonal set)
  - linear and circular phase gratings
  - smoothed random aberration (simulates “typical” component aberration)
  - unsmoothed random aberration (diffuser plate)
  - diffractive phase plates for far-field image synthesis
  - atmospheric aberration (Kolmogorov aberration)
  - thermal blooming (high power CW beams)
  - finite-element thermal distortion of components
  - kinoforms
  - binary phase plates
  - holographic elements — thin and volume holograms
  - lens and mirror arrays

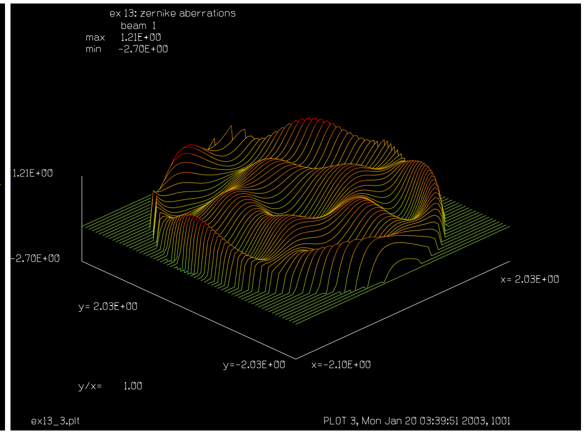
Miscellaneous aberrations

spherical aberration and astigmatism



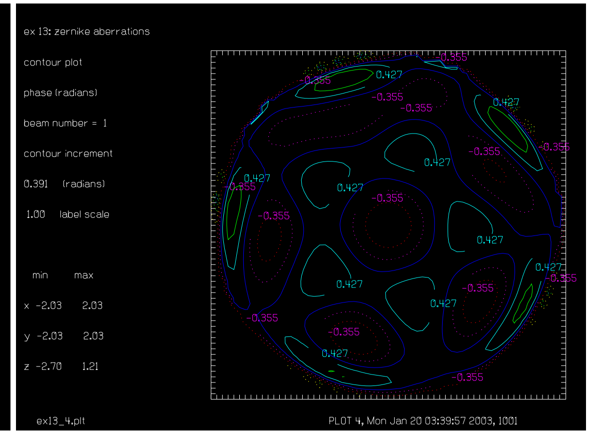

```
variab/dec/int Beam
Beam = 1; Ewav = .3; Azdeg =-45
abr/sph Beam Ewav
abr/ast Beam Ewav Azdeg
```

high order Zernikes



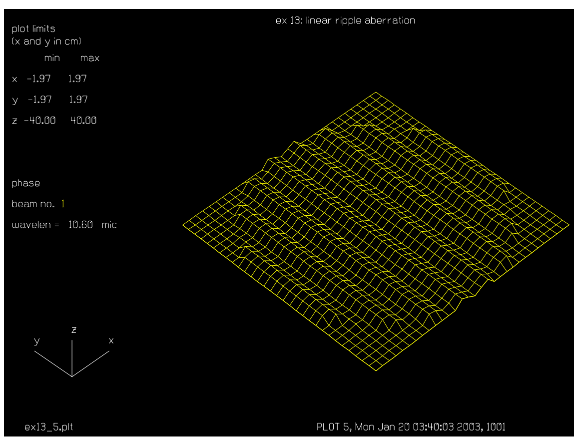

```
z = 1./-2/pi # in radians
abr/zern/rad 1 8 [-1.1*z]
abr/zern/cos 1 9 5 [1.2*z]
abr/zern/sin 1 8 6 [0.8*z]
```

high order Zernikes



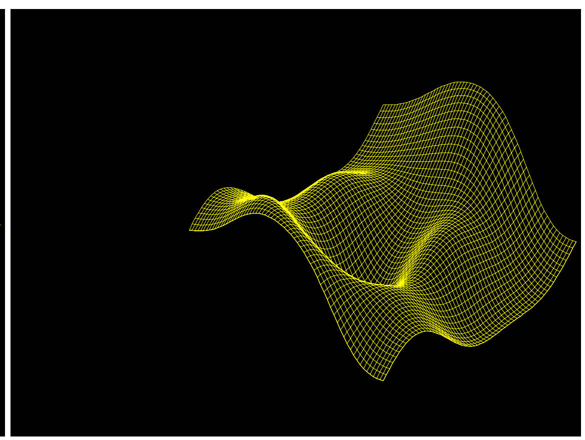

```
set/density 64
plot/contour/phase
```

linear phase grating



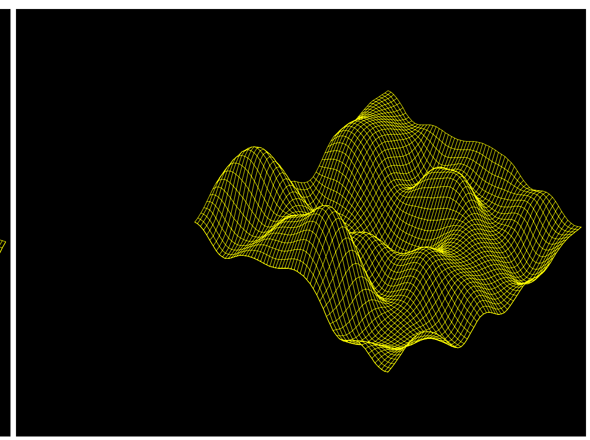

```
c Ibeams Ewav Wnbr Azdeg Phi
abr/lrip 1 3 4 90 90
```

smoothed random



```
phase/ran 1 .5 .15 is=1
```

smoothed random



```
phase/screen 1 .3 12
```

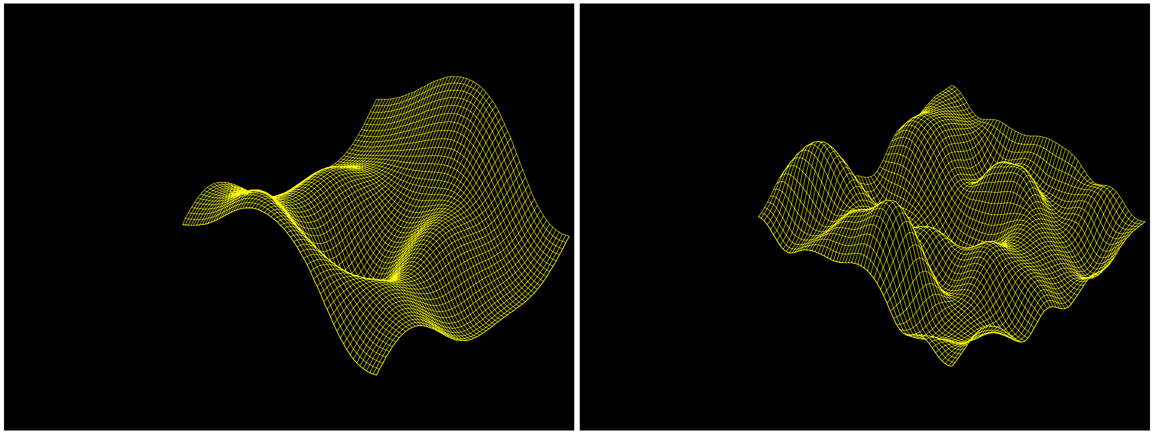
Creating random aberration

- random wavefronts simulate aberration that is defined by
  - from specifications in terms of rms wavefront error
  - from knowledge of “typical” components
- random wavefronts characterized
  - by rms error (standard deviation)

$$\sigma = \sqrt{\frac{\iint W(x,y)^2 dxdy}{\iint dxdy} - \left(\frac{\iint W(x,y)dxdy}{\iint dxdy}\right)^2} \qquad (4.1)$$

  - by autocorrelation width (typical width of constant phase region)

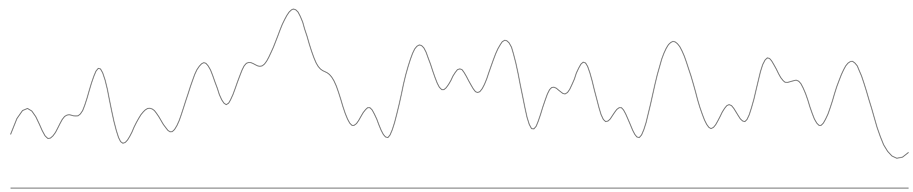
wide autocorrelation width   narrow autocorrelation width



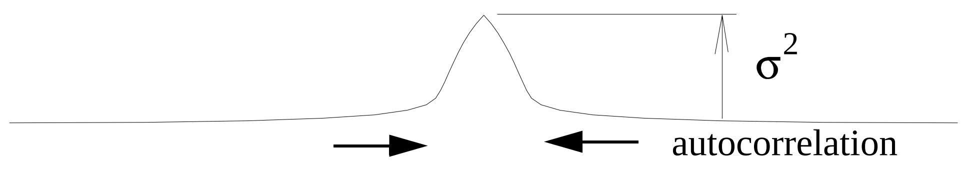
Autocorrelation function and autocorrelation diameter (unbounded speckle)

$$R(\Delta x, \Delta y) = \langle F(x, y) F(x + \Delta x, y + \Delta y) \rangle \text{ autocorrelation function} \tag{4.2}$$

unbounded speckle irradiance



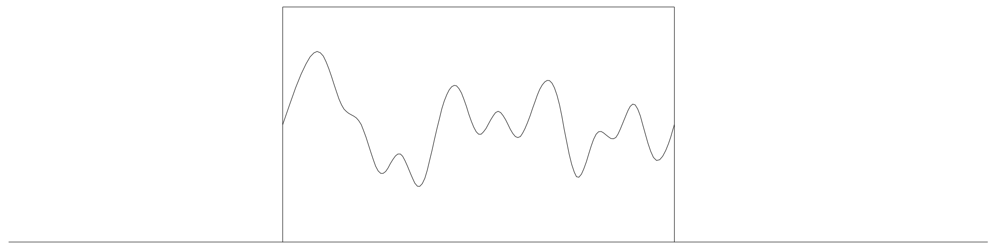
autocorrelation for unbounded speckle pattern



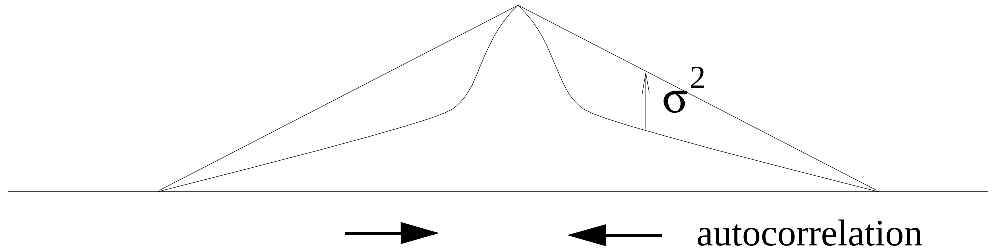
The autocorrelation function of an unbounded speckle pattern has a DC level determined by σ^2, the standard deviation of irradiance nonuniformity and a bump in the center which is determined by the typical speckle size.

Autocorrelation function and autocorrelation diameter (with aperture)

speckle irradiance bounded by a clear aperture



autocorrelation for bounded speckle pattern



For a finite size clear aperture the irradiance nonuniformity manifests itself as a drop from the autocorrelation of the uniformly filled aperture.

Constructing a smoothed random wavefront with specified autocorrelation width

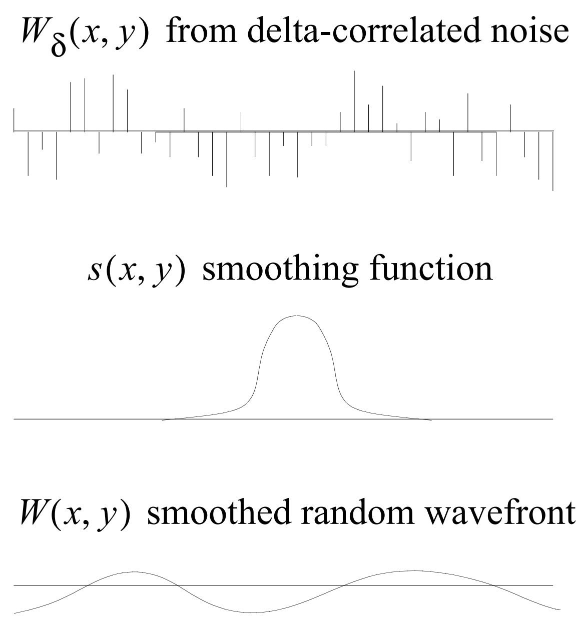
- construct random noise pattern for $W_\delta(x, y)$ (delta correlated)

$$R_\Delta(\Delta x, \Delta y) = \sigma^2 \delta(\Delta x, \Delta y) \text{ (autocorrelation function is delta function)} \tag{4.3}$$

where σ^2 is the wavefront variance.

- smooth with smoothing function $s(x, y)$

$$W(x, y) = W_\delta(x, y) ** s(x, y) \text{ (smoothed wavefront)} \tag{4.4}$$



Review

- pixel-to-pixel phase change must be less than π
- consider focus error of form $w(h) = w_0 h^2$
- wavefront slope is $\frac{dw}{dz} = 2w_0 h \rightarrow \Delta w = 2w_0 h \Delta x$
- Consider:

```
array/s 1 256
units 1 .01
abr/focus 1 Ewav rn=1 # Ewav is the wavefront error in waves
                      # at radius rn
```

- What value of Ewav should you choose to alias the phase at a radius of 1 cm?
- Use `plot/l/w`, `plot/i/w`, or `plot/x/w` to display phase.
- How do you recognize aliased phase from the plots?
- GLAD uses a 2π unwrapping algorithm to transform complex amplitude into continuous phase. The unwrapping algorithm fails when the phase becomes aliased so it is a good check.
- Try an array of 256 and units of .005 cm. At what radius does the phase alias now?

(4.5)

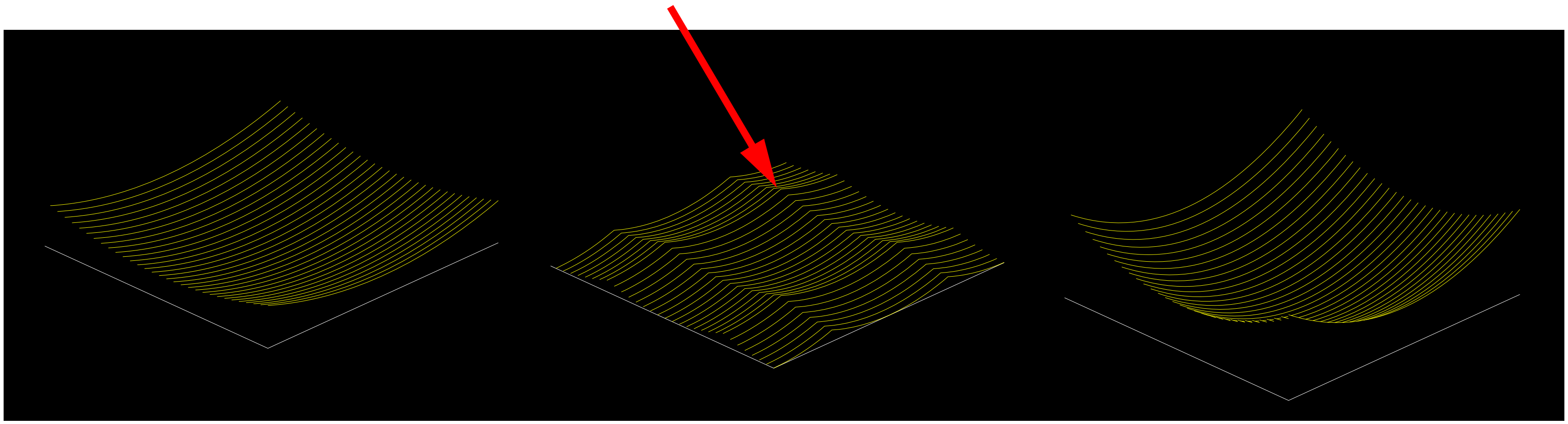
Aliasing of wavefront error

- pixel-to-pixel phase change must be less than π (1/2 wave per pixel)
- How do you recognize aliased phase from plots of the wavefront?

wavefront, 128 x 128

aliased point at which $\frac{d}{dr}W(r) = \frac{1}{2}$ waves per pixel

wavefront, 256 x 256



$W(r) = \left.\frac{1}{256}r^2\right|_{r=64} = 16\lambda$

$\frac{d}{dr}W(r) = \left.\frac{r}{128}\right|_{r=64} = \frac{\lambda}{2}$,

1/2 waves per pixel

$W(r) = \left.\frac{1}{128}r^2\right|_{r=64} = 32\lambda$

$W(r) = \left.\frac{1}{128}r^2\right|_{r=64} = 32\lambda$

$\frac{d}{dr}W(r) = \left.\frac{r}{128}\right|_{r=64} = \frac{\lambda}{2}$,

1/2 waves per pixel

Laying out the beam train

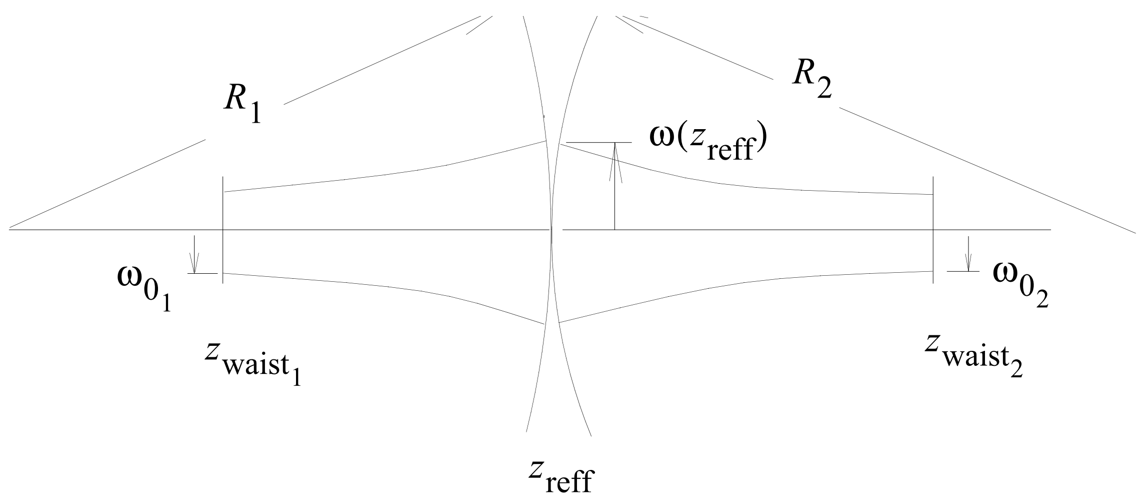
- geometrical optics is precise but not accurate
  - precise — numerical errors are low
  - not accurate — actual physics not well described
- physical optics is accurate but not precise
  - accurate — physics is well described
  - not precise — numerical errors are often noticeable and accumulate with each added component
- minimize the number of operations to improve both speed and accuracy
  - “unfold” the system to eliminate fold mirrors
  - lump closely spaced apertures
  - lump gain regions into gain sheets
  - where possible lump elementary elements into optical groups
- use idealized elements where possible
- begin model development with a simplified view
- add complexity in gradual stages
  - facilitates checking the model
  - aids in understanding the phenomenology

Lenses and mirrors

- many systems can be accurately modeled by considering optical components as idealized lenses and mirrors
- many systems can be “unfolded” removing folding flat mirrors
- idealized optics

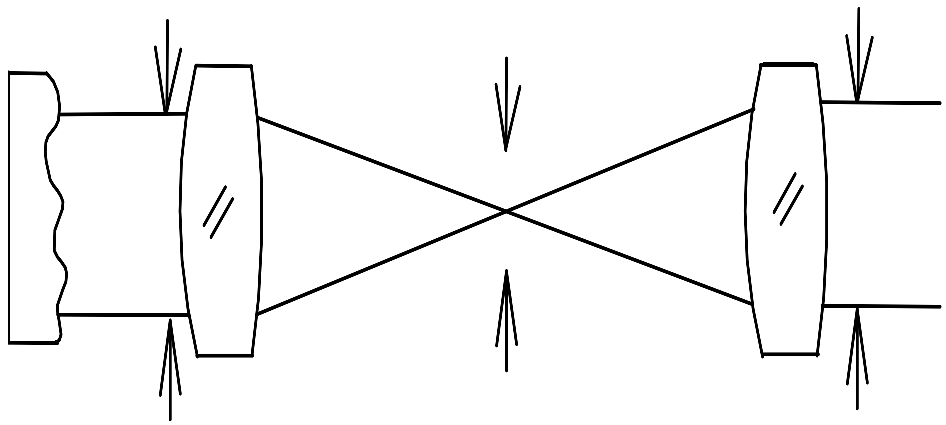
```
lens ibeams fl # idealized lens specified by focal length
mirror ibeams rad # idealized mirror
```

- aberrations may be explicitly added
- a new surrogate gaussian beam is calculated after each component with optical power



Spatial filter

- parts list: lenses, apertures, propagation
- aberration is removed by pinhole filter at focus plane and following aperture



starting phase

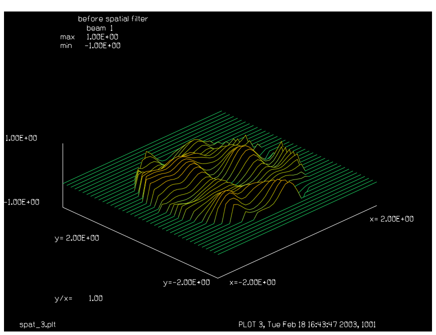


image before pinhole

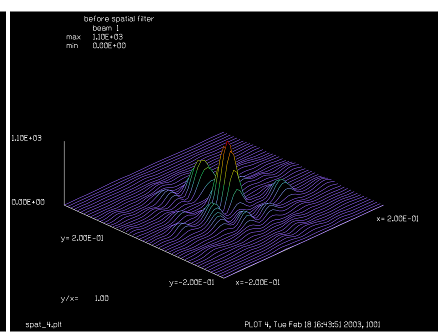
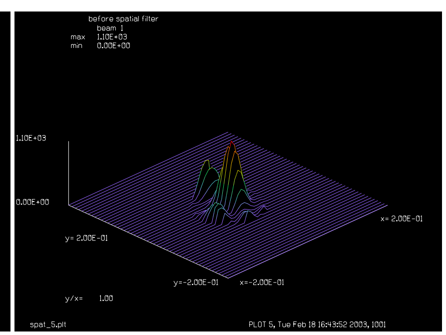
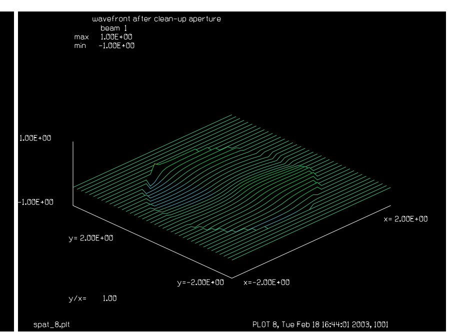


image after pinhole



final phase



Command file for simple spatial filter: spatial1.inp

```
array/s 1 128
View = 2.       # plot view in pupil
Apt = 1.5       # clear aperture in expanded beam
units 1 .1
phase/screen 1 .2 .25
title full field phase screen
plot/w spat_1.plt
plot/l/w
echo/on
# variance before aperture is exactly 0.2
variance
strehl
echo/off
clap/cir/con 1 Apt                  # aperture at end of spatial filter
plot/w spat_2.plt
title before spatial filter
plot/l xrad=View
pause
plot/w spat_3.plt
plot/l/w xrad=View max=1 min=-1
pause
echo/on
# note that the variance is not exactly 0.2 after aperture
variance
pause
echo/off
```

Command file for spatial filter (cont'd): spatial1.inp

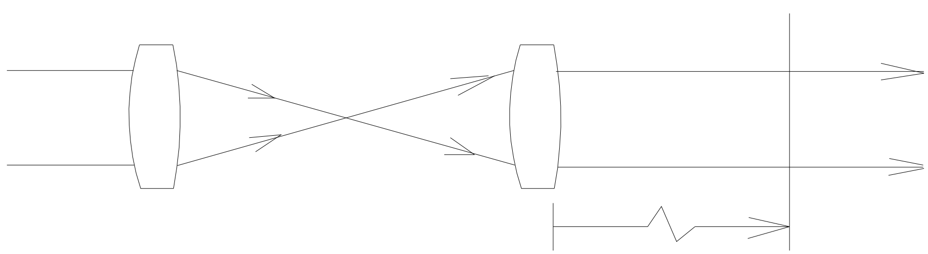
```
# spatial filter
lens 1 100
prop 100
plot/w spat_4.plt
plot/l xrad=.2
pause
clap/cir/con 1 .07
plot/w spat_5.plt
plot/l xrad=.2
pause
prop 100
lens 1 100
title after spatial filter
plot/w spat_6.plt
plot/l xrad=View
pause
plot/w spat_7.plt
plot/l/w xrad=View max=1 min=-1
variance
strehl
pause
clap/cir/con 1 Apt
echo/on
# variance and strehl after clean-up aperture
variance
strehl
```

Command file for spatial filter (cont'd): spatial1.inp

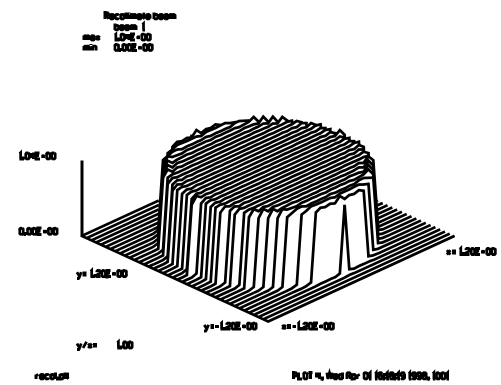
```
plot/w spat_8.plt
title wavefront after clean-up aperture
plot/l/w xrad=View max=1 min=-1
```

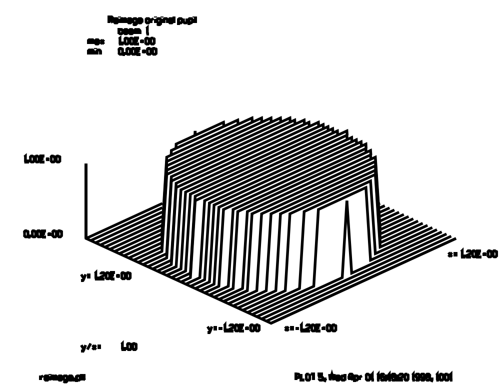
Recollimate the beam: focus2.inp

- Add recollimation lens to create a 1:1 telescope, f = 20
- add a plot to show distribution at second lens
- why is the initial pupil not recreated perfectly after the second lens?
- where is the pupil reimaged “perfectly”
- work from focus.inp to generate focus2.inp



what distance to reimage?





Making the telescope into a spatial filter: focus3.inp

- Add .75 wave of spherical aberration

```
abr/sph 1 .75 rnorm=Apt
Measure Strehl ratio and display wavefront error
strehl 1
plot/l/wave 1 xrad=1.2
```

Observe and plot image irradiance

Propagate to lens and measure Strehl ratio

Work from focus2.inp to create focus3.inp

  - Delete propagation to reimaged pupil. Why is this OK?
  - Delete second lens. Why is this OK?
  - Add aperture of size Apt at plane of second lens

- Impose aperture (pinhole filter) at image plane. Try radius of .001. What is the Strehl ratio after filtering?

Searching for best pinhole size using a macro: focus4.inp

- Finding the best pinhole size by trial and error
- We will use trial and error method to find aperture size that gives Strehl ratio of 0.8 for our 0.75 waves of spherical aperture
- Start from focus3.inp to build focus4.inp
- Encapsulate problem into macro for repetitive solution
- Make a copy of the beam at focus to avoid recalculating front end

```
nbeam 2 data
copy 1 2
macro/def search/o
 zreff 1 Focal_length
 copy 2 1
 clap/c/c 1 Pinhole
 plot/l 1 xrad=.002 # expand far-field
 prop Focal_length
 clap/cir/con 1 Apt
 strehl 1
macro/end
```

- Choose a value for Pinhole. Try Pinhole = 0.001
- Verify macro gives same value with repeated runs for Pinhole fixed
- Vary value of Pinhole and search for Strehl = 0.8

A more methodical search using udata: focus5.inp

Scan a range of pinhole values and plot with `udata`

- Add pass counter

```
variab/dec/int pass
pass = pass + 1
```

- Add increment to Pinhole with each pass

```
DeltaR = .0001
Pinhole = Pinhole + DeltaR
```

- Set variable to Strehl ratio

```
variab/set Strehl 1 strehl
```

- Set data into udata by row number, x-value, and up to 12 y-values

```
udata/set pass Pinhole Strehl
Plot udata
plot/udata min=0 max=1
```

- Begin with focus4.inp and build focus5.inp to plot Strehl ratio vs. pinhole size
- Make side-by-side plots of clipped image plane and `plot/udata`

Macro for generating Strehl vs. pinhole plot

```
Ntimes = 50
Pinhole_max = Ntimes*DeltaR
macro/def search/o
 Pinhole = Pinhole + DeltaR
 pass = pass + 1
 zreff 1 Focal_length
 copy 2 1
 clap/c/c 1 Pinhole
 plot/w plot1.plt 10 10 400 300
 plot/l 1 xrad=Pinhole_max # expand far-field
 prop Focal_length
 clap/cir/con 1 Apt
 strehl 1
 variab/set Strehl 1 streh
 udata/set pass Pinhole Strehl
 plot/w plot2.plt 410 10 400 300
 plot/udata min=0 max=1 left=0. right=Pinhole_max
macro/end
Pinhole = 0.
pass = 1
udata/set pass Pinhole 1.
macro search/Ntimes
```

Automatic search using optimization: focus6.inp

A simple optimization to find pinhole size giving Strehl ratio of 0.8

- Simplify macro to just calculate Strehl ratio vs. pinhole size

```
macro/def search/o
 zreff 1 Focal_length; copy 2 1
 clap/c/c 1 Pinhole; prop Focal_length
 clap/cir/con 1 Apt; variab/set Strehl 1 strehl
macro/end
```

- Add variable table and target table

```
opt/var/add Pinhole .0001# specify variable and increment of change
opt/tar/add Strehl .8      # target variable and value
```

- Define macro of system to be optimized

```
opt/name search            # specify name of system macro
```

- Set damping to 10 for this problem

```
opt/damp/mul 10            # increase damping for this problem
```

- Guess at Pinhole size

```
Pinhole = 0.001
```

- Run optimization process 10 times

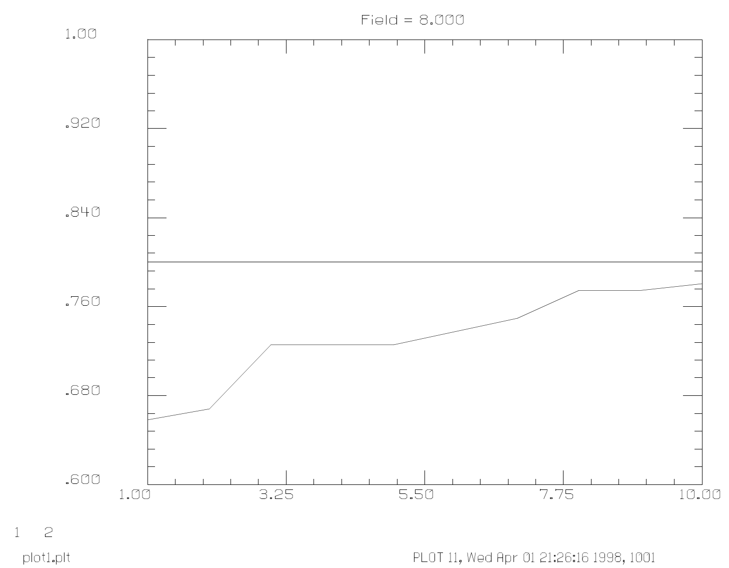
```
opt/run 10                 # run optimization 10 times
```

Adding progress plot to optimization: focus7.inp

Put optimization into macro with step counter and udata/set

```
macro/def opt/o
 pass = pass + 1
 opt/run 1
 udata/set pass pass Strehl .8
 plot/udata first=1 last=2 min=.6 max=1
macro/end
pass = 0
macro opt/10
```

- Try changing Field to 16 to get better resolution in far-field



Why are there jumps in the plot?

Wavefront variance and Strehl ratio

$$\sigma^2 = \frac{\iint W(x,y)^2 \mathrm{d}x\mathrm{d}y}{\iint \mathrm{d}x\mathrm{d}y} - \left(\frac{\iint W(x,y)\mathrm{d}x\mathrm{d}y}{\iint \mathrm{d}x\mathrm{d}y}\right)^2 \quad (4.6)$$

- Strehl ratio

The far-field intensity is $I(x,y)_{\mathrm{aberr}} = \frac{1}{\lambda^2 f^2}\left|\iint a(x,y)e^{j2\pi(x\xi + y\eta)}\mathrm{d}x\mathrm{d}y\right|^2$ (4.7)

The far-field intensity of the same intensity distribution, without aberrations is

$$I(x,y)_{\mathrm{noaberr}} = \frac{1}{\lambda^2 f^2}\left(\iint |a(x,y)| e^{j2\pi(x\xi + y\eta)}\mathrm{d}x\mathrm{d}y\right)^2 \quad (4.8)$$

Evaluating these at $\xi = 0$ and $\eta = 0$ the Fresnel kernel disappears, and we have

$$\text{Strehl ratio} = \frac{I(0,0)_{\mathrm{aberr}}}{I(0,0)_{\mathrm{noaberr}}} = \frac{\left|\iint A(x,y)\mathrm{d}x\mathrm{d}y\right|^2}{\left(\iint |A(x,y)|\mathrm{d}x\mathrm{d}y\right)^2} \quad (4.9)$$

- A well-known relationship between Strehl ratio and wavefront variance is

$$SR \approx e^{-4\pi^2\sigma^2} \text{ and } \sigma^2 \approx -\frac{\ln(SR)}{4\pi^2} \quad (4.10)$$

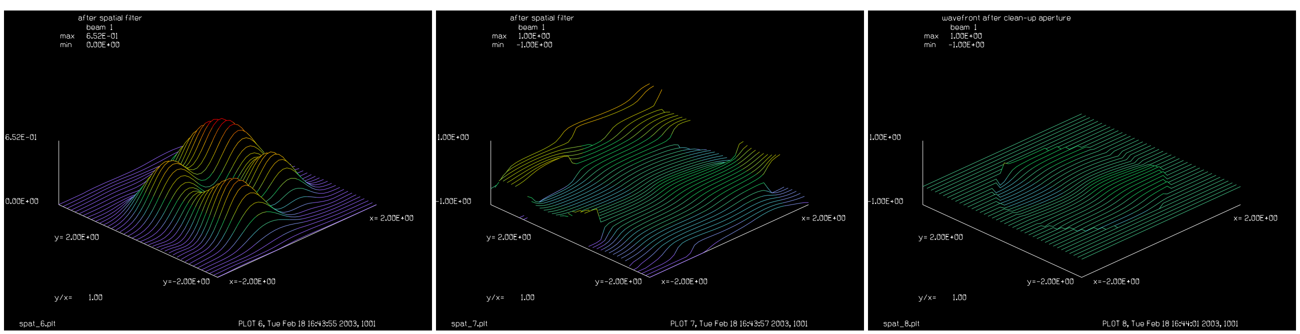
Working with the spatial filter model

- Verify the system will reimage the starting distribution if:
  - the pinhole aperture is “commented out”
  - the aperture after the second lens is commented out

Where is the image of the initial distribution relative to the second lens?
(hint: first order optical principles will help)
Verify the position of the image after the second lens.

- Why are the values of variance measured before and after the first aperture different?
- Why is the phase so disrupted at the edge just after the second lens?
- Does it matter that the phase is disrupted at the edge? If so why?, If not why not?
- Is Strehl ratio a better indicator of performance after the second lens? If so why

intensity after L2 phase before Apt 2 phase after Apt 2



Plotting Strehl ratio vs. pinhole size: spatial2.inp

- determine the variation of strehl ratio vs. size of the pinhole
- use a macro and udata
- loop over the system 30 times
- calculate and store Strehl ratio for each step, plot at the end
- note that the same random seed is used each time

Strehl ratio vs. pinhole size (spatial2.inp)

```
View = 2.       # plot view in pupil
Apt = 1.5       # clear aperture in expanded beam
variable/dec/int count
macro/def step/o
   count = count + 1
   array/s 1 128
   zreff 1 1
   units 1 .1
   phase/screen 1 .2 .25 3
   title full field phase screen
   clap/cir/con 1 Apt                    # aperture at end of spatial filter
   energy/norm 1 1
   lens 1 100
   prop 100
   Pinhole = .01*count
   clap/cir/con 1 Pinhole
```

Plotting Strehl ratio vs. pinhole size (cont'd): spatial2.inp

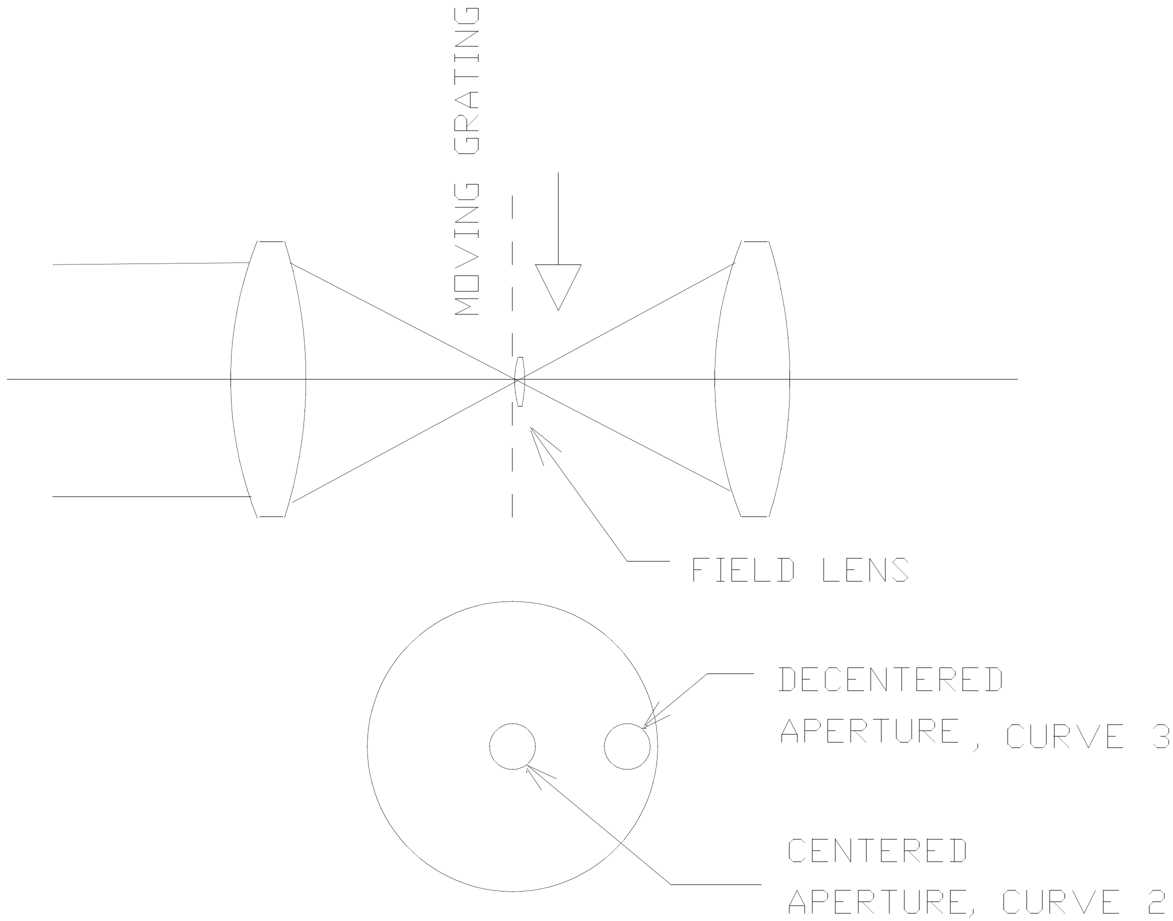
```
   prop 100
   lens 1 100
   strehl
   clap/cir/con 1 Apt
   variab/set Strehl 1 strehl
   udata/set count Pinhole Strehl
macro/end
macro/run step/30
udata/list
title Strehl ratio vs. pinhole size
plot/w spat_9.plt
plot/udata min=0 max=1.
```

Review

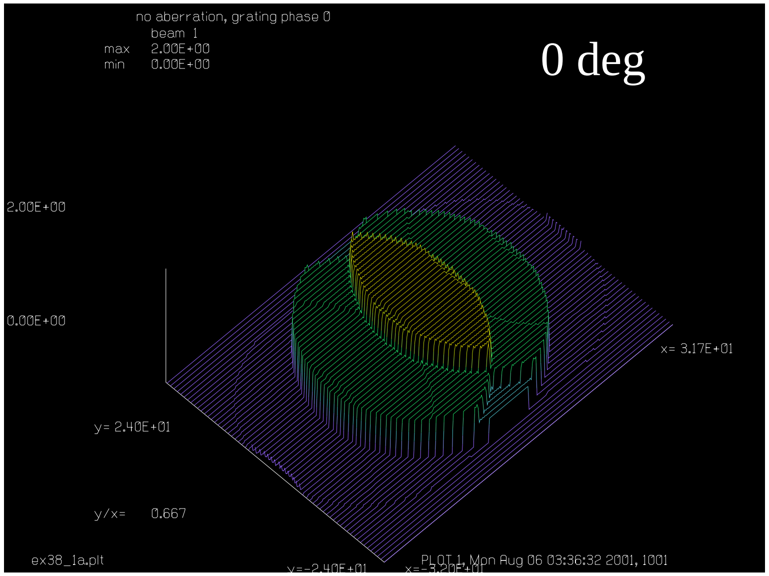
- Calculate the energy per step in addition to the Strehl ratio.
- Plot both Strehl ratio and energy at the same time.
- Where is the ‘best” trade off of Strehl ratio and energy transmission?
- What happens if different random seeds are used for each step?

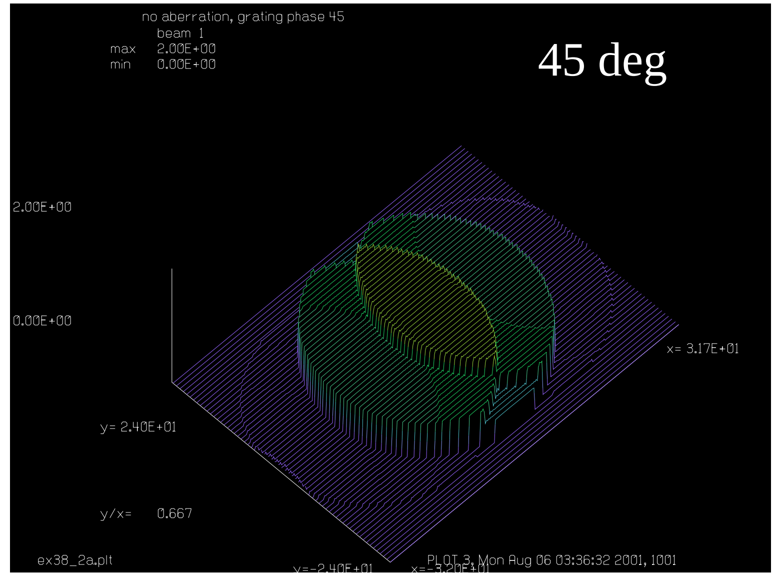
Lateral shearing interferometer: ex38x.inp

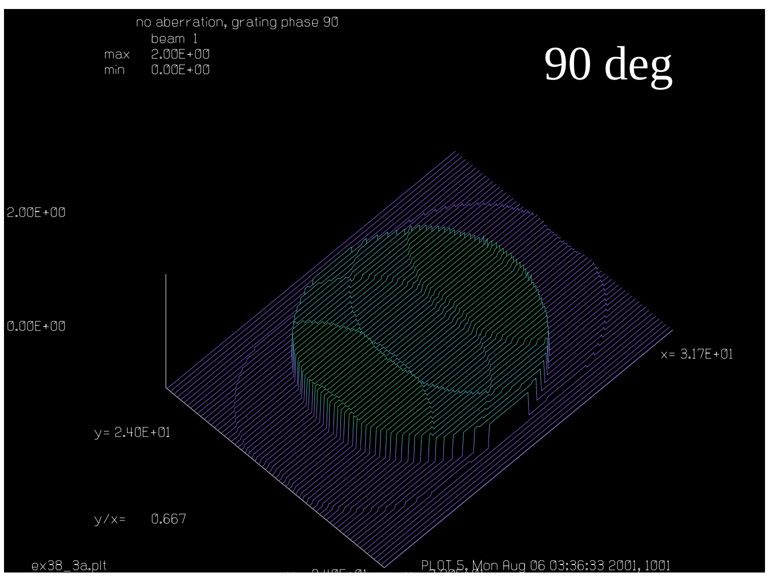
- a moving Ronchi ruling generates +1 and -1 diffraction orders which interfere with the zero order to make a shearing interferometer.

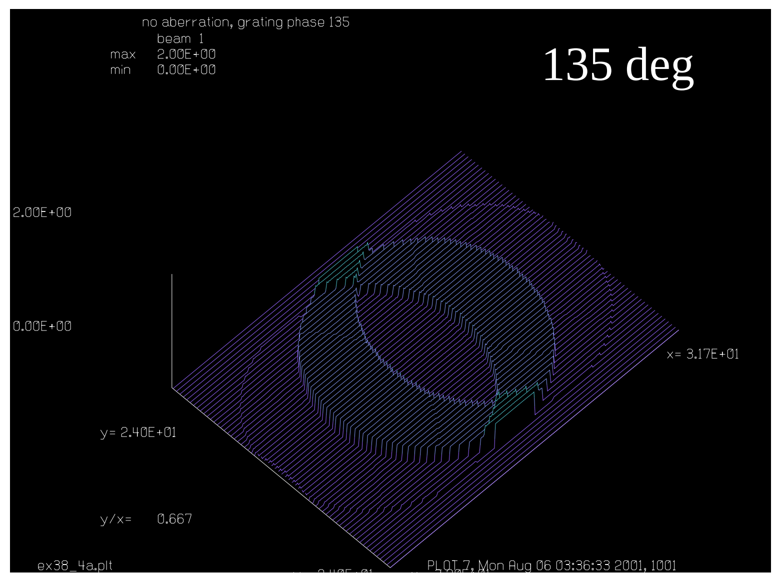


Ronchi ruling moves fractions of once cycle past the image: ex38x.inp

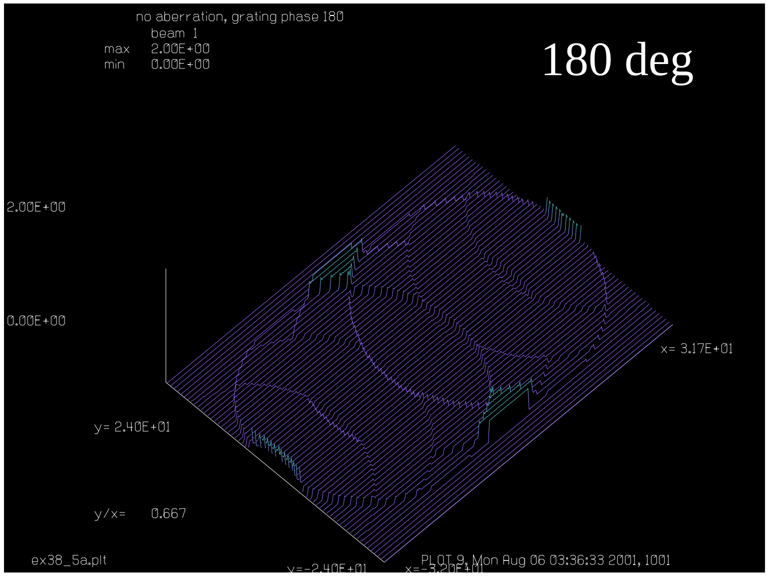


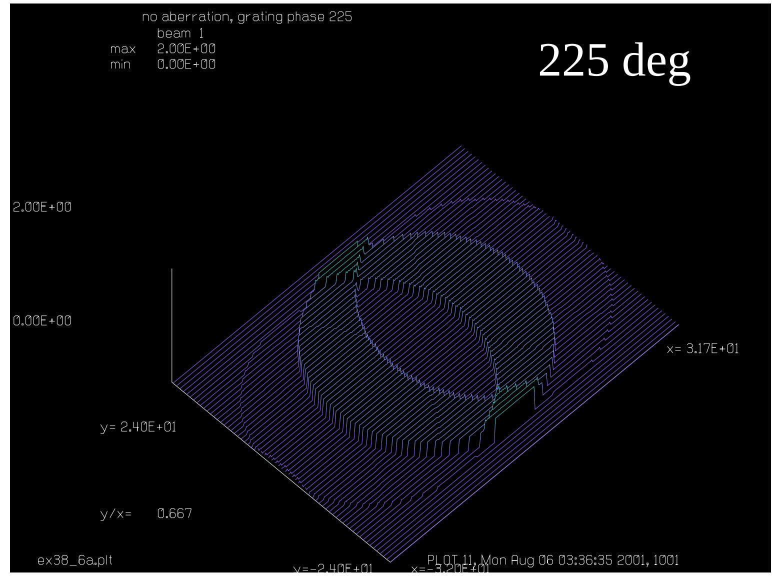


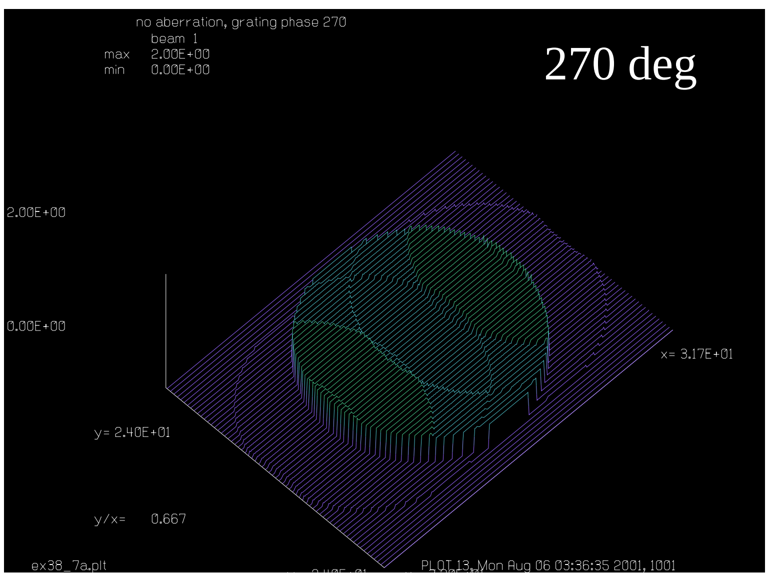


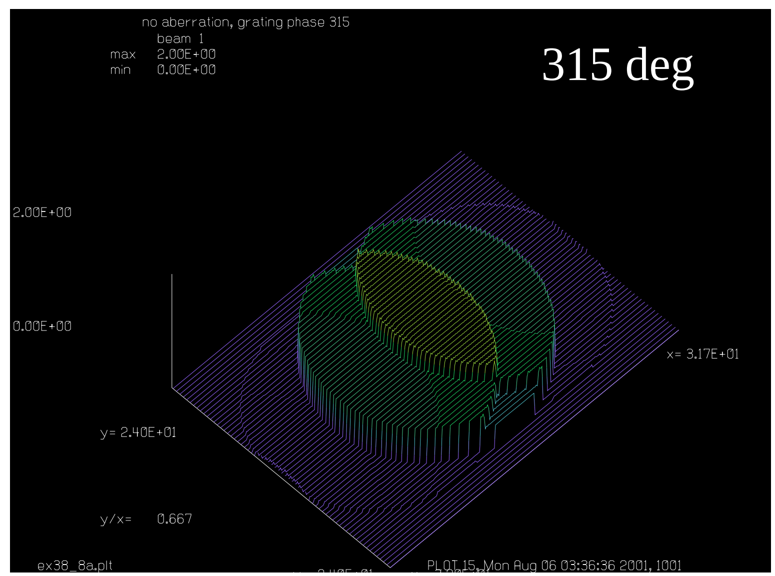


Ronchi ruling moves fractions of once cycle beyond (cont'd): ex38x.inp









Command file for shearing interferometer: ex38x.inp

```
c## ex38
c
c  Example 38:  Shearing interferometer
c
c  This example illustrates the modeling of a shearing interferometer by
c  using a moving amplitude grating.  A pupil of 40 cm diameter
c  is brought to a focus with a 100 cm lens.  An amplitude grating is moved
c  past the image, which causes a modulation in the reimaged pupil.
c  We calculate the total energy and the energy in a 2 cm radius circle
c  at the center of the aperture and in a 2 cm radius circle at the edge
c  aperture.  The total energy and energy in isolated
c  areas is, of course, always in phase.  The amplitude grating is moved
c  in 45 degree phase increments over one full cycle.
c
variab/dec/int pass phase
nbeam 3                       # Set up 3 beams, only 1 is active
array/s 0 256                 # Use 256 X 256 array
pass = 0                      # Initialize pass counter
phase = -45                   # Initialize grating phase
units 0 .25                   # Set units
clap/c/c 1 20                 # 40 cm diameter aperture
c abr/focus 1 6.5               # (insert this command to see aberration)
energy
lens 1 100                    # lens of 100 cm focal length
dist 100 1                    # propagate to focus
copy 1 2                      # save distribution in Beam 2
```

Command file for shearing interferometer (cont'd): ex38x.inp

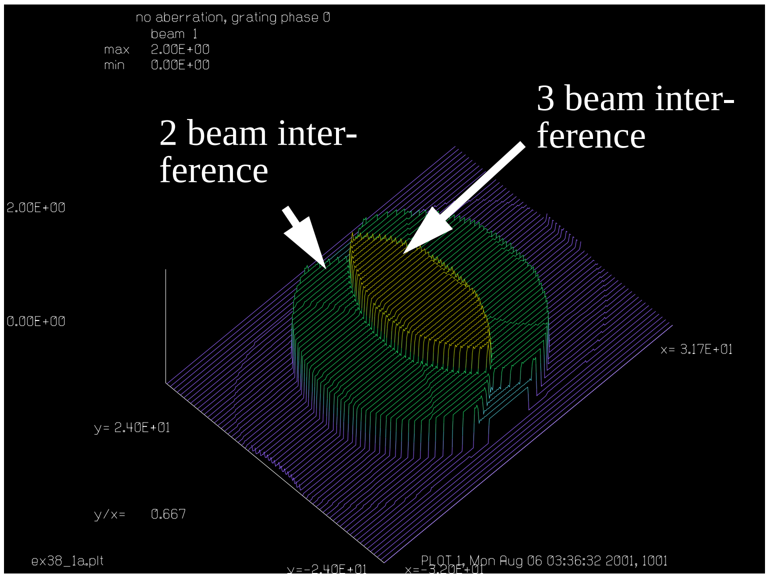
```
macro/define grat/o            # define macro
  pass = pass + 1                # increment pass counter
  phase = phase + 45             # increment grating phase
  copy 2 1                       # restore image distribution
  grat 1 8.281e-3 phi=phase      # grating of period = .0082813 cm
  variab/set energy1 1 energy    # Set energy1 to energy in Beam 1
  lens 1 50                      # field lens to reimage pupil
  dist 100 1                     # propagate to pupil image
  plot/watch ex38_@passa.plt     # set plot file name
  plot/l 1 xr=36 yr=24 ns=64 max=2 thet=40
  plot/watch ex38_@passb.plt
  plot/x/i 1 fmax=2
  copy 1 3                       # make a copy of the pupil in Beam 3
  clap/c/c 1 .31                 # 2 cm radius aperture in pupil center
  clap/c/c 3 .31 xdec=8          # 2 cm radius aperture, edge of pupil
  variab/set energy2 1 energy
  variab/set energy3 3 energy
  energy2 = energy2*2000         # linear scaling for PLOT/UDATA
  energy3 = energy3*2000
  udata/set pass phase energy1 energy2 energy3
macro/end
title/format f 3 0
title no aberration, grating phase @phase
macro/run grat/9
title summary of energy vs. phase
plot/watch ex38_10.plt
```

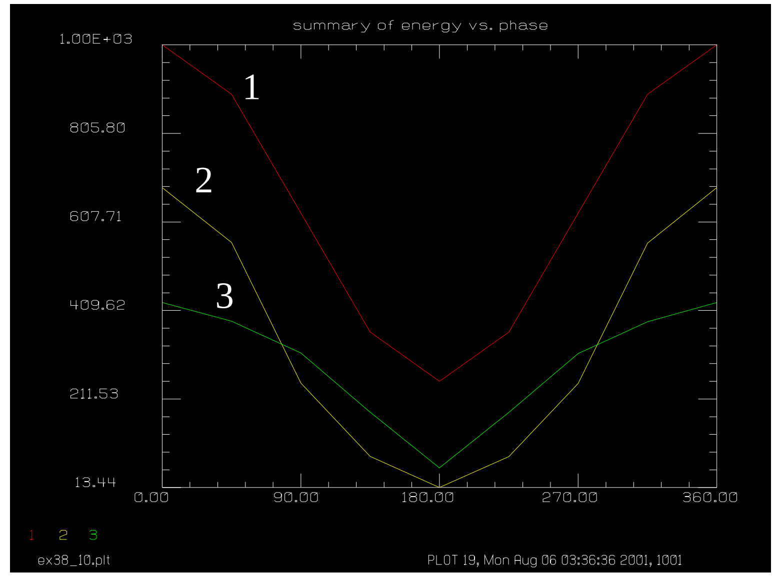
Command file for shearing interferometer (cont'd): ex38x.inp

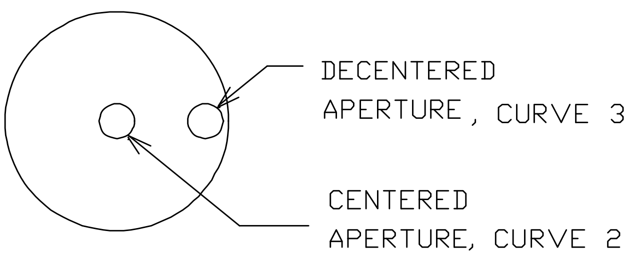
```
plot/udata 1 3
write/disk ex38.out/o
udata/list
end
```

Temporal waveform produced

- temporal waveforms for total energy, two beam, and three beam interference







Energy for the total aperture versus grating phase. Zero phase puts the peak of a cosine transmission pattern at the center of the image distribution. The energy in a small centered aperture (curve 2) and one at the edge of the aperture (curve 3) are also plotted. The energy from the smaller apertures is multiplied by 50 to make plotting easier. It can be seen that all energy curves are of the same frequency and phase.

Review (from Ex 38)

- What is the response of interferometer to astigmatism? Try adding a few waves?(Be sure the astigmatism varies in the x-direction)
- Increase the size of the arrays to 512×512. Remember to increase the memory for fastest speed. What value is best choice for the units?
- Are all three curves for temporal waveform cosine waves? Try increasing the number of optical cycles and use finer increments than 45 deg. to judge whether the waves are cosines.

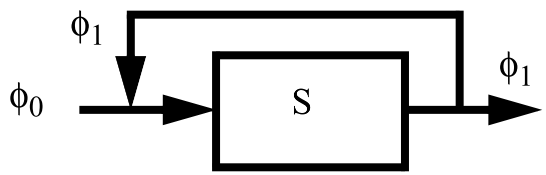
5. Resonators

Eigenfunctions and eigenvalues

- a resonator is a periodic operation

starting complex amplitude distribution: $\phi_0(x, y)$

after one pass: through the system S we have $\phi_1(x, y)$



$$\phi_1(x, y) = \mathbf{S}\phi_0(x, y) \tag{5.1}$$

- Eigenfunctions and eigenmodes

For special ψ functions called eigenfunctions,

$$\mathbf{S}\psi(x, y) = \lambda\psi(x, y) \tag{5.2}$$

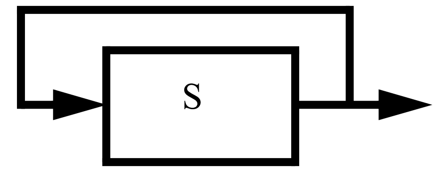
where λ is a scalar, complex coefficient called the eigenvalue

- eigenvalues may be complex, in bare cavity resonators (no gain) $|\lambda| < 1$
- eigenmodes are orthogonal and a complete set
- any arbitrary function may be represented as a summation of eigenmodes

$$\phi_0(x, y) = \sum_{n=0}^{\infty} a_n \psi_n(x, y) \tag{5.3}$$

Lowest loss mode

- Start with any arbitrary distribution



$$\phi_0(x, y) = \sum_{n=0}^{\infty} a_n \psi_n(x, y) \text{ (sum of eigenvalues)} \quad (5.4)$$

The coefficients a_n are unique to define the fit to $\phi_0(x, y)$

- If λ_n are the eigenvalues for the n modes
  - after *m* round trips the distribution in the resonator is

$$\phi_m(x, y) = \sum_{n=0}^{\infty} \lambda_n^m a_n \psi_n(x, y) \quad (5.5)$$

After a suitable number of passes, the mode with the largest eigenvalue will be the dominant term.

$$\phi_m(x, y) \approx a_j \lambda_j^m \psi_j(x, y) \text{ (the lowest loss mode)} \quad (5.6)$$

- Finding the lowest loss mode by making many passes through the resonator is called
  - Fox-Li method
  - Power method

Stable resonators

- geometric rays are trapped
- less sensitive to misalignment
- Hermite modes are eigenfunctions
- useful for low gain media
- bare cavity analysis is often used but does not accurately predict laser properties
  - real lasers do not converge to a single mode

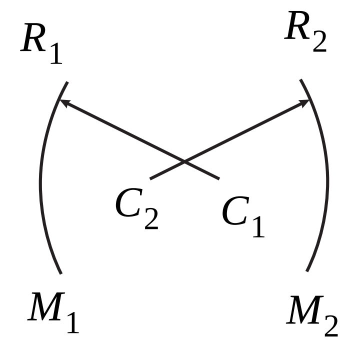
Modeling the stable resonator in GLAD

- numerical algorithms must be identical for each pass
- surrogate gaussian must, therefore, be identical for each pass
- use `resonator/eigen/test` command to force the surrogate gaussian to be an eigenmode of the paraxial resonator properties
- use `copy/con` to set starting distribution without changing surrogate gaussian beam

Elementary two-mirror systems

- two-mirror systems may be characterized simply by the "g" parameters

$$g_1 = 1 - \frac{L}{R_1},\; g_2 = 1 - \frac{L}{R_2} \tag{5.7}$$



- Using the g-parameters, the stability criterion is

$$0 < g_1 g_2 < 1 \tag{5.8}$$

Arbitrary systems described by ABCD

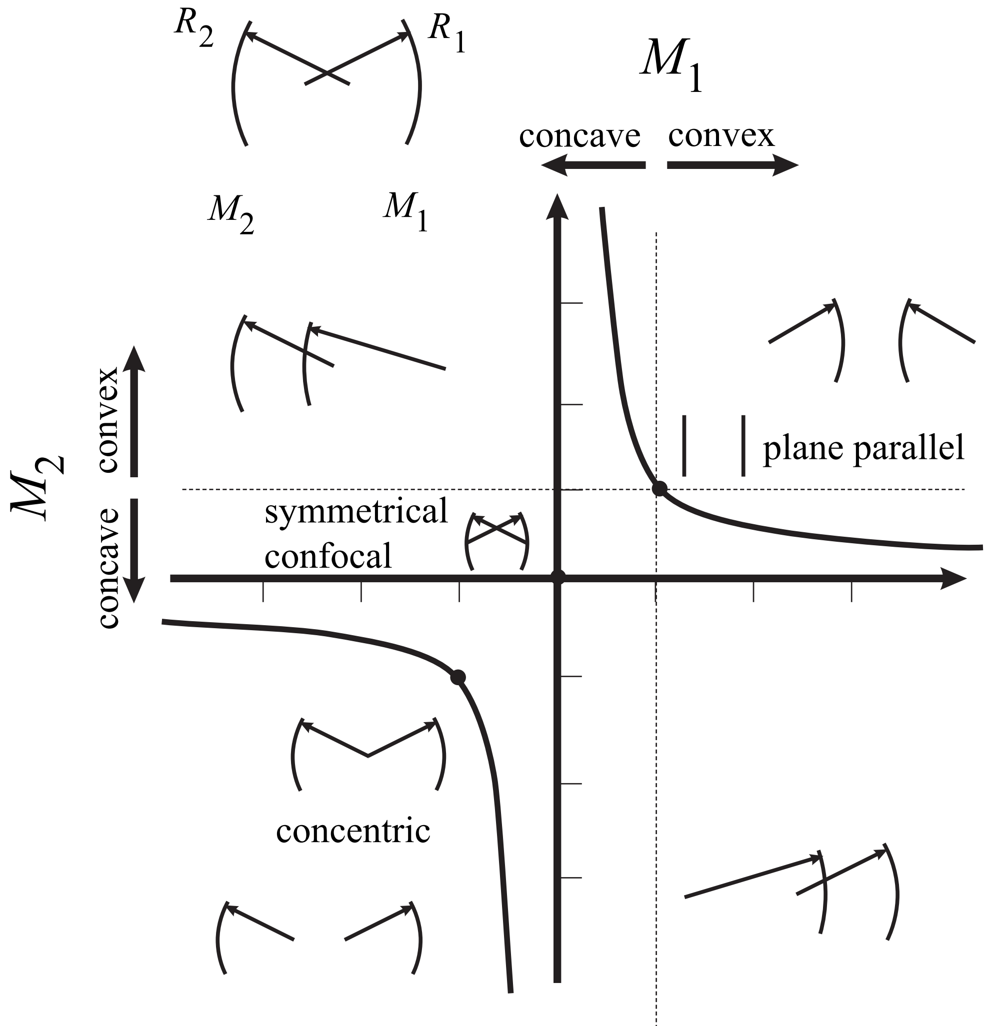
the eigenmode solution is a gaussian

$$\frac{1}{\tilde{q}} = \frac{D-A}{2B} \pm \frac{1}{B}\sqrt{\left(\frac{A+D}{2}\right)^2 - 1},\; R = \frac{2B}{D-A},\; \omega = \left(\frac{\lambda}{\pi}\right)^{1/2} \frac{|B|^{1/2}}{\left[1 - \left(\frac{A+D}{2}\right)^2\right]^{1/4}}, \tag{5.9}$$

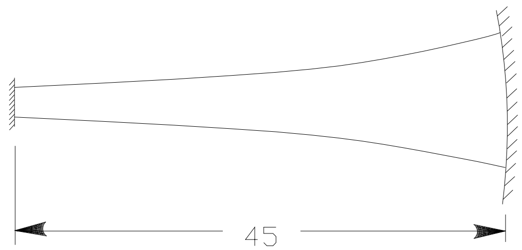
and the waist properties determined by,

$$z_{\text{waist}} = \frac{R}{1 + \left(\frac{\lambda R}{\pi\omega^2}\right)^2},\; \omega_0 = \frac{\omega}{\sqrt{1 + \left(\frac{\pi\omega^2}{\lambda R}\right)^2}} \tag{5.10}$$

The classical stability map for g_1 and g_2



Half symmetric stable cavity resonator: ex33x.inp



Simpler esonator consisting of a flat mirror and a concave mirror. The waist will form at the flat mirror.

Table. 5.1. Parameters or stable resonator example.

length	45 cm
mirror radius	50 cm
wavelength	1.064 µ
Rayleigh range	15 cm
waist radius	0.02253936 cm
aperture radius	0.14 cm

```
macro/def reson/o
   pass = pass + 1 list                # increment pass counter
   step = step + 1                     # increment step number
   prop 45                             # propagate 45 cm.
   mirror/sph 0 -50                    # mirror of 50 cm. radius
   clap/c/c 0 .14                      # .14 cm. radius aperture
   prop 45                             # propagate 45 cm. along beam
   mirror/sph 0 1.e15                  # flat mirror
   energy                              # calculate energy in the beams
   variab/set Energy 1 energy
   Energy = Energy - 1                 # calculate energy difference
   udata/set pass step Energy          # store energy differences
   energy/norm 1 1                     # renormalize energy
   plot/watch ex33_1.plt
```

Half symmetric stable cavity resonator (cont'd): ex33x.inp

```
   plot/udata left=10 right=100 min=-.05 max=.05
   plot/watch ex33_2.plt
   plot/l
macro/end
wavelength 0 1.064                     # set wavelengths
units 0 .005
resonator/name reson
resonator/eigen/test 1
resonator/eigen/set 1                  # set beam 2 to eigen mode
clear 1 1                              # start with a plane wave in beam 2
energy/norm 1 1                        # normalize energies
status/p
pass = 0                               # initialize variables
step = 0                               # for pass counters
title Energy loss per pass
reson/run 100
```

Review

- Modify the example to start with random noise
  - make a separate beam of same size and attribute "data"
  - clear the array to zero
  - put in random noise with the `noise` command
  - use `copy/con` to copy the noise into beam 1 without changing geodata variables
  - observe convergence
- Does starting with random noise change the solution after full convergence?
- Measure the mean radius of the converged mode using `fitgeo`.
- Add an internal lens of -150 cm focal length at a distance of 10 cm from the curved mirror. Keep the resonator length at 45 cm.
- Check convergence with a smaller clear aperture, e.g., of about 0.1 cm. Try again with aperture at 0.12 cm.
- how much does the converged mode differ in size?
- estimate the relative rate of convergence for the 0.1 vs. 0.2 aperture

Stable resonator, bare cavity analysis: stable.inp

- half symmetric resonator, r1 = 50 cm (clap of .13 cm), r2 (flat), length = 45 cm, wavelength 1.064 micron, reflectivity of flat = .98. No aperture for now.
- energy normalization replaces true laser gainvariab/dec/int pass

```
macro/def reson/o
 pass = pass + 1 list      # increment counter
 prop 45                   # propagate 45 cm.
 mirror/sph 0 -50          # mirror of 50 cm. radius
 prop 45                   # propagate 45 cm. along beam
 mirror/flat 1             # flat mirror
 variab/set Energy 1 energy
 udata/set pass pass [Energy-1]# store energy differences
 energy/norm 1 1           # normalize energy each pass
 plot/l 1 xrad=.15         # picture of cavity mode
macro/end
```

Setup surrogate gaussian beam

Analyze resonator to determine ideal gaussian mode

- Guess at units (in this case)

```
units 0 .005
wavelength 1 1.064
resonator/name reson
resonator/eigen/test
```

- Set resonator to eigenmode and normalize

```
resonator/eigen/set        # set beam to eigenmode
energy/norm 1 1            # normalize energies
Test for one pass with geodata and plot/l
geodata
plot/l 1 xrad=.15
pause
reson/run 1
geodata
```

- Check that beam is, indeed, unchanged

A classical bare-cavity analysis: stable1.inp

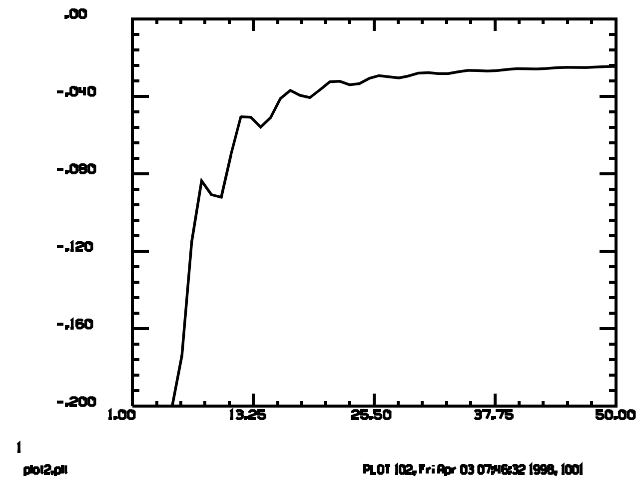
- Add an aperture of radius 0.13

```
clap/c/noadjust 1 .13
```

- start from noise

```
clear 1 0
noise 1 1
```

- Run resonator 50 times
- Observe the convergence for 50 cycles
  - explain the oscillatory behavior in the `udata` curve
- Does this device ever converge to TEM(0,0). Try another 450 cycles?
- Will this laser really converge in this manner?

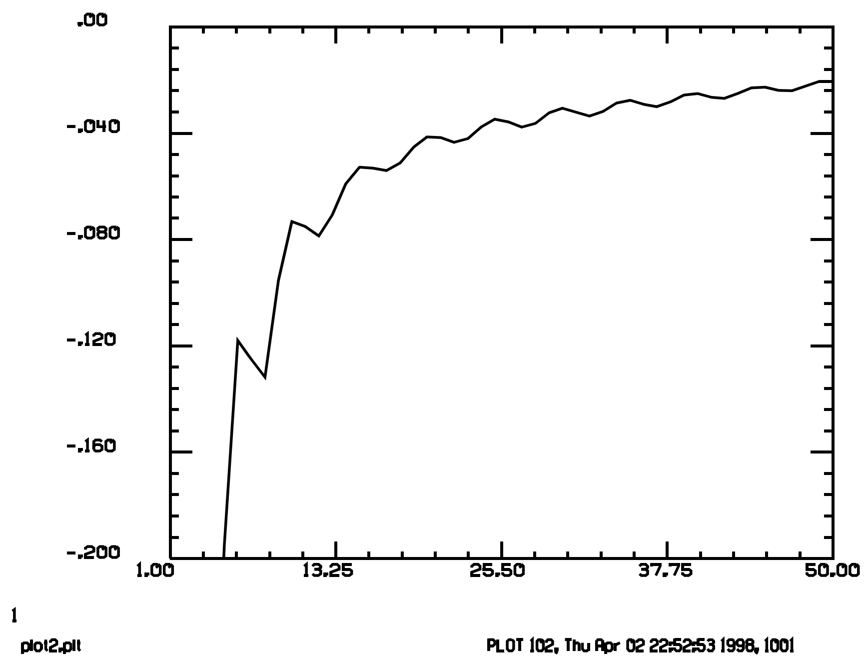


A more realistic bare cavity analysis: stable2.inp

- Bare cavity analysis with noise seeding
- Lasers are constantly reseeded by stimulated emission. The laser never truly stabilizes -- except statistically
- Put noise inside the macro

```
noise 1 1e-6
```

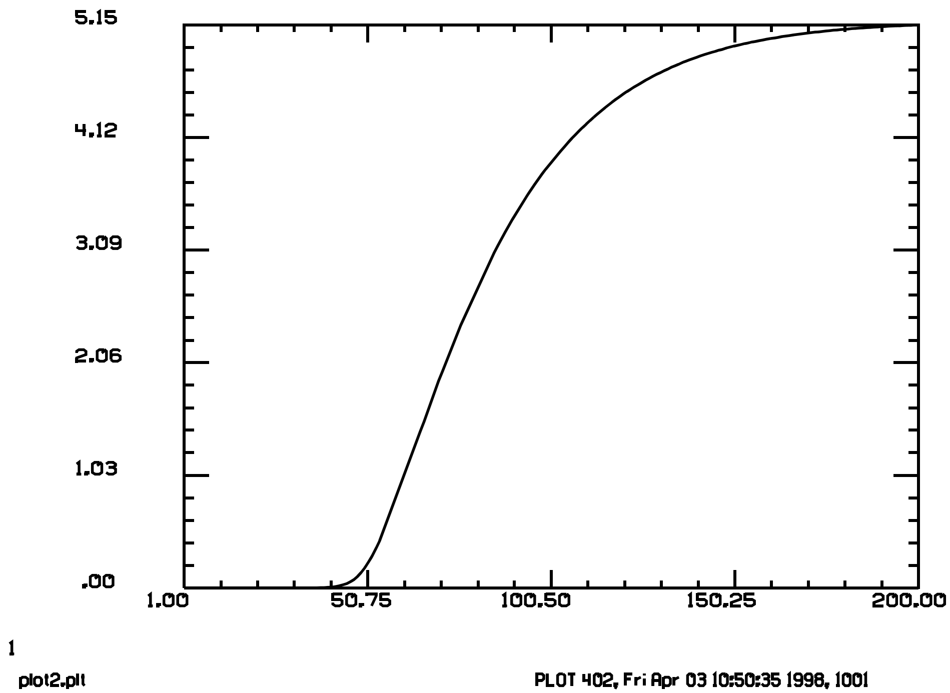
- Try 50 cycles
- Try another 400 cycles. Do you think it converges?



Now replace energy normalization with Beer's Law gain: stable3.inp

Adding Beer's Law saturated gain

- Choose g0 = 0.04, Es = 100, length = 6 cm, Reflectivity = .98
- About how many passes for mode to stabilize?
- About how many passes for energy to stabilize?
- Estimate converged energy



Optimization of the mirror reflectivity: from stable3.inp create: stable4.inp

Use a single-variable optimization of mirror reflectivity to optimize output power

```
Build an outer macro to allow settle time
macro/def settle/o
 pass = 0                  # initialize variable
 clear 1 0                 # clear to zero
 x = srand(1)              # set random seed
 reson/run Settle_time# Settle_time is the number of settling passes
 Energy_settle = Energy_trans
 inverse = 1./Energy_settle# make merit function
macro/end
```

- Set random seed to be the same every time

```
x = srand(1)
```

- Make another outer macro to run optimization

```
macro/def optimize/o
 opt_pass = opt_pass + 1
 opt/run 1
 plot/w plot2.plt 410 400 400 300
 udata/set opt_pass opt_pass Energy_settle
 plot/udata 1
macro/end
```

Continue resonator optimization

- Set up variable table, target table, name of macro to be optimized

```
opt_pass = 0
opt/name settle
opt/var/add Reflectivity .002
opt/tar/add inverse 0.
opt/damp 2
Settle_time = 200
macro optimize/1
```

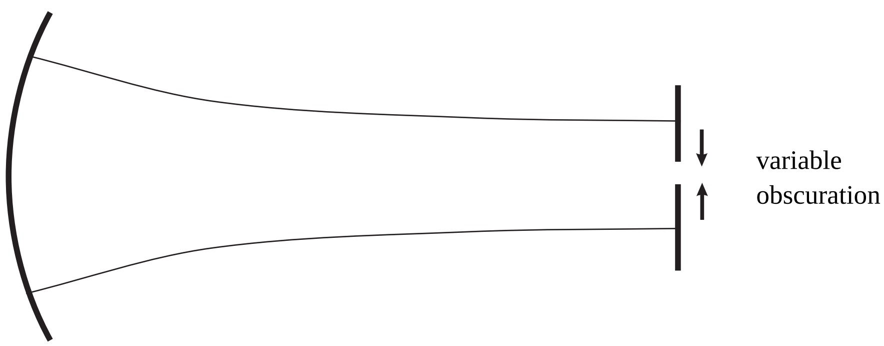
- Choose to optimize inverse of Energy. Why?
- Start with high value of Reflectivity, so optimization starts decreasing it.
- Start with a low value of Settle_time to check out optimization for mistakes
- Make sure optimization has "captured" the problem, i.e., is making progress toward as solution before increasing Settle_time

Stable resonator with central obscuration: ex57x.inp

- a central obscuration of variable size is placed in the half-symmetric resonator
- for a very small hole TEM(0,0) prevails and is simply less efficient
- for larger holes the donut mode begins to compete and inhibits convergence when the two modes have similar eigenvalues.
- a series of mode types are generated as the hole is made bigger with transition points showing poor convergence

Table. 5.2. Parameters for resonator with hole outcoupling, i.e., central obscurations.

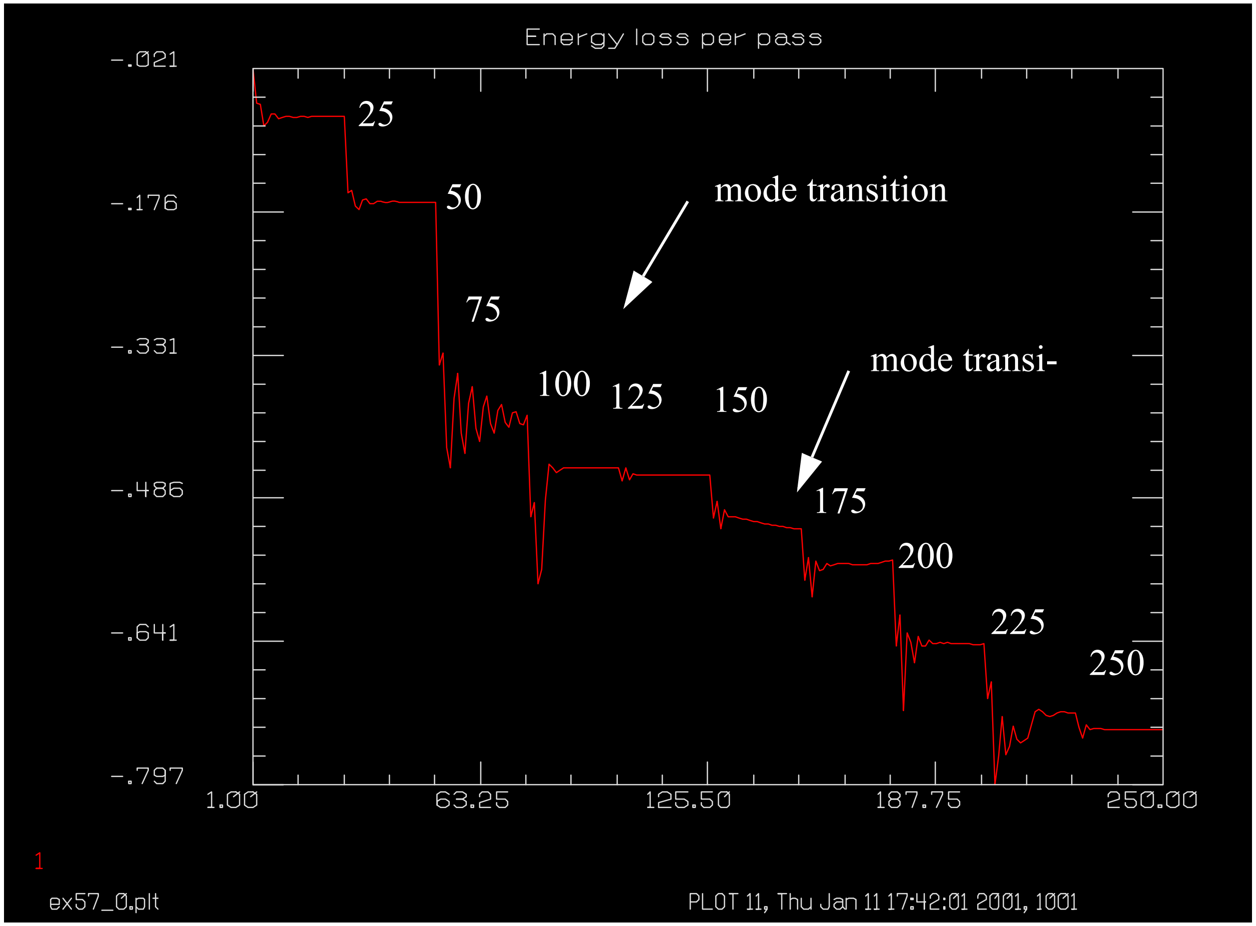
length	45 cm
mirror radius	50 cm
wavelength	1.064 microns



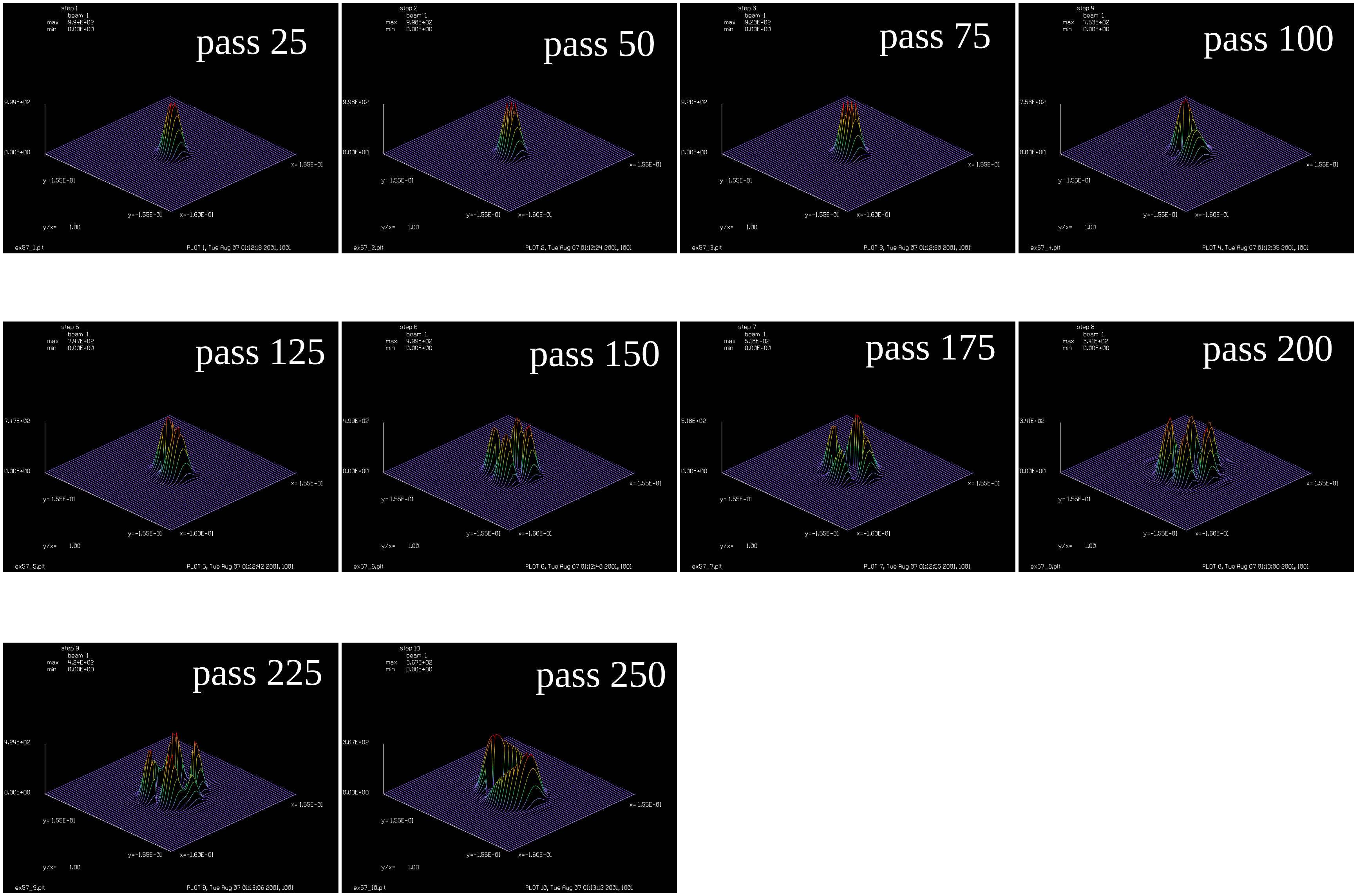
Concave and flat mirror resonator with variable size hole in the flat which forms an obscuration in the beam.

Loss per pass vs. number of passes

- size is increased by .005 cm every 25 passes
- significant mode competition at 75 passes, 150 passes, 250 passes

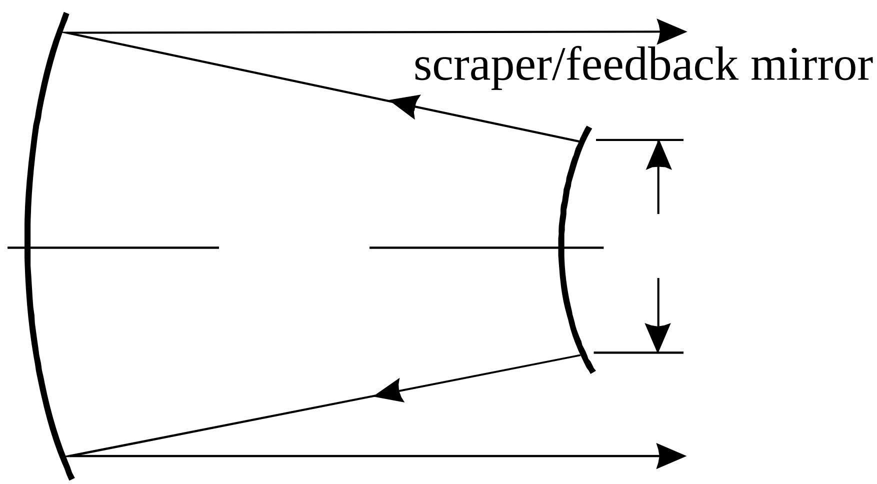


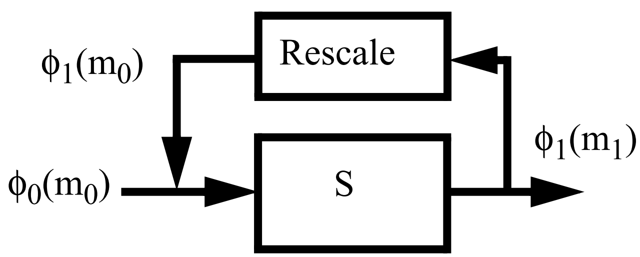
Modes for first 10 obscuration sizes: ex57x.inp



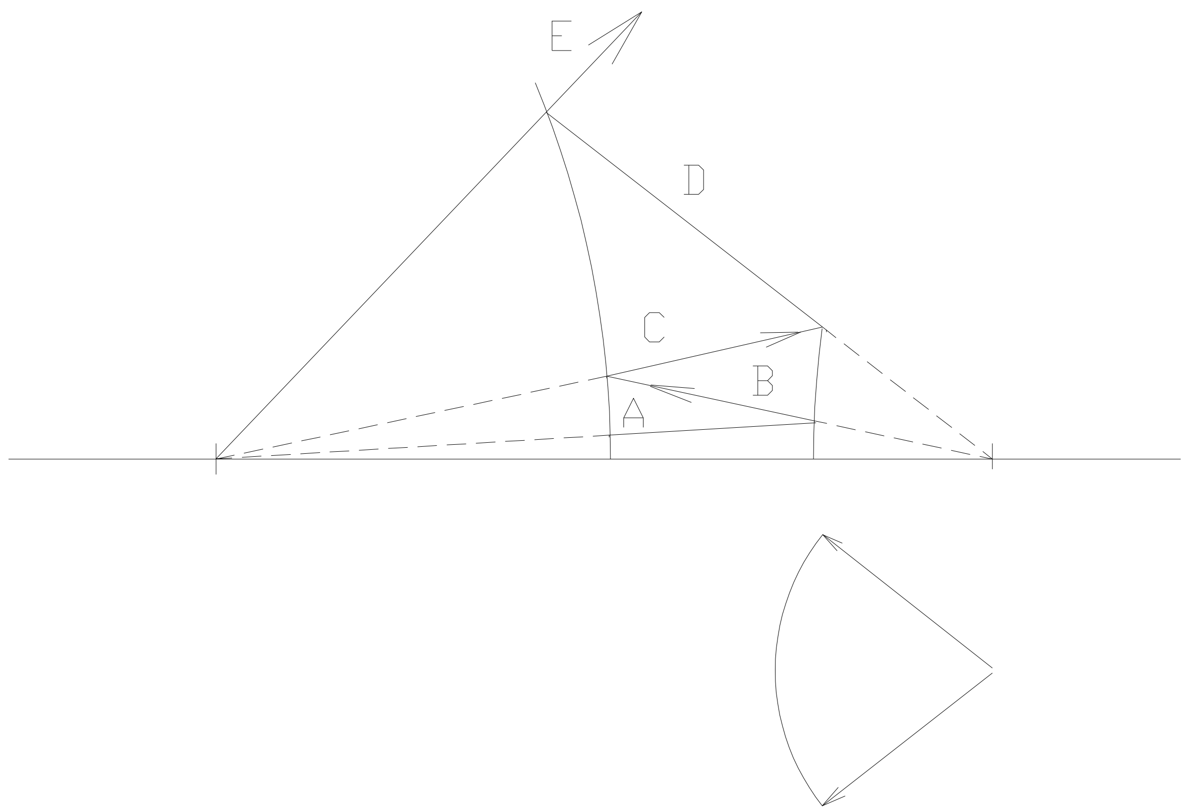
Unstable resonators

- Siegman originally suggested using unstable resonators to achieve good beam quality with high gain media
- modes must generally be found by numerical calculations
- Net round-trip magnification requires a “rescale” operation
- GLAD automatically handles the rescale with the resonator command
- GLAD also calculates the appropriate surrogate gaussian beam





The unstable resonator exhibits an eigenradius



Ray picture of unstable resonator. The ray goes through segments A through E, increasing in magnification by about 3 each time. The ray shown is set to originate from the center of curvature of the eigenradius. In this example, the inner third of the outgoing wavefront is fed back during each pass. In numerical analysis, the units will expand by the magnification in one round trip. After aperturing by the scraper mirror, leaving only the inner part of the array, we rescale the array to the original units, discarding the outer parts of the distribution, which are, of course, zero after aperturing.

Positive and negative branch unstable resonators

$$\text{unstable if } |m| = \left|\frac{A+D}{2}\right| > 1. \tag{5.11}$$

If m > 1, resonator is a positive branch (even number of internal foci, usually zero).

If m < -1, the resonator is a negative branch unstable resonator and has an odd number of internal foci.

We may find eigenvalues for the unstable resonator, as given below.

$$\lambda_a, \lambda_b = m \pm \sqrt{m^2 - 1} = M \text{ or } \frac{1}{M}, \tag{5.12}$$

Find eigenfunctions by numerical analysis.

Find eigenradii by

$$\frac{1}{R_a} = \frac{D - \lambda_a}{B}, \frac{1}{R_b} = \frac{D - \lambda_b}{B} \tag{5.13}$$

In most cases we choose the solution with magnification greater than 1.

The demagnifying solution will collapse into the optical axis and ultimately, because of diffraction, will come back out as a magnifying solution, so it is usually sufficient to analyze the magnifying wave.

Unstable resonators: ex11x.inp

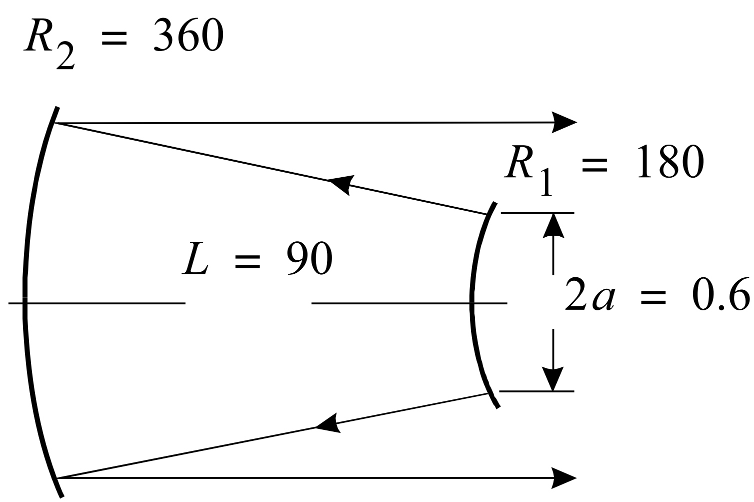
Confocal resonator — foci of two mirrors are coincident
confocal resonator produces collimated output

$$N_c = \frac{Ma^2}{L\lambda},\ N_{eq} = \frac{M^2-1}{2M}\frac{a^2}{L\lambda} \qquad (5.14)$$

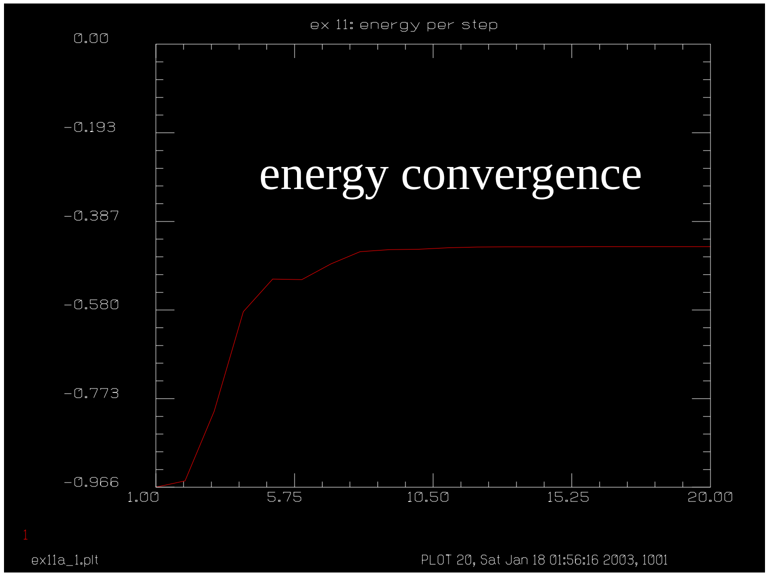
where a is the aperture radius, L is the resonator length, λ is the wavelength, and M is the magnification.

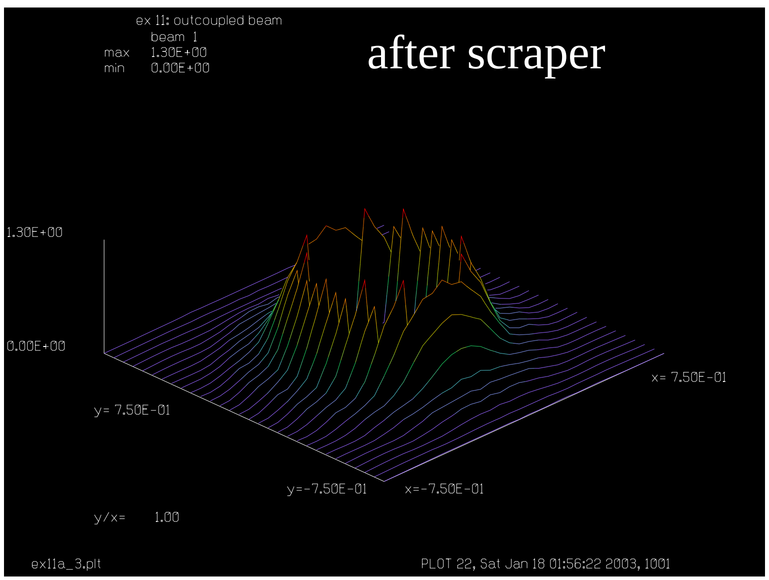
The parameters that are used are L = 90 cm, a = 0.3 cm, M = 2, λ = 10 micron

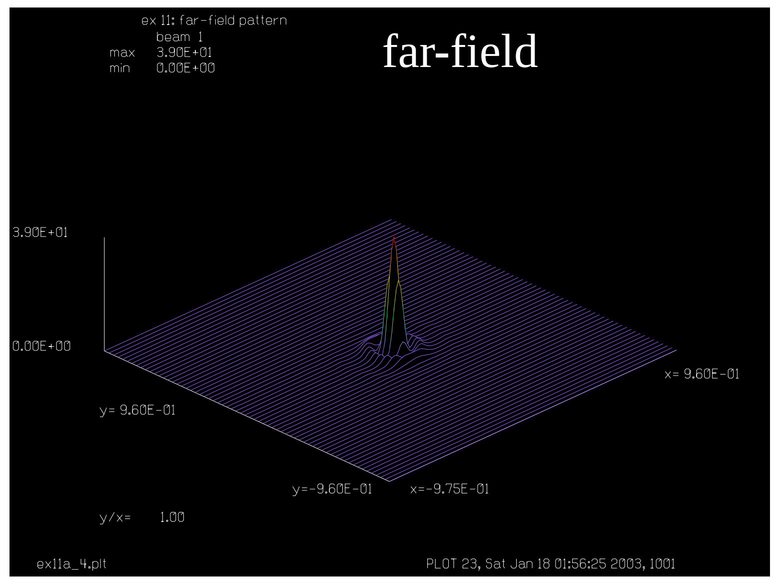
This results in $N_c = 2$ and $N_{eq} = 0.75$.



Unstable resonator from: ex11x.inp







Summary of macro and resonator commands for unstable resonator

```
macro/def conres/over
 pass_number = pass_number + 1 # increment pass counter
 clap/cir/con 1 .3 # aperture of .3 cm
 mirror rad=180 # convex mirror
 prop 90 # propagate 90 cm backward
 mirror rad=360. # concave mirror
 clap/cir/con 1 .7 # aperture of .7 cm
 prop 90 # forward propagation
 variable/set Energy 1 energy
 udata/set pass_number pass_number [Energy-1]
 energy/norm 1 1
 energy/norm 1 1
 title resonator mode pass = @pass_number
 plot/l xrad=.75
macro/end
.
.
.
 resonator/name conres
 resonator/eigen/test 1
 resonator/eigen/set 1
 clear 1 0
 noise 1 1
 resonator/run 30
```

Review of unstable resonators

- Does the converged mode change if we start with flat amplitude rather than noise?
- Keeping the concave mirror at a radius of 360, shorten the cavity length to 80 cm
  - what radius for the secondary (convex) mirror is required for a confocal configuration?
  - what is the magnification per round-trip? Check this magnification with `resonator/eigen/list`
  - change ex11a to use these new parameters
  - use the resonator commands to test and initialize the device
  - test the resonator for one pass to see that geodata does not change over one round-trip
  - run the resonator to convergence, check the wavefront of the converged mode by Strehl ratio
- use `udata/list` to determine the loss-per-pass [energy-1]
- add 0.2 waves of astigmatism (normalizing radius = 0.7) rerun the problem and note the change of round-trip loss. Note the mode shape.
- try adding +0.2 waves of Seidel spherical aberration. Try adding -0.2 waves of Seidel. Can you explain the changes in loss-per-pass relative to the addition of 0.2 waves of astigmatism?

References

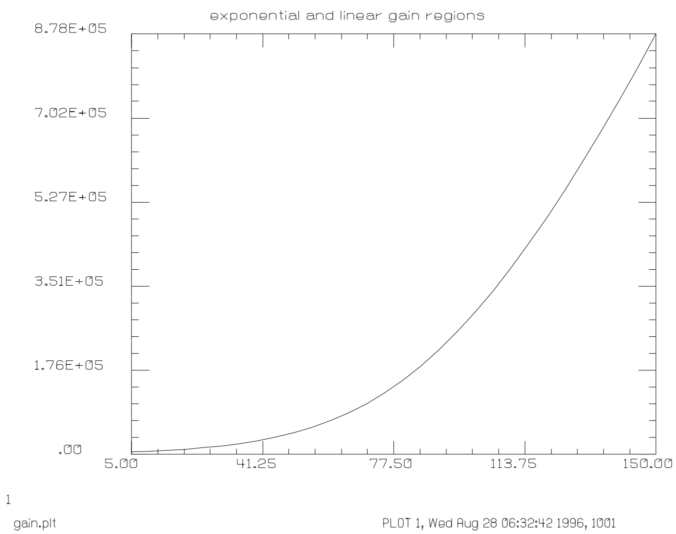
1. G. Fox and T. Li, “Resonant Modes in a Maser Interferometer,” Bell System Technical Journal, Vol. 46, 453 (1961).
2. A. E. Siegman, *Lasers*, University Science Books, Mill Valley, CA (1986).
3. A. E. Siegman and H. Y. Miller, “Unstable Optical Resonator Loss Calculations Using Prony Method,” Appl. Opt. Vol. 9, No. 12, p. 2729 (1970).

6. Laser Gain

Saturated Beer's Law gain

$$I(z+\Delta z) = I(z)\exp\frac{g_0\Delta z}{\left(1+\frac{I(z)}{I_{sat}}\right)^q}, \qquad (6.1)$$

- where g_0 is the small signal gain, I_{sat} is the saturation intensity, and $q = 1/2$ for inhomogeneously and $q = 1$ for homogeneously broadened gain.
- The gain grows exponentially at low values.



$$\frac{dI}{dz} \approx g_0 I. \qquad (6.2)$$

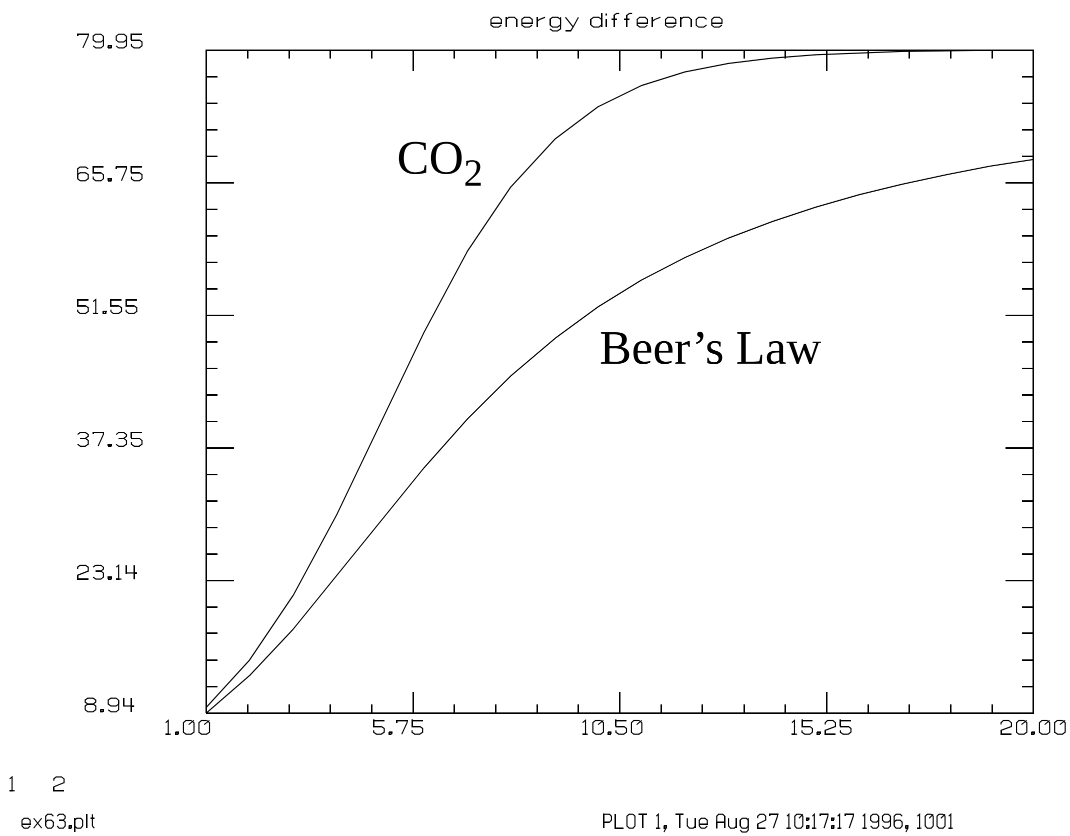
- The characteristic gain length is $1/g_0$. When I is comparable to I_{sat}, the homogeneously broadened gain takes the form

$$\frac{dI}{dz} \approx g_0 I_{sat}, \qquad (6.3)$$

Franz-Nodvik gain

- appropriate for short pulses with square temporal profiles propagating through an amplifying medium

$$I(x, y, z + \Delta z) = I_{sat} \ln\left\{ 1 + \exp[g_0(x, y)] \exp\left[\left(\frac{I(x, y, z)}{I_{sat}}\right) - 1\right]\right\}, \quad (6.4)$$



Rate equation formulation for two-level system

- The rate equations are [Siegman, *Lasers*]

$$\Delta N_2 = \left[R_2 - \frac{N_2}{t_2} - (N_2 - N_1)W_i(\nu)\right]\Delta t, \tag{6.5}$$

$$\Delta N_1 = \left[R_1 - \frac{N_1}{t_{10}} + \frac{N_2}{t_{spont}} + (N_2 - N_1)W_i(\nu)\right]\Delta t, \tag{6.6}$$

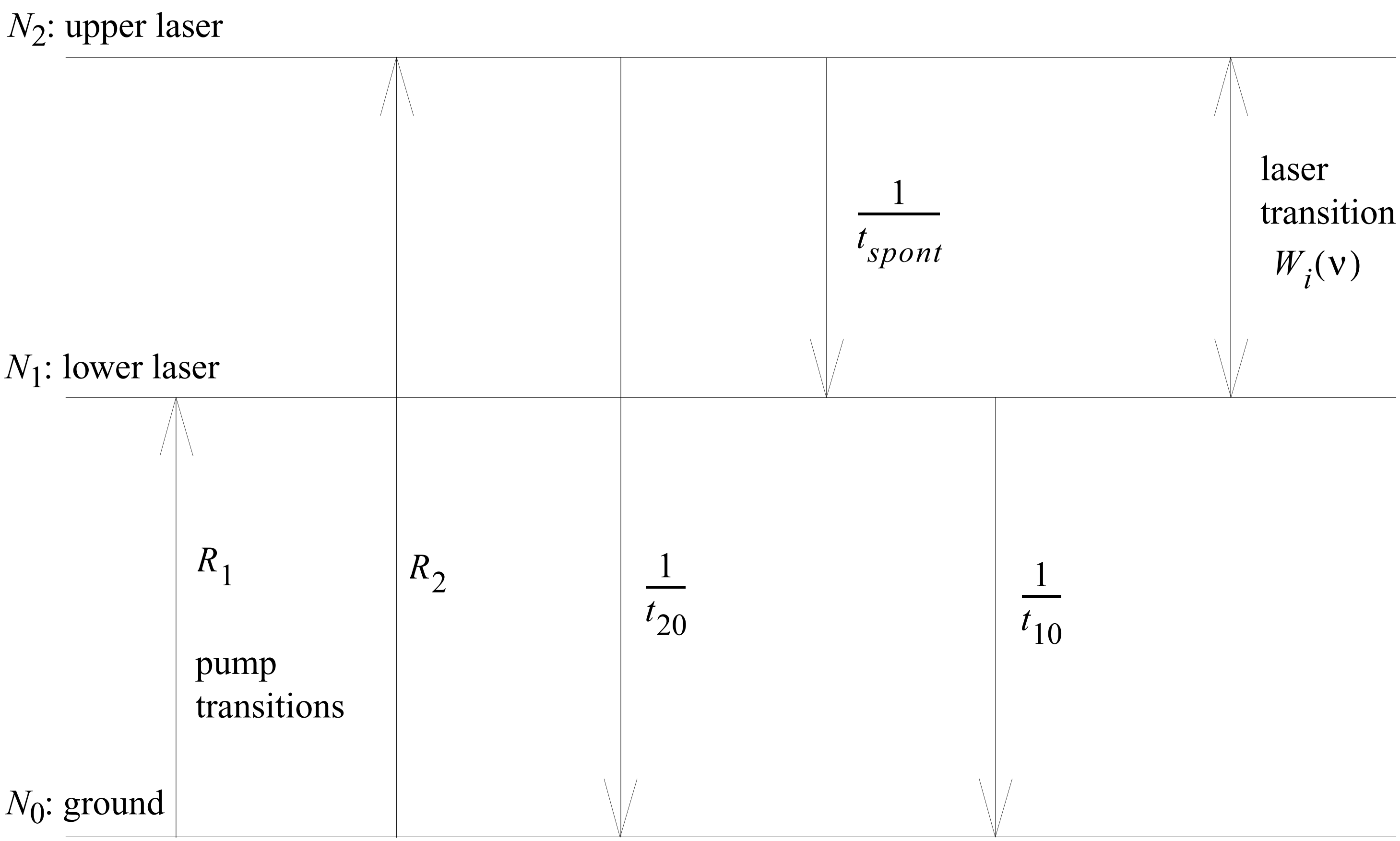
ΔN_1, ΔN_2 change in population of lower and upper levels, atoms/cm^3,

R_1, R_2	pump rate for lower and upper level, excitations/sec/cm^3,
t_{spont}	spontaneous decay lifetime, sec,
t_{20}	decay time from upper level to ground, sec,
t_2	total decay time from upper level to ground, sec,

$$1/t_2 = 1/t_{20} + 1/t_{spont},$$

t_{10}	decay time from lower level to ground, sec,
$W_i(\nu)$	transition probability density, probability/sec/cm^3,
Δt	elapsed time.

Energy diagram for two-level system



Transition probability and small signal gain

The transition probability density is,

$$W_i(\nu) = \frac{\lambda^2 g(\nu)}{8\pi n^2 h\nu t_{spont}} I, \tag{6.7}$$

where

λ	wavelength,
$g(\nu)$	normalized lineshape,
n	index of refraction,
h	Planck's constant,
ν	frequency of the radiation,
I	irradiance of the radiation.

The transition propability may be written in terms of the Einstein B-coefficient:

$$W_i(\nu) = B(\nu)\frac{I}{h\nu} \tag{6.8}$$

$$\text{where } B(\nu) = \frac{\lambda^2 f(\nu)}{8\pi n^2 t_{spont}} I. \tag{6.9}$$

The small signal amplification takes the form

$$I(z) = I(0)e^{B\Delta Nz}. \tag{6.10}$$

Franz-Nodvik

- Gain depends on $I(z)$, $N(z)$, and the cross section B

irradiance [w/cm^2]

gain [cm^{-1}]

$$\frac{\partial I(z)}{\partial z} = B\Delta N(z)I(z) = g_0 I(z) \tag{6.11}$$

cross section [cm^2]

population inversion [cm^{-3}]

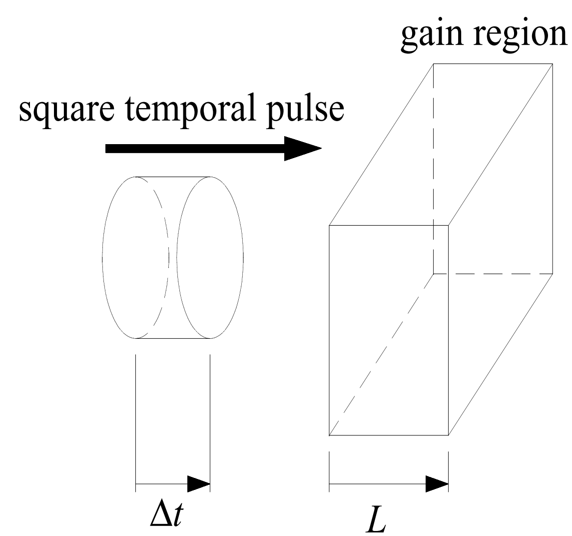
Assume a constant temporarl pulse of duration Δt and short section of gain of length L.
Total initial energy = final energy

$$\Delta t\left[I_{max} = I(0) + \Delta N(0)\frac{h\nu L}{2\Delta t}\right] = \text{total energy (const.)} \tag{6.12}$$

$$\frac{h\nu L}{2}\left[\Delta N_{max} = \Delta N(0) + \Delta I(0)\frac{2\Delta t}{h\nu L}\right] = \text{total energy (const.)} \tag{6.13}$$

$$\frac{\partial I(z)}{\partial z} = BN(z)I(z) \Rightarrow B\left[\Delta N_{max} - \frac{2I(z)\Delta t}{h\nu L}\right]I(z) \tag{6.14}$$

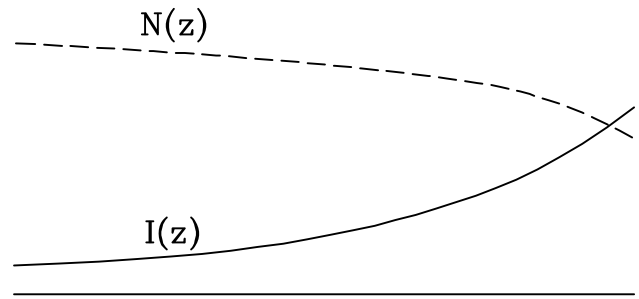
$$I(L) = \frac{I_{max}I(0)}{I(0) + [I_{max} - I(0)]e^{-B\Delta N_{max}L}} \quad \text{Franz-Nodvik} \tag{6.15}$$



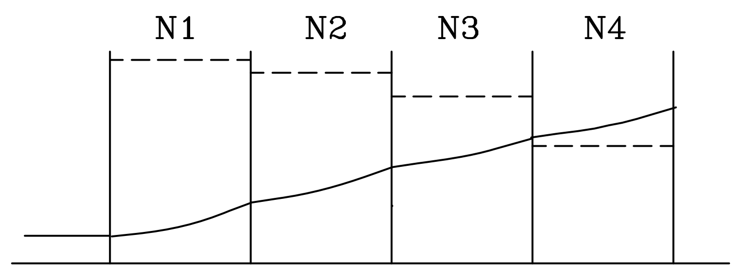
Franz-Nodvik, axial sampling with multiple gain sheets

- The population inversion ΔN is constant for each gain sheet
- For extremly strong one-pass gain, multiple gain sheets may be needed

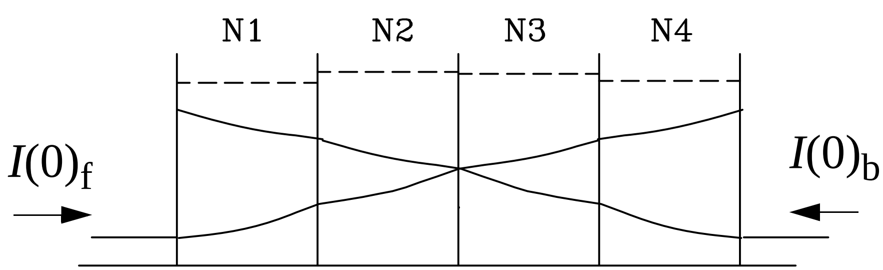
extremely strong one-pass gain



approximation with four gain sheets



double pass levels out the population inversion, fewer gain sheets required

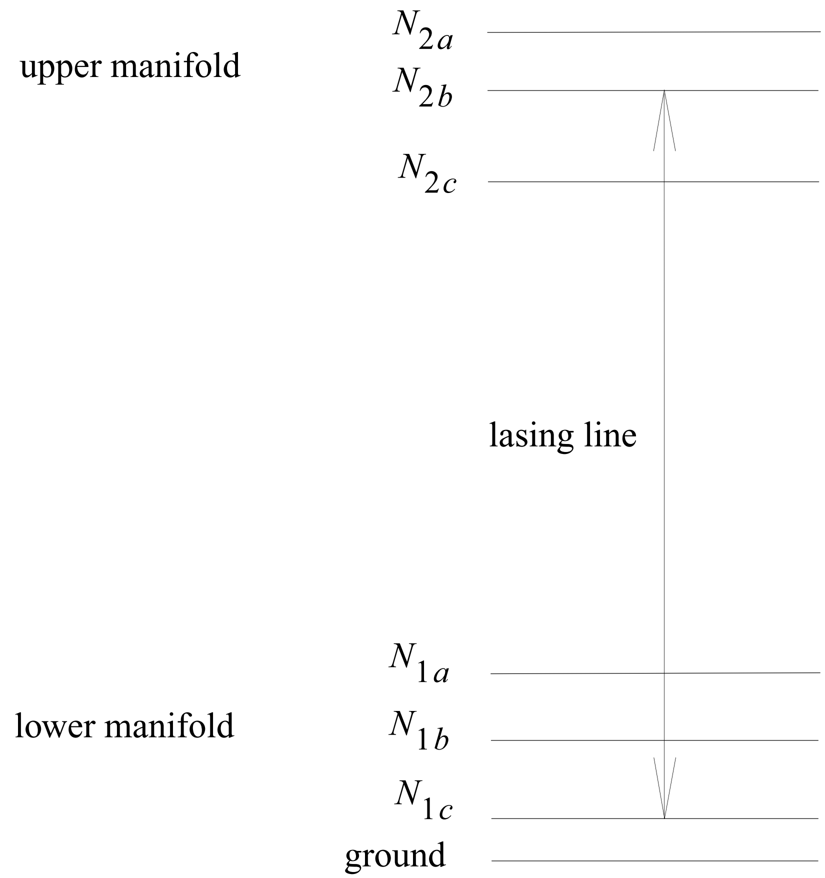


Franz-Nodvik, Three-level gain

- In three-level gain, N_1 is at ground level (or very closely coupled to ground level)
- Upper and lower lasing lines may be part of manifolds that are coupled by thermal equilibration
- The total population for N_2 and N_1 is fixed.
  - general manifolds

$$N_{tot} = N_1 + N_2 = \sum_{p=0}^{P} N_{1p} + \sum_{q=0}^{Q} N_{2q}$$

  - ruby laser $N_2 = N_2(2A) + N_2(E)$
  - single upper and lower states
    $N_{tot} = N_1 + N_2$
- Solve with Franz-Nodvik after substitution of variables and consideration of Boltzmann thermal equilibrium conditions



Steady-state condition derived from rate equations

Under steady-state conditions, the irradiance of the optical field is constant and the rate equations of Eq. (6.5) and (6.6) leads to the steady-state solution for the population inversion:

$$\Delta N_0 = R_2 t_2 - \left(R_1 + \frac{t_2}{t_{spont}} R_2\right) t_{10} . \tag{6.16}$$

The small signal gain coefficient is

$$g_0(\nu) = B(\nu)\Delta N_0 \tag{6.17}$$

The gain coefficient for homogeneous broadening and for arbitrary irradiance magnitude is,

$$g(\nu) = \frac{g_0(\nu)}{1 + \frac{I}{I_s}} \tag{6.18}$$

where,

$$I_s = \frac{8\pi n^2 h\nu}{\left(\frac{t_2}{t_{spont}}\right)\lambda^2 g(\nu)} = \frac{h\nu}{B(\nu) t_2} \tag{6.19}$$

Steady-state and strong saturation

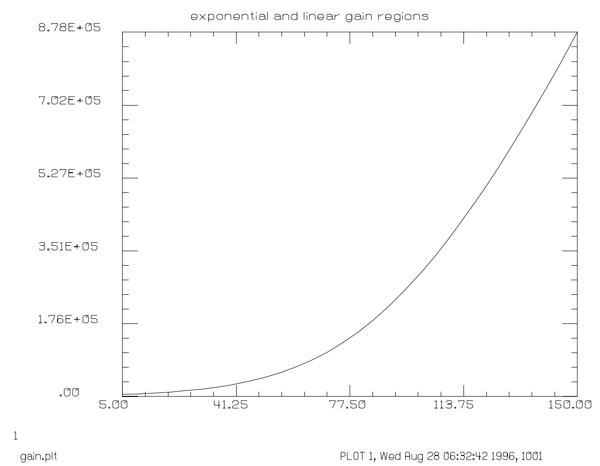
In the case of strong saturation the change in intensity per unit length is

$$\frac{dI}{dz} \approx g_0(\nu) I_s = (B \Delta N_0)\left(\frac{h\nu}{B(\nu) t_2}\right) = \frac{\Delta N_0 h\nu}{t_2}. \tag{6.20}$$

Where the pumping rate into the upper level dominates the process, Eqs. (6.10), (6.16), and (6.17) give the saturated gain coefficient as

$$\frac{dI}{dz} = R_2 h\nu,\ g(\nu) = R_2 h\nu, \tag{6.21}$$

showing, in the case of saturated steady-state gain, a linear growth of irradiance with distance based on the pumping flux density.



Off line center effects

The gain and off-line index of refraction effects are represented by a complex index of refraction using χ'_m and χ''_m such that

$$n \to n\left(1 + \frac{\chi'_m}{2n^2} + j\frac{\chi''_m}{2n^2}\right) \tag{6.22}$$

$$\chi''_m = (N_1 - N_2)\frac{\lambda^3}{16\pi^3 t_{spont} n} g(\nu),\ \chi'_m = -\frac{2(\nu_{off} + m\Delta\nu_c)}{\Delta\nu}\chi''_m, \tag{6.23}$$

$$g(\nu_m) = \frac{\Delta\nu}{2\pi\left[(\nu_{off} + m\Delta\nu_c)^2 + \left(\frac{\Delta\nu}{2}\right)^2\right]}, \tag{6.24}$$

where $\nu_m - \nu_{cen} = \nu_{off} + m\Delta\nu_c$, and m is the mode number. The optical field, under steady state conditions varies as

$$a_m(x, y, \Delta t) = a_m(x, y, 0) e^{(jk\chi'_m + k\chi''_m)\frac{L}{2n^2}}, \tag{6.25}$$

where $\chi_m = \chi'_m - j\chi''_m$ is the electric susceptibility and n is the index of refraction.

Spontaneous emission

- Spontaneous emission arises from the decay of the population inversion.
- This spontaneous emission is a noise source for many laser processes.
- The noise power injected into each mode in a distance is

$$\Delta I_{\text{noise}} = \frac{(N_2 - N_1)h\nu\Delta z}{nct_{spont}} \frac{\lambda^2}{4\pi\Delta x\Delta y}, \quad (6.26)$$

$\Delta\Omega = \lambda^2/(\Delta x\Delta y)$ is the solid angle subtended by the computer array $(\Delta x, \Delta y)$.
Noise is delta-correlated, normally distributed complex random numbers.

Spontaneous emission and the finite lifetime of photons in the cavity limit the degree of convergence.

Lasers always run on a collection of modes rather than the single lowest-loss mode, as predicted by bare cavity analysis.

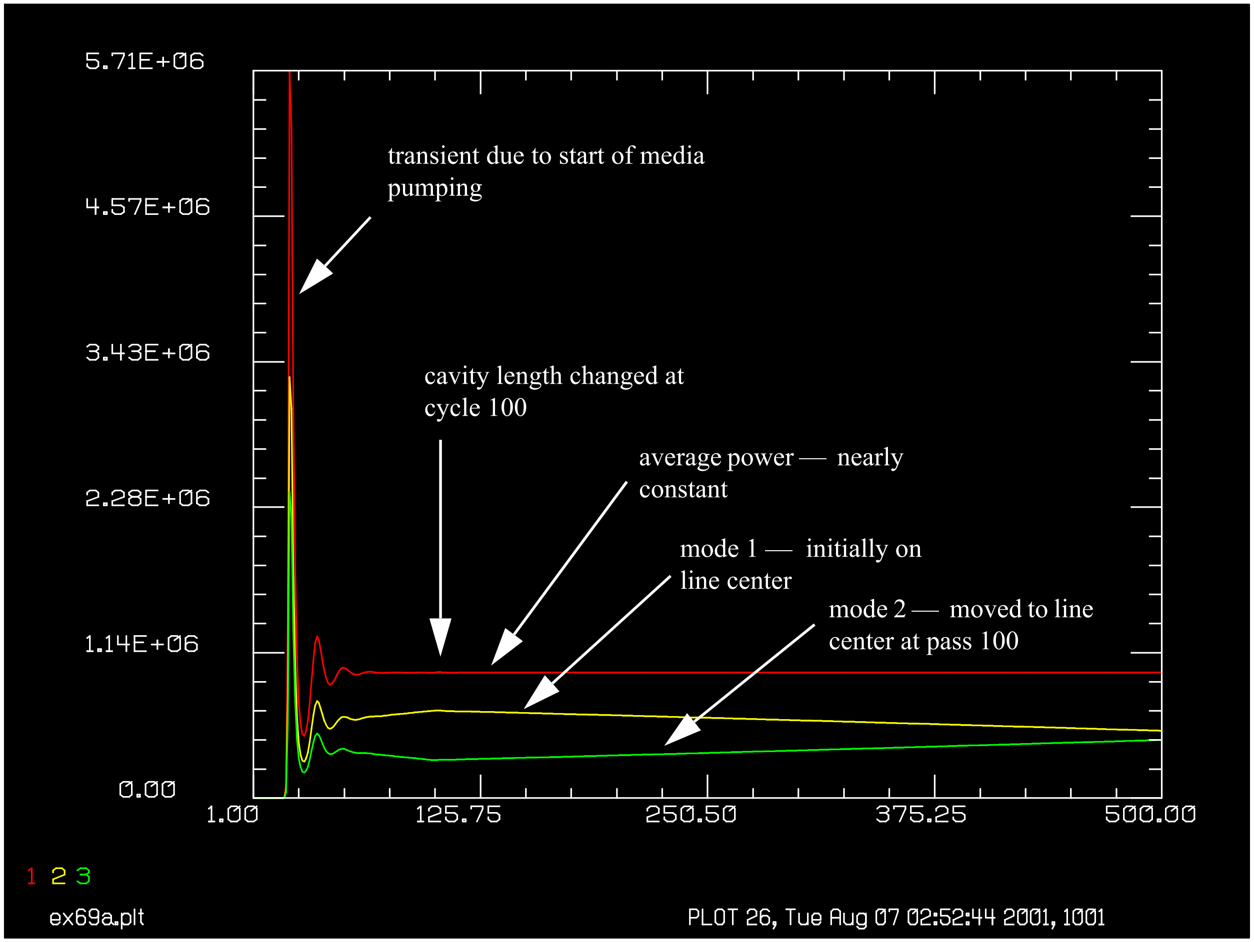
Simple example of mode competition with rate equation treatment

Table. 6.1. Parameters for rate equation calculation.

wavelength	0.84 micron
index of refraction	3.35
cavity length	0.89 cm
atomic line width	1.45×10^{13} Hz
atomic line center	3.57×10^{14} Hz
spontaneous decay time	3×10^{-9} sec
transition rate for Level 2 to 0, t_{20}	0.3 sec
transition rate for Level 1 to 0, t_{10}	1×10^{-11} sec
longitudinal mode separation	5×10^{11} Hz
Fresnel reflection loss	0.396
cavity length change	-5×10^{11} Hz

Mode competition

two modes of frequency $f_1(0)$ and $f_2(\Delta\omega)$ compete for gain, at pass = 100 the cavity length is changed causing $f_1(-\Delta\omega)$ and $f_2(0)$, so mode 2 now prevails



Temporal response of competing modes from the start of medium pumping. The cavity length is changed at pass 100 causing the relative loss on the two longitudinal modes to reverse.

Rate equation gain: ex69x.inp

```
c## ex69x
c
c  Example 69a: Rate equations
c
c  This is an example of the use of rate equations
c  The resonator is similar to that of a laser diode
c  with flat faces.
c
c  The device will be allowed to converge for 100 passes
c  and then the length changed to put Beam 2 on the line
c  center causing Beam 1 to decay and Beam 2 to increase.
c
c  Establish initial units and a gaussian field distribution
c
c  Beams   Use
c  1       longitudinal mode on line center
c  2       slightly off line center
c  3       pump distribution array
c
   variab/dec/int pass switch
   array/s 1 4
   nbeam 3                      # set number of beams
   array/s 3 4 4 1              # make medium array polarized
   units/s 0 1
   wavelength/set 0 .84 3.35    # set wavelength and index
   clear 1 1e-4                 # set initial irradiance
   clear 2 1e-4
   clear 3 1.93e6               # set pump rate in
                                # watts per cm**2
   gain/rate/n2pump 3 3         # Modify beam 3 to put
                                # the pump rate in the real
                                # word
```

Rate equation gain (cont'd I): ex69x.inp

```
                                  # inversion will go in the
                                  # imaginary word
c
width = 1.45e13         #  Set line width, hz
center = 3.57e14        #  Set line center, hz
tspont = 3e-9           #  Set spontaneous emission rate, sec
t20 = 3e-1              #  Set decay rate for level 2, sec
t10 = 1e-11             #  Set decay rate for level 1, sec
mode_sep = 5e11         #  Set longitudinal mode separation, hz
c
c  Set offset from line center of first mode, 0 hz
c  Set fractional pumping into level 1, n1pump, 0
c
   gain/rate/set width center tspont t20 t10 mode_sep 0 0
   gain/rate/list
   pack/set 1 2 3               # pack all 3 beams
   pass = 0                     # initialize pass counter
macro/def ex69a/o
   pack/in                      # pack beams
     pass = pass + 1            # increment pass counter
c
c    Set gain length of .89 cm
c    Set pump time of 1e-10
c    Set number of steps over which distance is to be taken
c
     gain/rate/step .89 1e-10 10 # implement rate eq. gain
   pack/out                     # unpack beams
   mult 1 .396                  # Fresnel reflection losses
   mult 2 .396
   variab/set peak1 1 peak      # record peak irradiance
   variab/set peak2 2 peak
   sum = peak1 + peak2          # record sum of Beam 1 and 2
   udata/set pass pass sum peak1 peak2 # store for the record
```

Rate equation gain (cont'd II): ex69x.inp

```
   switch = mod(pass,20)          # plot every 20th pass
   if switch = 0 then
      plot/udata/seq              # make udata plot
   endif
macro/end
   plot/udata/set y01 y02 y03     # specify beams to be displayed
   plot/watch ex69a.plt           # display record
   mac ex69a/100                  # stabalize for 100 passes
c
c  shift cavity length
c
c  Set frequency offset so Beam 2 is now on line center
c
   gain/rate/set width center tspont t20 t10 mode_sep -5e11 0
   mac ex69a/400                  # run 400 passes to let
                                  # Beam 2 to begin to grow
   plot/udata/seq
   end
```

Calculate gain: ex69fy.inp

Calculate actual gain and small signal gain for the parameters of Table 6.2. For small signal gain use Eqs 6.12 through 6.15.

Table. 6.2. Parameters for rate equation calculation.

wavelength	0.84 micron
index of refraction	3.35
gain length	0.89 cm
pulse time	1×10^{-12} sec
atomic line width	1.45×10^{13} Hz
atomic line center	3.57×10^{14} Hz
spontaneous decay time	3×10^{-9} sec
transition rate for Level 2 to 0, t_{20}	0.3 sec
transition rate for Level 1 to 0, t_{10}	1×10^{-11} sec
initial irradiance	1×10^{8} w/cm^2
pumping	0.0 w/cm^2
inital N_2 population	1.89818778×10^{-4} j/cm^3
inital N_1 population	0.0 j/cm^3

Pulse propagation, Franz-Nodvik (cont'd) ex69fx.inp

$$\text{Optical amplification } \frac{\partial I(z)}{\partial z} = B\Delta N(z)I(z) \tag{6.27}$$

Equation (6.27) now takes the form,

$$\frac{\partial I(z)}{\partial z} = BN(z)I(z) \Rightarrow B\left[\Delta N_{max} - \frac{2I(z)\Delta t}{h\nu L}\right]I(z) \tag{6.28}$$

Using

$$\text{total energy density as irradiance: } I_{max} = I(0) + \frac{\Delta N(0)h\nu L}{2\Delta t} \tag{6.29}$$

$$\text{total energy density as population inversion: } \Delta N_{max} = \Delta N(0) + \frac{2I(0)\Delta t}{h\nu L} \tag{6.30}$$

Equations (6.28)-(6.30) have the exact solution,

$$I(L) = \frac{I_{max}I(0)}{I(0) + [I_{max} - I(0)]e^{-B\Delta N_{max}L}} \tag{6.31}$$

Gain has built-in energy conservation!

Calculate the Franz-Nodvik gain from Eq. (6.31) and complare with GLAD calculation for the parameters of Table 6.2.

7. Waveguides and Fiber Optics

GLAD model dielectric and reflecting-wall waveguides

- Dielectric waveguides
  - describe the index of refraction in (x,y) slices (phase sheets)
  - vary the phase sheets along the z-axis
  - result: arbitrary n(x,y,z) defined
- Physical optics argument
  - light tends to follow the higher index, beam tends to focus
  - diffraction tends to cause the beam to spread
  - balance of focusing and diffraction spreading make steady-state mode propagation
- Geometrical explanation
  - angles less than the critical angle reflected at core by total internal reflection (TIR)(do not attempt to analyze dielectric waveguides by rays)

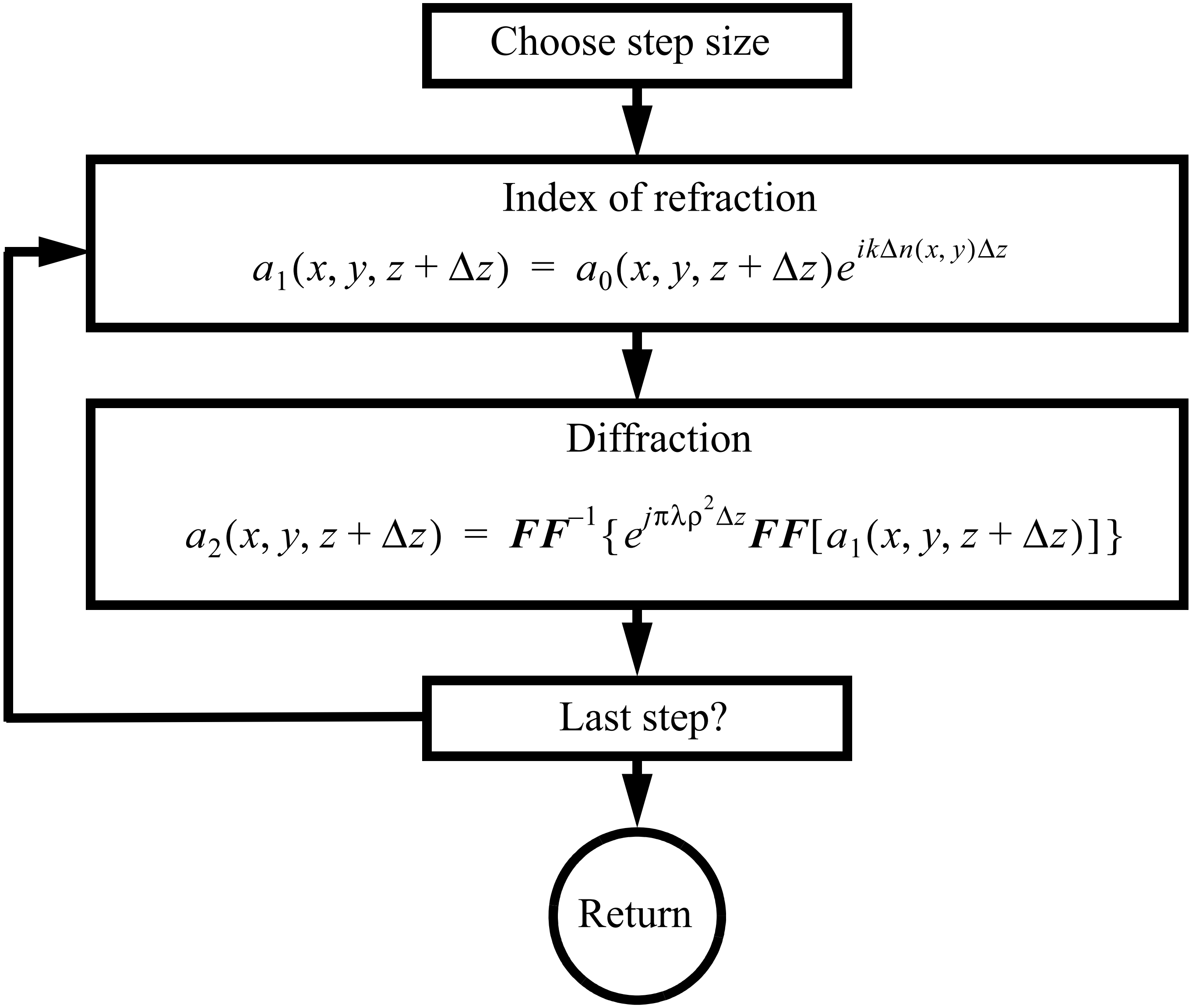
3D treatment or 2D treatment?

- One may perform detailed 3D modeling of fibers and waveguides in GLAD
- For slab waveguides (various rib waveguide forms):
  - 3D treatment includes detailed rib structure
    - scalar BPM for modest core-cladding index differences
    - vector and/or wide angle BPM for high core-cladding index differences
  - 2D treatment, effective index method allows treatment by one-directional arrays with an index n(x) to define the gradient index waveguide structures.

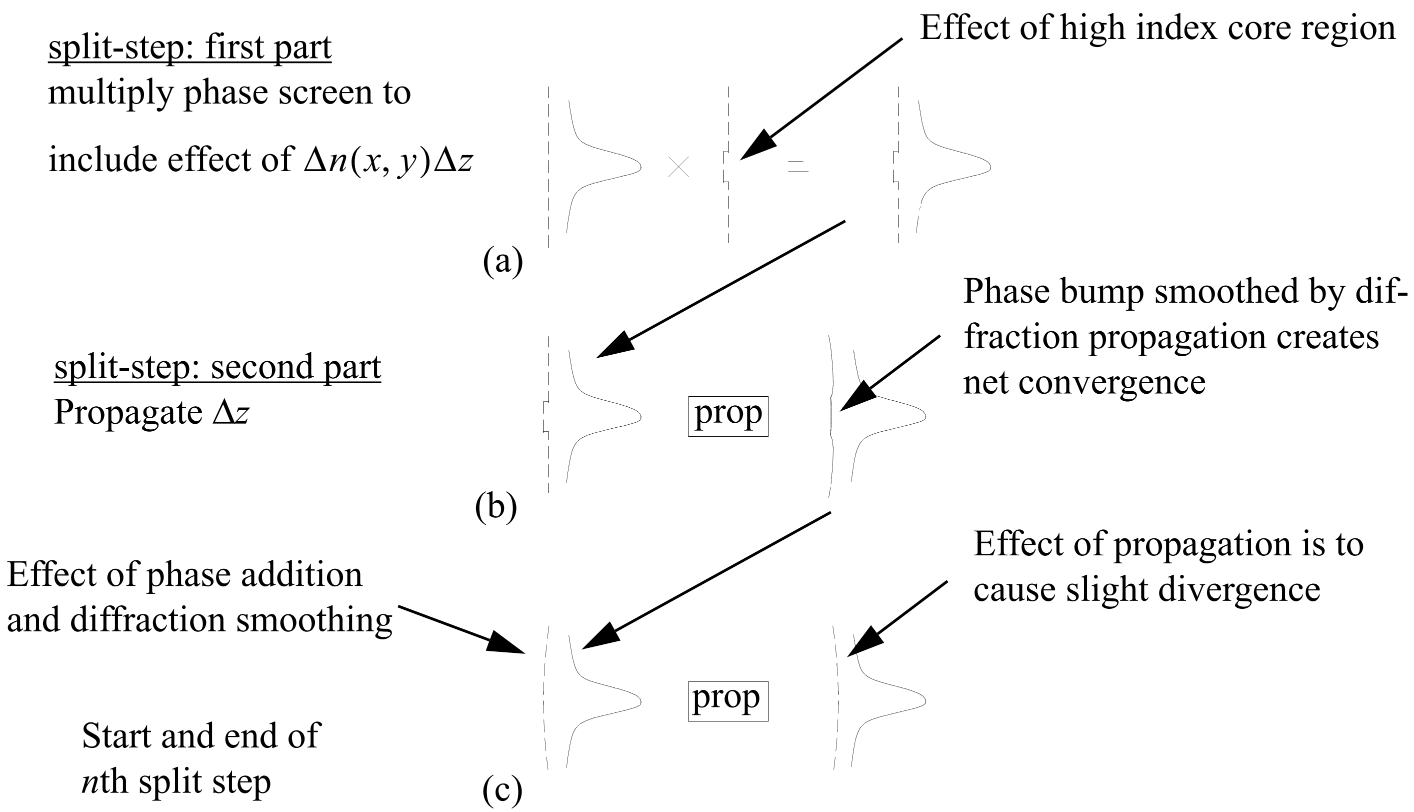
Straight, circular, step index waveguide: 3D treatment

- Ex86a, straight fiber
  - 3D treatment
  - transient and steady state behavior observed
  - implement through command language
    - allows bends, tapers, splitting, joining, coupled cores, fiber lasers, directional couplers, in- and out-coupling
    - may be accelerated by built-in function
    - may be accelerated by effective index, 2D treatment
  - implemented by split-step method

Split-step implementation of waveguide

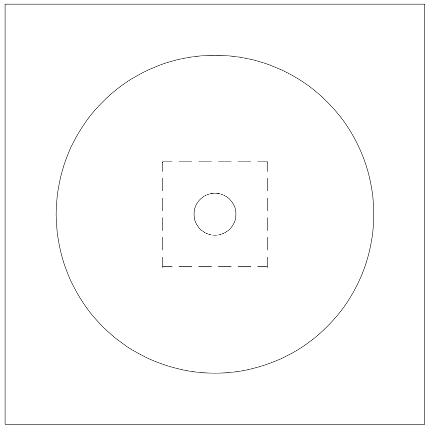


Split-step method for waveguide propagation

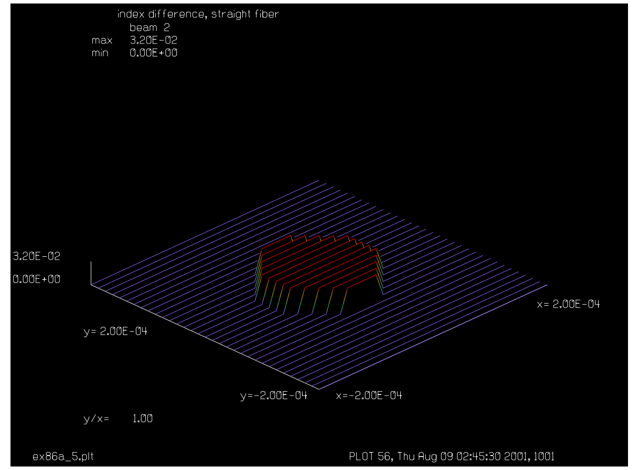


Split step method. Fig. (a) shows the first part of the first split-step. The high index region creates a phase bump on the wavefront. Fig. (b) shows the second part of the first step where both the phase pump and the amplitude are smoothed out somewhat. Fig. (c) shows that the accumulated effect of phase addition and diffraction spreading creates a very slightly converging wavefront that becomes very slightly diverging after the propagation step of Δz. Converging and diverging effects exactly balance for an eigenmode.

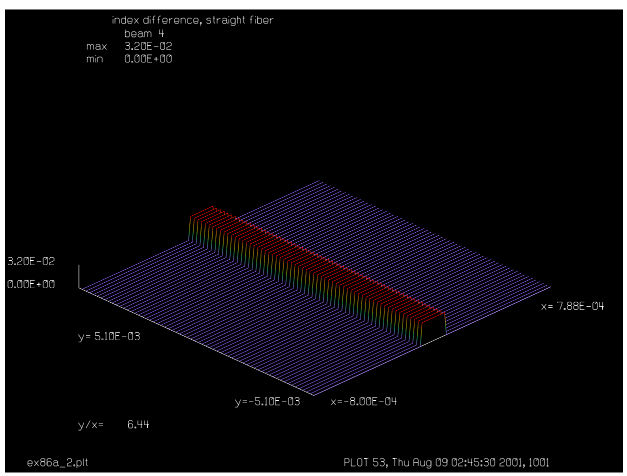
Straight fiber



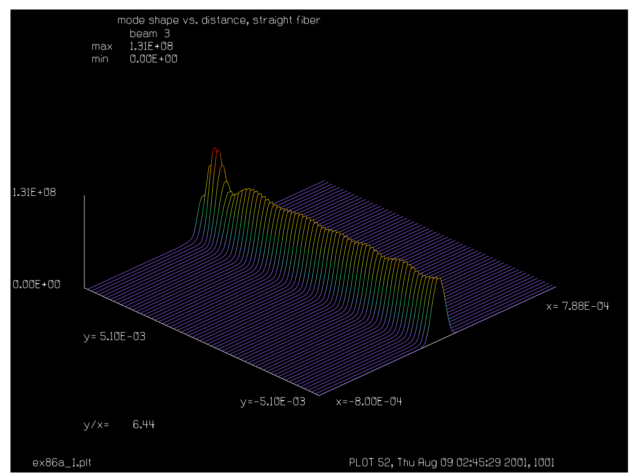
Core region dia. 1.6µ, absorbing wall dia. 12µ, and array of width 16µ.



Index distribution of .032 to scale of dashed square.



Index difference distribution. Array is 100µ long by 16µ wide and n = 0.032.



History of irradiance profiles. Mode stabilizes in about 20µ.

Speeding up the calculation

- time ex86a using:

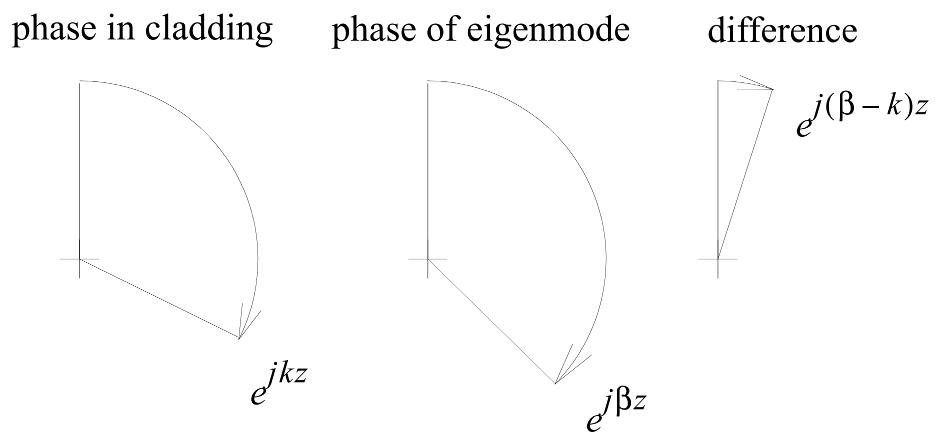
```
time/i
macro step/512
time
```

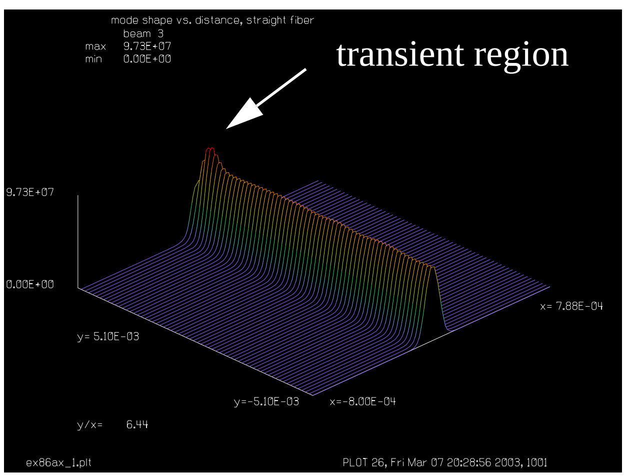
- Eliminate output by adding `write/off` above macro step and write/on after
  - check time without writing to screen or making plots
- move operations on beam 2 outside the macro (see ex86ay.inp). Repeat time test.
  - `clear`, `clap`, `int2phas`, `split` to form beam 2 operations
  - use `mult/beam 1 2` in macro
- convert ex86a to 2d form (see xex86axx.inp). The 2D model is the equivalent of propagation in for a rib waveguide on a slab (not a round fiber).
  - create a new file
  - change 512 x 512 array to 512 x 1 array for beam 1 and beam 2
  - change `plot/liso` to `plot/x/i` for beam 1 and beam 2
  - change the `copy/row` command to replace nline/2 + 1 with 1
  - time the 2D code
- What is the ratio of improvement between 3D and 2D models. To what do you attribute the degree of improvement in speed?

Direct observation of the propagation constant: ex86ax.inp

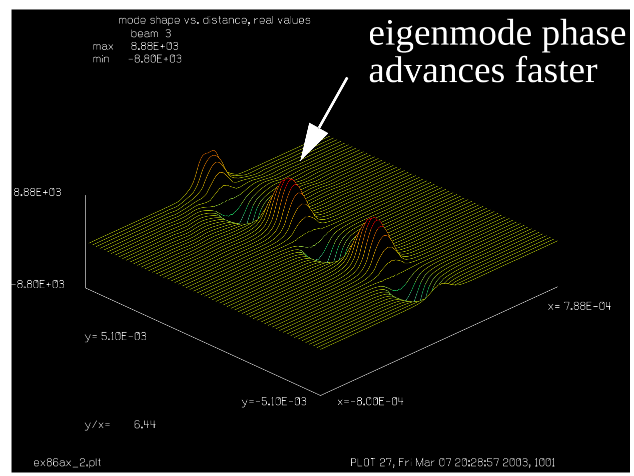
- Typically we use a coordinate system that moves with the beam, e^{jkz}, based on the index of the cladding, n_0
- The eigenmode phase advances fasterthan coordinate system—display with

$$\mathrm{Re}[e^{jk(\beta - k)z}] = \cos[(\beta - k)z] \qquad (7.1)$$





Intensity of beam, $|A|^2$, in straight fiber.



Real part of complex amplitude showing cyclical behavior at the rate $e^{jk(\beta - k)z}$

Displaying the phase advance of the propagation vs. base index, 3D case

- Change `plot/liso/intensity` to `plot/liso/real`
- Estimate the phase advance per step

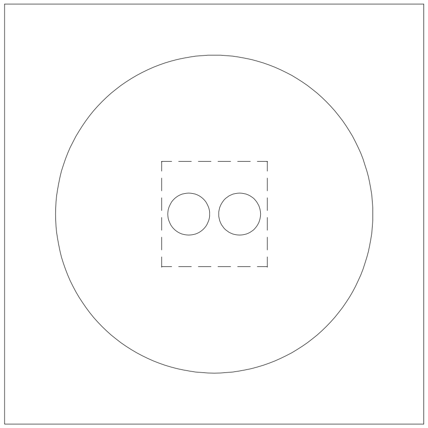
About 2.75 cycles in 512 steps -2 degrees per step

- Try to compensate the phase advance using `phase/piston ibeams phsdeg`
  - Try +2 degrees and -2 degrees

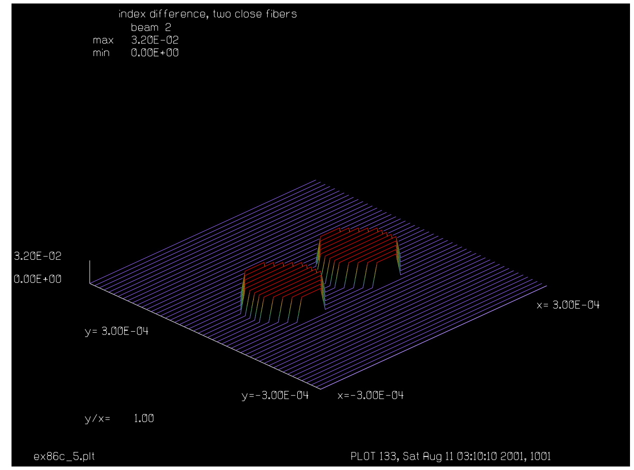
Repeat the experiment for the 2D case: xex86axx.inp

- Is the phase advance the same as the 2D case? What does this mean?

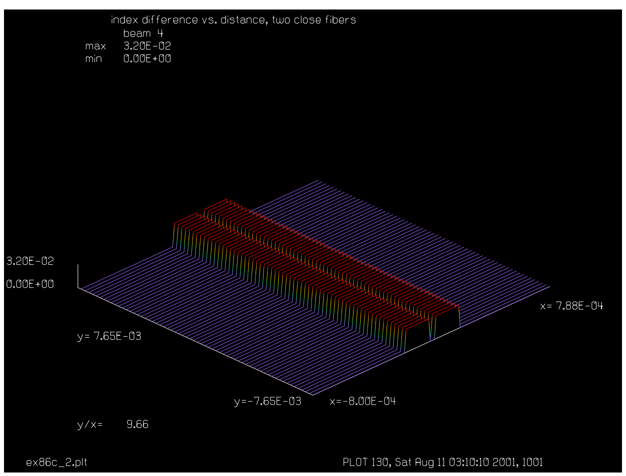
Closely spaced straight cores



Two core regions of dia. 1.6μ , separated by 4μ.

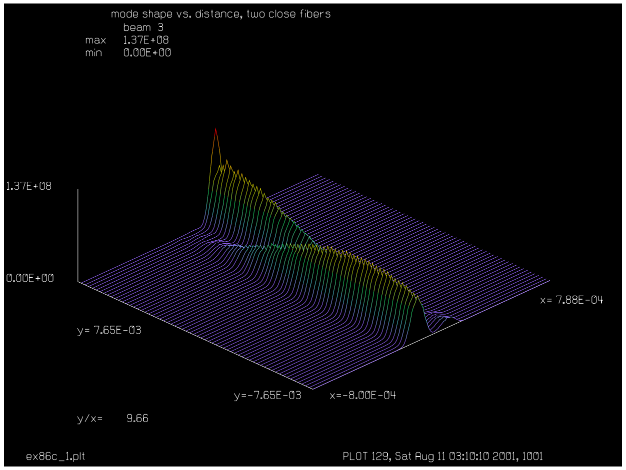


Index distribution of .032 to scale of dashed square in figure to the left.

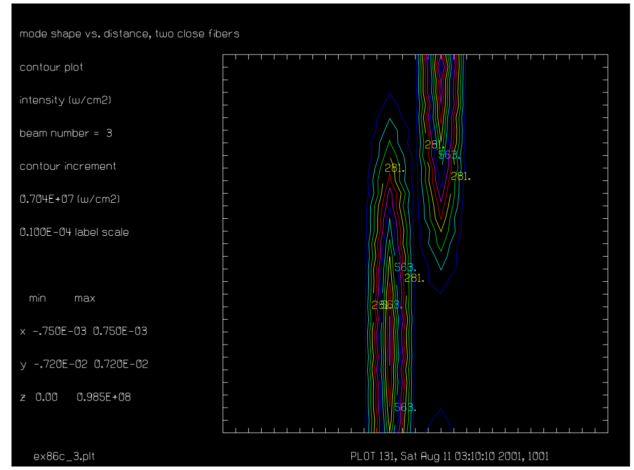


Index difference distribution. Array is 300μ long by 16μ wide and n = .032.

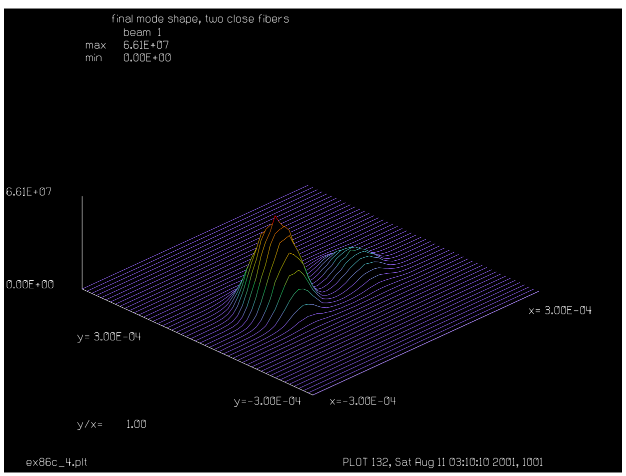
Closely spaced straight cores exchange energy



History of irradiance profiles. Mode oscillates between the two core regions.



Contour plot of irradiance profiles, showing oscillation of mode between two cores.



Final mode shape, showing mode partly in each of the cores. Same scale as top right, last page.

Code for two closely spaced cores: ex86cx.inp

```
mem/set/b 3                             # requste 3 megabytes
variab/dec/int pass count nlinex nliney # define variable names
dist = 3e-5                             # step length
nlinex = 128                            # propagating array width
nliney = 512                            # length of history arrays
shift = 1.e-4
title mode shape vs. distance, two close fibers
plot/watch ex86c_1.plt
array/s 1 nlinex                       # beam 1, propagating beam
nbeam 2 data                           # beam 2, index distribution
nbeam 3 nlinex nliney data             # beam 3, history of mode shape
nbeam 4 nlinex nliney data             # beam 4, history of index profile
units/field 0 8e-4                     # field half-width of 8 microns
variab/set units 3 units
units/set 3 units dist                 # set units of history arrays
units/set 4 units dist
wavelength 0 .6328 1.5                 # set wavelength and cladding index
clear 1 .032
clear 2 .032
clap/c/c 1 .8e-4 xdec=[-shift]         # make left core
clap/c/c 2 .8e-4 xdec=shift            # make right core
add/inc/con 2 1                        # sum the two cores
clear 1 1
clear 3 0
clear 4 0
set/density 64
alpha = 2.*pi*1.5*dist/.6328e-4        # phase coefficient per step
macro/def step/o
   pass = pass+1
   count = count+1
   zreff 1 0
   geodata/set 1 0 0 1e-4 1e-4 1 1
c
```

Code for two closely spaced cores (cont'd): ex86cx.inp

```
c  take two steps together
c
   int2phas/two 1 2 alpha             # implement index in beam 2 on beam 1
   split/cir/in 1 6e-4 6              # absorbing boundary for the array
   dist dist 1
   int2phas/two 1 2 alpha             # implement index in beam 2 on beam 1
   split/cir/in 1 6e-4 6              # absorbing boundary for the array
   dist dist 1
   if pass = 1 energy/norm 1
   copy/row 1 3 [nlinex/2+1] pass     # store mode in history array
   copy/row 2 4 [nlinex/2+1] pass     # store index profile in history array
   if count = 4 then
      plot/watch plot1.plt
      plot/l 3 ns=64                 # plot every 10 steps
   endif
   if count = 8 then
      plot/watch plot1.plt
      plot/c 3 ilab=0                 # plot every 10 steps
      count = 0
   endif
macro/end
set/density 32 32
gaus/c/c 1 1 1e-4 decx=shift          # inject gaussian mode into one core
pass = 0
macro step/nliney                     # propagate nliney times
plot/watch ex86c_1.plt
plot/l 3 ns=64
plot/watch ex86c_2.plt
title index difference vs. distance, two close fibers
plot/l 4 ns=64 h=.1
plot/watch ex86c_3.plt
title mode shape vs. distance, two close fibers
plot/con 3 con=14
```

Code for two closely spaced cores (cont'd): ex86cx.inp

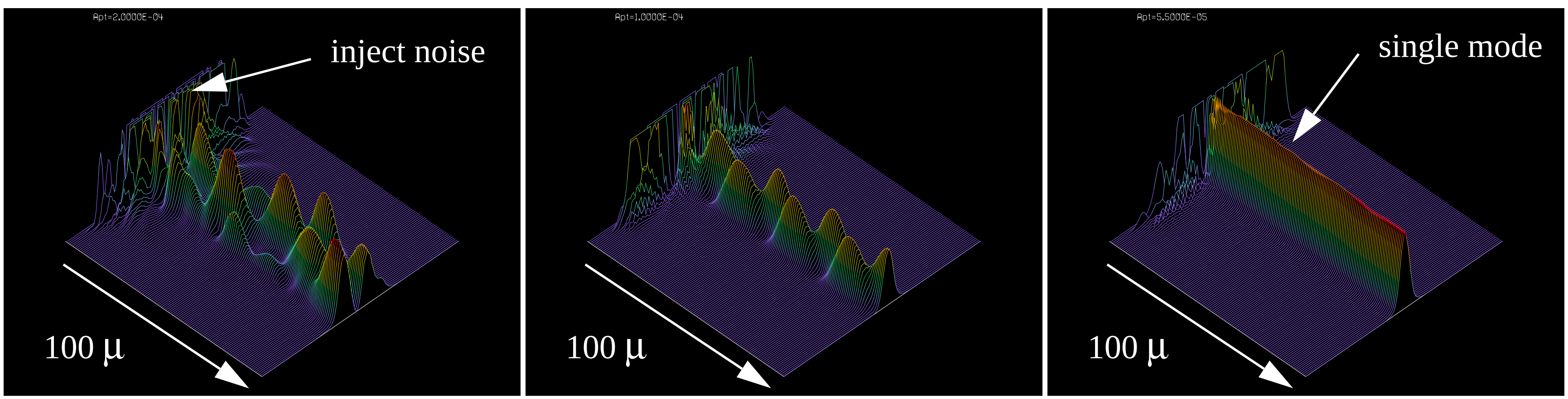
```
plot/watch ex86c_4.plt
title final mode shape, two close fibers
plot/l 1 xrad=3e-4 ns=64
plot/watch ex86c_5.plt
title index difference, two close fibers
plot/l 2 xrad=3e-4 ns=64 h=.1
energy 1
```

Effect of guide width on number of modes, transient regime: ex86a55.inp

- Reducing the width of the waveguide to achieve single mode performance
- Inject random noise into the waveguide
- Mode behavior stabilizes in about 20 microns (for this device)

$$\upsilon < \frac{\pi}{2} \rightarrow \left(\frac{2\pi a}{\lambda}\right)^2 (n_f^2 - n_s^2) \rightarrow a < \frac{\lambda}{\sqrt{8\pi(n_f^2 - n_s^2)}} \qquad (7.2)$$

For $\lambda = 0.6238\mu$, $n_f = 1.532$, $n_s = 1.500$, $a < 0.405\mu$ for single mode operation



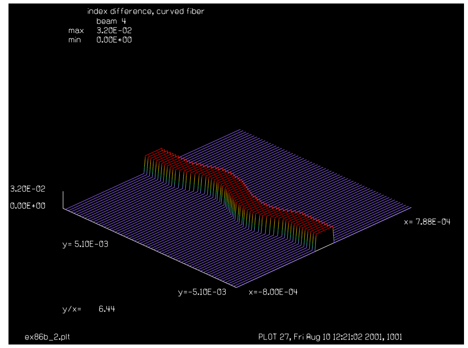
$\lambda = 0.6238\mu$, $2a = 3.2\mu$, $\Delta = 0.21$. About 3 modes.

$\lambda = 0.6238\mu$, $2a = 2.0\mu$, $\Delta = 0.21$. About 2 modes.

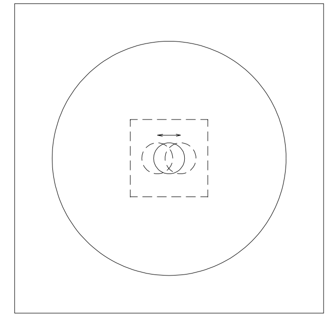
$\lambda = 0.6238\mu$, $2a = 0.8\mu$, $\Delta = 0.21$. About 1 mode.

Waveguide with “S” bend

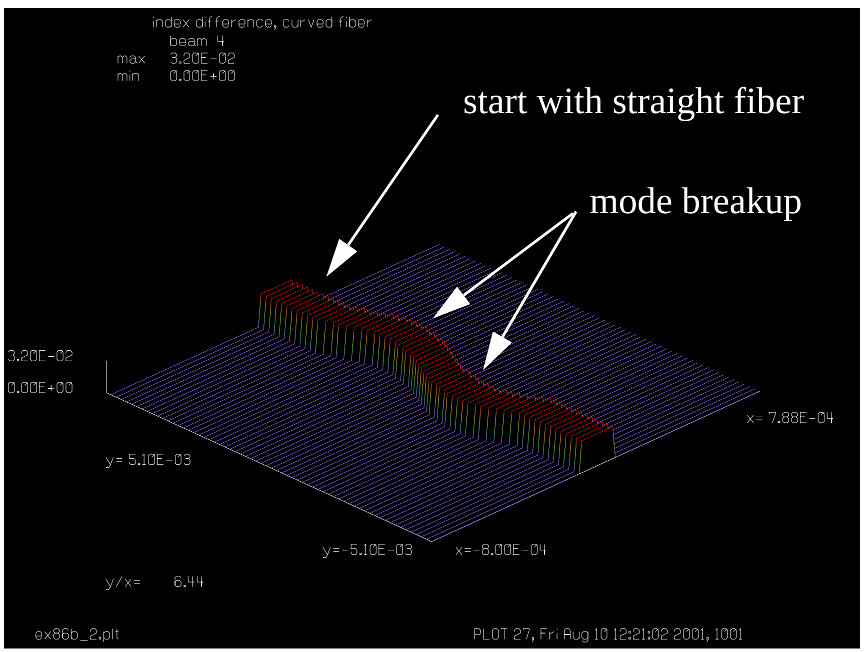
- High index core is moved right and then left to make an “S” bend
- Bend induces coupling into higher-order modes that radiate out of waveguide

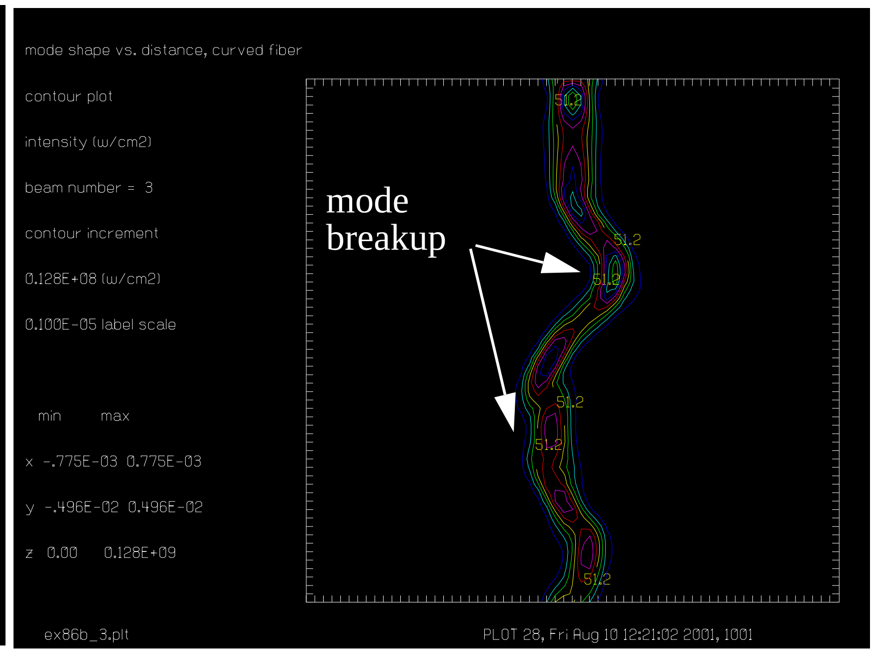


index profile with sinusoidal bend



Core region moves back and forth 6μ to form “s”.

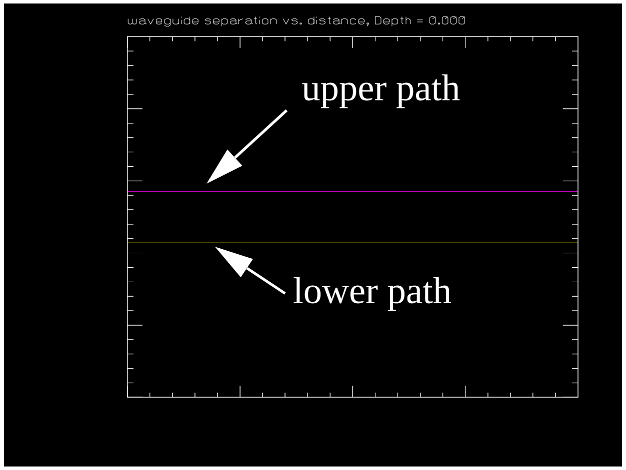




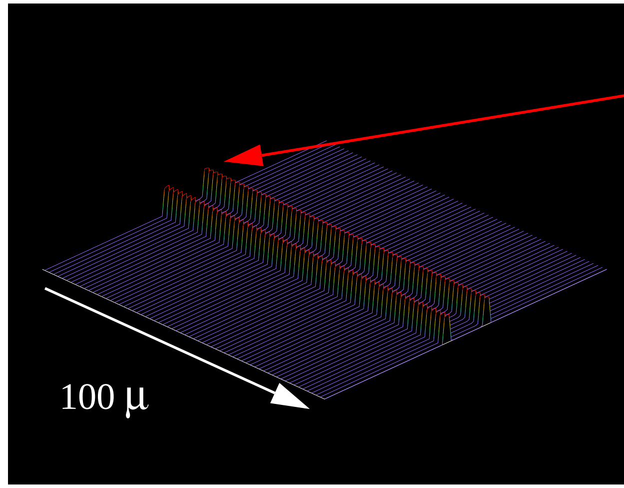
Enhancements

- `Extrude` treats a constant cross section waveguides with path deflections
- Easy definition of bent waveguide paths
- `slab/waveguide`
- Time for a typical slab problem reduced from about 250 times

Coupling between two straight guides: separation of 7µ: ex87a.inp

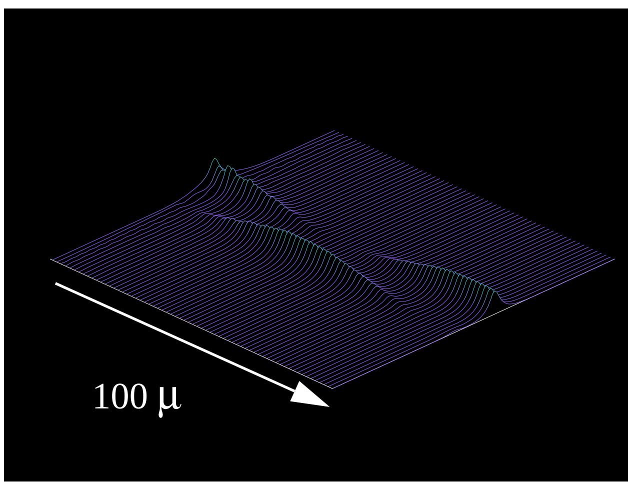


Paths of two close waveguides on a slab.

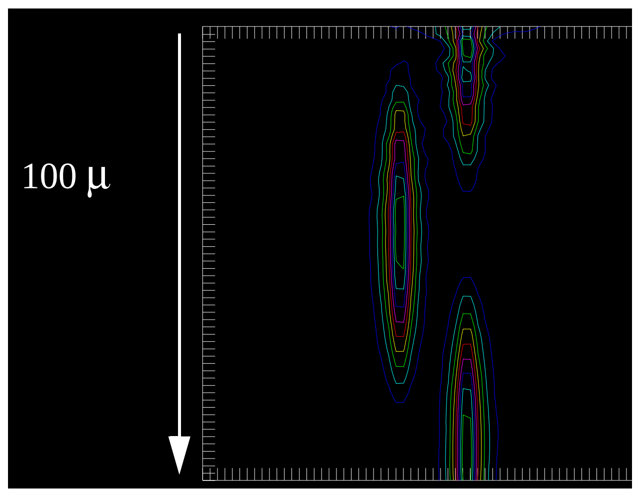


inject light into this waveguide

Index distribution for two close waveguides on a slab.



Light injected into the upper guide is coupled to the lower guide and then back.

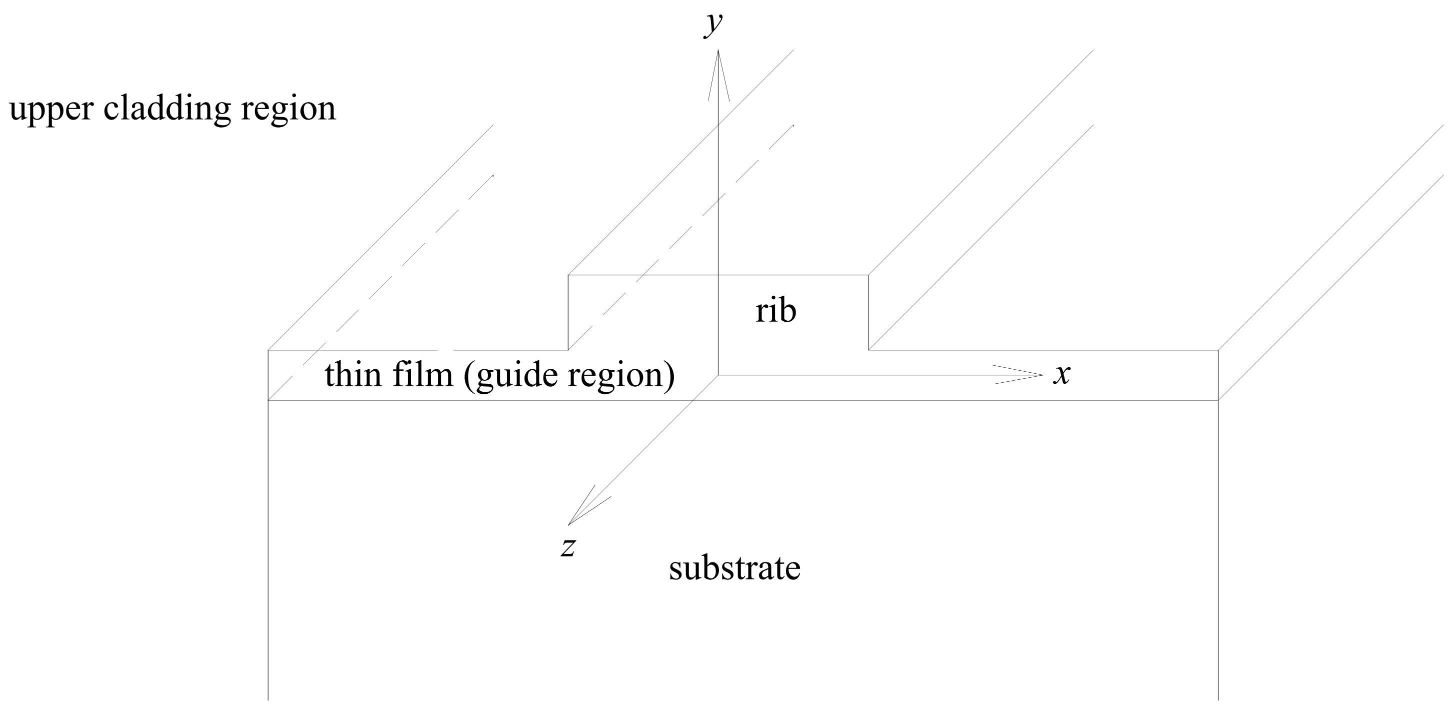


Contour of coupled waveguides shows power coupling between the waveguides.

Waveguides operate by total internal reflection (TIR). Close waveguides couple by evanescent tails of waveguide modes—a form of frustrated total internal reflection (FTIR).

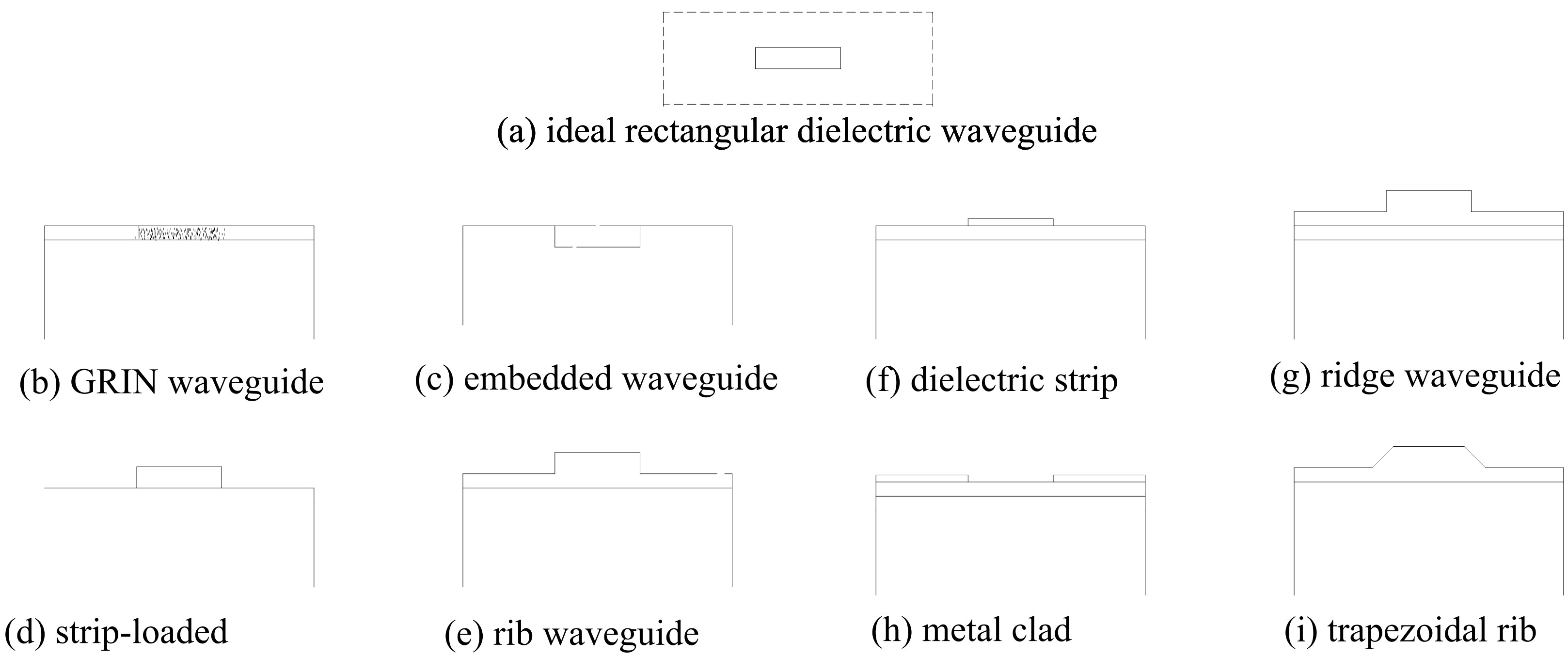
Typical slab waveguide with rib to generate high effective index region

- Effective index method:
  - assume rib just raises index locally in slab
  - treat slab as 2D calculation with gradient index region under slab



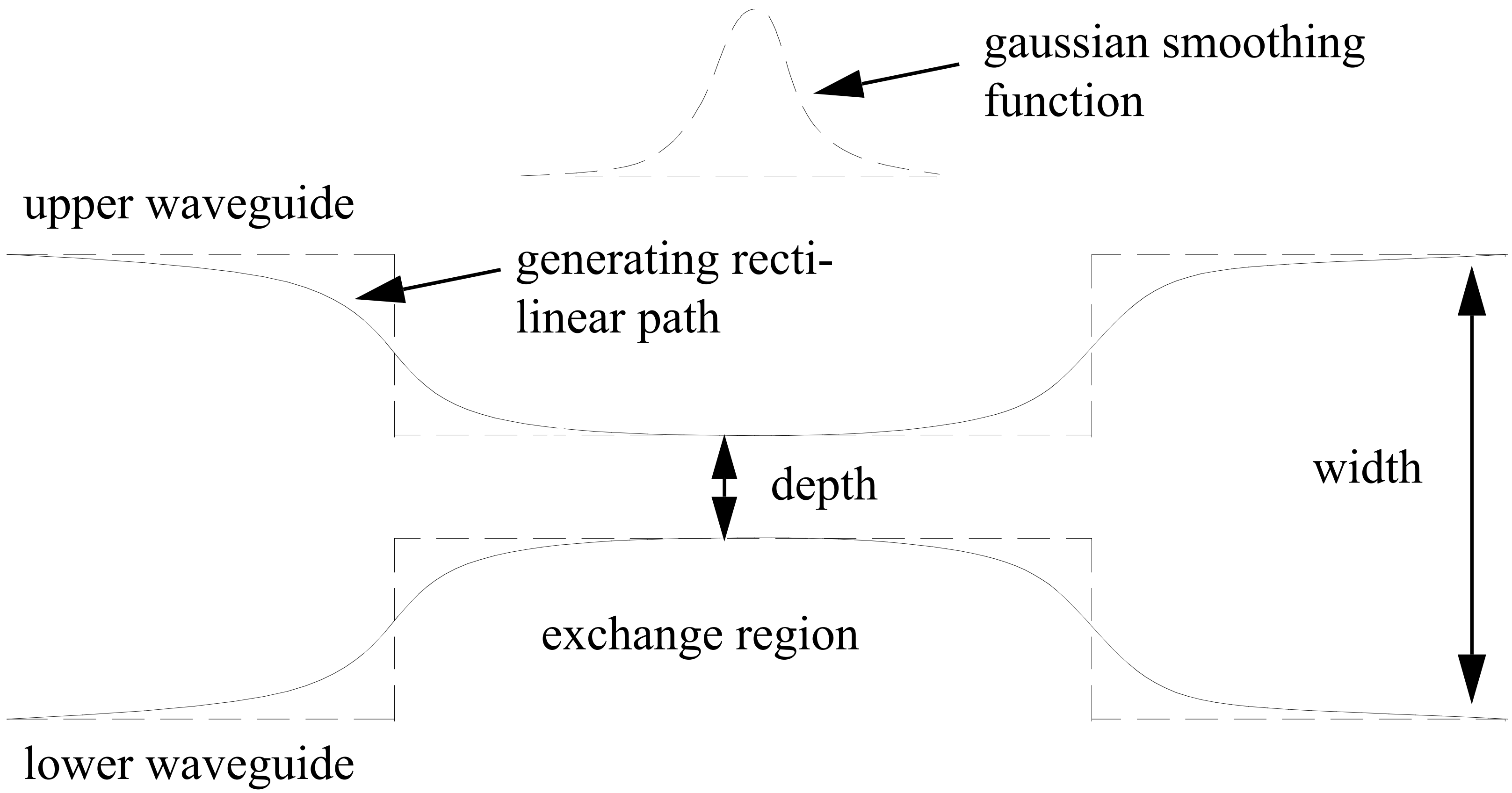
Waveguide with rib to create high index region. The substrate (lower index of refraction) supports the thin film (higher index) waveguide region which confines the optical mode in the y-direction. A thicker region, forming a rib, creates a region of higher index in the thin film waveguide region with the lower index substrate below and the lower index upper cladding layer above. The upper cladding region is often air. The greater thickness of the thin film in the region of the rib leads to a higher propagation coefficient and a corresponding higher effective index. From Koshiba.

Some slab waveguides with ribs to guide in the slab direction



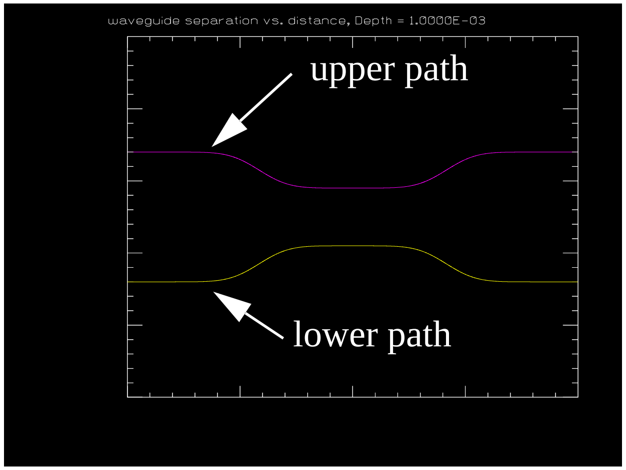
(a) An ideal rectangular waveguide surrounded by cladding. (b) A slab waveguide with a thin film waveguide to confine the light in the y-direction and a high index GRIN region to confine the light in the x-direction. Doping the material to create the GRIN region may not be practical. (c) An embedded high index region — difficult to manufacture. (d) A strip-loaded waveguide traps the light in the x-direction. Does not use a uniform thin film coating of the substrate. (e) The rib waveguide uses a localized thick region of the thin film coating. The thicker region creates a localized higher effective index. (f) A deposited strip acts to locally increase the effective index. (g) Ridge waveguide adds a lower thin film layer. (h) Metal coating of the region in the non-guide areas lowers the effective index to create the guided area. (i) An alternate form of the rib waveguide with trapezoidal cross section. Some figures adapted from Koshiba.

Directional coupler

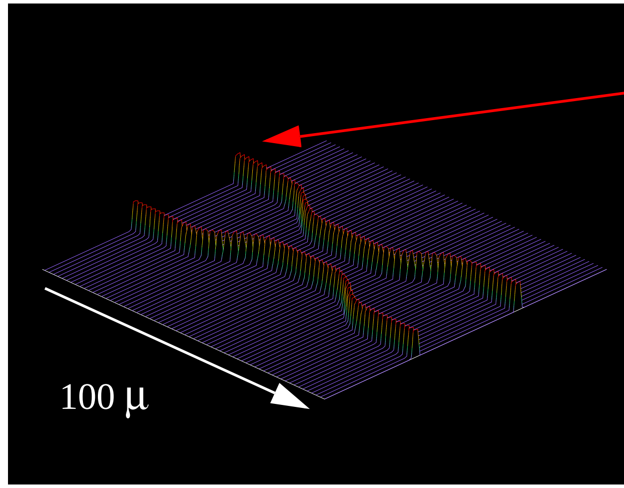


Generating a smooth curve for a waveguide directional coupler to minimize losses in the bend region. The paths for upper and lower waveguides is constructed from rectilinear generating path (dashed) and convolved by a gaussian (shown above as dashed figure) to create smooth waveguide path. The width of the gaussian may be made narrower or wider for tighter or looser curved regions.

Directional coupler, equal power in both legs, depth = 10µ: ex87b.inp

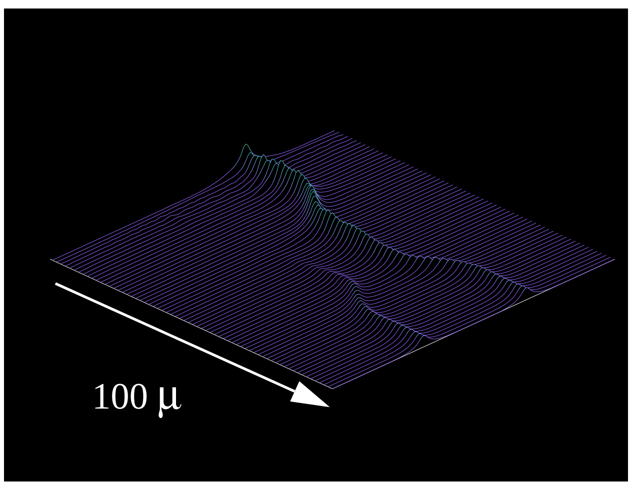


Change depth and length to control coupling efficiency to 50%. Total width 18µ, depth 10µ.

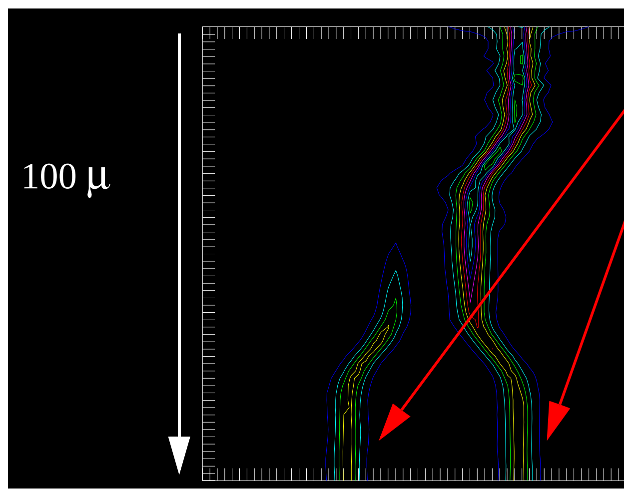


Index distribution.

inject light into this waveguide



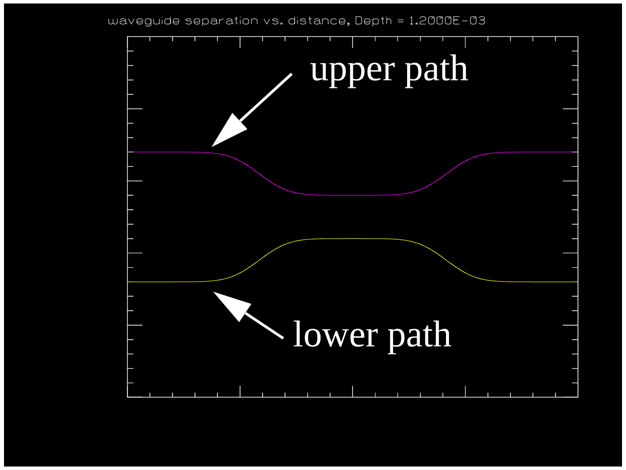
Light injected into the upper guide is evenly divided at the output.



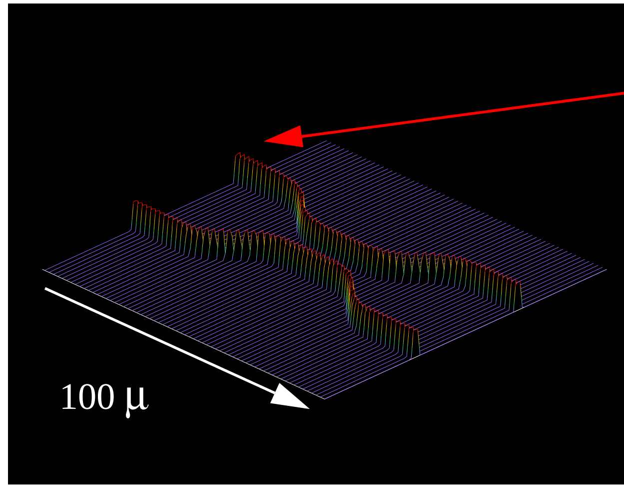
Contour of coupled waveguides.

Depth and length of coupling region controls coupling efficiency. In this case, beam is divided equally into both output waveguides.

Directional coupler set for 100% power conversion, depth = 12μ: ex87c.inp

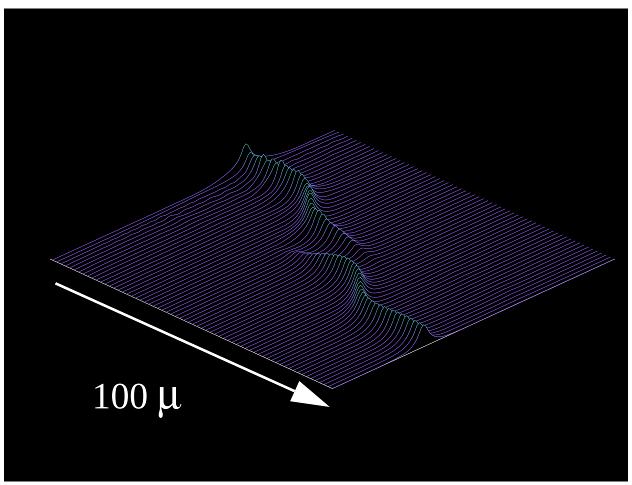


Paths of two waveguides, total width 18μ, depth 12μ.

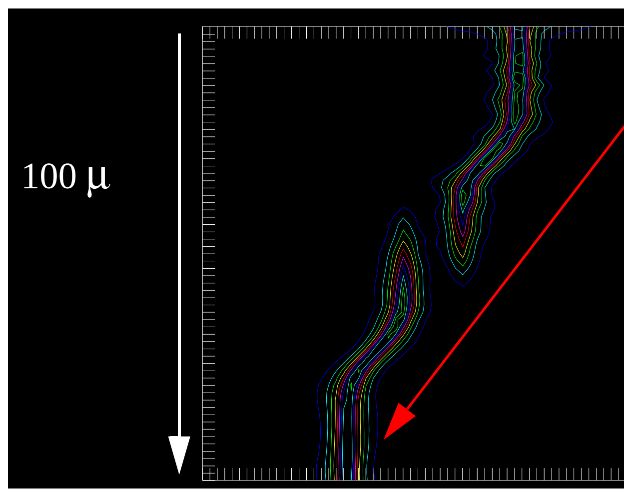


Index distribution.

inject light into this waveguide



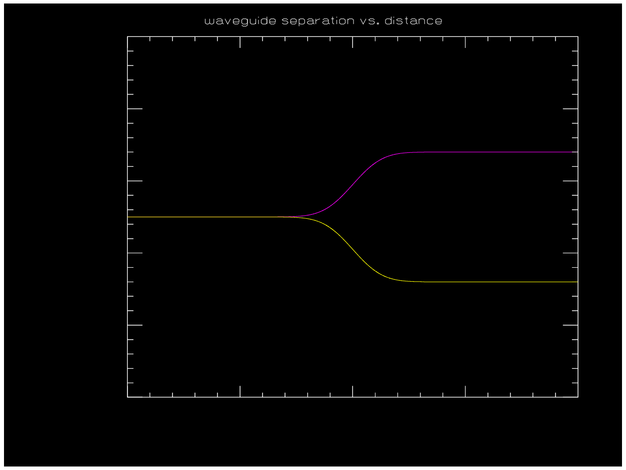
Light injected into the upper guide is completely converted to lower guide.



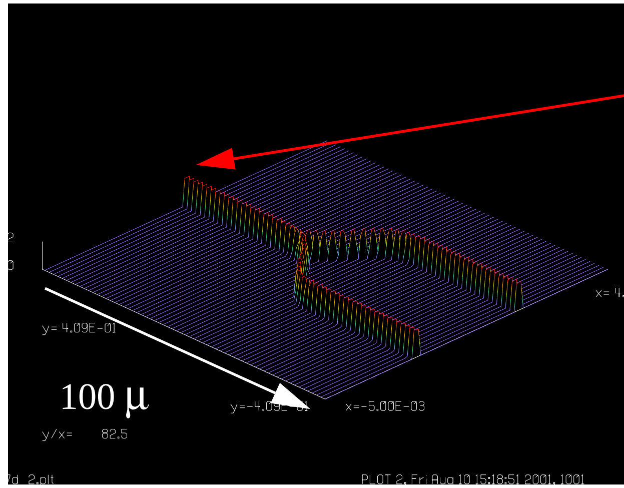
Contour of coupled waveguides.

nearly all light exits in lower guide with depth = 12μ.

Y-splitter, nearly 100% efficient: ex87d.inp

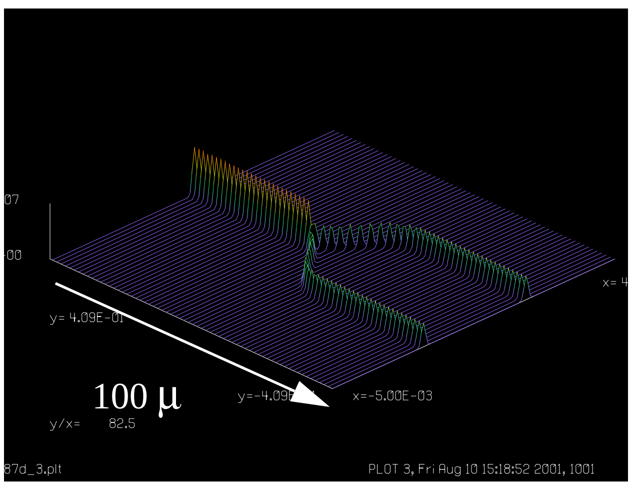


Paths of split waveguide.

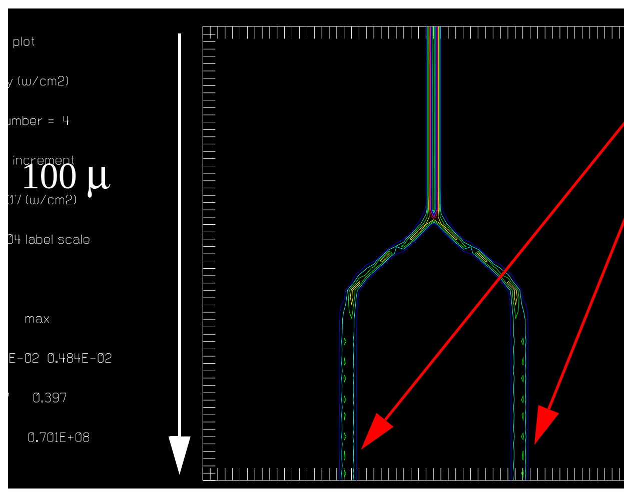


Index distribution.

inject light here



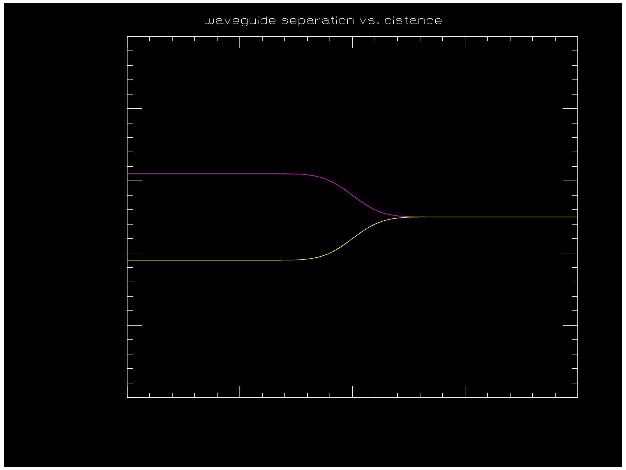
The propagating modes for the Y-junction. Efficiency is 98%.



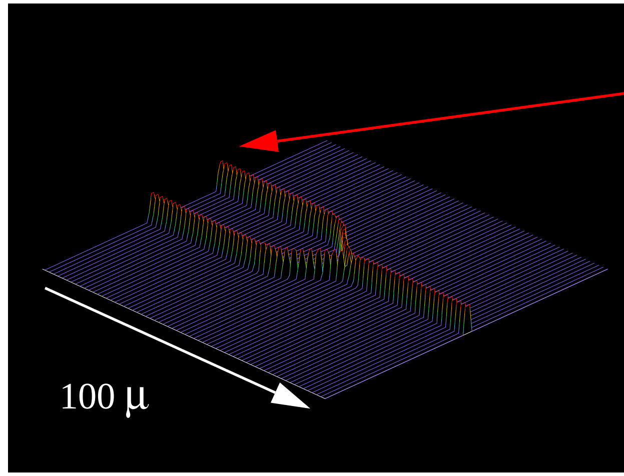
Contour of coupled waveguides.

Energy equal in the two output guides

Y-combiner, single input (not mode matched): ex87e.inp

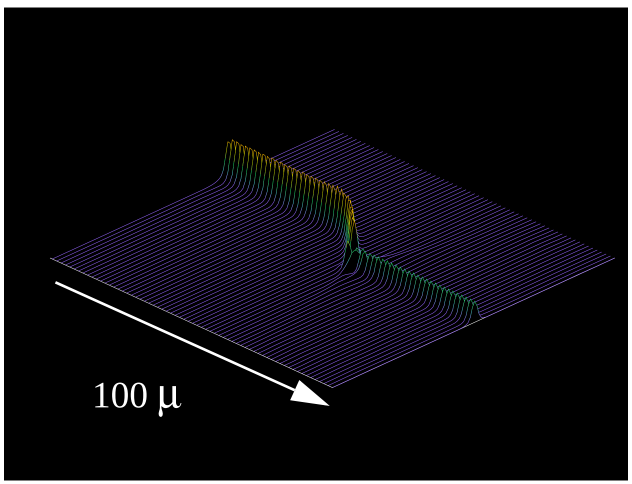


Paths of Y-combiner waveguide.

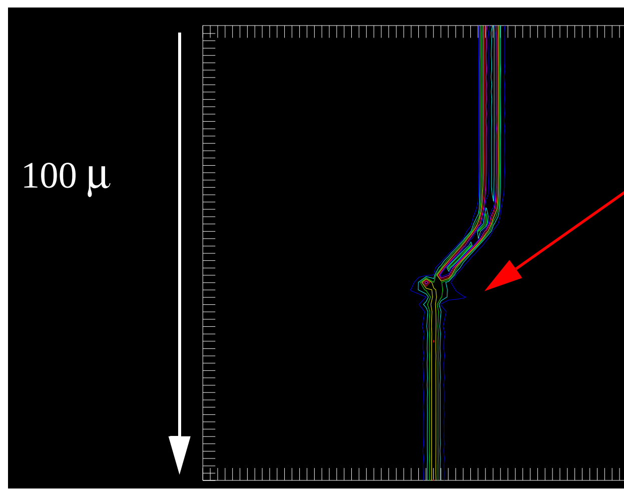


Index distribution.

inject light into this waveguide



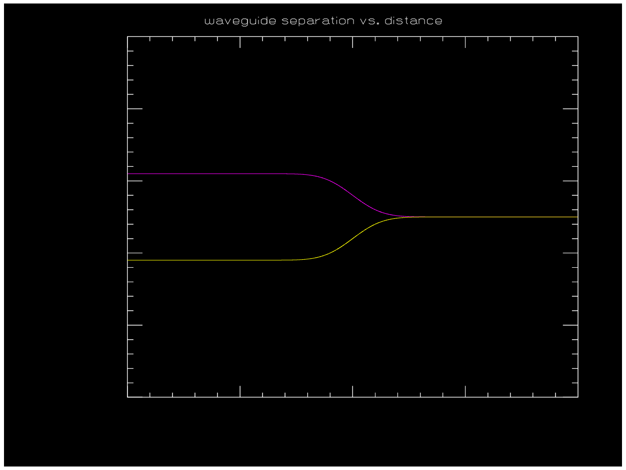
Injection in only the upper guide. Imperfect coupling. Efficiency about 50%.



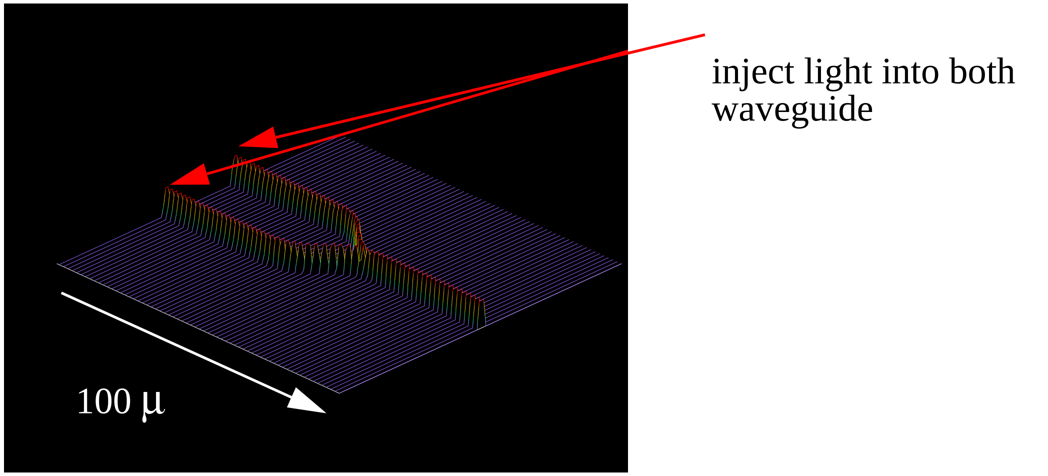
Mode coupling mismatch at after intersection of empty guide.

“missing” input causes loss

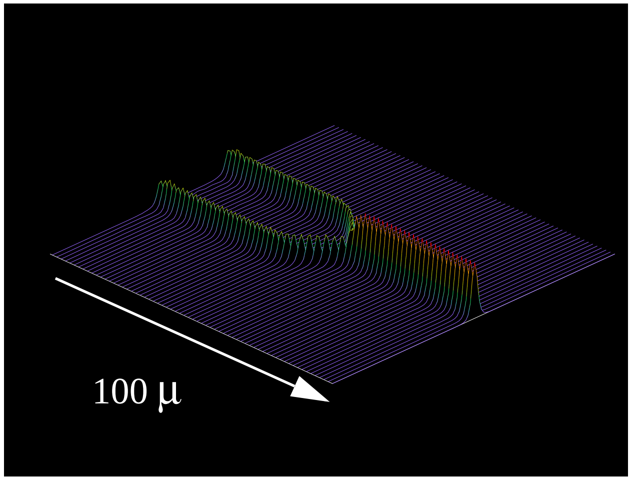
Y-combiner, double input, perfect mode matching: ex87f.inp



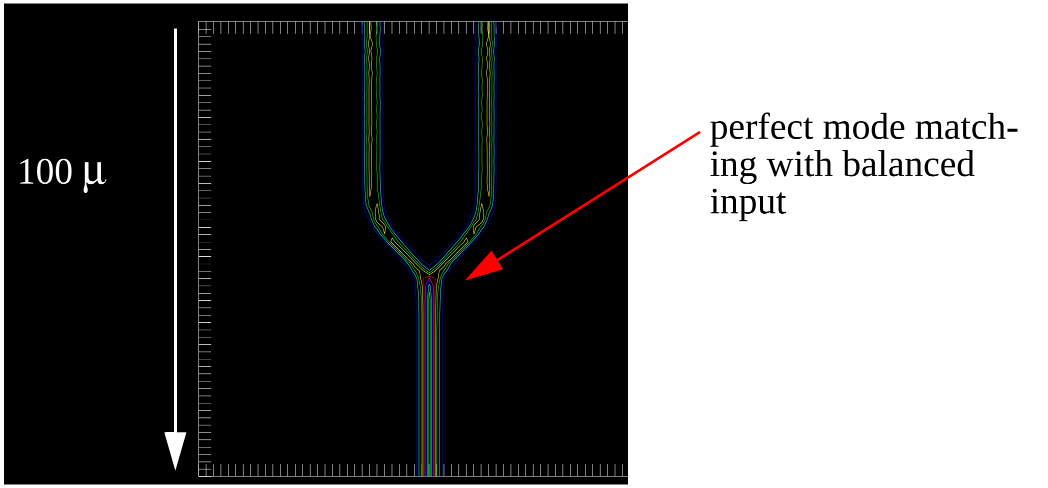
Paths of Y-combiner waveguide.



Index distribution.

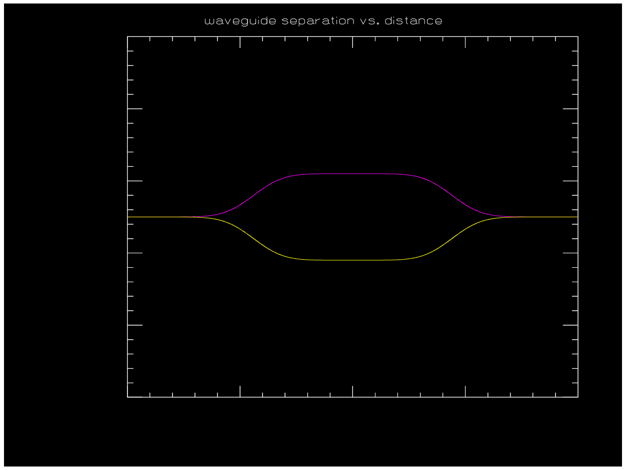


Injection of identical modes into the Y-combiner. Efficiency about 100%.

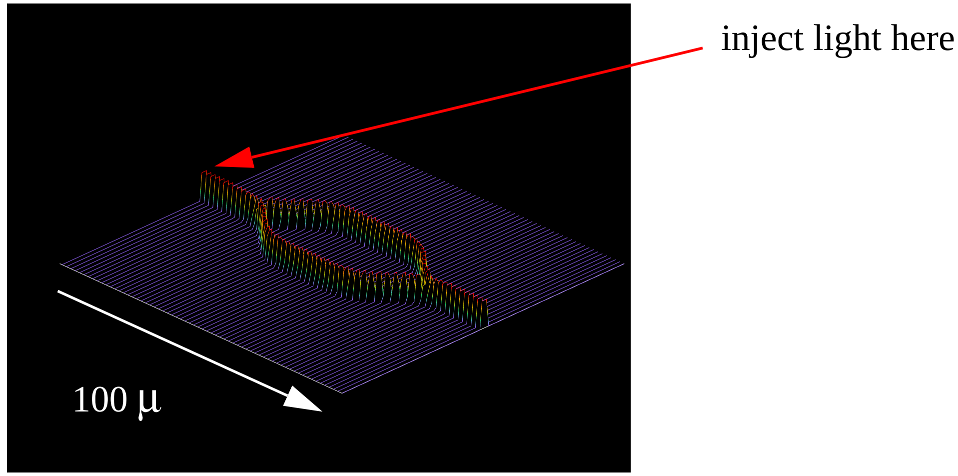


Injection of identical modes into the Y-combiner.

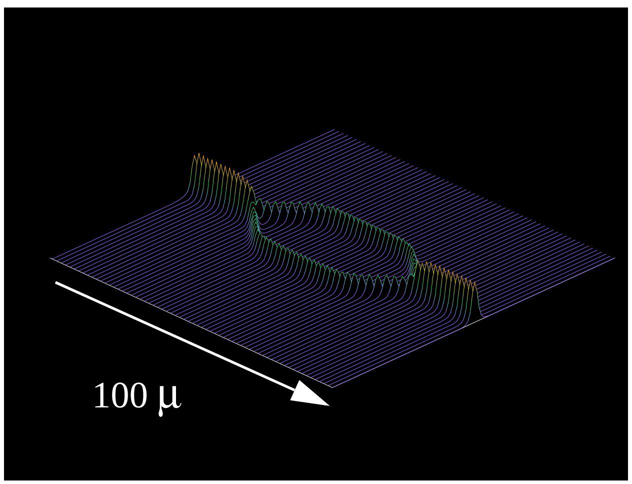
Y-combiner, double input, perfect mode matching: ex87f.inp



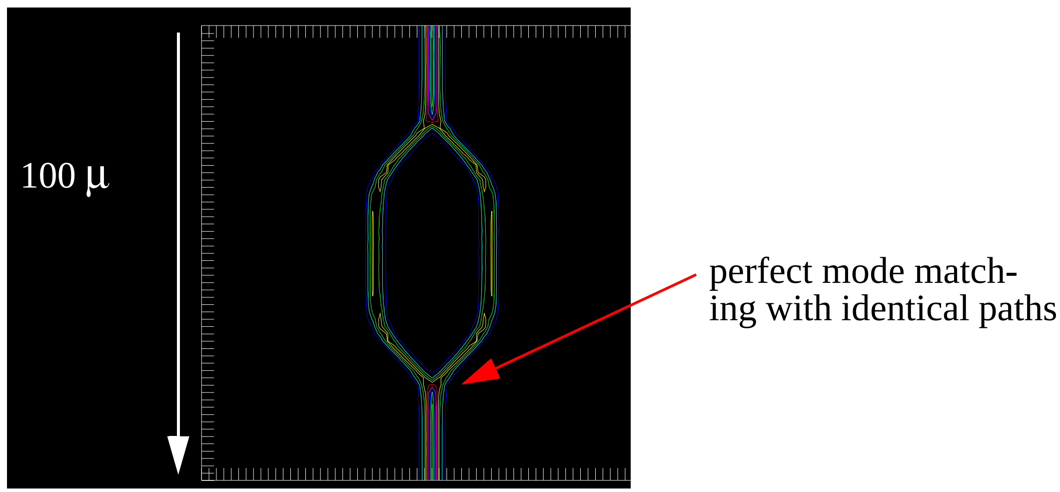
Paths of Y-combiner waveguide.



Index distribution.

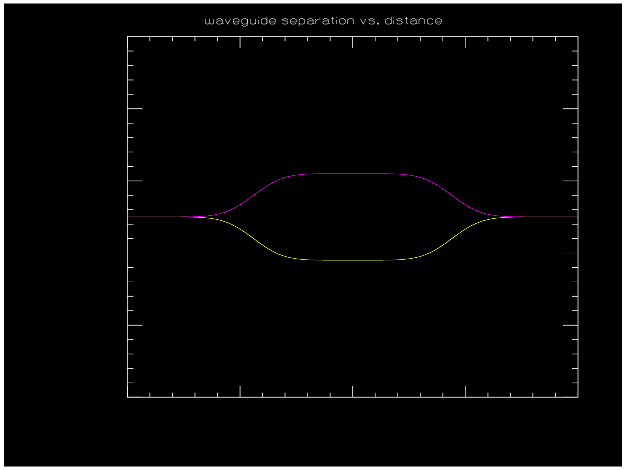


Injection of identical modes into the Y-combiner. Efficiency about 100%.

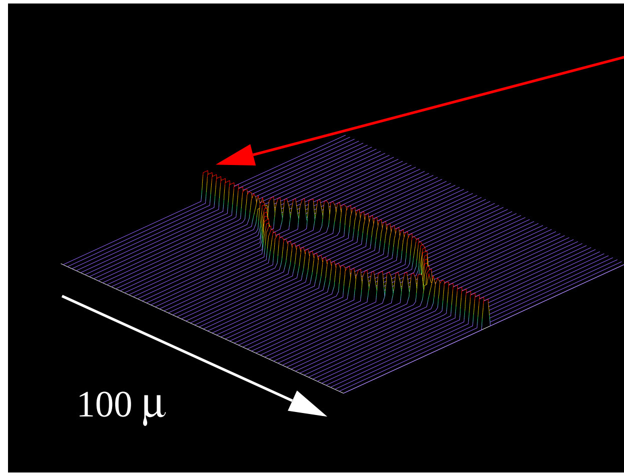


Injection of identical modes into the Y-combiner.

Optical switch: Y-splitter and Y-combiner, switch is <u>on</u>: ex87g.inp



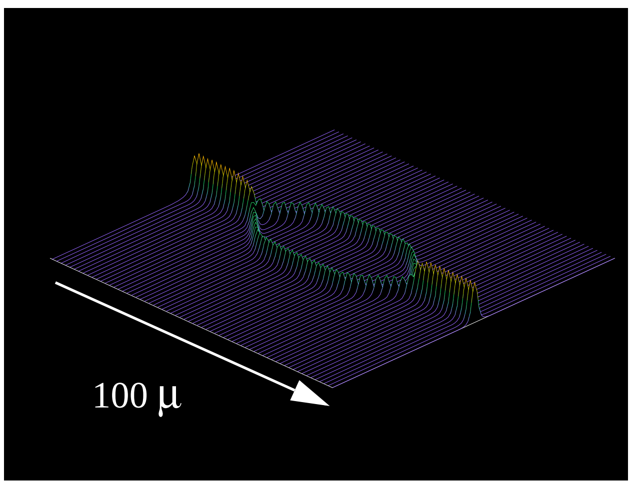
Paths of Y-combiner waveguide.



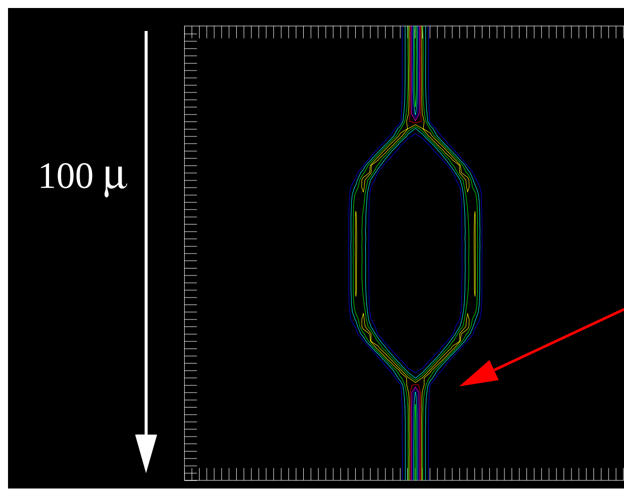
Index distribution.

inject light here

Similar to Mach-Zender interferometer



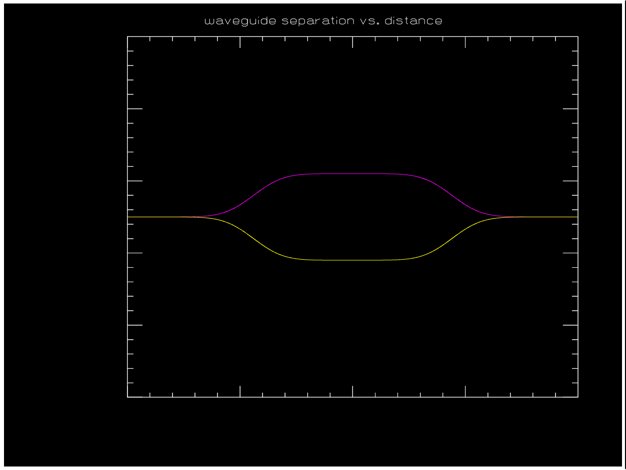
Injection of identical modes into the Y-combiner. Efficiency about 100%.



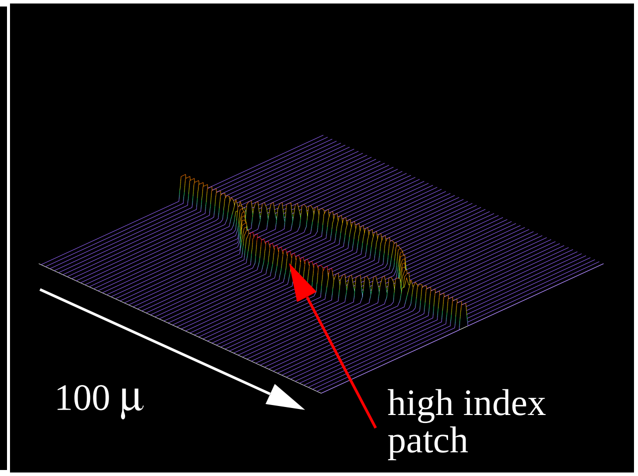
Injection of identical modes into the Y-combiner.

perfect mode matching with identical paths

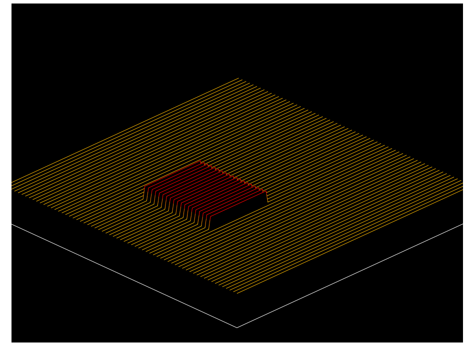
Optical switch: Y-splitter and Y-combiner, switch is <u>off</u>: ex87h.inp



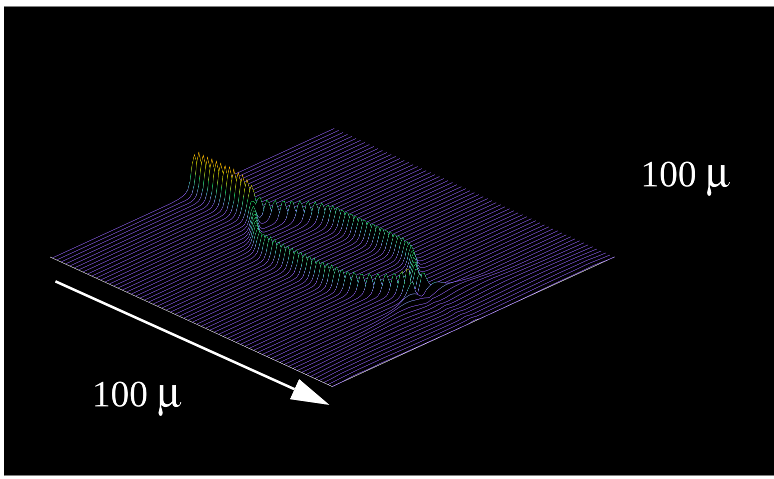
Paths of optical switch in <u>off</u> position, identical to on position.



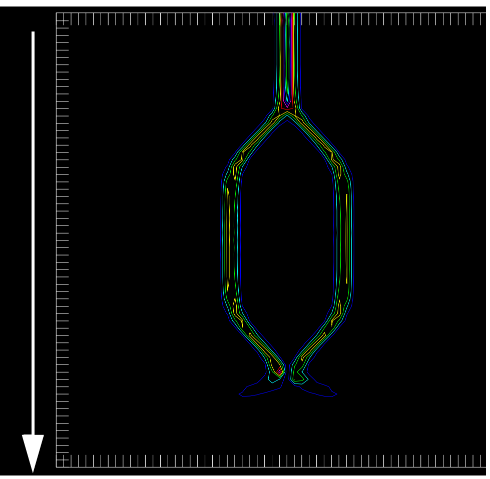
Index distribution.



Region changed to higher index to turn switch off.



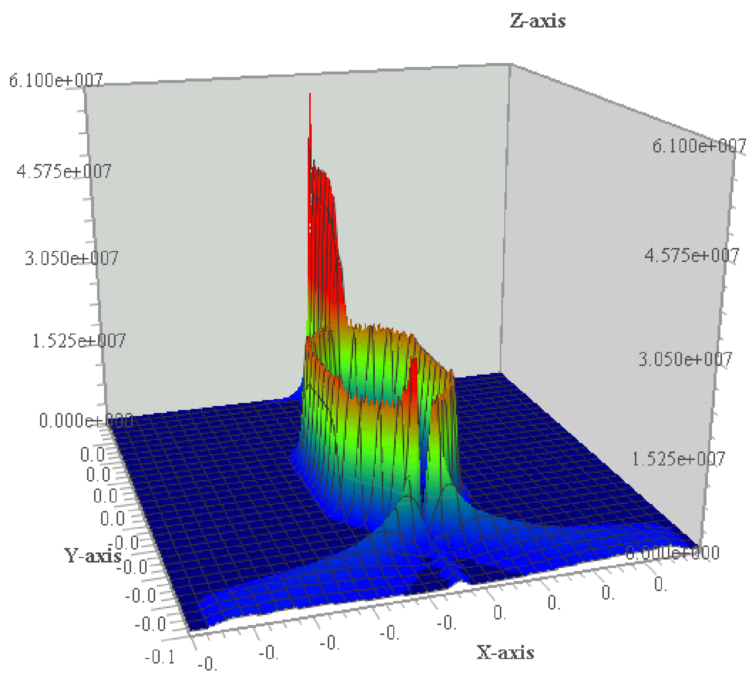
Waveform of optical switch in <u>off</u> position.



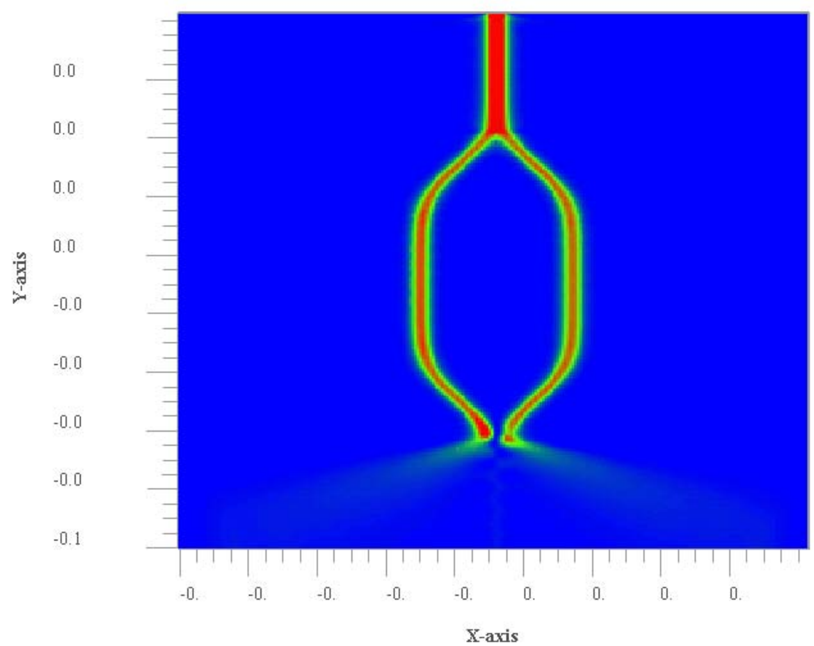
Waveform of optical switch in <u>off</u> position.

Index is raised on left side to change phase. The intensity is identical in the two sides, but the left side is π out of phase, so that cancellation occurs when the beams are recombined.

Photonic switch in the Off position: `plot/bitmap/intensity/arrayvisualizer`



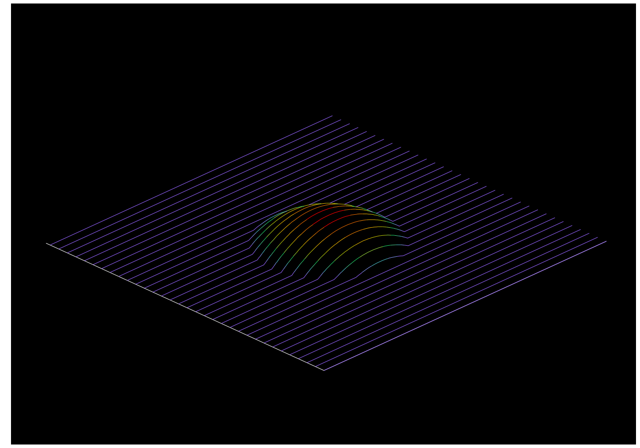
Isometric (view may be changed by dragging).



Bitmap style from Array Visualizer.

Waveguide lens: ex87i.inp

- Direct control of or deform surface (geodesic lens)

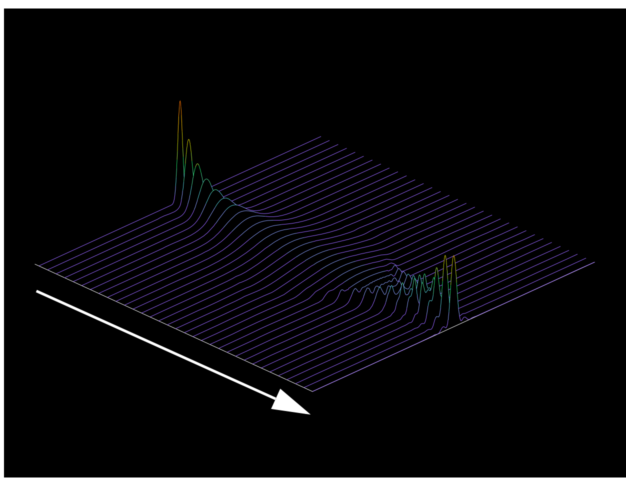


Region changed to higher index for geodesic lens.

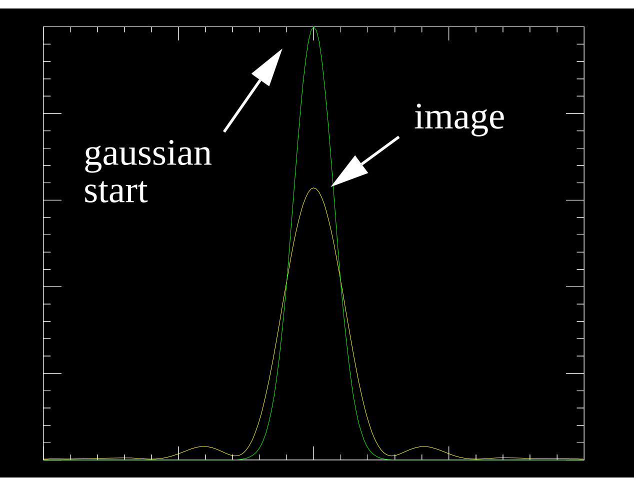
Maxwell's fisheye lens (a perfect imaging device for all conjugate points)

$$n(r) = \frac{n_0}{1 + \frac{r^2}{f^2}} \qquad (7.3)$$

Luneburg lens: $n(r) = n_0\sqrt{1 - r^2/f^2}$ (7.4)



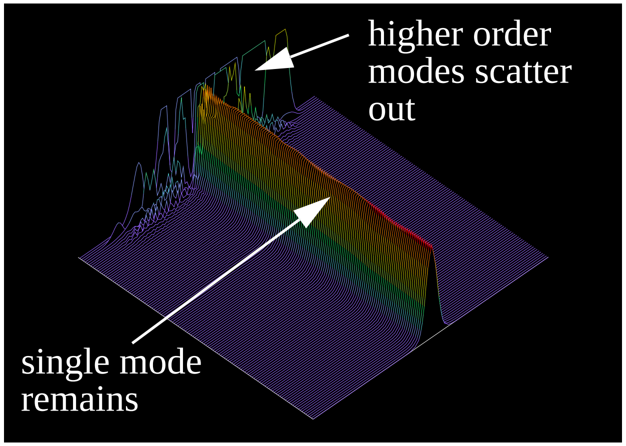
Optical mode vs. distance starting from a gaussian and ending with a near-gaussian.



Comparison of starting gaussian with ending beam—note side lobes due to truncation by lens.

Coupling into a single mode waveguide, overlap integral

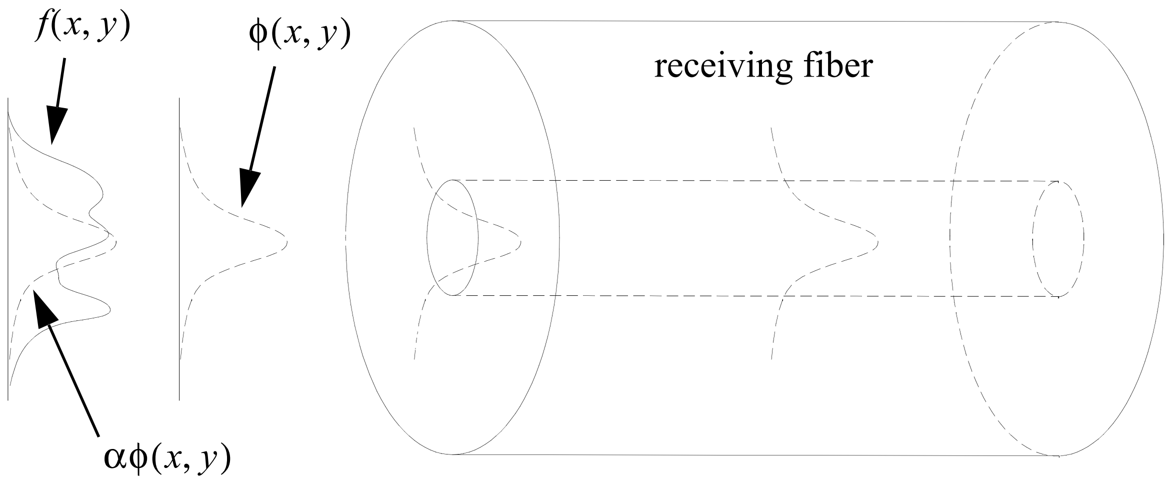
- Arbitrary incident beam $f(x, y)$
- Only the eigenmode $\phi(x, y)$ is sustained in the waveguidenon supported, higher order modes scatter out of the waveguide
- The overlap integral determines the coupling coefficient (same as correlation coefficient)



Region changed to higher index for geodesic lens.

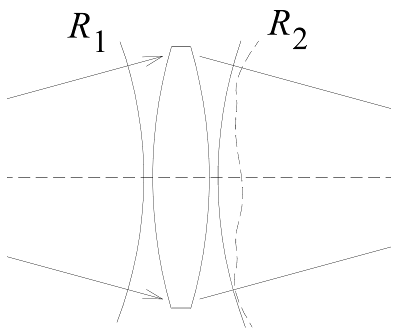
$$\alpha = \frac{\int \phi(x, y)^* f(x, y) dA}{\sqrt{\int |\phi(x, y)|^2 dA \int |f(x, y)|^2 dA}} \tag{7.5}$$

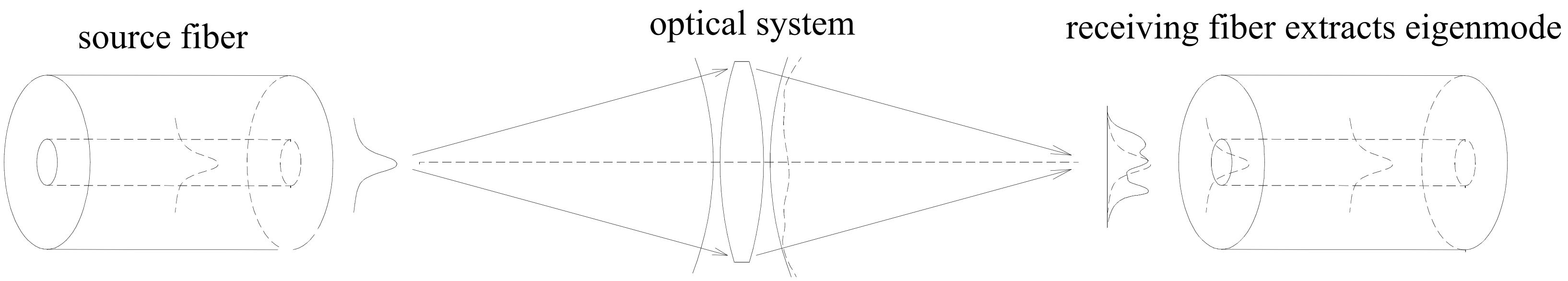
In-coupled field is $|f(x, y)| \rightarrow \alpha\phi(x, y)$ (7.6)



Coupling optics

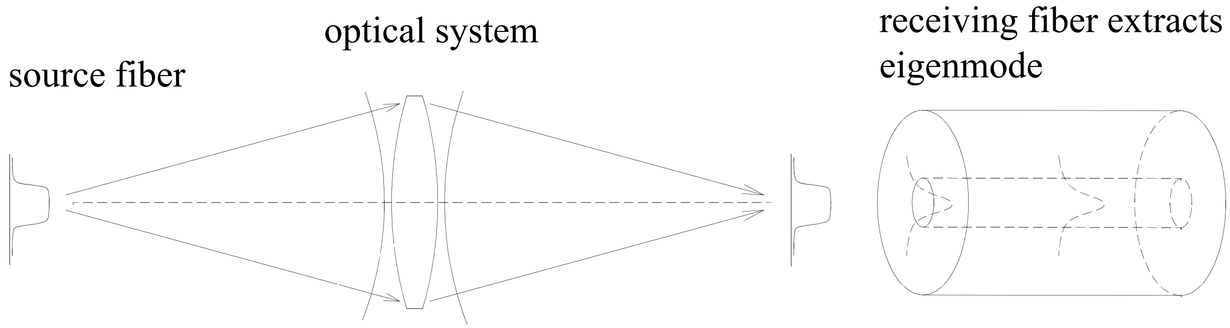
- Mode exists from fiber, propagates through lenses, apertures, aberrations
- Perform overlap integral at entrance plane to in-coupling fiber
- Optical design: correct aberrations in transforming R_1into R_2 —optimizes input coupling

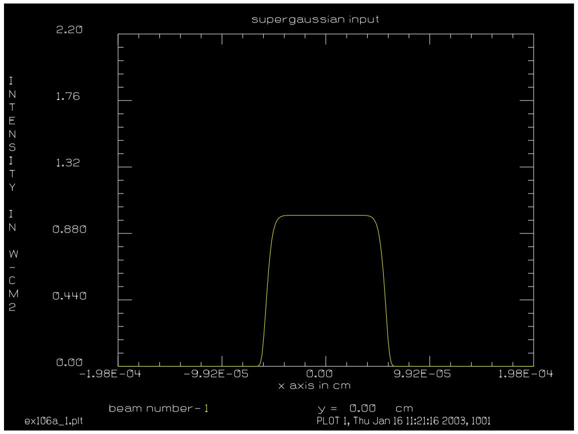




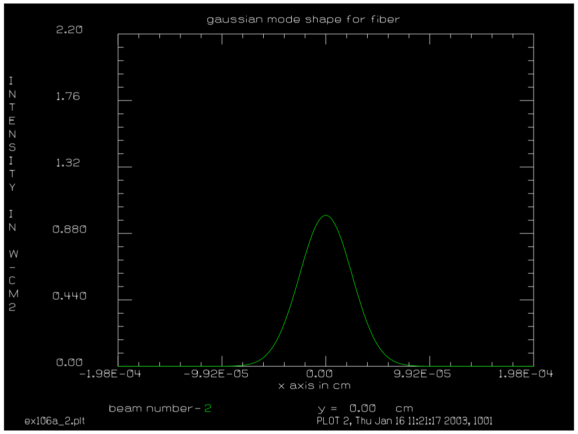
Elementary example of fiber-to-fiber coupling: couple.inp

- hypothetical case, couple a supergaussian output into a fiber with a gaussian mode shape

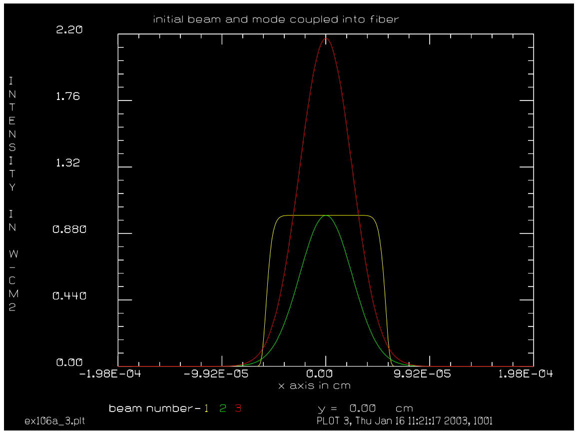




Hypothetical starting distribution from initial fiber.



Assumed gaussian mode shape for receiving fiber.



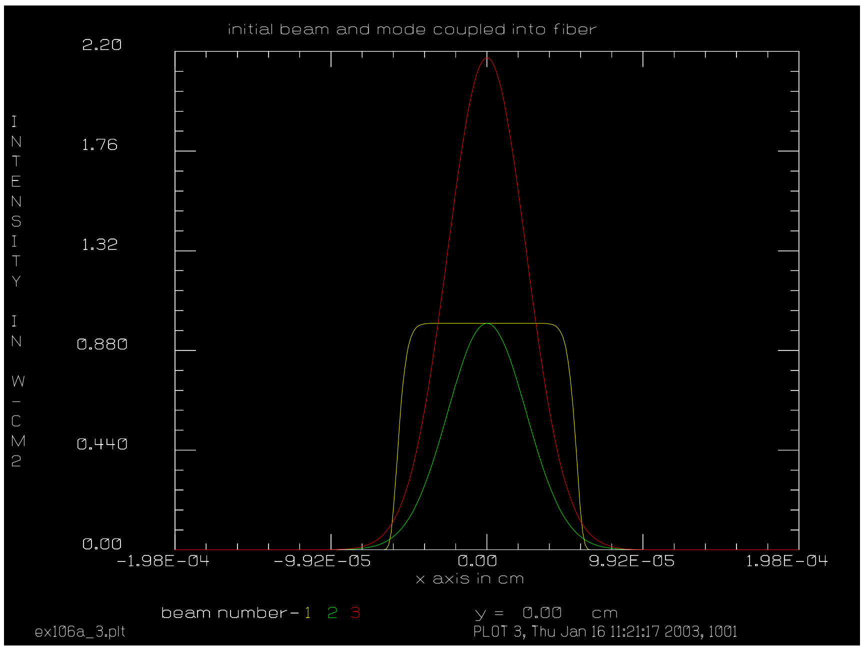
Extracted mode (red): 87.3% energy conversion from start (yellow).

Investigate tilt of fiber: couplea.inp

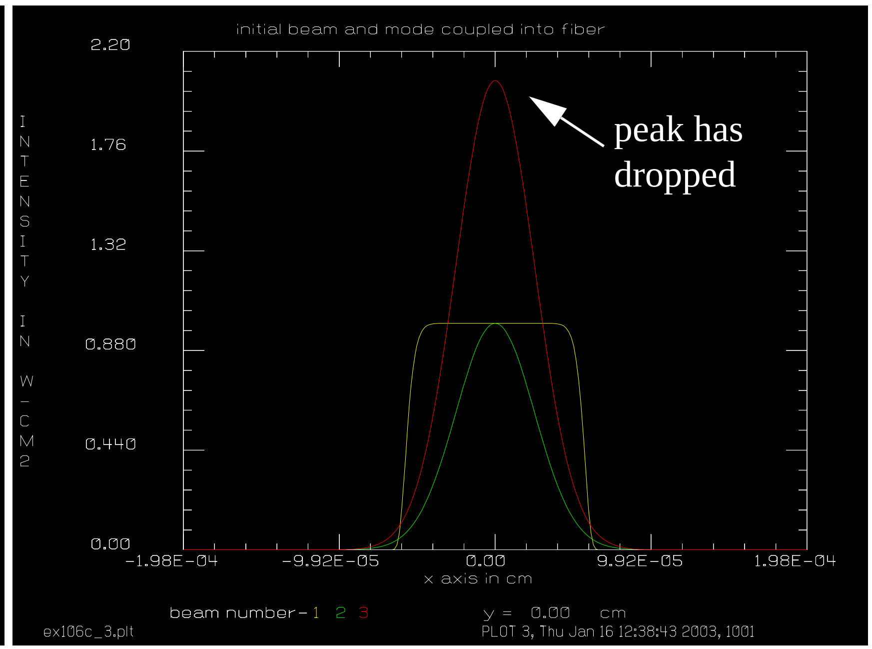
- Tilt fiber by 10 degrees. Use `abr/tilt`

```
Rnorm = 1.
Waves = pi*Angle_Degrees/180./Lambda*Rnorm
abr/tilt kbeam Waves rnorm=Rnorm
```

- Calculate the energy coupling with no tilt and 10 degree tilt.



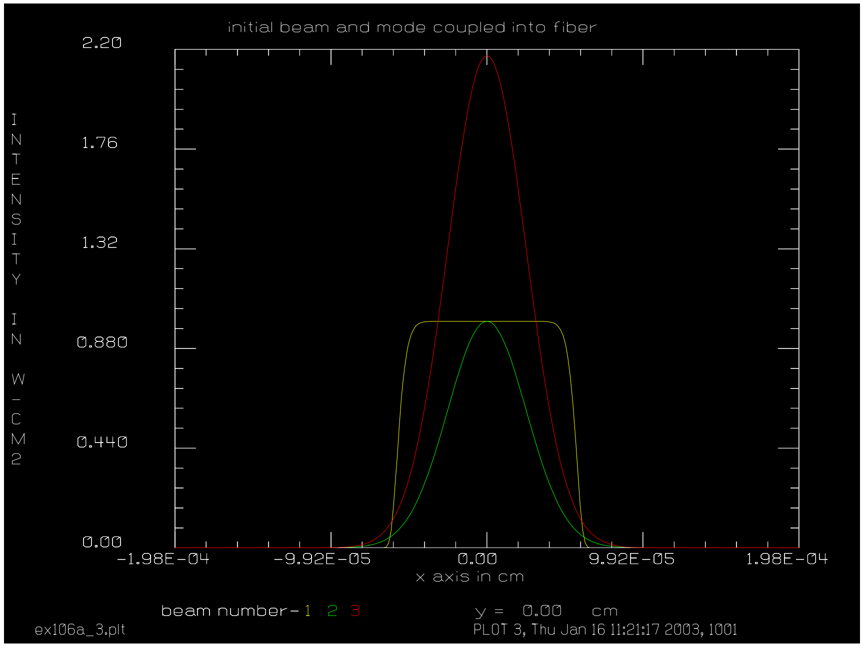
Extracted mode (red) with no tilt.



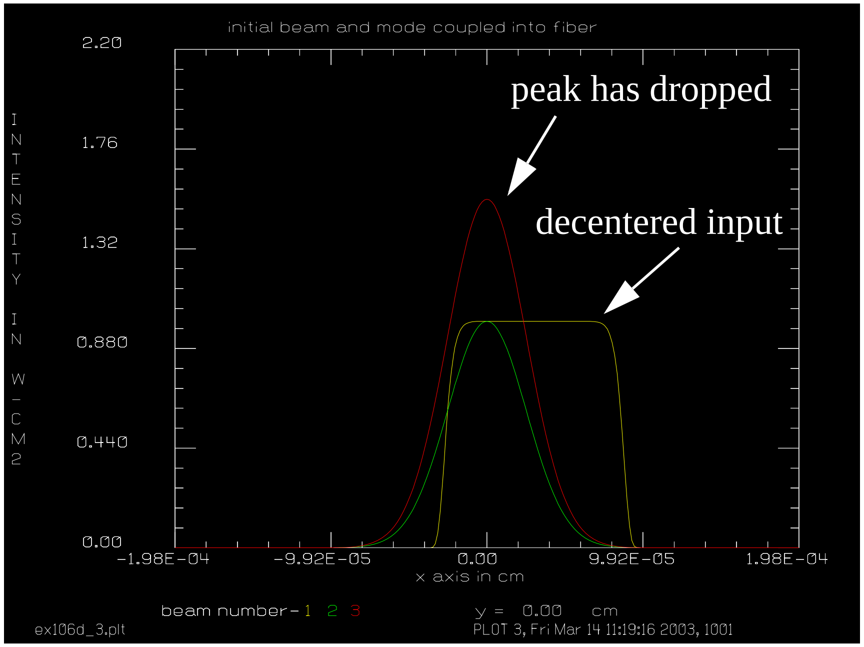
Extracted mode (red) with 10 degree tilt.

Investigate decenter of beam: coupleb.inp

- Decenter the beam by 0.3 micron
- Calculate efficiency

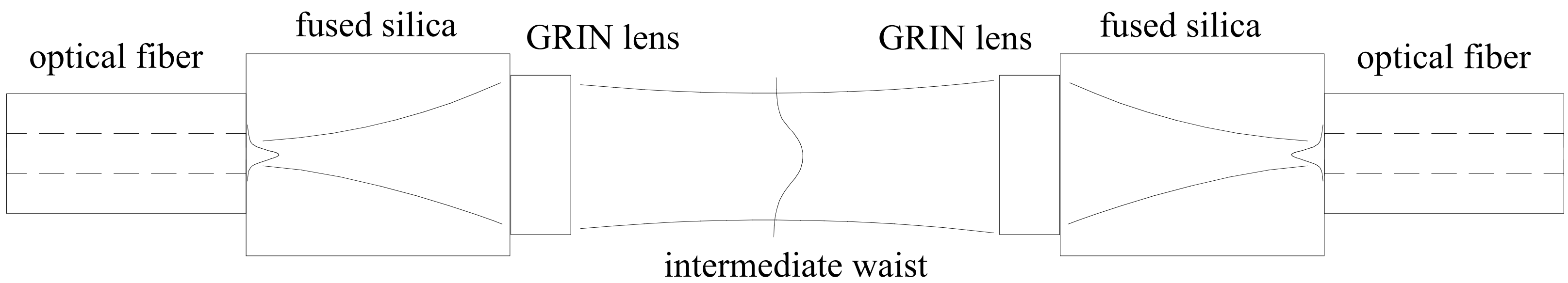


Extracted mode (red) with no decenter.

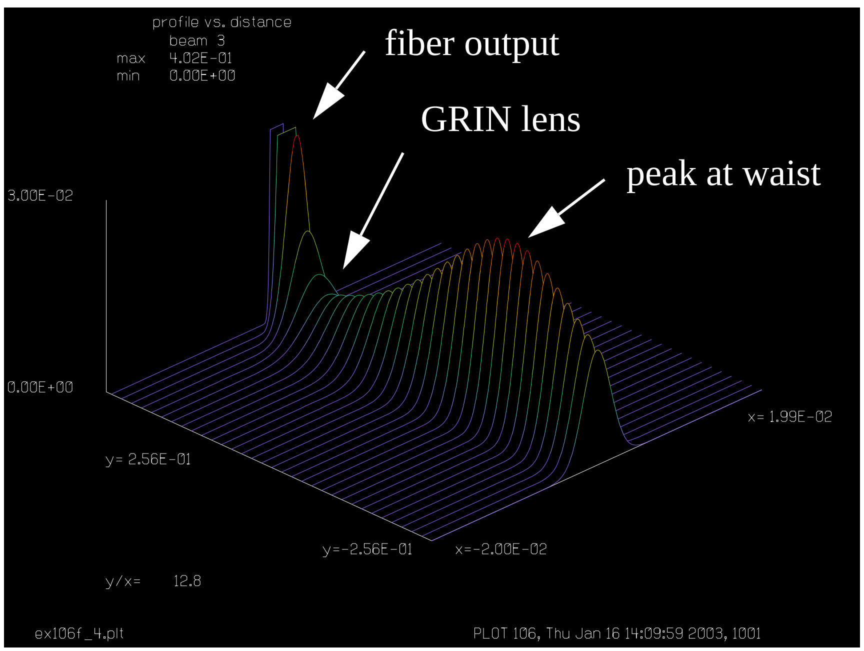


Extracted mode (red) with 10 degree tilt.

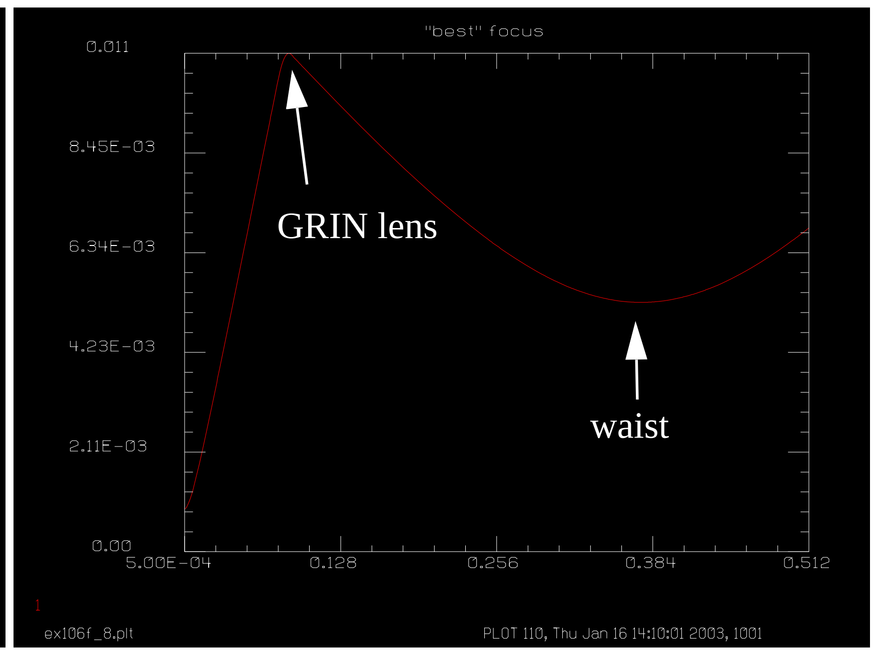
Fiber, GRIN lens, long waist: ex106d.inp



Light is emitted from a single mode fiber into a fused silica rod. At the end of the rod a GRIN lens creates a large diameter waist. The beam is collected by the mirror-image system.

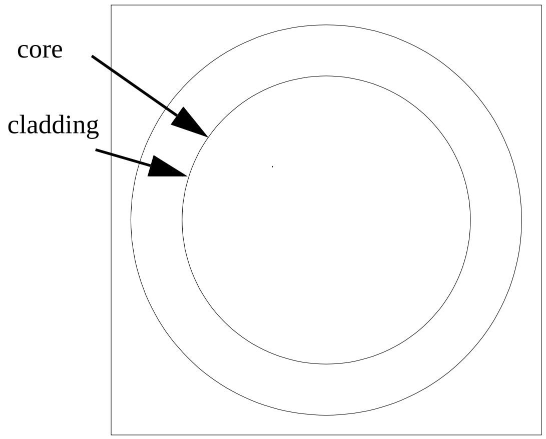


Through-focus display.

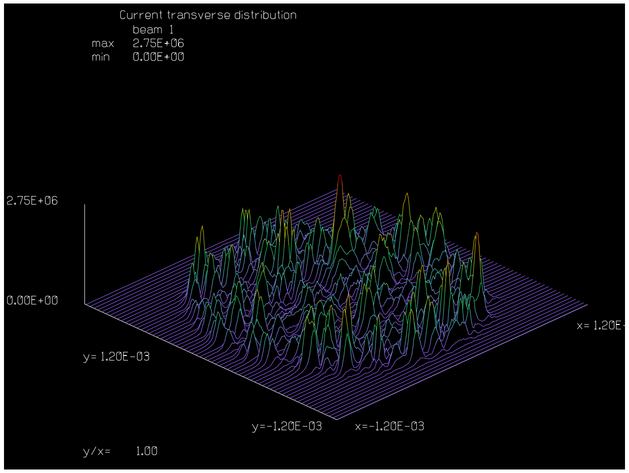


Beam width vs. axial position.

Multimode core: ex86d.inp

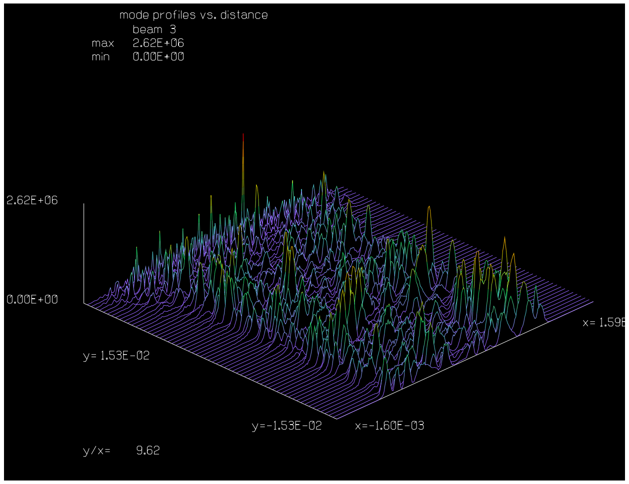


Ex86d models a multimode fiber of 20µ diameter. The 256 x 256 array spans 32µ.

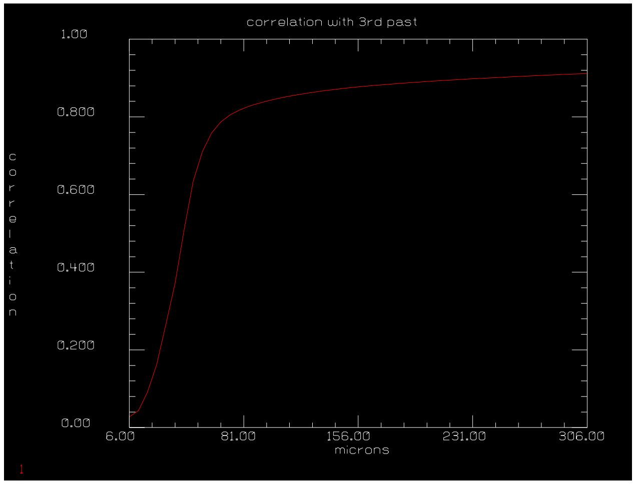


Speckle pattern after 300µ propagation. The speckles change but are statistically stationary.

instantaneous speckle stabilizes to uniform statistics



History of the speckle pattern over 300µ.



Correlation of the nth and n-3 passes shows convergence.

correlation of nth and n-3 passes stabilizes indicating steady case condition

Comparison of analysis: integrated optics vs. fiber optics

Issues	Integrated optics	Fiber optics
shape	slab	round
propagation distances	short (millimeter scale)	long (kilometer scale)
propagation losses	less important	very important
importance of dispersion	less important	very important
nonlinear optic effects	negligible	Brillouin scattering, other
Bragg effect	negligible	important application
method of analysis	BPM, finite difference, finite element	analytical modes, perturbation analysis
lasers	laser diode	fiber laser

Acceptance angle, numerical aperture, relative refractive index

- critical angle, angle of TIR, $\varphi_c = \sin^{-1}\left(\frac{n_1}{n_2}\right)$
- acceptance angle θ_a measured from optical axis, $n_0 \sin\theta_a = n_1 \cos\varphi_c$
- numerical aperture (NA): $\mathrm{NA} \equiv n_0 \sin\theta_a = \sqrt{n_1^2 - n_2^2}$ (independent of waveguide diameter), $\theta_a = \sin^{-1}\left(\frac{1}{n_0}\sqrt{n_1^2 - n_2^2}\right)$
- relative refractive index difference: $\Delta = \frac{n_1^2 - n_2^2}{2n_1^2} \approx \frac{n_1 - n_2}{n_1}$ for $n_1 \approx n_2$
- $NA = n_1\sqrt{2\Delta}$

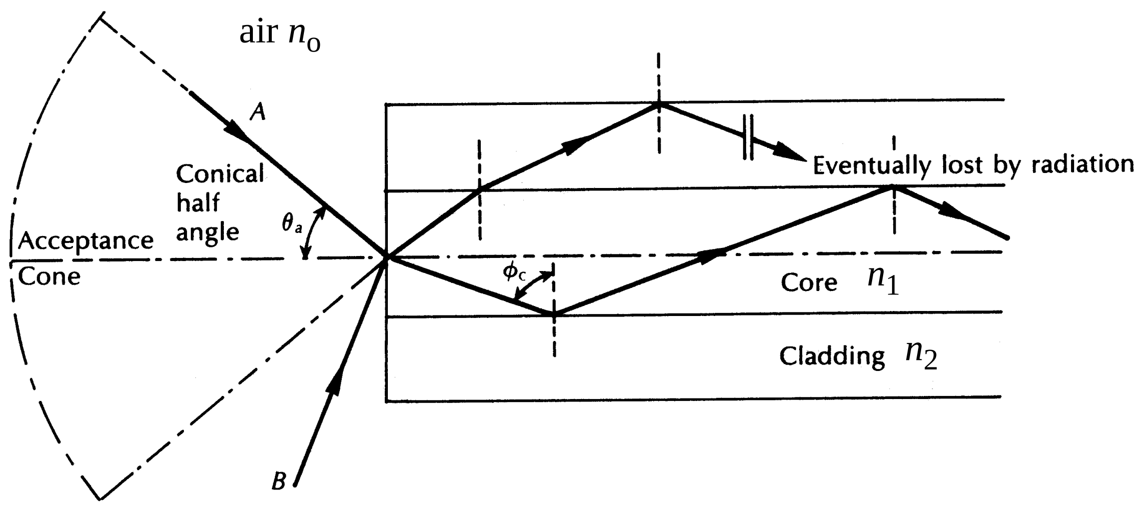
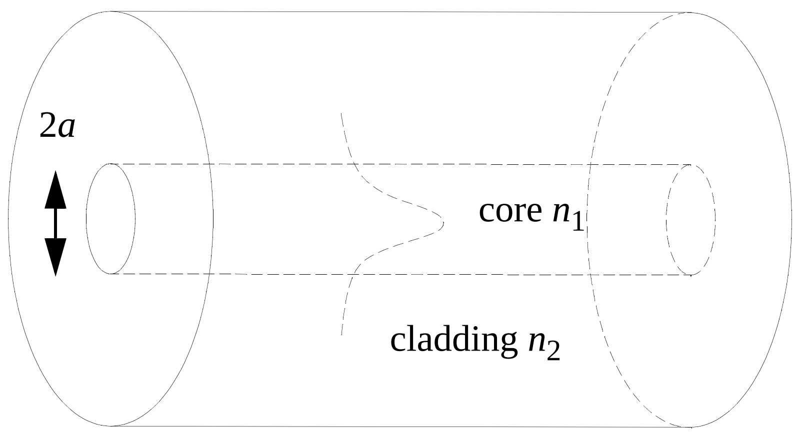


Figure 2.4 The acceptance angle θ_a when launching light into an optical fiber.

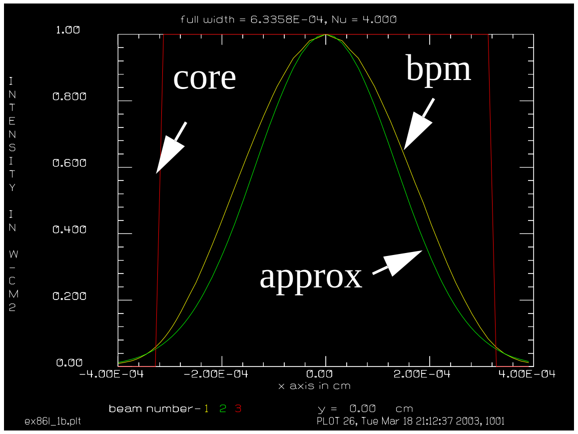
Gaussian approximation for circular, step function fiber

- Exact solution is rather complicated.
- Evaluation of Bessel functions inconvenient: J_0, J_1, J_2, K_0, K_1, K_2.
- For normalized frequency $\upsilon > 2$ (guiding not too weak), the HE11 mode shape may be approximated by a gaussian distribution.
- for step index fiber: $\omega_0 \approx a(0.65 + 1.619\nu^{-3/2} + 2.879\upsilon^{-6})$
- for parabolic index fiber $\omega_0^2 \approx \frac{a}{2n_2k}\sqrt{\frac{2}{\Delta}}$, $\beta^2 = n_1^2k^2\left(1 - \frac{2\sqrt{2\Delta}}{n_2ka}\right)$,

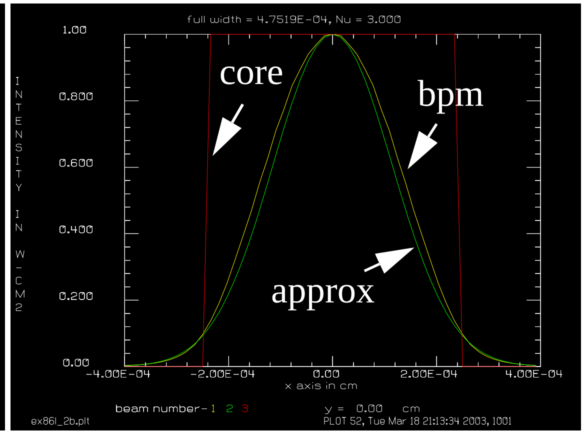
where $\Delta = \frac{n_1^2 - n_2^2}{2n_1^2} \approx \frac{n_1 - n_2}{n_1}$ for $n_1 \approx n_2$



HE11 fiber mode vs. gaussian approximation: ex86l.inp

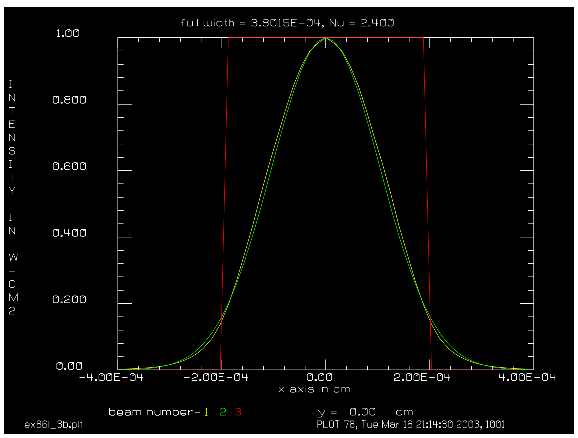


v = 4.0, well confined, width = 6.32μ.

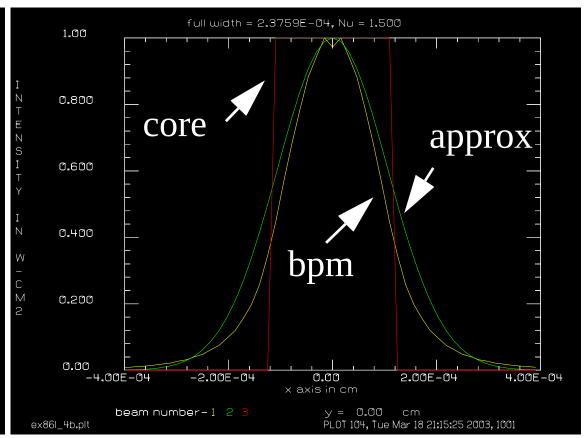


v = 3.0, well confined, width = 4.75μ.

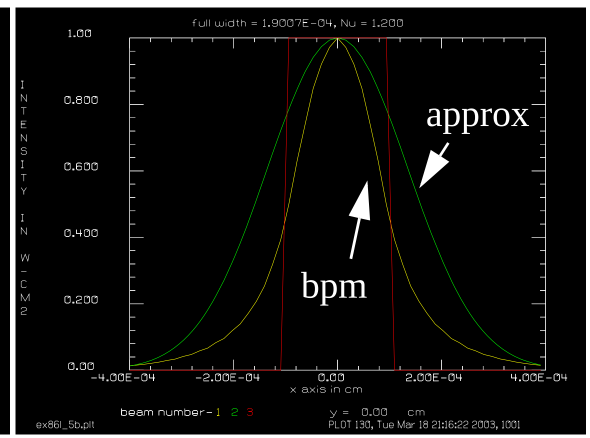
Core region (red)
BPM calculation (yellow)
gaussian approximation (green)
$\lambda = 1.55\mu$, $n_f = 1.532$, $n_c = 1.500$.



v = 2.4, good agreement, width = 3.8μ.



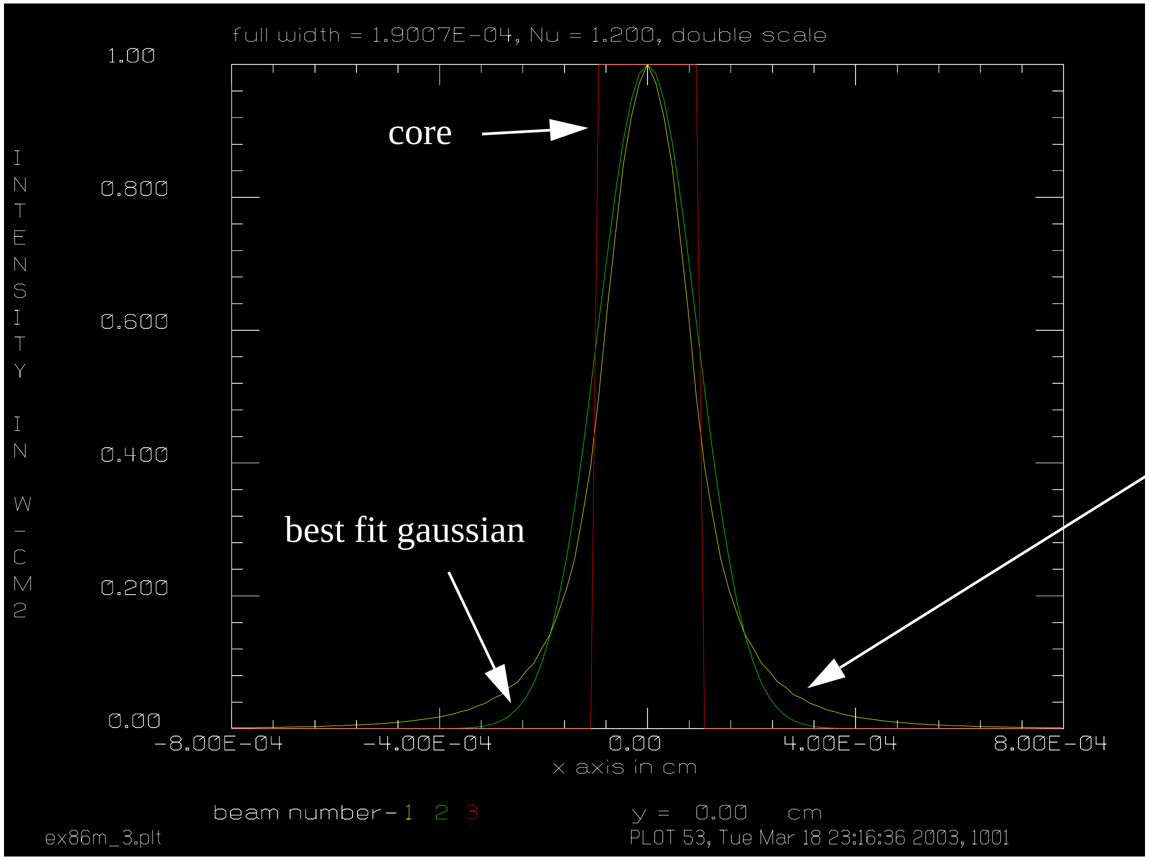
v = 1.5, separation of BPM and gaussian, width = 2.4μ.



v = 1.2, strong separation of BPM and gaussian, width = 1.9μ.

True HE11 mode decays slower than gaussian for low *v*: ex86m.inp

- For weak guiding a strong exponential decaying tail is formed. (Gaussian approximation not suitable for lateral coupling effects)



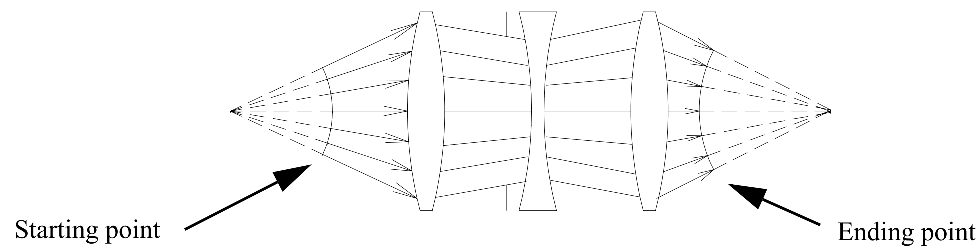
bpm, wide skirt, $e^{-\sigma x}$

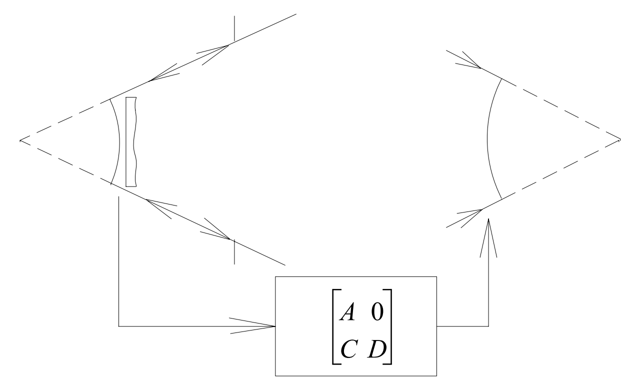
v = 1.2, at critical frequency, good match, width = 1.9m., double scale, with "best fit" gaussian. Note the true curve calculated by BPM method has a much wider skirt. The true curve is close to the form $\exp(-\sigma x)$ outside the core and decays slower than the gaussian.

8. Lensgroup

LENSGROUP: Geometrical optics within GLAD

- analyze lensgroup with rays
- build physical optics equivalent
  - aberrations
  - optical propagation
  - global positioning





The aberrations of the optical system are determined by probing with rays. Generally, the rays will be started at some intermediate point on the beam in object space and terminating in image space. The system may be represented by an aberration plate and the paraxial optics behavior.

LENSGROUP: lens.inp

- Construct a lens using radius-thickness-glass prescription

Table. 8.1. λ = .55, aperture radius = 0.5 cm,

surface	radius	thickness	glass
1	2.18490397	0.326326343	sk16
2	-17.2730754	0.516337239	air
3	-2.01031718	0.099769834	f2
4	2.33565226	0.424584729	air
5	22.8173221	0.1956842	sk16
6	-1.70906459	0. (paraxial solve)	air

1) Add 7th surface as image, figure aberrations from paraxial focus: `image focus`.

2) What is Strehl ratio (relative to reference surface centered at paraxial focus),

3) Propagate to paraxial focus: `focus/apply`.

4) Check peak value: `peak`.

5) Search for peak value by taking small propagation steps around paraxial focus and checking peak value.Try steps of about .0001 and stay within Rayleigh range of about .0033 cm of paraxial focus.

6) How much focus shift is needed to maximize peak value? Compare Zreff (local position) with Zbound (paraxial focus).

7) Question: What could we do to find the focus position giving the optimum peak value without taking multiple diffraction propagation steps?

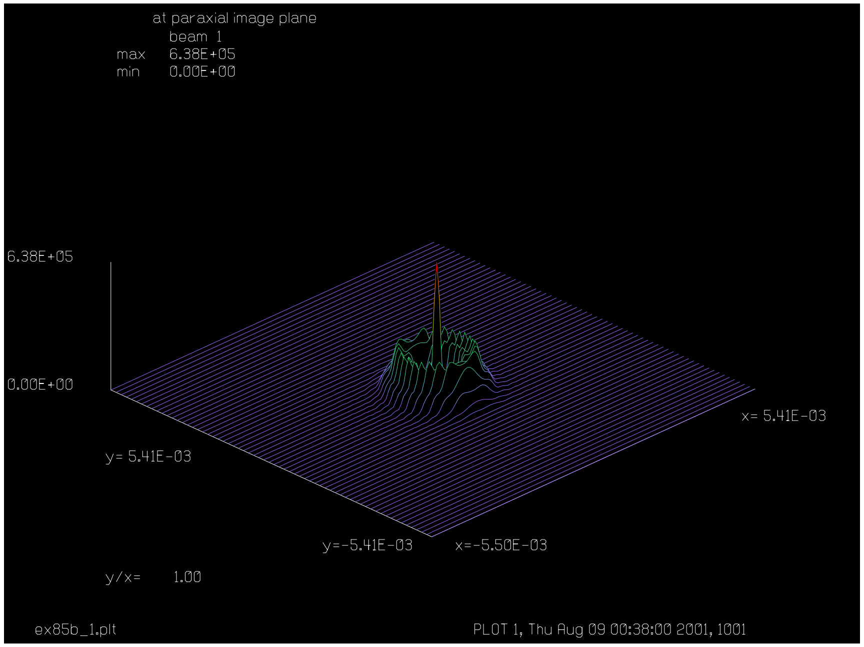
Triplet lens: ex85bx.inp

```
c## ex85b
c
c  Example to illustrate a Cooke triplet
c
c  This particular lens was designed to work at 20 deg. half angle field
c
array/s 1 128
lensgroup/def cooke/o
  object lambda=.55 yfield=yo yna=.5e-10 zsurf=obj_dist zpupil=pupil_dist air
  surface 1e10 0. air
  surface  2.18490397 .326326343 sk16
  surface -17.2730754 .516337239 air
  surface -2.01031718 .099769834 f2
  surface 2.33565226 .424584729 air
  surface 22.8173221 .1956842 sk16
  surface -1.70906459 4.17485291 air
  image
lensgroup/end
  uc = tan(pi*20/180)                 # chief ray paraxial angle
  obj_dist = 1e10                     # set object distance, to a large number
  yp = -.3711                         # set chief ray height at 1st surface
  pupil_dist = -yp/uc                 # calculate pupil displacement from
                                      # 1st surface
  yo = uc*(obj_dist + pupil_dist)     # calculate object height
variab                                # list variables
color 1 .55                           # select wavelength
units 1 .025                          # specify units
clap/c/c 1 .5001                      # specify clear aperture
c
c Trace a single ray, in global coordinates
c
lensgroup/trace/oneray/beam cooke 1 0 1 global
c
```

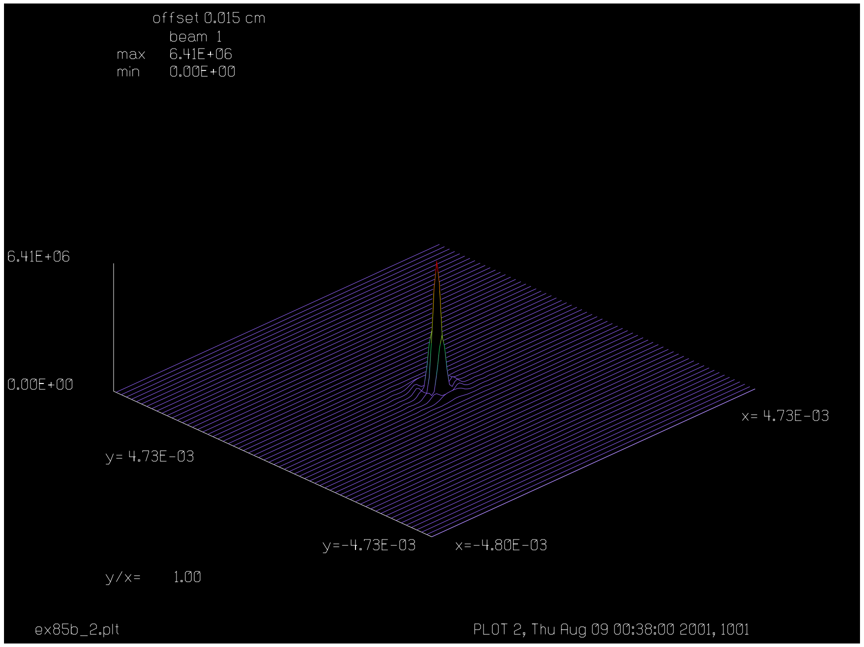
Triplet lens (cont'd): ex85bx.inp

```
c Trace a ray fan, using 5 rays
c
lensgroup/trace/yfan/beam cooke nrays=5
c
c Trace a spot diagram and display x- and y-fans
c
lensgroup/trace/spot/beam cooke
c
c Implememt the lens. The beam propagates to the last surface of the lens.
c
lensgroup/run cooke 1
geodata
focus/a 1                           # propagate to paraxial image
c
title at paraxial image plane
plot/watch ex85b_1.plt
plot/l 1 ns=64
title offset 0.015 cm
prop .015
plot/watch ex85b_2.plt
plot/l 1 ns=64
pause 5
end
```

Image formed by triplet lens:



Far-field for Ex. 85b, paraxial focus.



Far-field for Ex. 85b, best focus.

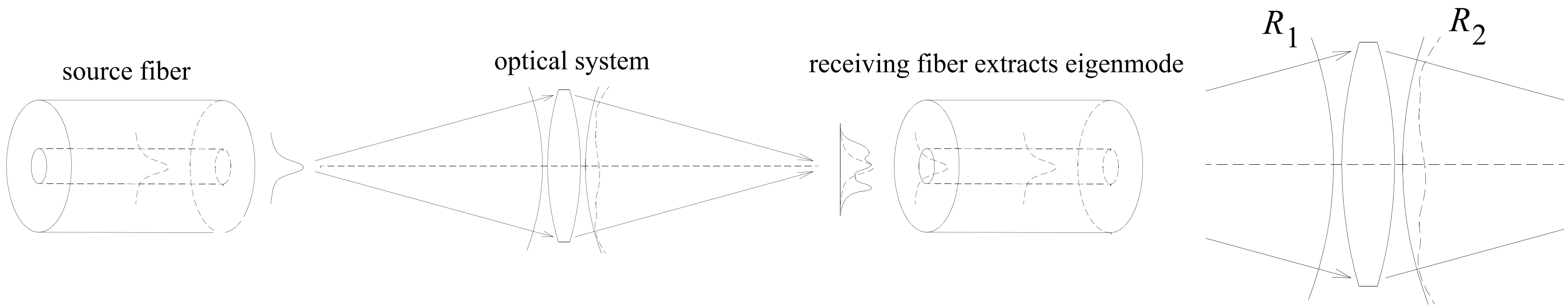
Finding the "best" focus without focal plane search: lens1.inp

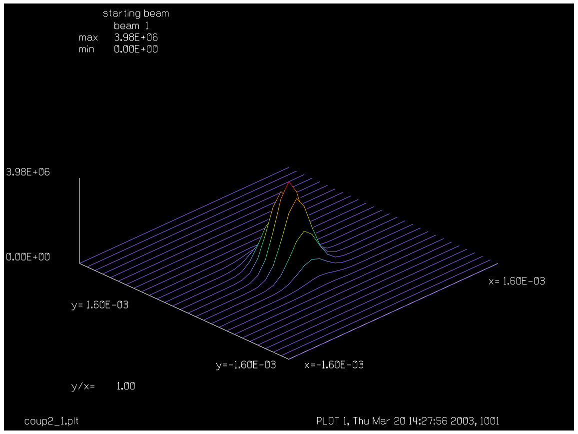
- Build a macro called "search" to optimize the Strehl ratio

1) make a second "data" array to store original pupil
2) try different values of defocus: `abr/focus 1 Waves`
3) after optimizing Strehl ratio, propagate to paraxial focus: `focus/apply/abcd 1`
4) how well does the peak compare with focal plane search?

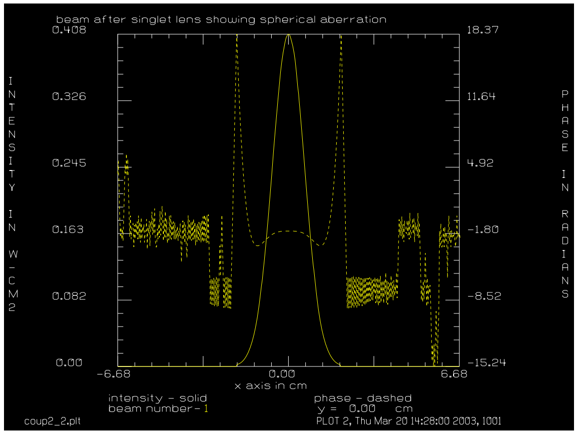
Optimizing in the pupil is faster because the search does not require diffraction propagation.

Fiber-to-fiber relay lens with lensgroup: couple2.inp

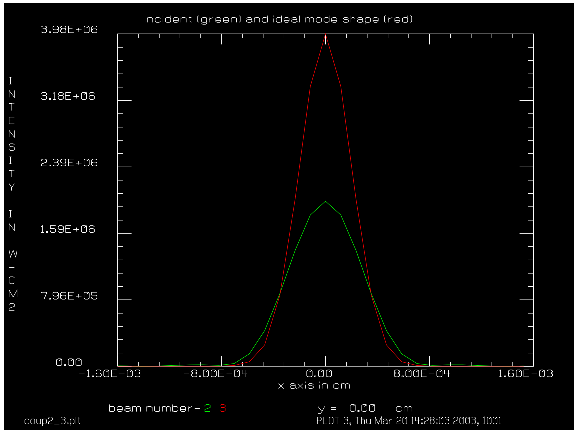




Source



Aberration of lens



Overlap integral formed

Fiber-to-fiber relay lens with lensgroup: couple2.inp

```
c## couple2

# example of fiber-to-fiber coupling with a singlet

# A gaussian beam representing the output of an optical fiber
# is imaged by a plano-convex lens onto an identical fiber.
# The lens has a conic surface to remove most of the spherical
# aberration

echo/on
mem/set/b 8            # set memory to more than enough
array/s 1 512          # set array size
Waist=4e-4             # waist of 4 microns
nbeam 3 data           # make two extra arrays for data
units/field 0 .03      # field half-width of array
wavel 0 1.55           # set wavelength to 1.55 micron
gaus/c/c 1 1 Waist     # gaussian start
energy/norm 1 1        # set energy to unity
plot/w coup2_1.plt
title starting beam
plot/l 1 xrad=4*Waist
c
c  Define a lensgroup to reimage the beam
c
lensgroup/def singlet/overwrite
#          radius thickness glassname
   surface 1e10   .2        bk7
#          radius thichness conic-constant [aspheric terms]
   surface -2.537 .0         cc=-1.132                           air
   image focus
lensgroup/end
vertex/locate/abs 0 0 10             # locate lensgroup 10 cm from start
prop/vertex
```

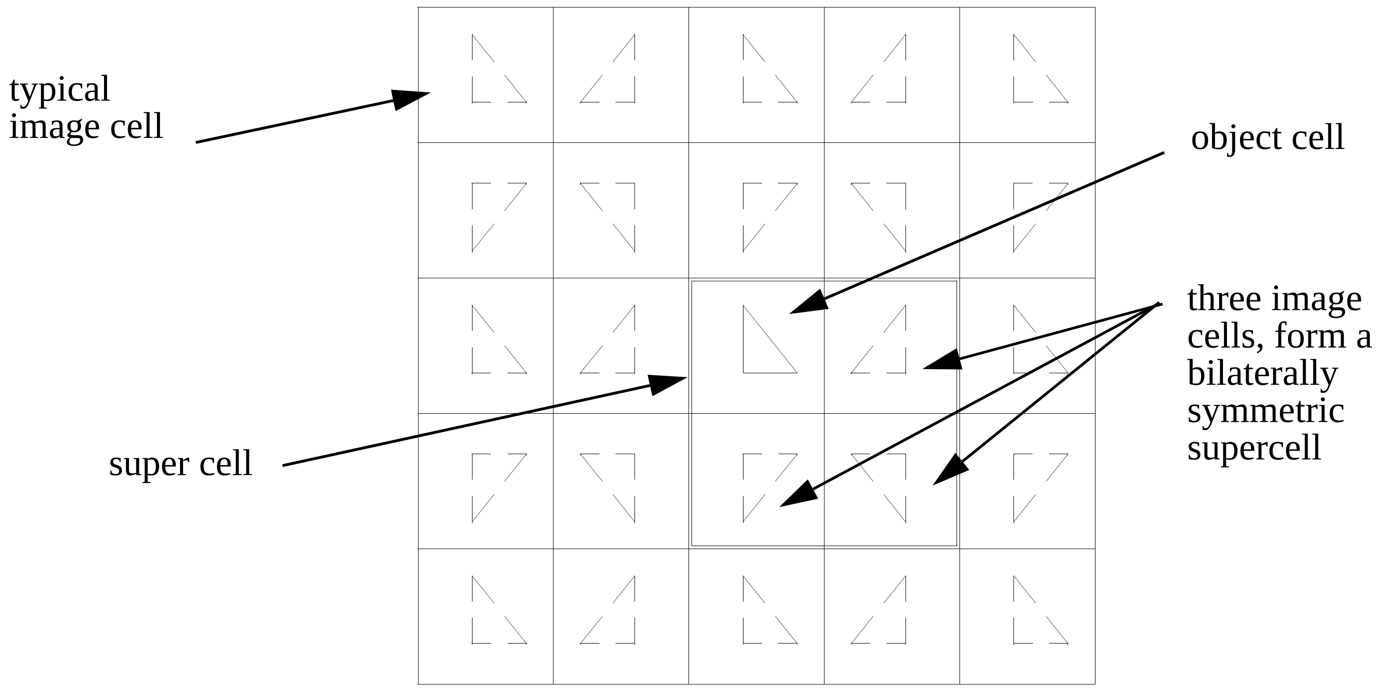
Fiber-to-fiber relay lens with lensgroup (cont'd): couple2.inp

```
clap/c 1 2                          # insert clear aperture
lensgroup/run/radial singlet 1       # call lensgroup
str
plot/w coup2_2.plt
title beam after singlet lens showing spherical aberration
plot/x 1
focus/apply/waist                    # propagate to gaussian waist
c
c  Make a copy of beam 1 in beam 2
c
prop -.0030
copy 1 2
c
c  make gaussian beam to represent single mode fiber
c
units/beam 3 1
gaus/c/c 3 1 Waist
energy/norm 3 1
plot/w coup2_3.plt
title incident (green) and ideal mode shape (red)
plot/x/i fi=2 la=3 le=-4*Waist ri=4*Waist
c
c  Mult beam 1 by beam 3
c
mult/mode/parallel 1 3
energy 1                # How much energy got into fiber mode?
```

9. Reflecting wall waveguides

Using aliasing to model wall reflection

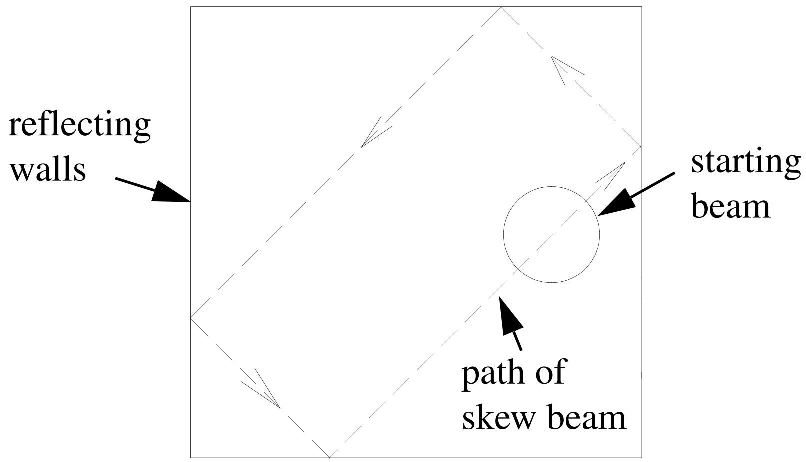
- Reflecting walls may be modeled by an array of images of the object
- For even cells, aliasing effects are identical to reflections from the wall
- Make an odd cell to be even by constructing a super cell of four quadrants
- Set array size to just fit the super cell



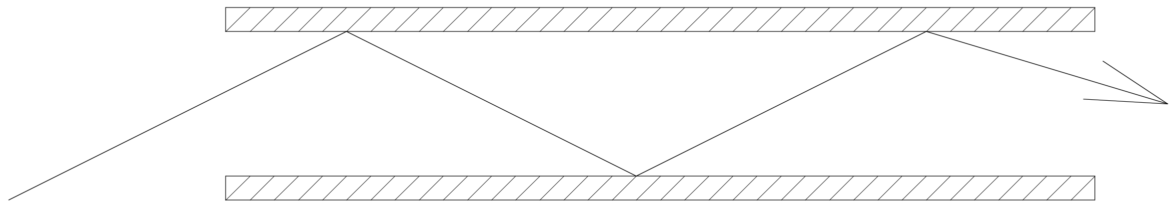
A number of examples of reflecting wall waveguides

- Straight reflecting wall waveguide: **ex77.inp**
- Tapered waveguide, converging beam: **ex77b.inp**
- Tapered waveguide, collimated beam: **ex77c.inp**
- Curved waveguide: **ex77d.inp**
- Waveguide used as optical integrator: **ex77e.inp**
- Waveguide in a resonator: **ex77f.inp**

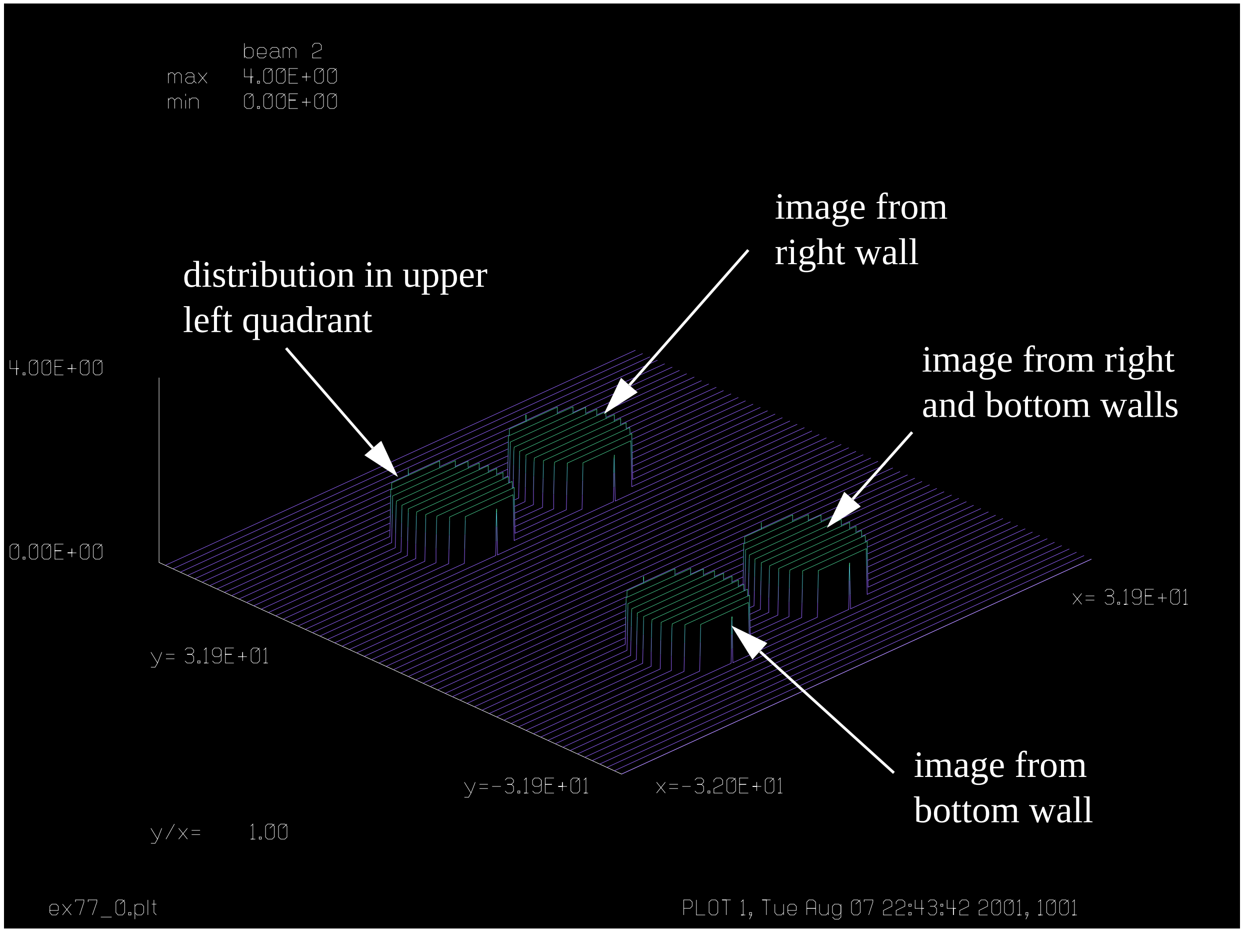
Inject a skew pencil of light into a reflecting wall waveguide



Consider a beam injected into a hollow waveguide with reflecting walls. The beam is given a tilt which sends it toward the upper right. The beam will reflect around the walls while expanding because of diffraction.

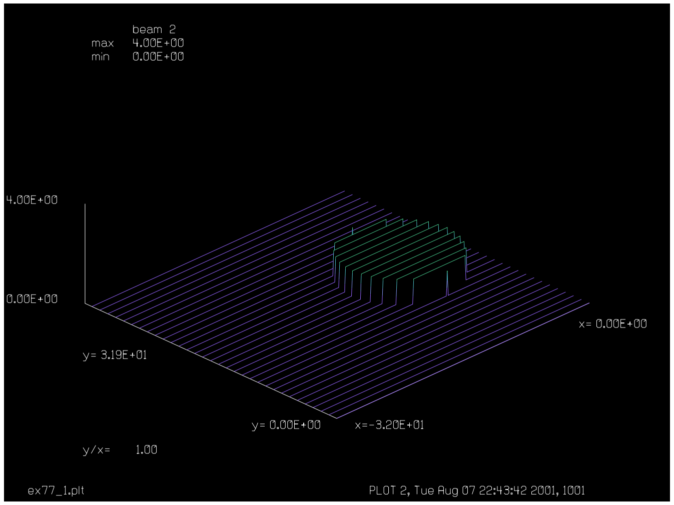


Setup beam and three images to force function to be even: ex77.inp

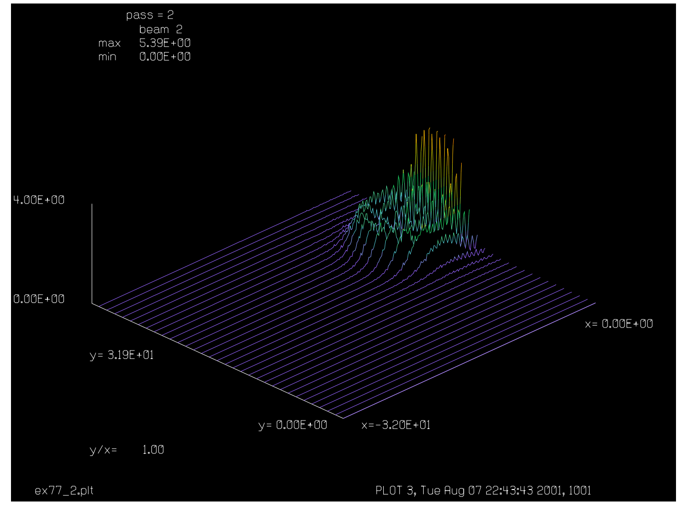


A hollow waveguide may be represented by placing the distribution in one quadrant (here the upper left quadrant) and placing images as formed by the walls in the other three quadrants. The starting distribution is offset and has a tilt which directs the beam initially toward the upper right.

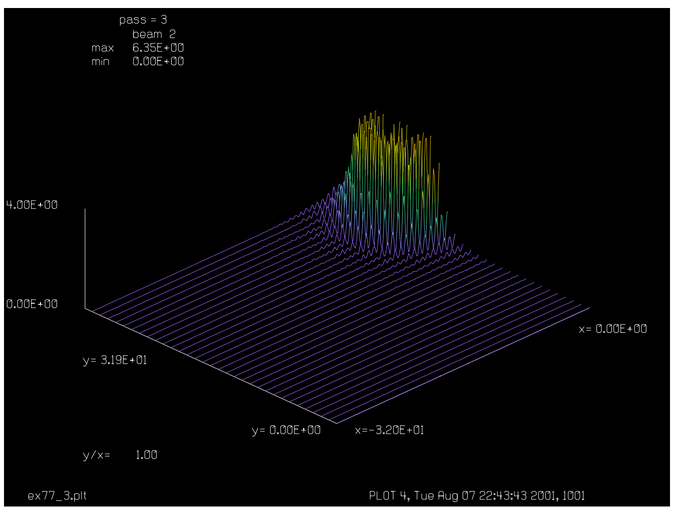
Skew ray hits right, top, left, bottom walls with self interference:



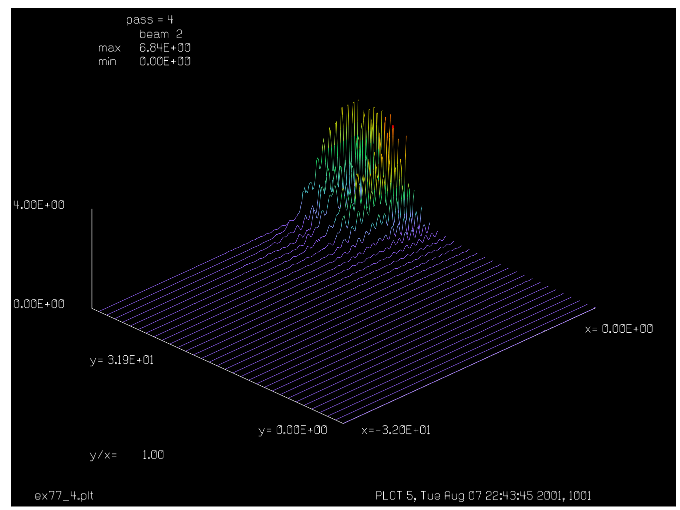
Start with tilt aberration, showing only upper left quadrant, as with rest of figures.



Beam is tilted to upper right and hit right wall.



Beam collides with right wall and is deflected toward top wall.

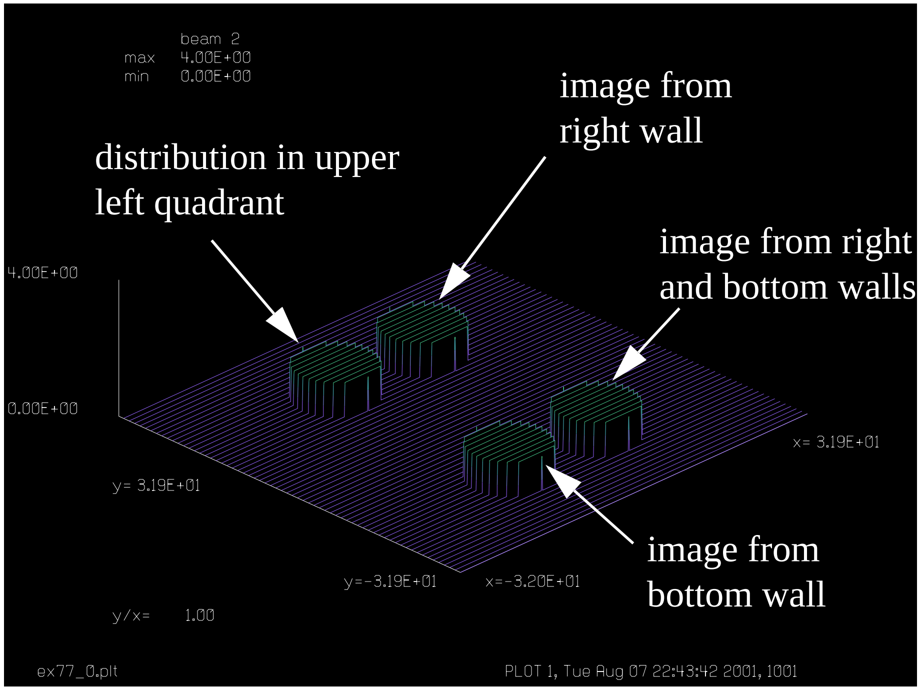


Beam is passing from right to top wall.

Making the super cell: image upper left quadrant into three other quadrants

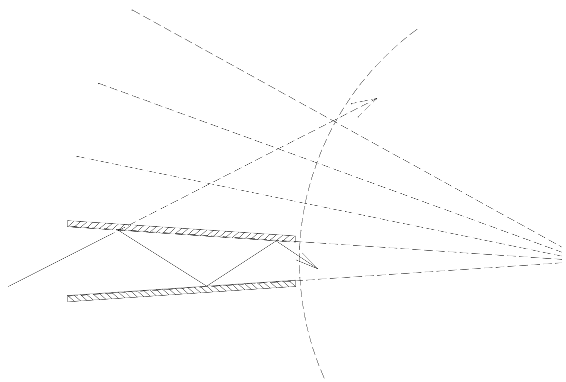
```
# beam 1 is the propagating mode 128 x 128
# beam 2 is super cell 256 x 256

macro/def waveguide/o
 copy/con 1 2 -128 -128 # copy to upper left
 flip/x 1 # flip about x-direction
 copy/con 1 2 128 -128 # copy to upper right
 flip/y 1 # flip about y-direction
 copy/con 1 2 128 128 # copy to lower right
 flip/x 1 # flip about x-direction
 copy/con 1 2 -128 128 # copy to lower left
 prop WaveguideLength 2 # propagate desired distance
 copy/c 2 1 # copy back to cavity mode array
macro/end
```

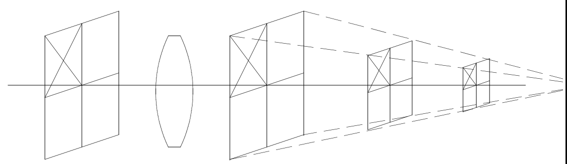


Tapered waveguides: add a lens to the super cell: ex77b.inp

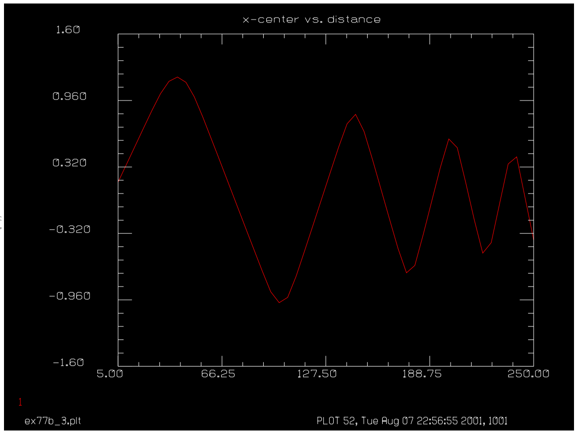
- “chirped” wall intersections in tapered waveguide



Tapered waveguide. The intersections points become closer together and the angles steeper as the ray moves into the waveguide. The intersection points are “chirped”. We may consider the multiple reflections of the waveguide faces. A ray that is too steep will miss the sphere formed by rotating the grating about the perspective point.

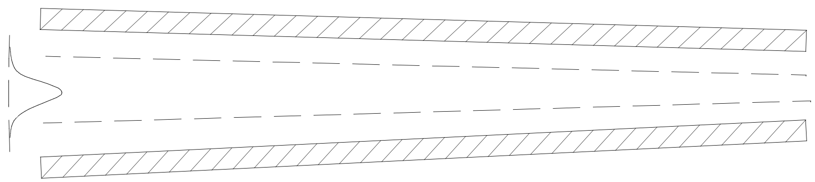


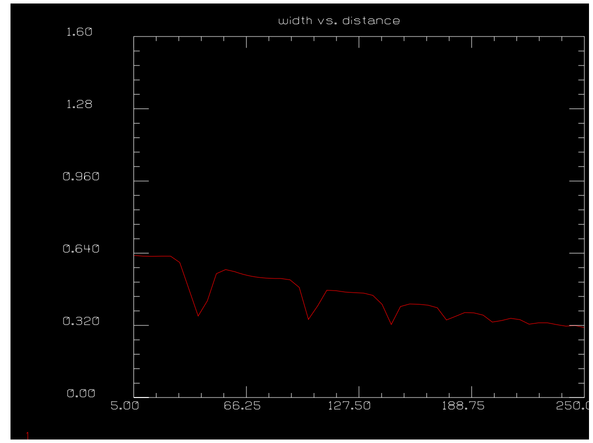
The taper is achieved by adding a lens to the composite array containing the four imaged quadrants. The coordinate system converges to the focus point of the lens which becomes the perspective point of the tapered waveguide.



Trajectory of beam hitting waveguide walls shows chirped behavior for Ex77b.inp. Beam is overfilling waveguide reducing the apparent centroid motion.

Injected beam may be converging toward perspective point: ex77c.inp

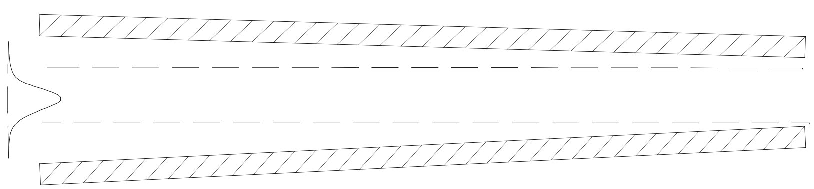


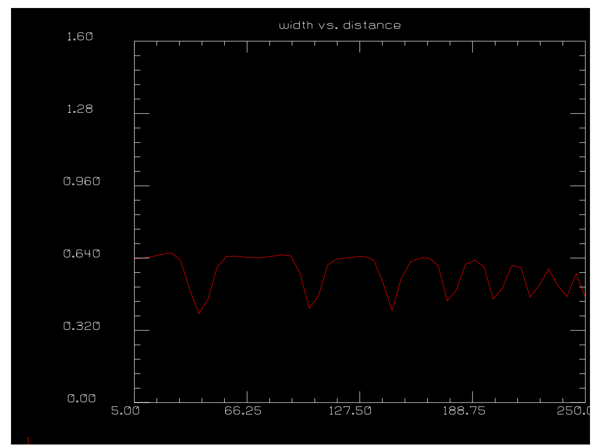


The lens tends to produce a converging beam with width decreasing with distance. Ex77b.

Width decreases for convergent input beam. Ex 77b.inp.

Injected beam may be collimated: add defocus, compensate lens: ex77c.inp



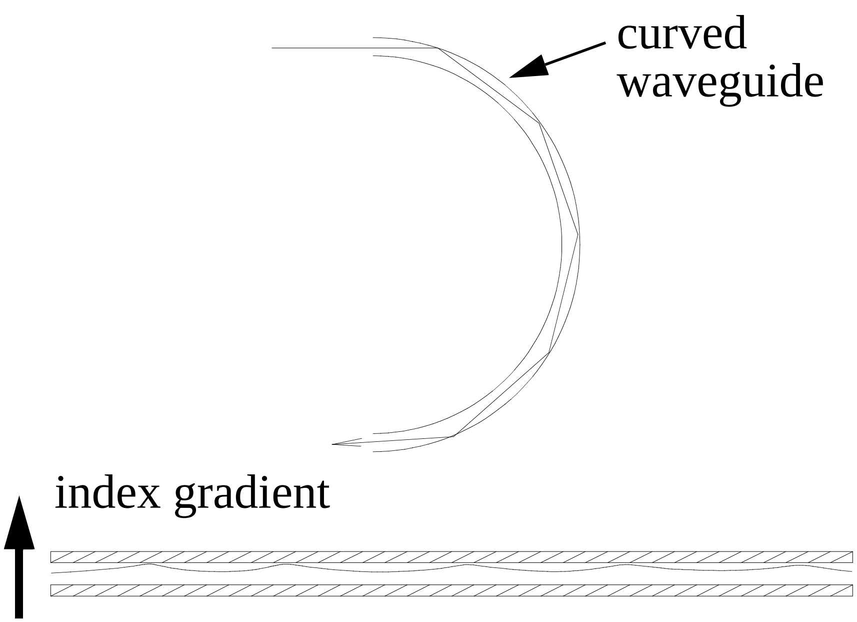


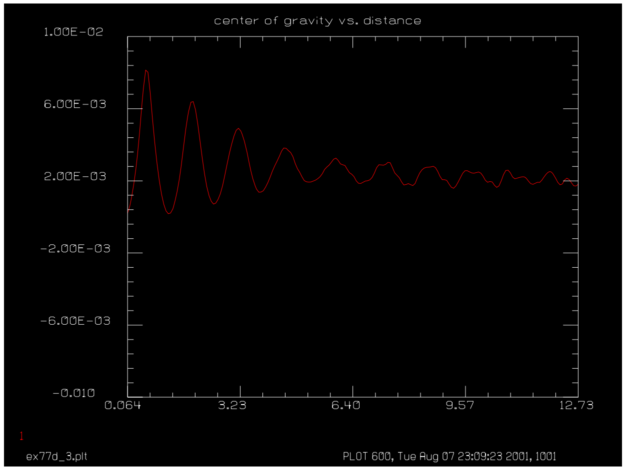
A collimated input may be modeled by adding appropriate divergence to exactly compensate for the focusing of

Width remains constant for collimated input, except at the intersection points. Ex 77c.inp.

Curved waveguide: ex77d.inp

keep guide straight add curve to medium by adding tilt each for each step to simulate index gradient across waveguide

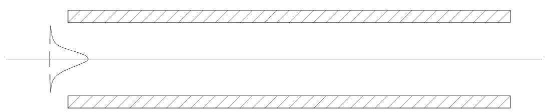




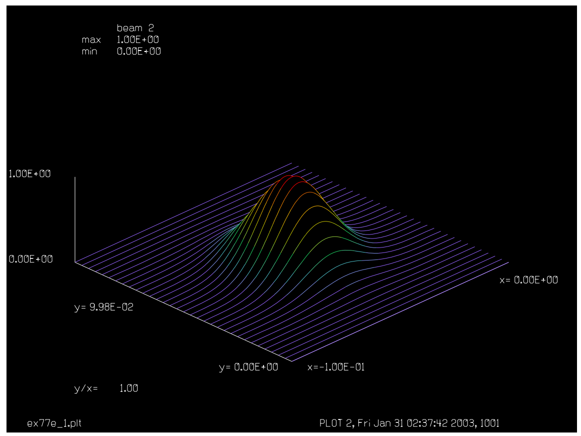
Light injected into a curved waveguide bounces around the circumference in a “square-the-circle” manner. We may treat this with a straight waveguide model where the coordinate system of the beams is constantly changed by adding tilt aberration.

Motion of the center of the beam relative to the walls. The apparent motion decreases as the beam diameter increases.

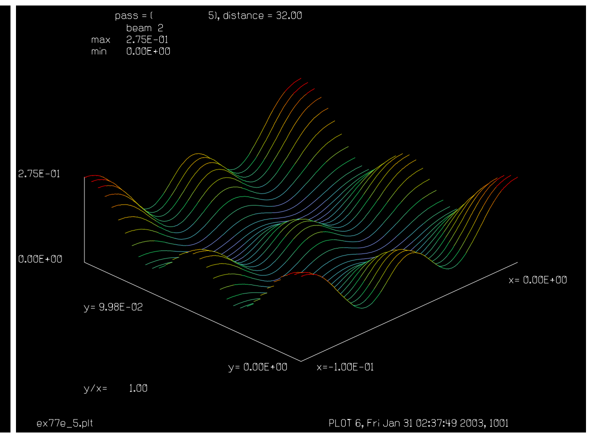
Waveguide used as a homogenizer for a gaussian beam



Waveguide used as homogenizer for gaussian beam. Ex 77e.



Starting gaussian for waveguide homogenizer 0.1 x 0.1 cm square and 32 cm long.

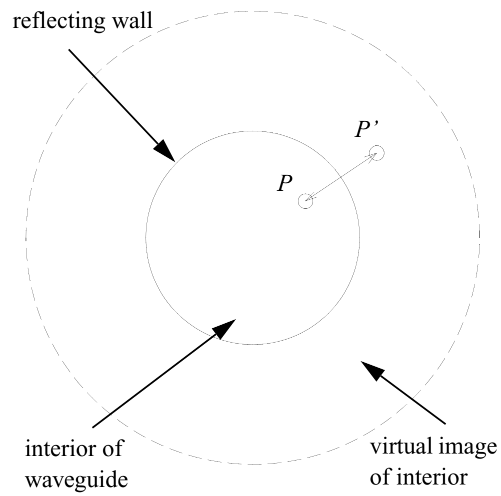


Output of homogenizer. Shows moderate smoothing. Ex77e.inp.

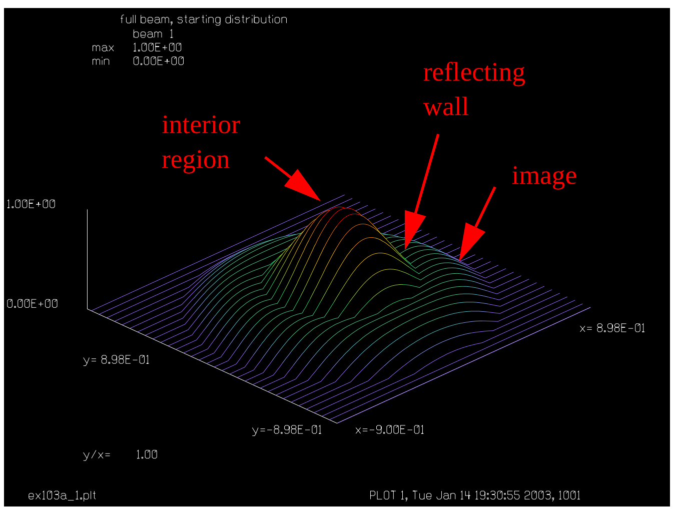
Round and pentagonal reflecting wall waveguides

- Cylindrical rod, single image, 1024 x 1024 array: **ex103a.inp**
- Cylindrical rod, 1st and 2nd images, 1024 x 1024 array: **ex103b.inp**
- Cylindrical rod, 2048 x 2048 array, small memory model: **ex103c.inp**
- Cylindrical rod, 2048 x 2048 array, large memory model: **ex103d.inp**
- Pentagonal rod, implemented with a macro: **ex103e.inp**

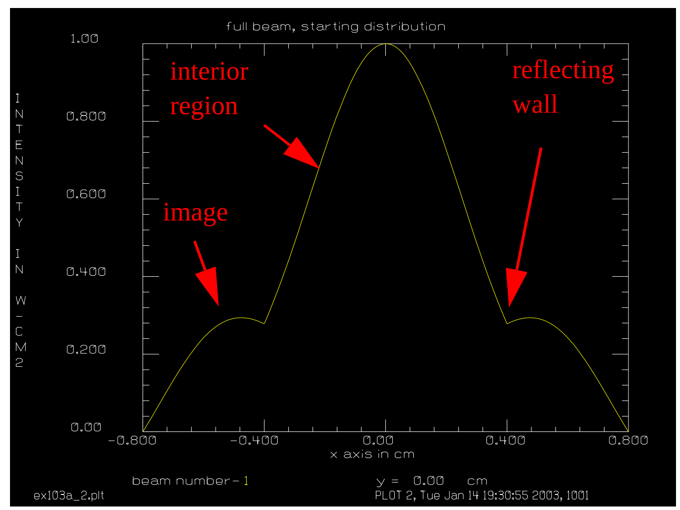
Cylindrical waveguide, ex103a.inp , use **rod kbeam radius**



A virtual image of the interior of the waveguide region is formed into an annular region. The point *P* is imaged into *P'* equidistant from the wall.

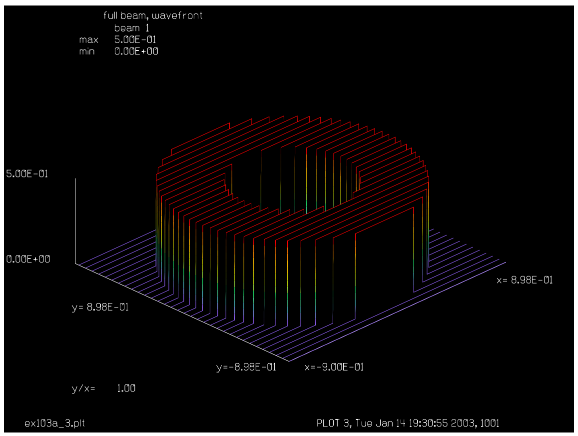


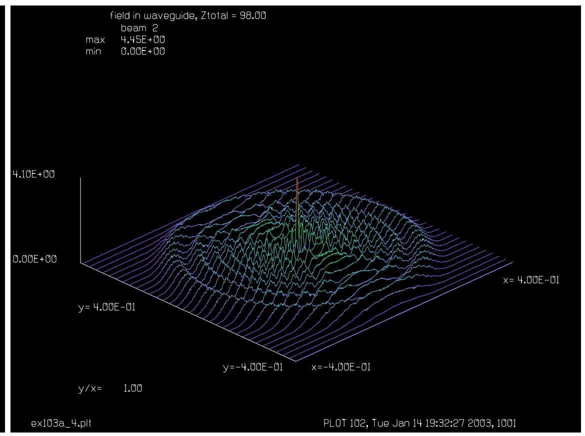
A gaussian distribution and its image formed by the reflecting wall (with reduced intensity).

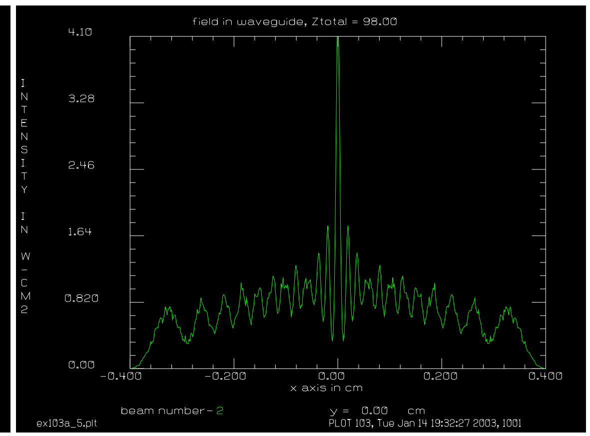


A gaussian distribution and its image formed by the reflecting wall (with reduced intensity).

Phase change at the reflecting wall must be considered







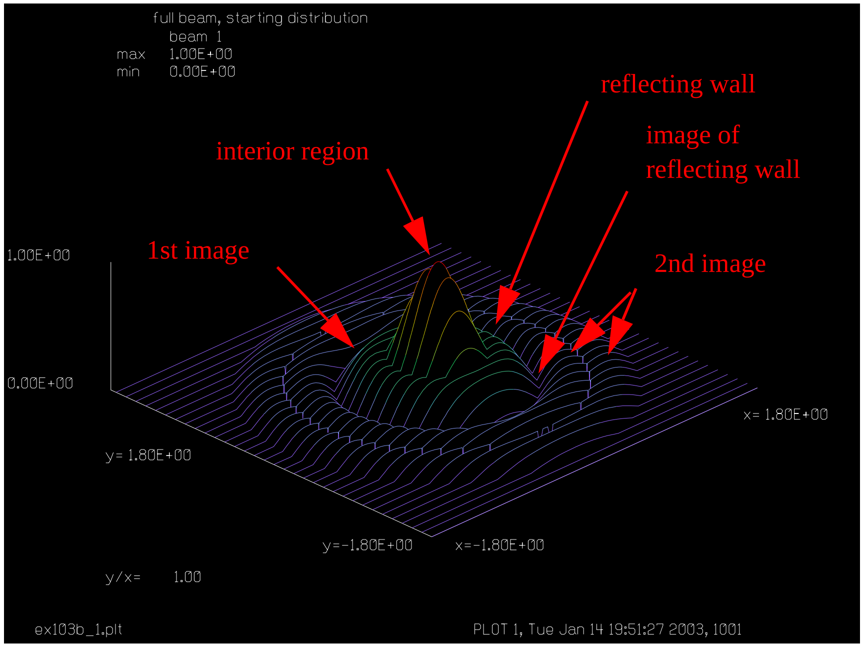
Phase of starting distribution. Grazing light reflected by TIR has a π phase change.

The cavity mode after propagating 98 cm. The center point appears at every odd Fresnel number.

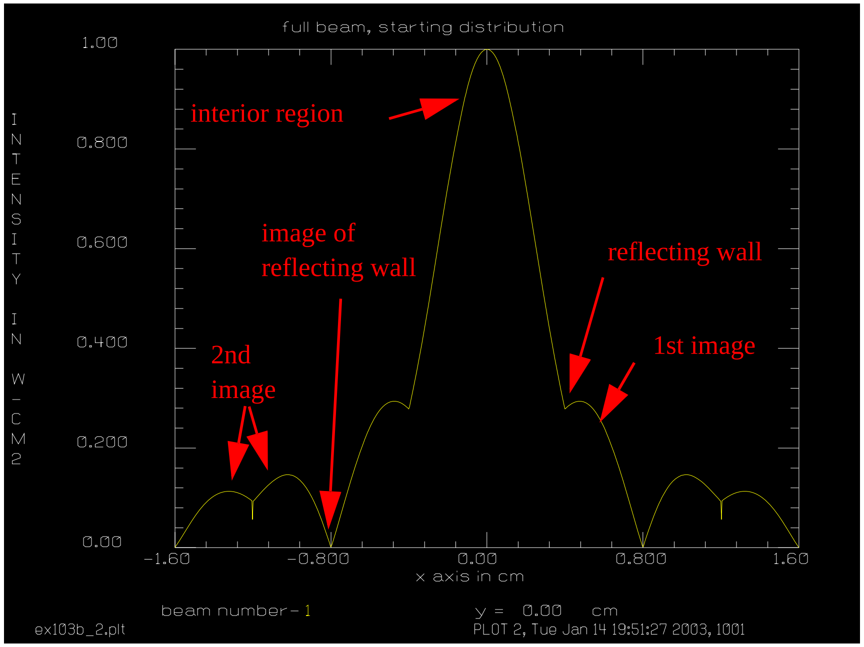
Profile of distribution at z = 98 cm. The beam peaks at odd Fresnel numbers and has a zero at even Fresnel numbers.

Considering the 2nd nearest neighbor

```
rod 1 Radius # 1st nearest neighbor
rod 1 [2*Radius] # 2nd nearest neighbor
```



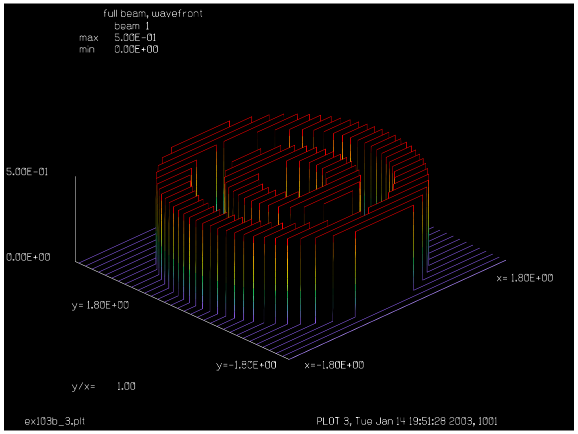
A gaussian distribution and 1st and 2nd images.



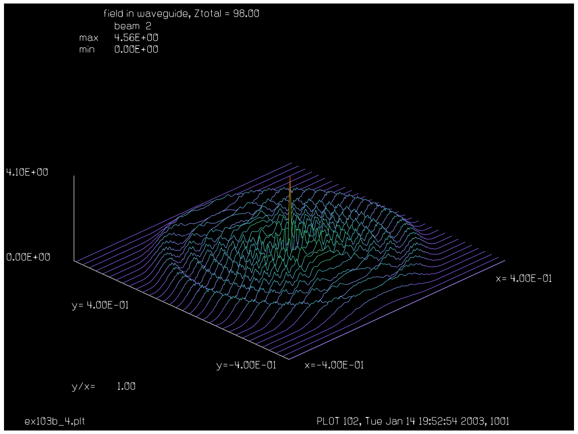
1st and 2nd images by considering the reflecting wall and the image of the reflecting wall.

2nd nearest neighbor allows longer propagation

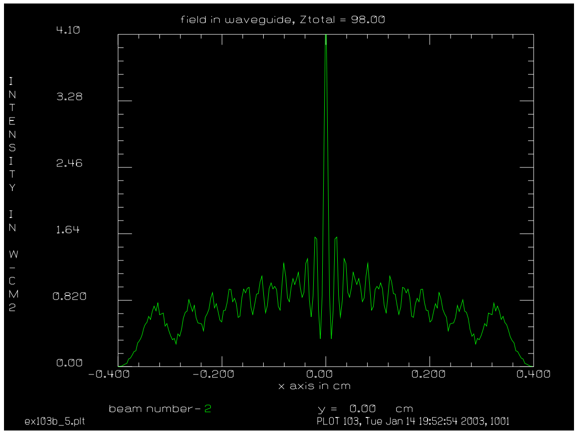
2nd nearest neighbor not needed at z = 98 cm for this case.



Phase of starting distribution. Phase are 0, interior of waveguide, π for first image, 0 and π for images of interior and 1st image, making four zones in all.



Distribution at z =98. The use of the 2nd image does not change distribution.



Profile of distribution at z = 98 cm. The beam peaks at odd Fresnel numbers and has a zero at even Fresnel numbers.

10. Propagation: thick elements, tilted surfaces

Outline

- consider Fresnel approximation and Fourier condition
- review elementary diffraction
- examples of common systems which exhibit non-Fourier behavior
- how to observe non-Fourier behavior in the laboratory
- diffraction behavior of some simple systems
- relation to well-known elementary optical principles
    - reduced length $\frac{L}{n}$
    - tunnel diagrams
    - Petzval curvature
- efficient calculation of non-Fourier behavior

Fresnel approximation

- modest diffraction angles (cosine terms may be neglected)
- parabolic wave approximation
- stationary phase approximation limits the effective width of influence function
- applicable for most common cases

Fourier condition

- influence function has constant shape and size over aperture
- near-field diffraction may be characterized by influence response functions
- propagation only allowed between concentric surfaces
- Fourier transforms or convolution may be used
- highly efficient, calculation time scales as $2N^2 \log_2 N$

Fourier Methods

convolution $a(x_1, y_1) = \iint h(x_0 - x_1, y_0 - y_1) a(x_0, y_0) dx_0 dy_0$ (10.1)

influence function $h(x, y) = \frac{\exp(jkz)}{j\lambda z} \exp\left(\frac{jk}{2z} r^2\right)$ (10.2)

transfer function $H(\xi, \eta) = \exp(-j\pi\lambda z\rho^2)$, and $\rho^2 = \xi^2 + \eta^2$ (10.3)

where ξ and η are spatial frequency parameters

Influence function

- influence function develops along optical ray
- stationary phase limits the effective size of the “influence”
- influence size is on the order of $\sqrt{\lambda z}$

- influence function $h(x, y) = \frac{\exp(jkz)}{j\lambda z}\exp\left(\frac{jk}{2z}r^2\right)$

Generalized transfer function for optical components and systems

generalized transfer function

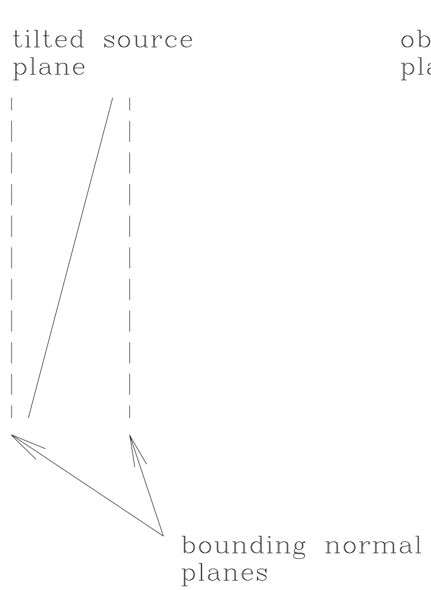
$$H(\xi, \eta) = \exp(-j\pi\lambda z\rho^2) \rightarrow \exp\left(-j\pi\lambda\int\frac{\rho(s)^2}{n(s)}ds\right) \rightarrow \exp(-j\pi\lambda z_{eff}\rho^2) \quad (10.4)$$

$$\int\frac{\rho(s)^2}{n(s)}ds = \rho^2\int_{s_0}^{s_1}\frac{1}{n(s)M^2(s)}ds = \lambda z_{eff}\rho^2 \quad (10.5)$$

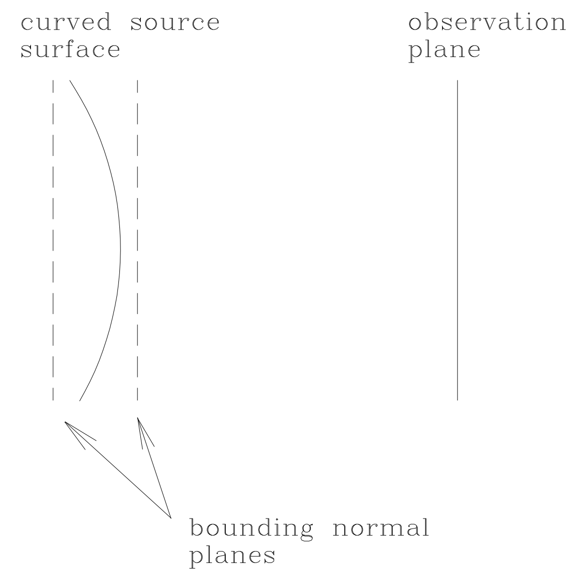
- z_{eff} effective propagation distance
- $n(s)$ is the index of refraction along the ray
- $M(s)$ is the magnification along the ray
- s is the path length parameter along the ray

Effective propagation distance

- determines evolution of local diffraction effects
- determines size of influence function $\sqrt{\lambda z}$
- must be constant across aperture for Fourier condition(requires concentric surfaces)
- the extremes of the surfaces will always satisfy Fresnel condition



tipped plane

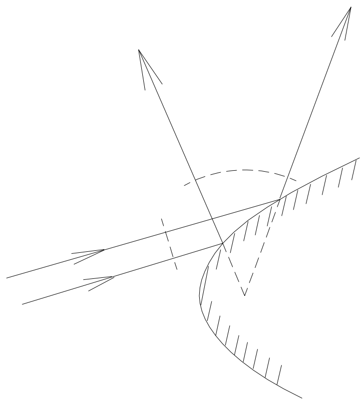


dished surface

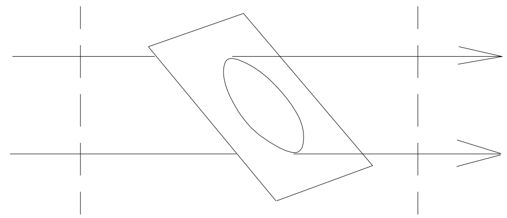
Some non-Fourier elements and systems

- influence function width varies significantly

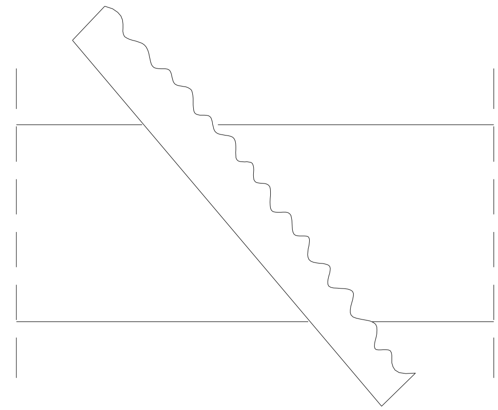
grazing incidence mirror



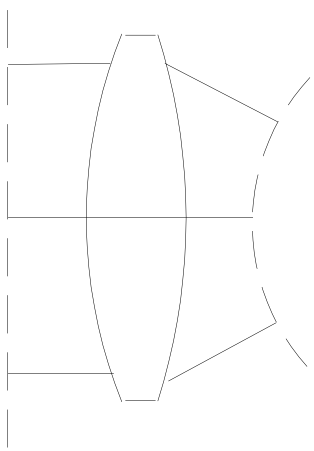
tilted aperture



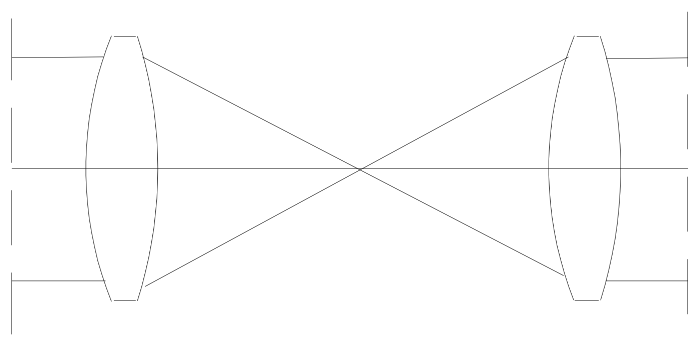
tilted phase plate



thick lens



telescope



How to observe non-Fourier behavior

- high spatial frequency structure in the image is most affected
  - diffraction ringing
  - phase plates with fine structure, Ronchi ruling, phase gratings

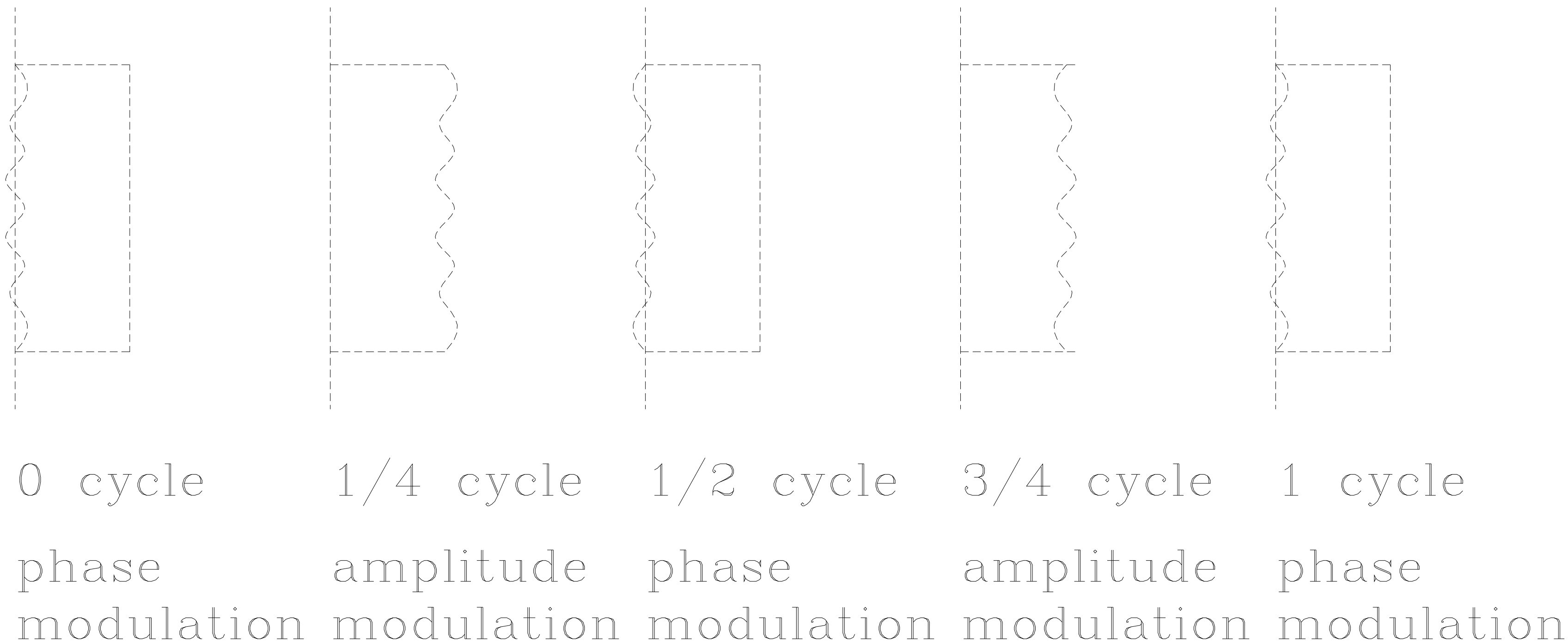
phase gratings

- small amplitude phase gratings
- phase converted to irradiance modulation in one-quarter Talbot cycle
  - one-quarter Talbot cycle = $z_{1/4T} = \dfrac{1}{2\lambda f^2}$
- modulation vs. distance = $\sin\left(\dfrac{z}{z_{1/4T}}\right) \approx \dfrac{z}{z_{1/4T}}$ for $z \ll z_{1/4T}$

  (modulation varies linearly for small distance)

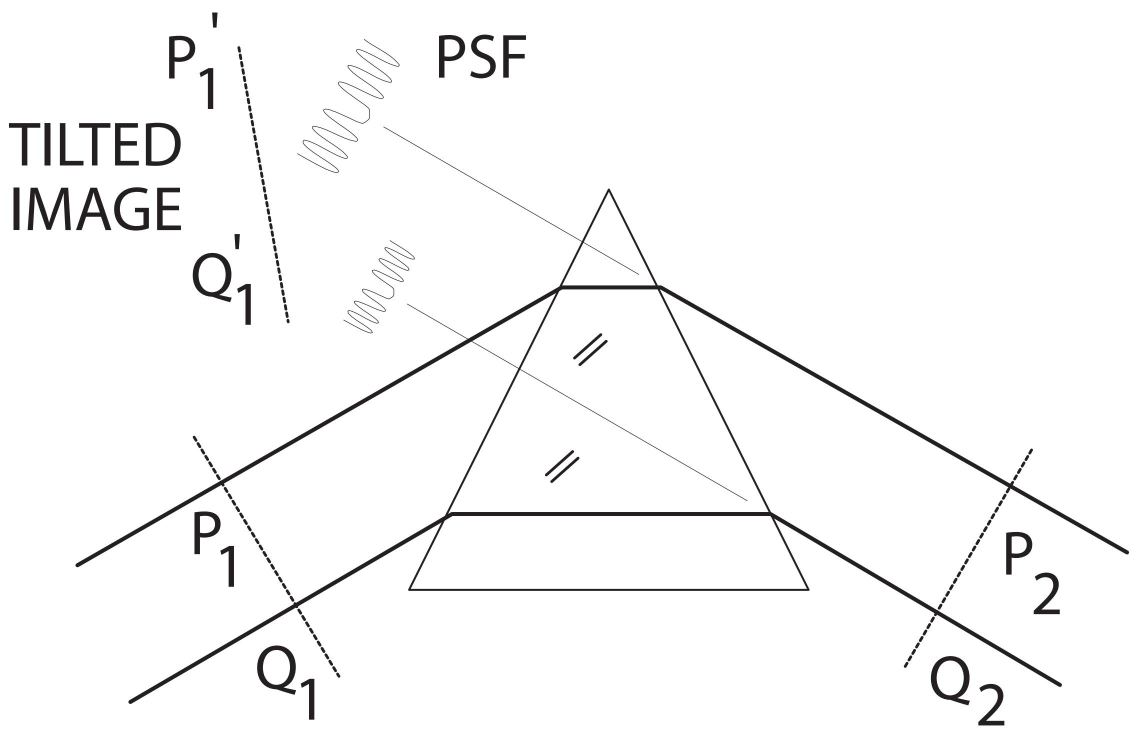
Talbot imaging

- bands of irradiance form in collimated, free space propagation
- sheared Talbot bands form for tilted planes
- curved Talbot bands form for “dished” surfaces



Perfect, thick prism

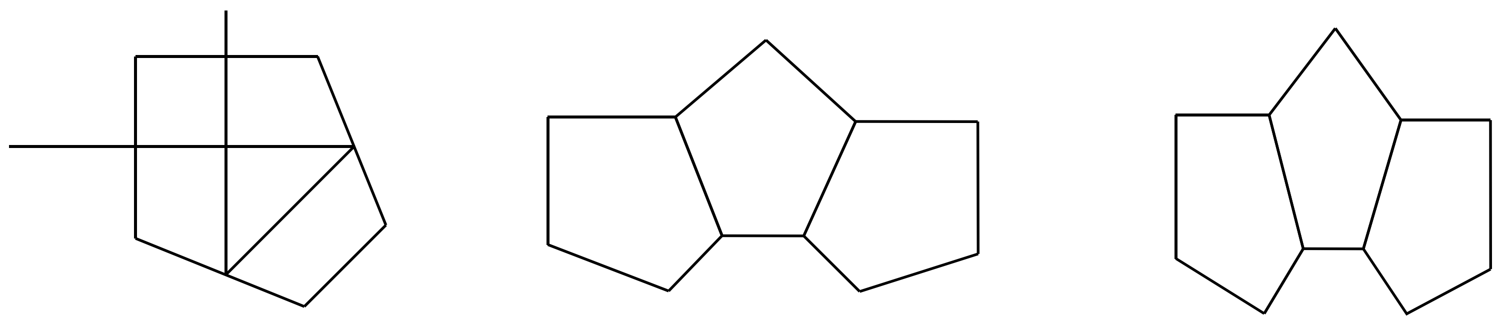
- optical path difference errors are zero, *nL*
- geometrical optics predicts perfect performance
- reduced length $z_{eff} = \int \frac{1}{n(s)} ds \approx \frac{L}{n}$ varies over aperture(reduced length is shortest for longest glass path)
- reduced length is equivalent air path



Tunnel diagram

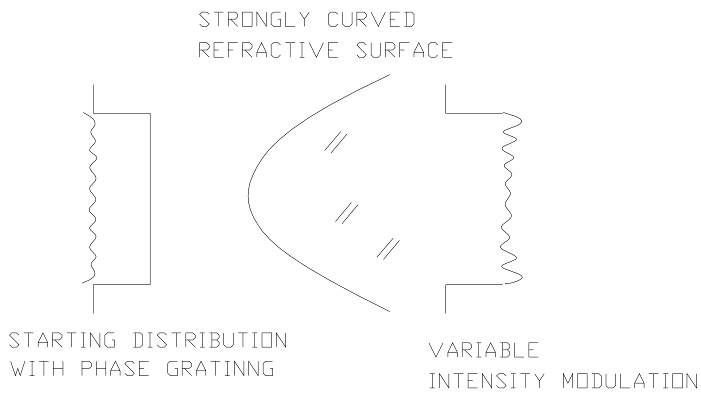
- A prism may be represented by an equivalent air path whichhas identical diffraction properties
- effective propagation distance => reduced length => equivalent air path

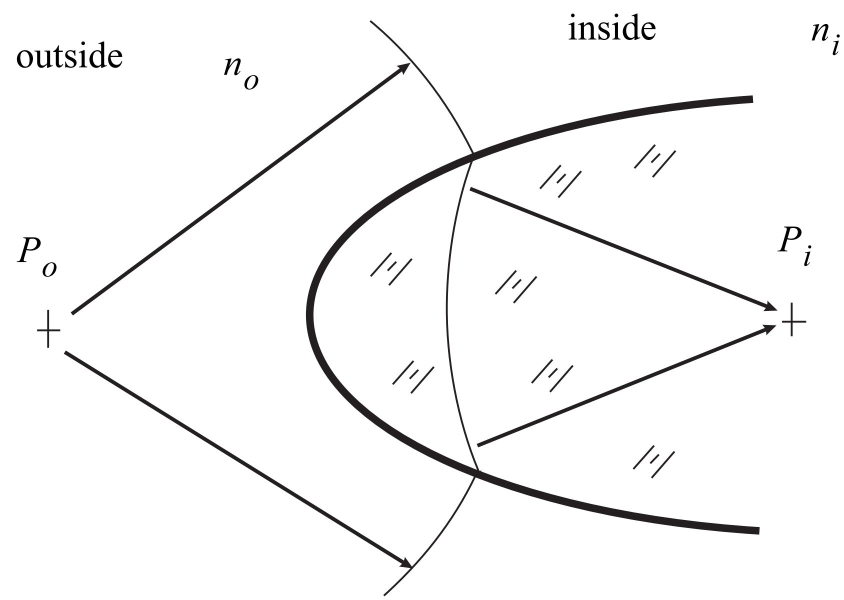
actual prism — unfolded prism — air-equivalent path



Thick refractive surface

- deep refractive surface with small index difference generateslarge variations of z_{eff} while staying within Fresnel approximation
- outer region has longest air path and longest reduced length
- center region has shortest reduced length





A beam partially into a medium with a spherically refracting lens surface.

Methods of calculation

- general treatment

$$a(x_1, y_1) = \iint h(x_0, y_0; x_1, y_1) a(x_0, y_0) dx_0 dy_0 \tag{10.6}$$

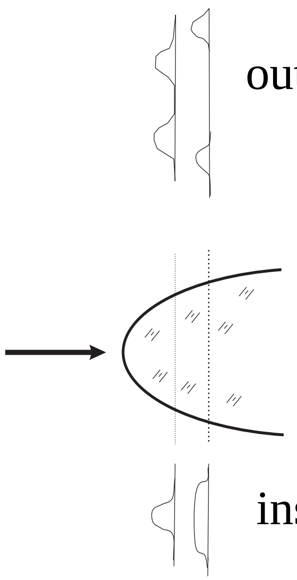
$$h(x_0, y_0; x_1, y_1) = \frac{1}{j\lambda r_{01}} \exp(jkr_{01}) \tag{10.7}$$

- For a two-dimensional array of size $N \times N$, we will need approximately N^4 elementary calculations (add-multiply pairs)
- aperture division
- nonlinear mapping

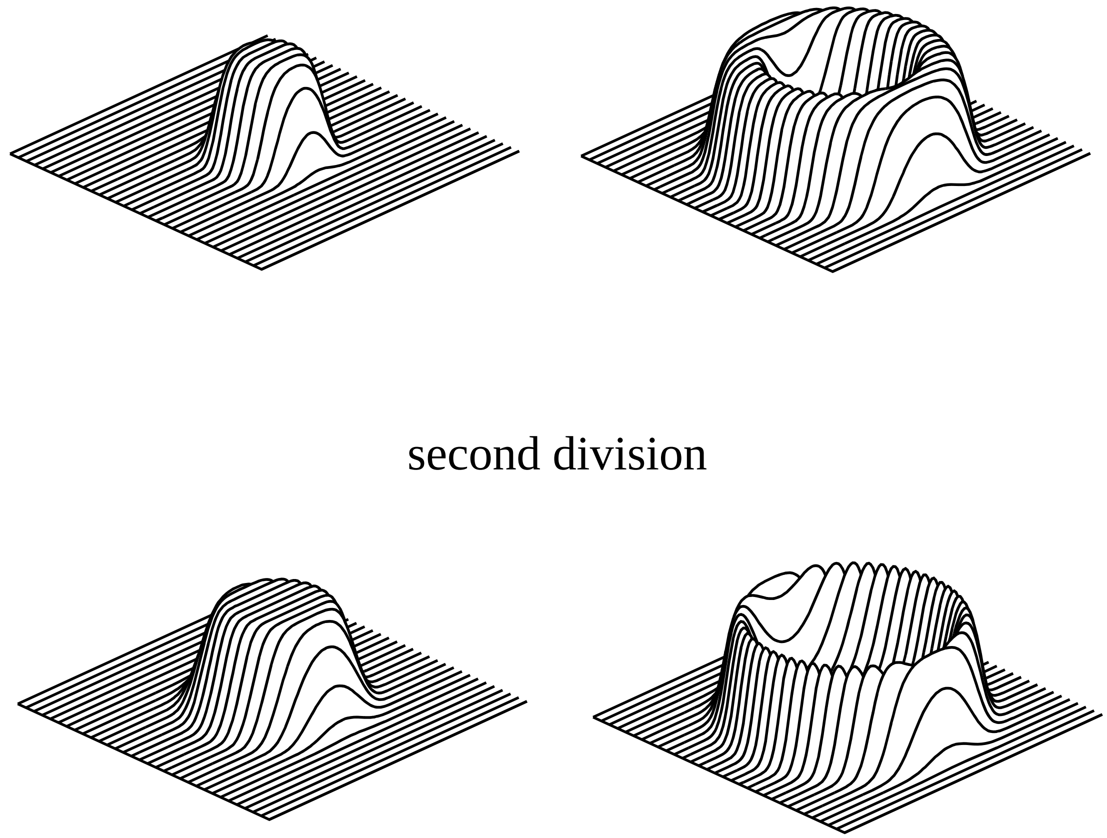
Non-Fourier propagation by aperture division

- divide aperture into regions
- each region shares (almost) an effective propagation distance
- propagate each region separately — *N* propagations required
- solution is approximate
  - tilted source is represented by stair step pattern
  - curved source is represented by radial zones
- soft-edged divisions reduce artifacts at zone boundaries

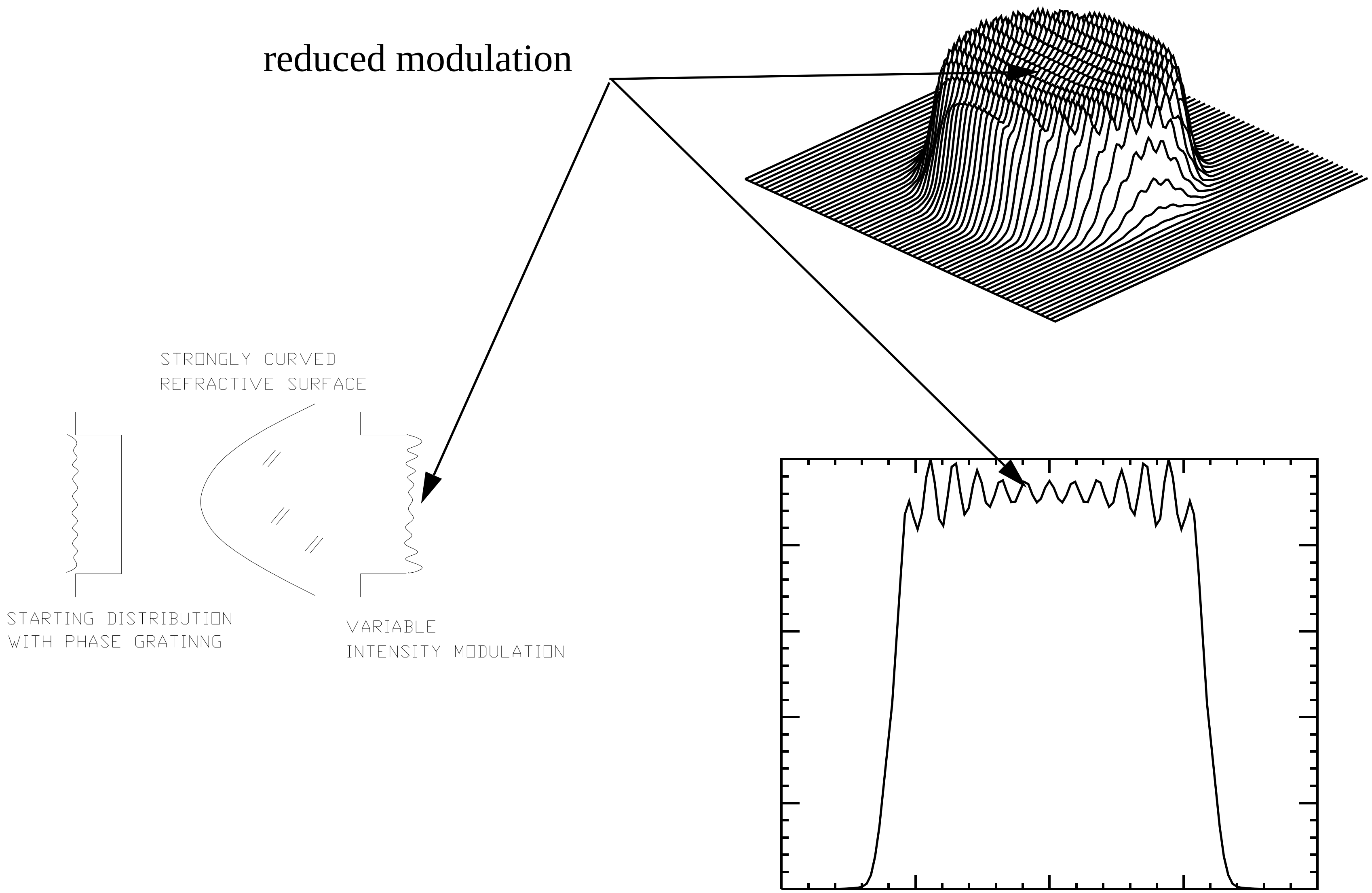
Aperture division



The aperture is split into inside and outside regions with a soft aperture such as a supergaussian function and its complement. The inner portion is extended until all of the incident beam is inside the medium.

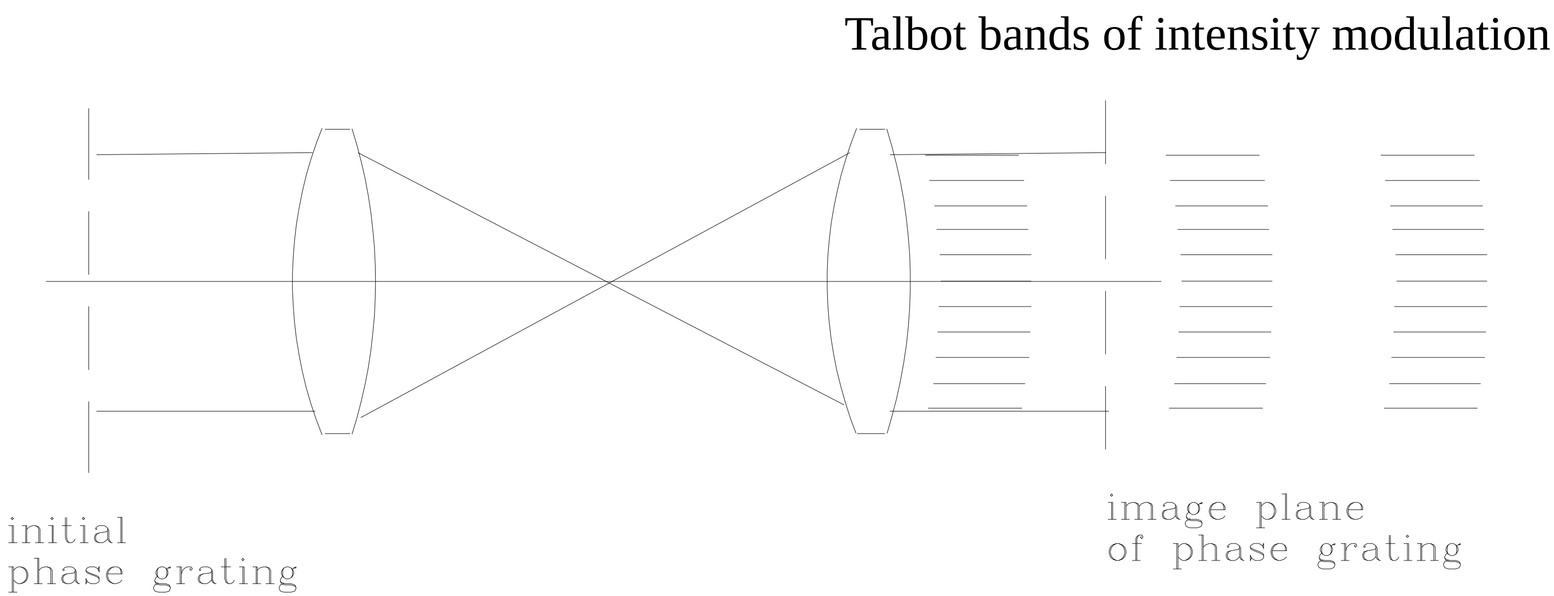


Thick lens has short reduced length on axis: ex35b.inp



Perfect 1:1 telescope with thick elements

- free of phase aberrations
- reduced distance is shortest in center
- curved Talbot bands follow locus of equal reduced length
- curved bands conform to curved Petzval surface from aberration theory



Optical elements and systems

- elementary optical theory can be used to image of diffracting source
  - image position
  - tilt of image
  - curve of image surface
- any complex system (with no intermediate apertures) may be representedby an air-path equivalent
- only equivalent air-path systems need be considered
  - concentric
  - tipped
- dished systems

Nonlinear mapping

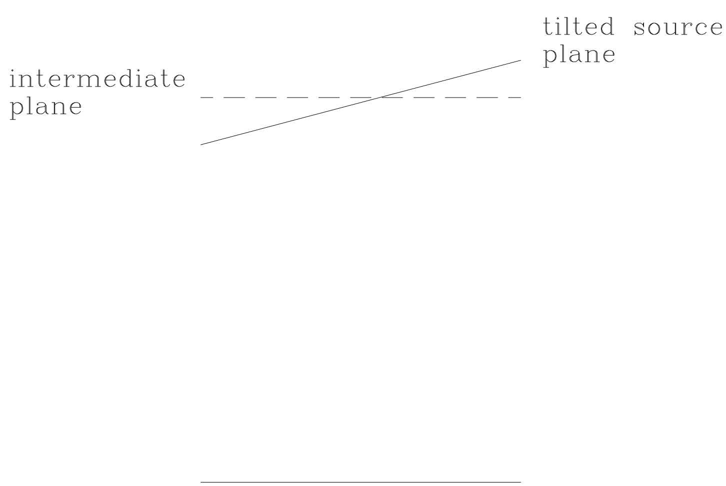
- unequal effective propagation distances cause variation in sizeof influence function
- Fourier methods require constant-size influence functions $\sqrt{\lambda z_{eff}}$
- degree of diffraction evolution depends on relative size of featuresto size of influence function
- solution:
  - select intermediate value of effective propagation distance $\sqrt{\lambda z_{eff_0}}$
  - remap source distribution by local magnification,

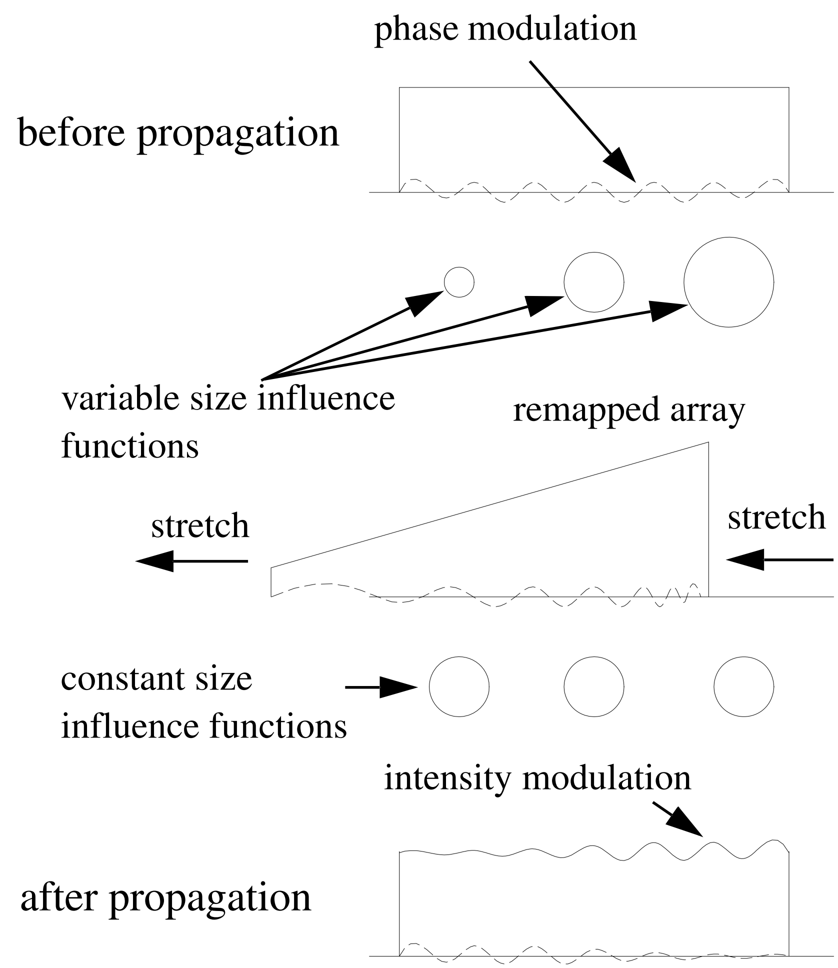
    $$m(x, y) = \sqrt{\frac{z_{eff_0}}{z_{eff}(x, y)}}$$

    where $z_{eff}(x, y)$ is the local effective distance
  - propagate z_{eff_0} by Fourier methods
  - inverse mapping by $\frac{1}{m(x, y)}$
  - two interpolation steps take about same time as one propagation step
  - allow extra guardband because of interpolation “growth”

Nonlinear mapping for tilted plane

- regions with short $z_{eff}(x, y)$ are expanded
- regions with long $z_{eff}(x, y)$ are compressed
- Fourier methods used for propagation





Conclusion

- many ordinary optical elements and systems depart significantly from Fourier condition in near-field propagation
- high spatial frequency phenomena are most affected
- non-Fourier effects are readily observed in the laboratory
- the effective propagation distance is a useful measure of diffraction evolution(similar to reduced length and air-equivalent tunnel diagrams)
- calculation methods offer high efficiency
  - aperture division — soft sided apertures
  - nonlinear mapping

11. Statistical Optimization: Gerchberg-Saxton and Simulated Annealing

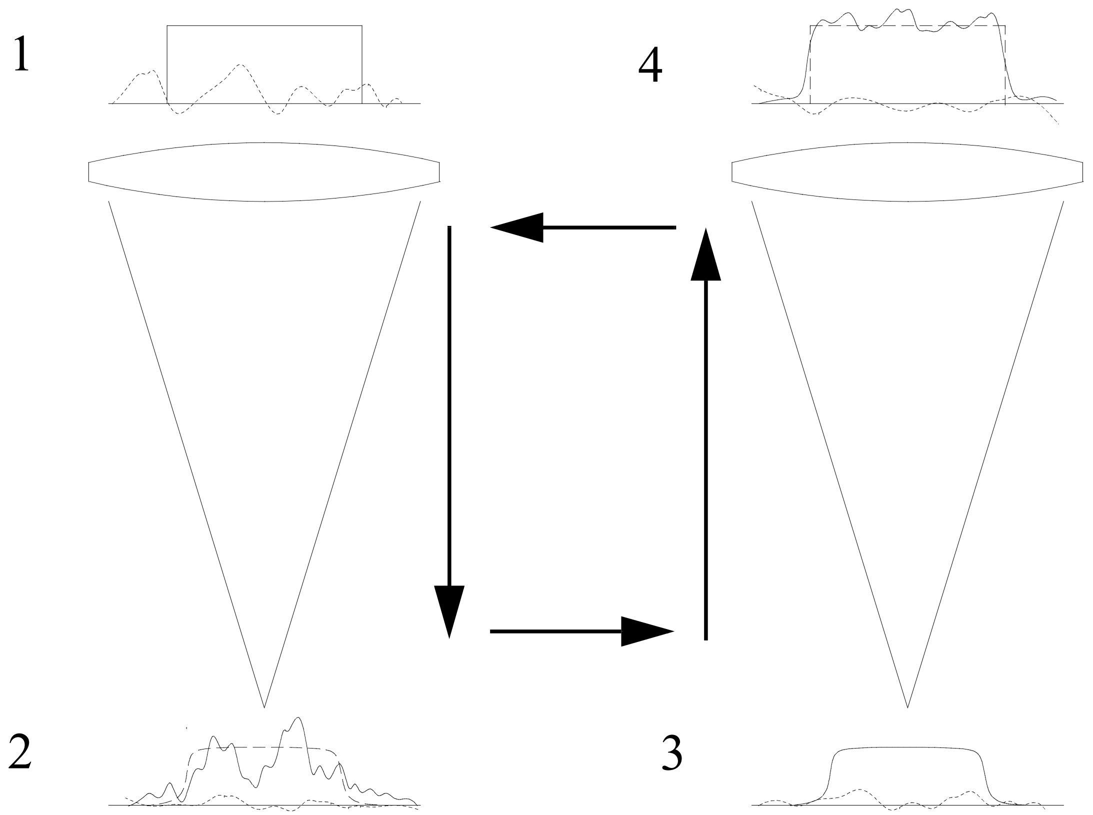
Gerchberg-Saxton (also known as phase retrieval method)

- Solve for phase in pupil to achieve desired far-field intensity envelope
- Used successfully for Hubble Space Telescope to determine aberration error
- May be used for synthesizing far-field envelope(you can write your initials in the far-field)
  - solution has speckle
  - near-field phase may have discontinuities (poles) leading to scatter
- Does not allow special constraints
- Fast execution

Simulated annealing

- Achieves “optimum” solution (global method)
- Accomodates special constraints
- Any target definitions
- Very slow
- Rather strange

Gerchberg-Saxton Phase Retrieval (Phase Retrieval)



Step 1. Normalize near-field irradiance to match input (flat-top). Leave phase unchanged.

Step 2. Calculate far-field irradiance

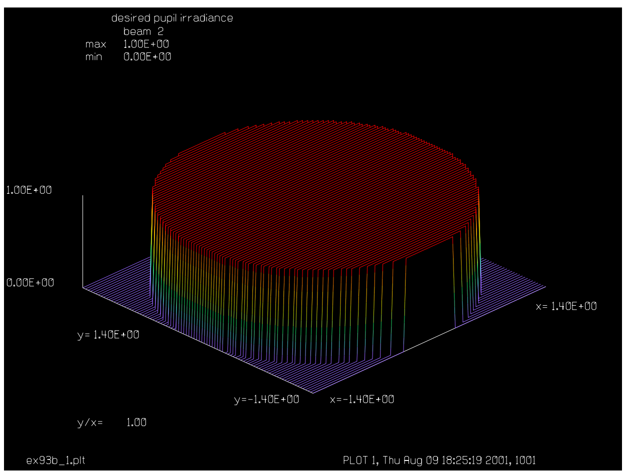
Step 3. Normalize far-field to target envelope. Leave phase unchanged.

Step 4. Back-propagate to near field.

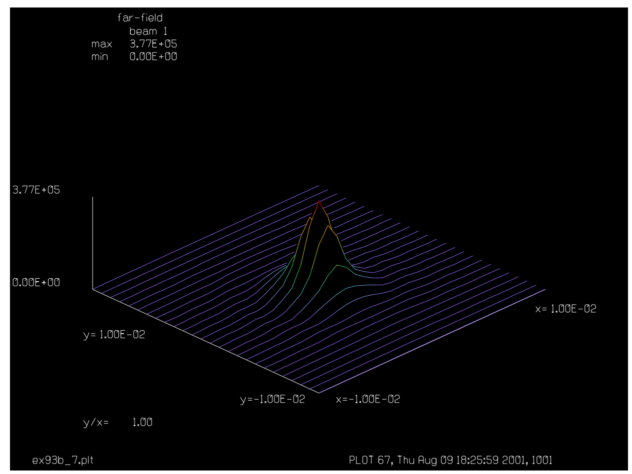
(repeat to Step 1-4)

Gerchberg-Saxton Examples: ex93a.inp , ex93b.inp

- Similar to Hubble Space Telescope problem
- Start with far-field irradiance envelope of star image with 0.5 waves of spherical aberration.
- Step 1. Assume flat-top irradiance and and flat phase in the pupil.
- Step 2. Propagate to far-field.
- Step 3. Normalize to far-field irradiance of aberrated image, leaving phase unchanged.
- Step 4. Back-propagate to near-field.
- (Repeat Steps 1-4).

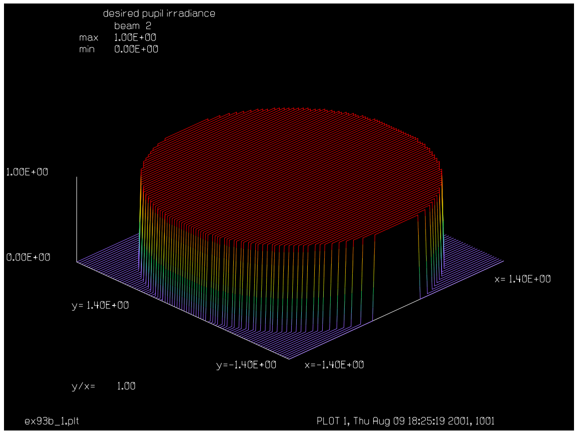


Flat-top pupil irradiance.

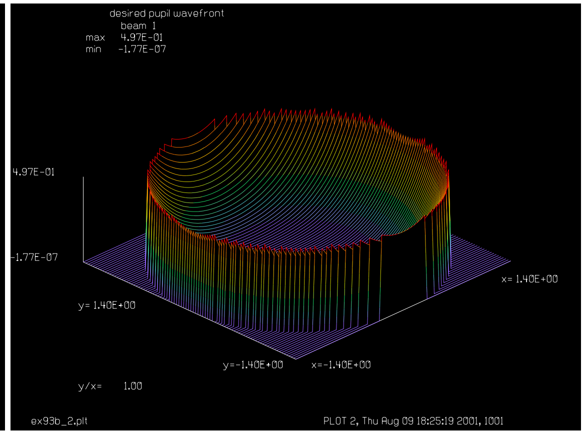


Far-field irradiance target.

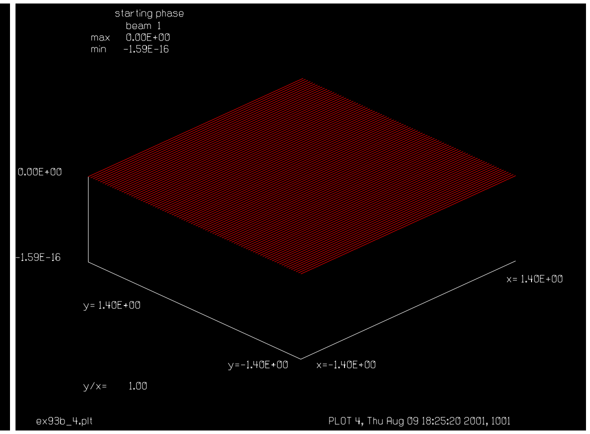
Gerchberg-Saxton: Determining Phase from Far-Field Irradiance



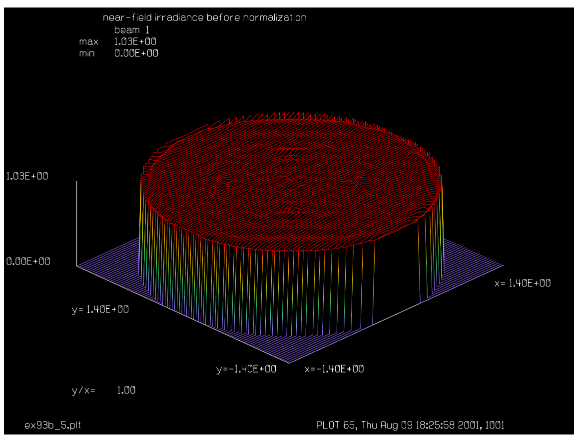
Start. Flat-top pupil irradiance.



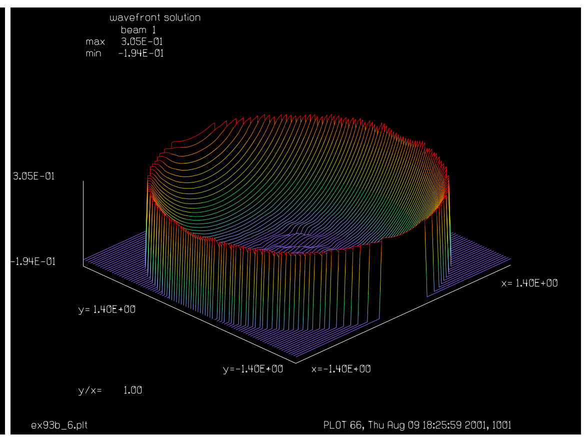
Actual pupil phase is 0.5 waves of spherical aberration.



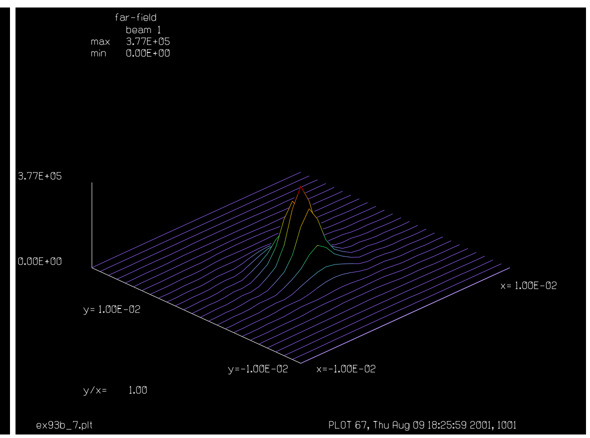
Starting phase is flat.



Recreated pupil irradiance after 60 passes.



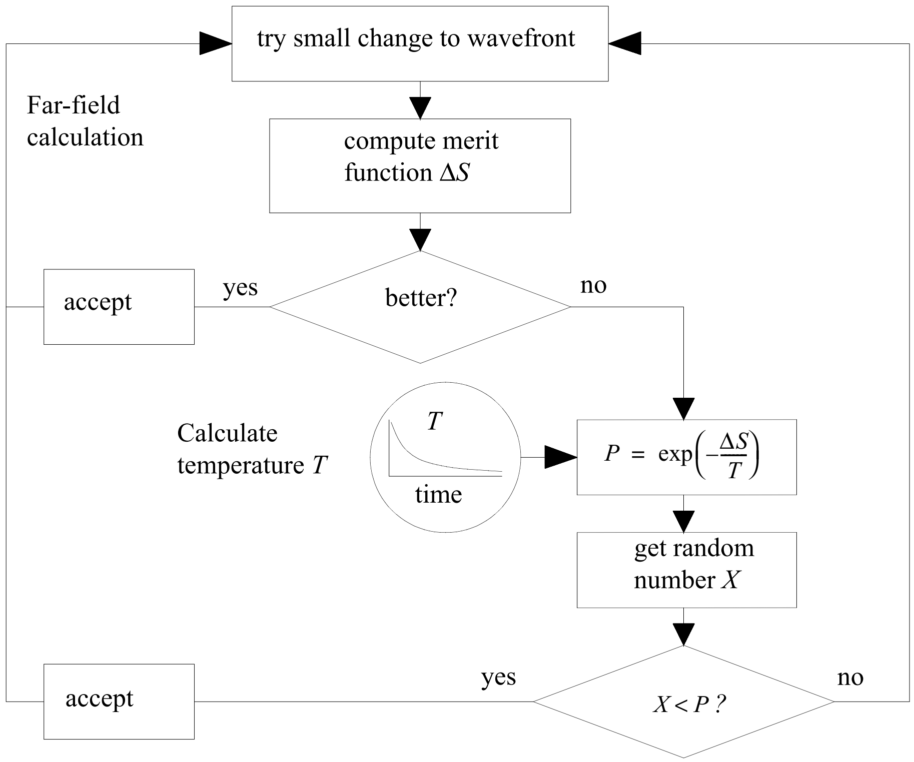
Recreated pupil phase.



Far-field irradiance target.

Simulated Annealing

- Define target function — far-field envelope to be supergaussian in this example
- Define merit function — perhaps RMS errors with respect to target function
- Start with a deliberately bad fit
- Guess at an incremental improvement
- compute merit function
  - If better, accept
  - If worse, sometimes accept

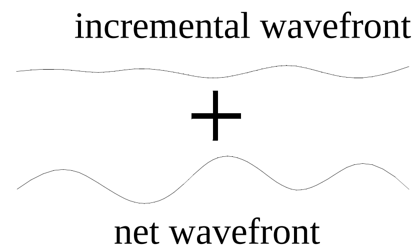


Simulated Annealing: ex98a.inp and ex98b.inp

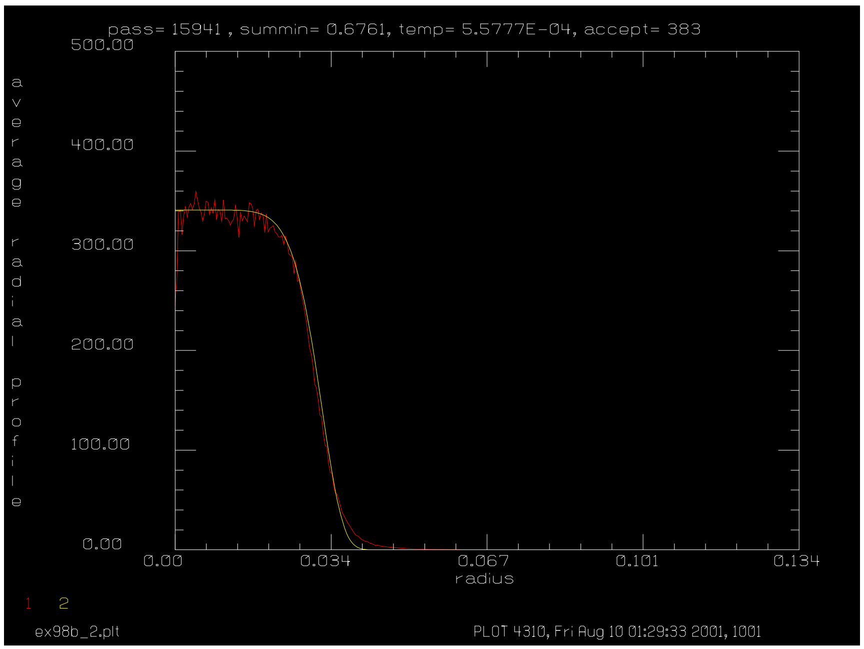
- Examples: ex98a.inp start, and ex98b (continue)
- seek a supergaussian envelope in the far field:

$$\text{target shape} = t(r) \;=\; \exp\left[-2\left(\frac{r}{r_0}\right)^M\right]$$

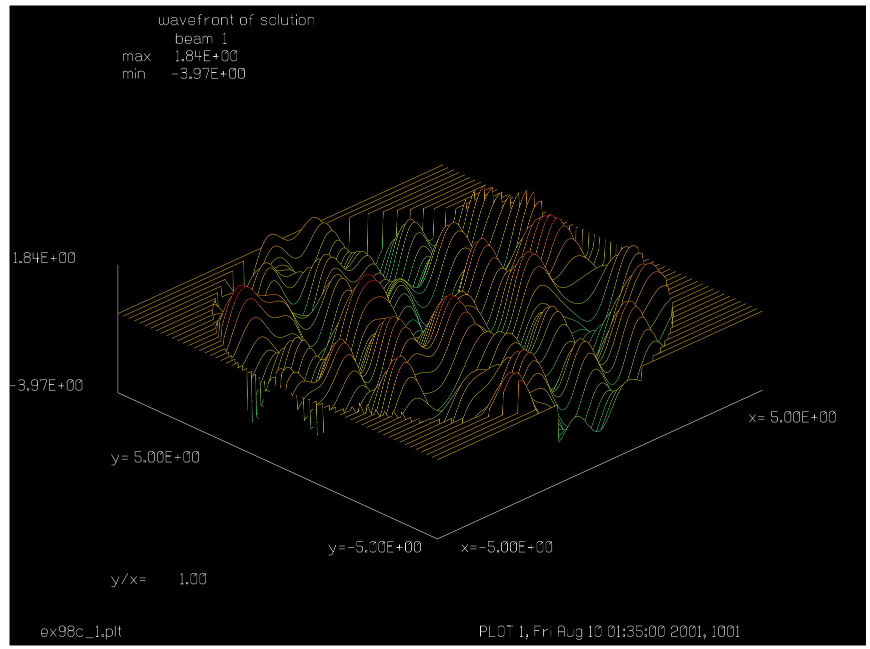
- use smoothed random phase plate
- evaluate average radial profile, circular pattern



An incremental wavefront is added to the net wavefront, if the system is improved, the incremental change is kept.



Average radial profile (red) fit to target supergaussian (yellow), after 16,000 iterations.



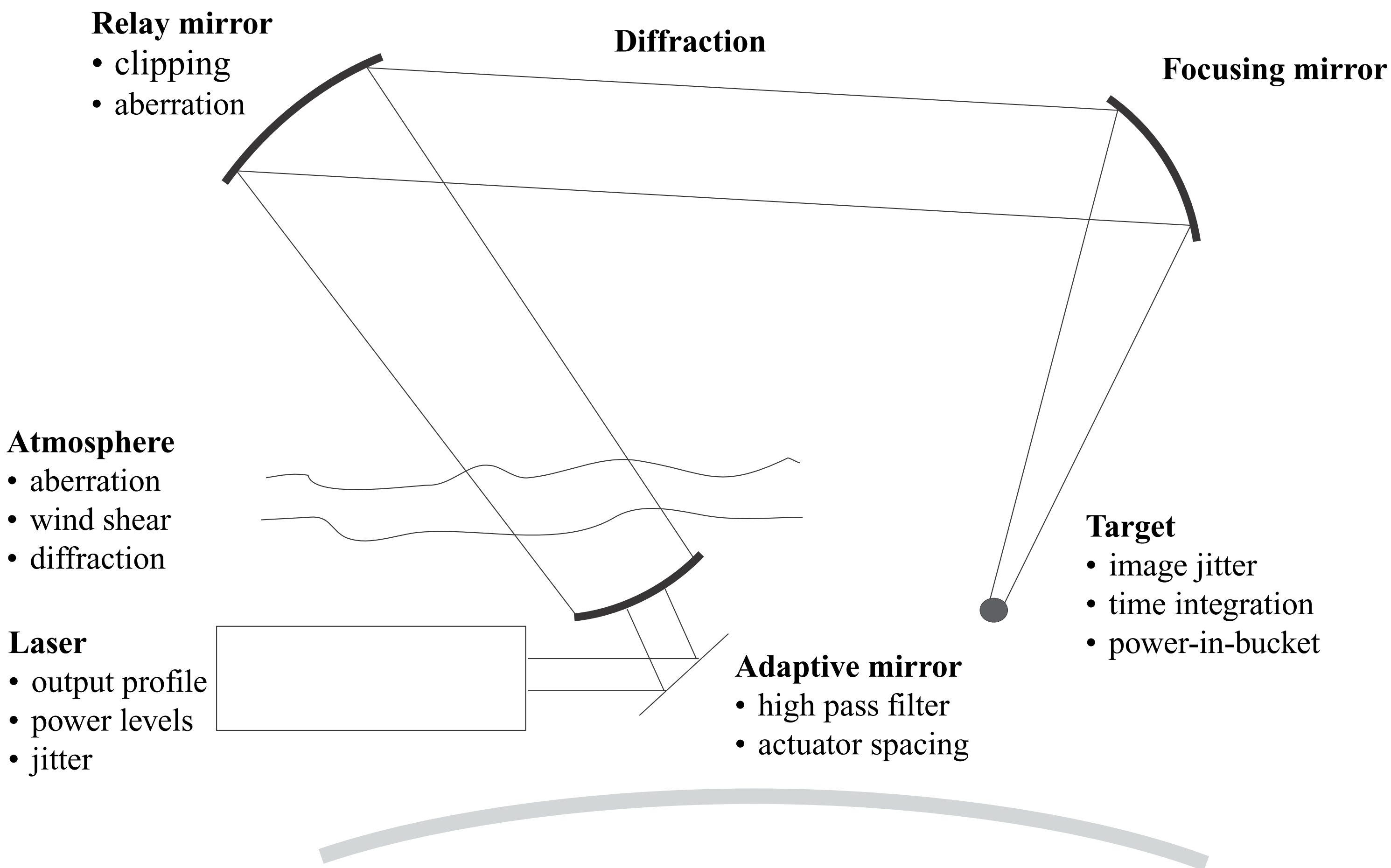
Wavefront after 16,000 iterations — the solution yields the result on the left.

12. Atmospheric Effects

Atmospheric phenomena effecting optical propagation

- absorption
  - Beer's Law $I(z) = I_0 e^{-\alpha z}$
- scattering
  - Beer's Law $I(z) = I_0 e^{-\alpha z}$
- aberration due to turbulence
  - smoothed random, phase sheets based on Kolmogorov statistics
  - phase sheets and propagation via split step method
  - adaptive mirrors can partially compensate
- thermal blooming
  - heated air lowers refractive index to create self-defocusing: phase screens
  - wind and beam skewing create relative shear. burns a trough of low index air
  - adaptive optics can partially compensate

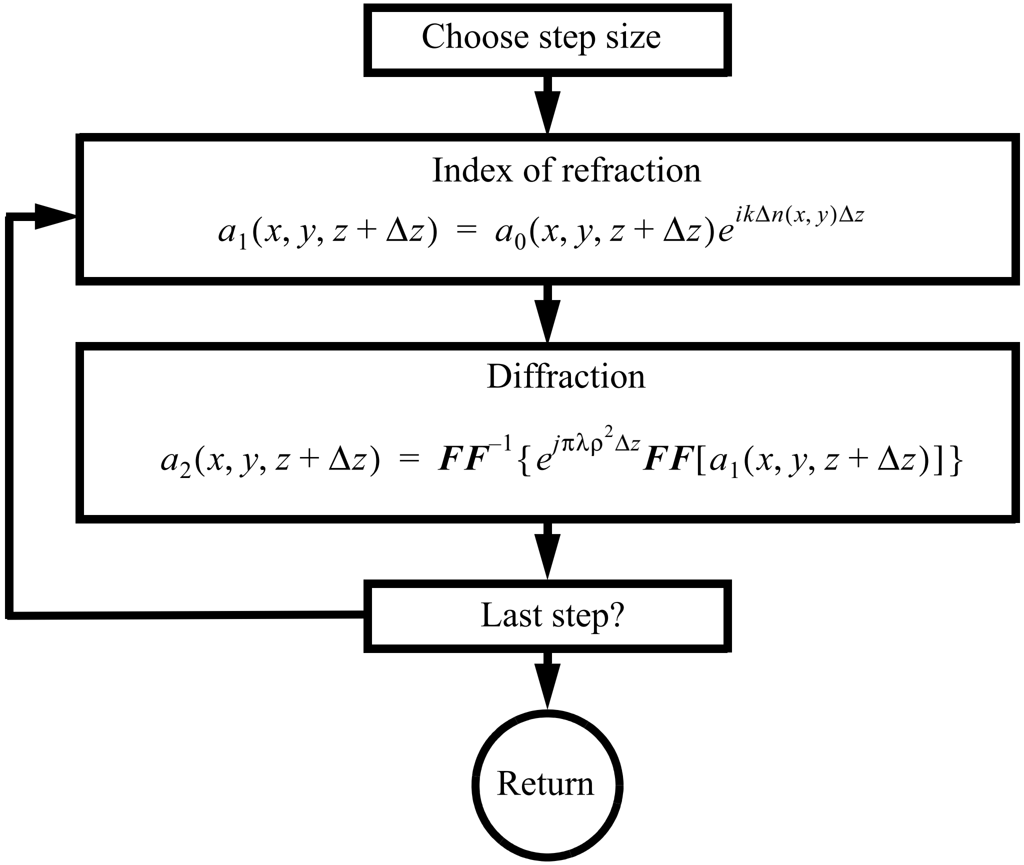
Large system



Split-step implementation of atmospheric model

- Turbulent flow in the atmosphere is may be characterized by a distribution of blobs ("turbs") of regions of relatively constant temperature.
  - atmosphere represented as gradient index distribution $n(x, y, z)$
- Construct a series of phase screens in a split step solution with diffraction
  - each phase screen represents the index distribution over a modest distance Δz

Split step applied to propagation through atmosphere with *n*(*x*,*y*,*z*)



Flow chart for split-step method of treating diffraction and the refractive index function $\Delta n(x, y)$. For a small step Δz, the effect of the refractive index is implemented as a phase screen of the form $\exp[jk\Delta n(x, y)\Delta z]$ to change the initial complex amplitude $a_0(x, y, z)$ into the intermediate result $a_1(x, y, z)$. A diffraction step is applied to the intermediate result, implemented by FFT methods.

Kolmogorov wavefront power spectrum

Statistics of atmospheric aberration

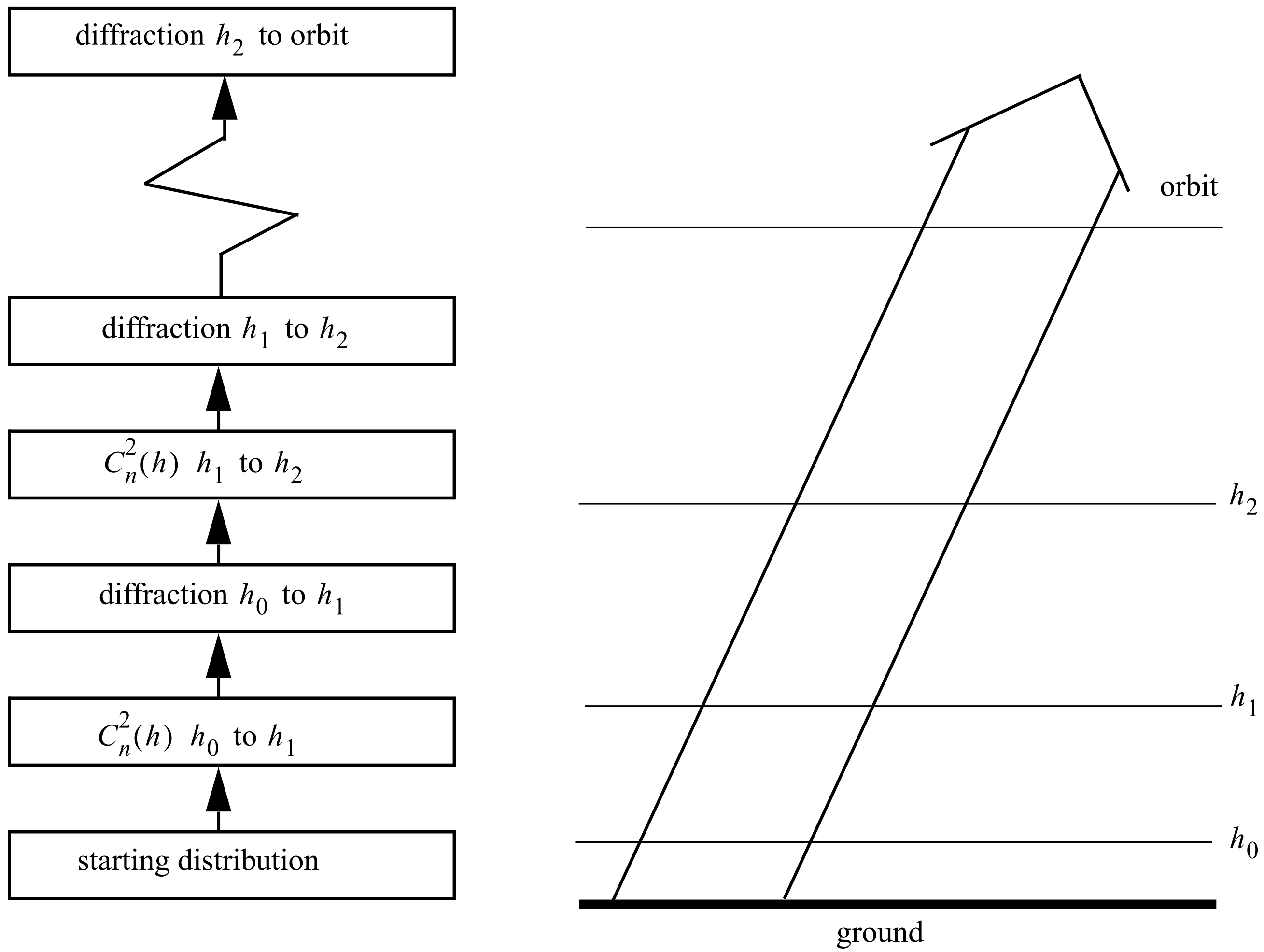
$$W^2(\rho) = \frac{0.38}{\lambda^2 \rho^{11/3}} \int_{h_{\min}}^{h_{\max}} C_n^2(h)\,dh \tag{12.1}$$

$W^2(\rho)$	wavefront power spectrum
$C_n^2(h)$	atmospheric structure at height h
ρ	spatial frequency

- An approximate expression for night time adequate for estimation purposes is

$$C_n^2(h) = 2\times10^{-12} h^{-4/3}. \tag{12.2}$$

Propagation through levels of atmosphere



Schematic of propagation through the air where aberration and diffraction propagation must be considered. The propagation is done in relatively short steps with the diffraction propagation and addition of aberration alternated to achieve an approximation to a continuous process.

Use of Fried's parameter to characterize atmospheric "seeing"

Fried's parameter characterizes atmospheric "seeing"

$$r_0 = \left[0.423k^2 \int_{h_{\min}}^{h_{\max}} C_n^2(h)dh\right]^{-5/3} . \tag{12.3}$$

Fried's parameter may be used to calculate the wavefront power spectrum

$$W^2(\rho) = \frac{0.23}{r_0^{5/3}\rho^{11/3}} . \tag{12.4}$$

Lutomirski and Yura correction factors

$$W^2(\rho) = \frac{0.023e^{-\rho^2 L_i^2}}{r_0^{5/3}\left(\rho^2 + \frac{1}{L_o^2}\right)^{11/6}} \tag{12.5}$$

L_o is the outer scale

L_i is the inner scale.

Using Fried's parameter

- The summation of the r_0's of different levels takes the form

$$r_{\text{total}} = [r_1^{-5/3} + r_2^{-5/3} + \ldots]^{-3/5}. \tag{12.6}$$

Propagation through N layers of equivalent aberration of r_0, which might occur in horizontal propagation, will result in

$$r_{\text{total}} = \frac{r_0}{N^{3/5}}. \tag{12.7}$$

For a wavelength of 0.5×10^{-6} meters and the simplified expression for $C_n^2(h)$,

$$C_n^2(h) = 2\times10^{-12} h^{-4/3}. \tag{12.8}$$

Eq. (12.8) can be evaluated. For a lower limit of $h_1 = 10$ meters, $r_0 \approx 3$ cm. For day time conditions, $r_0 \approx 0.7$ cm. The seeing may be significantly improved by raising the height of the laser source and by designing the laser station to minimize local turbulence.

Characteristic diffraction length

From Talbot imaging the quater Talbot cycle determines the propagation distance for conversion of phase to intensity

$$z_{char} = \frac{T^2}{2\lambda} \approx \frac{r_0^2}{2\lambda} \qquad (12.9)$$

For $r_0 \approx 3$, typical of night time seeing, and 0.5 μ wavelength, z_{char} = 900 meters.

For $r_0 \approx 0.7$, typical of day time seeing, and 0.5 μ wavelength, z_{char} = 49 meters.

As series of N propagation steps less than z_{char}, with Fried's parameter $r_0 = r_{\text{total}} N^{3/5}$ for each step, are required to model the path.

- Wind: Given a wavefront of $w(x, y)$ and velocity components v_x and v_y, the shift in the atmospheric aberration with time is

$$w(x - v_x t, y - v_y t) = \mathrm{FF}^{-1}\left[e^{-2\pi(v_x\xi + v_y\eta)t}\mathrm{FF}[w(x, y)]\right], \qquad (12.10)$$

where FF indicates a two-dimensional Fourier transform.

Atmospheric aberration with steady wind (x-direction) ex29.inp

.

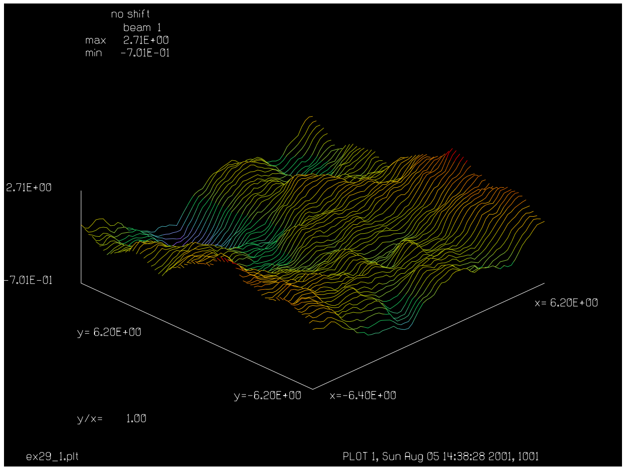


Fig. 12a.1. No shift.

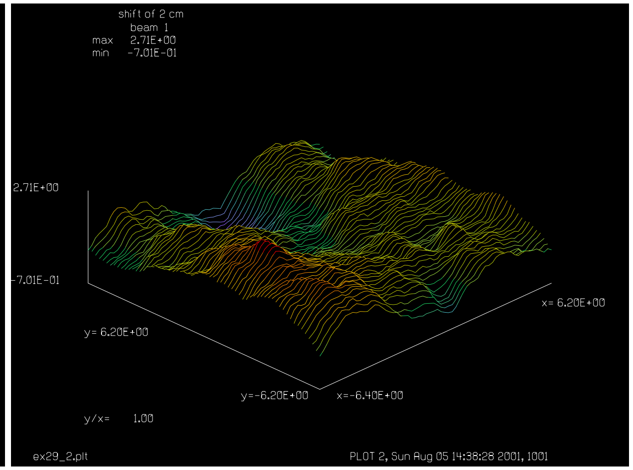


Fig. 12b.1. Shift of 2 cm.

wind shift of 0, 2, 4, and 6 cm for wind blowing in the x-direction

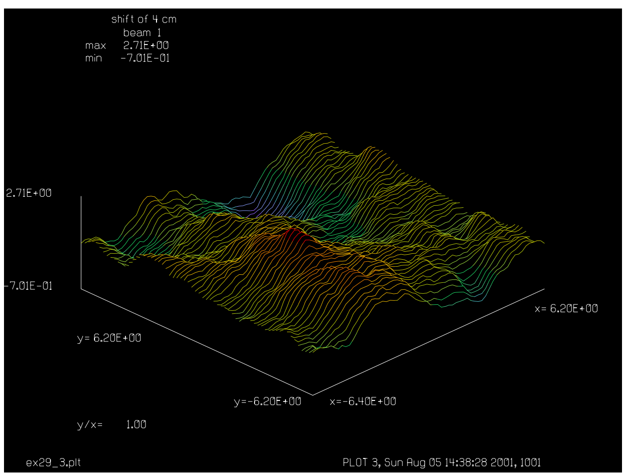


Fig. 12c.1. Shift of 4 cm.

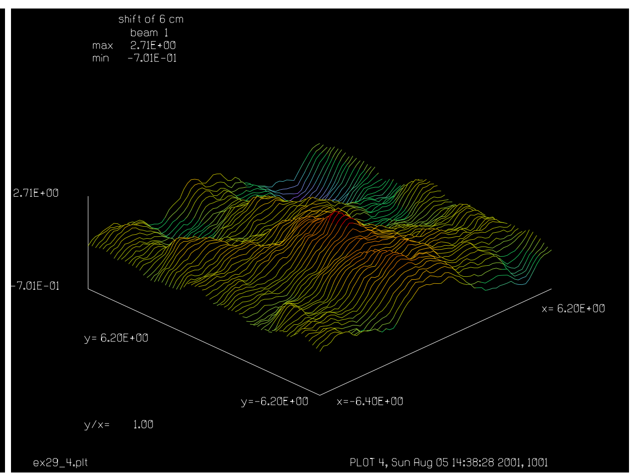


Fig. 12d.1. Shift of 6 cm.

Atmospheric aberration continuously changing with no wind

- Changing pattern to model dynamic atmosphere change (no wind)
  - current wavefront is decreased $\exp(-\Delta t/\tau)$
  - new wavefront, with a proportional reduction $[1-\exp(-\Delta t/\tau)]$ is added
  - wavefront aberration level is statistically constant
  - wavefront changes

$$W(x, y, t_1) = W(x, y, t_0)e^{-\frac{\Delta t}{\tau}} + W(x, y)_{new}\left(1 - e^{-\frac{\Delta t}{\tau}}\right) \tag{12.11}$$

$W(x, y)$ is the current wavefront

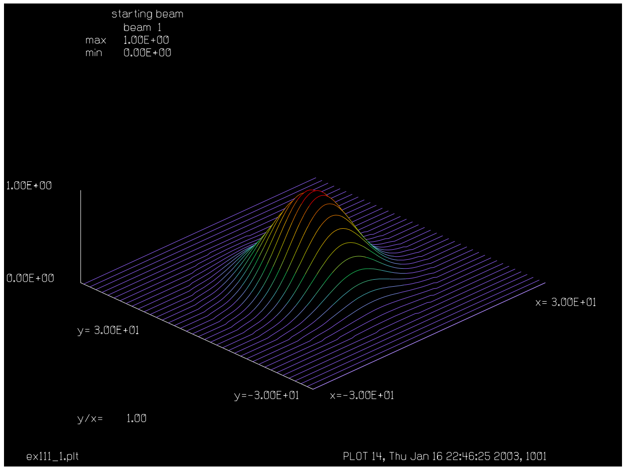
$W_{new}(x, y)$ is a new instance of a Kolmogorov wavefront

Δt is a time interval

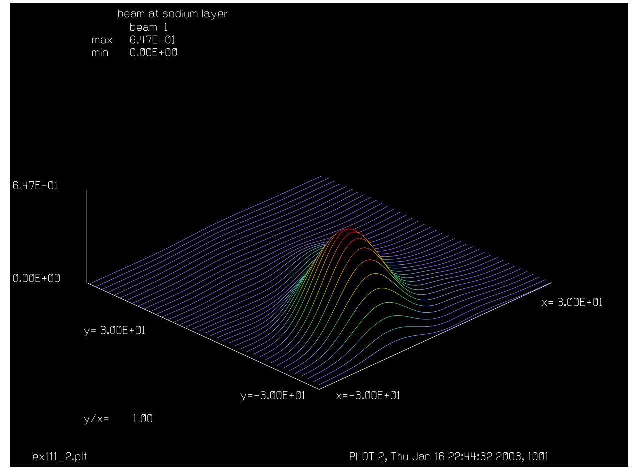
τ is the time constant for change of the atmosphere

Movie shows dynamic change **imsicap.avi**

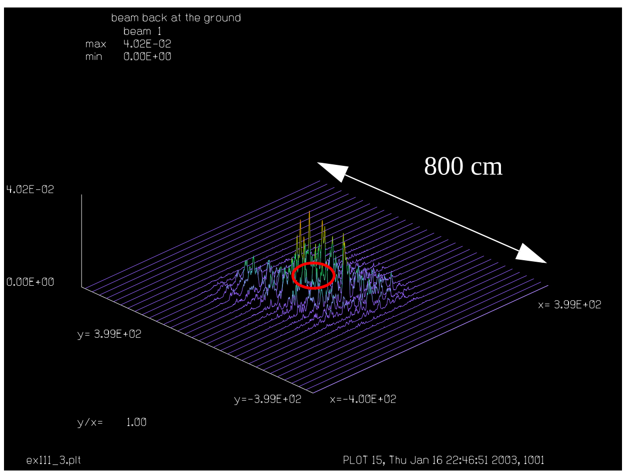
Propagation to sodium layer and backscatter to ground



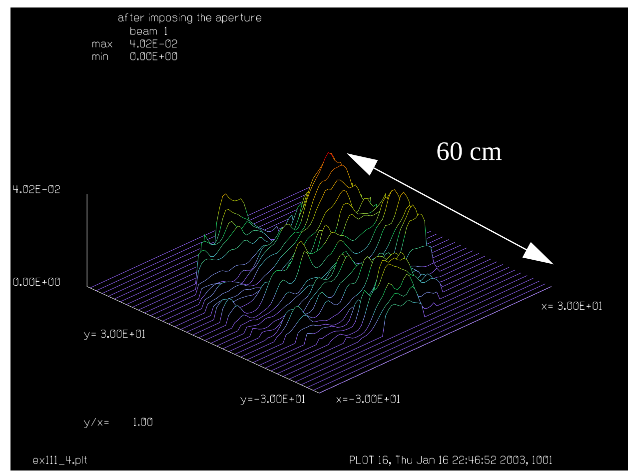
Starting gaussian beam



Beam at 90 km altitude, distorted by atmospheric aberration, creates back scatter.



The back scattered light covers the ground. We approximate this over a 800 x 800 cm section, overfilling the aperture.

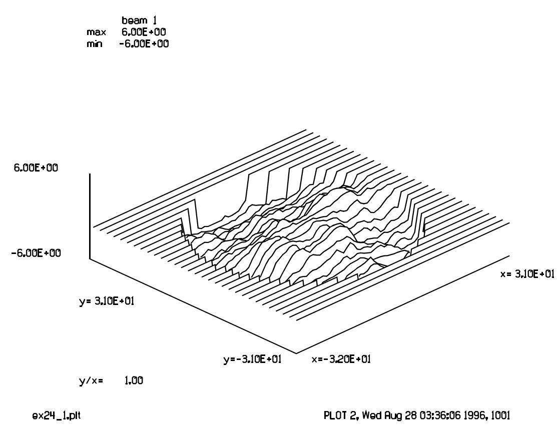


Light intercepted by 50 cm diameter receiving aperture.

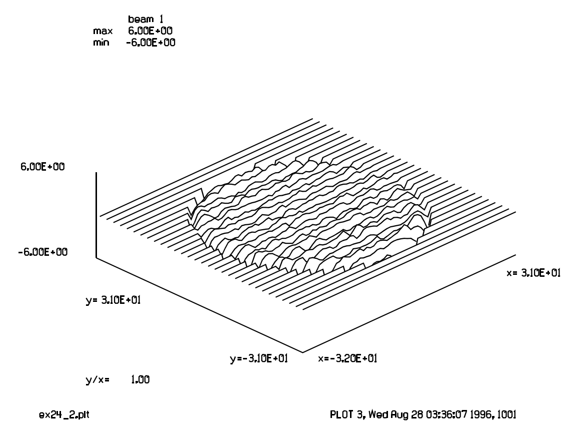
Correction of atmospheric aberration ex24a.inp

- for strong aberration and relatively long propagation distance
  - $L > z_{char}$
  - wavefront aberration will role into intensity modulation
  - the beam is described as speckled
  - adaptive mirrors only correct phase aberration — can not fully correct for specled beams
  - nonlinear phase conjugation such as stimulated Bruillion scattering (SBS) can conjugate speckled beams.

    (SBS may be explained by Talbot imaging)
  - Zonal adaptive optic model

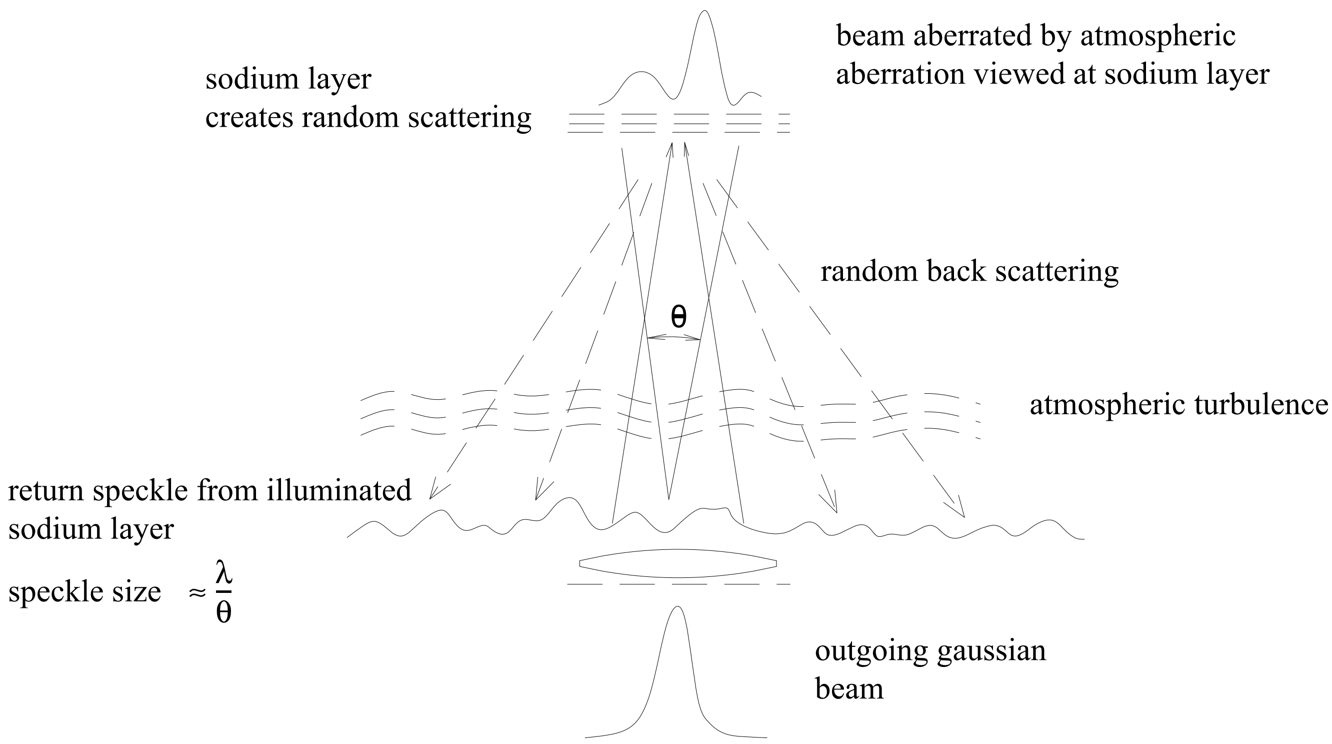


Wavefront before correction.



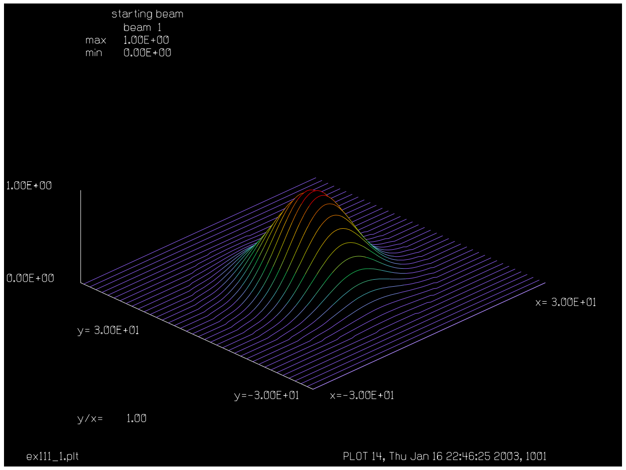
Wavefront after correction. Leaves high order residuals with actuator printthrough.

Astronomical guide star by scattering from sodium layer ex111.inp

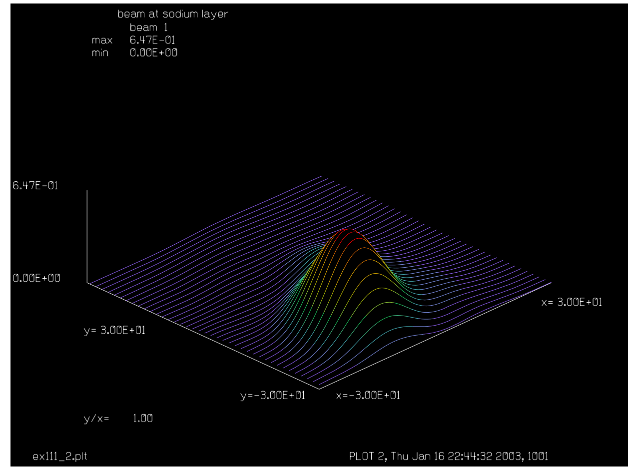


The outgoing beam is distorted by atmophere to create aberrated beam in upper atmosphere at sodium layer at 90 km altitude. Sodium layer is a random scattering source. The back scattered radiation creates a speckled pattern at the ground where the speckle size is $\approx \lambda/\theta$, where θ is the subtended angle of the illuminated region at the sodium layer.

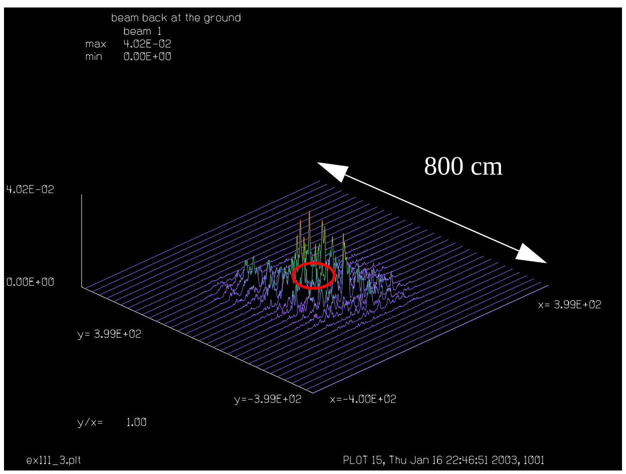
Propagation to sodium layer and backscatter to ground



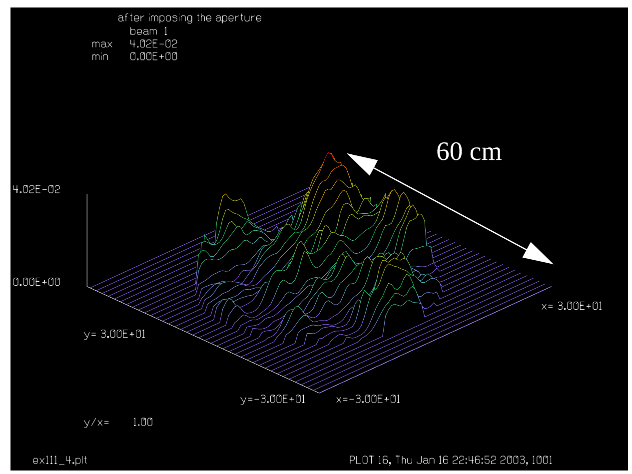
Starting gaussian beam



Beam at 90 km altitude, distorted by atmospheric aberration, creates back scatter.



The back scattered light covers the ground. We approximate this over a 800 x 800 cm section, overfilling the aperture.



Light intercepted by 50 cm diameter receiving aperture.

13. Some Examples from Technical Support

1) Tilt does not work!

I added tilt angle of Theta = 6.3 milliradians, but the far-field did not move. I doubled the tilt angle of Theta = 6.3*2 milliradians, but the far-field image still did not move.

I tried tripple the tilt Theta = 6.3*3 milliradians, but the far-field images just sits there.

What is wrong with the code? **tilt.inp**

2) Why does GLAD not make a gaussian beam in far-field?

I take a gaussian beam and bring the light to a focus in the far-field. Why do I not see a gaussian -- just see a spike. **gauss.inp**

3) No propagation occurs?

I tried to propagate beam 1 by 100 cm, but did not get any propagation. **noprop.inp**

4) Why does the calculation not stop at the proper point?

Why does this calculation not stop at the 99th pass? How can get this problem be fixed? **not_stop.inp**

5) Resonator does not converge.

Why does this resonator not converge? **bad_reson.inp**

14. Future Tasks

1) Enhancements for Ver. 5.6 and 5.7

- Coherent treatment of gain
  - More powerful than rate equation gain.
  - Short pulse capability.
  - Formation of longitudinal modes.
  - Proper treatment of Q-switch rise time.
- Extended cavity laser diode.
- 64 bit version of GLAD.
- Support for more than 2 CPU’s.

Index