A new FFT code: FOG
Fast Fourier Transform Optical Simulation of Gravitational Wave Interferometers

Richard DAY

European Gravitational Observatory

Why another FFT code ?

Already many proven FFT simulators used in the GW community for AdV Virgo

- DarkF : M. Pichot, written in Fortran
- SIS : H. Yamamoto , written in C++
- OSCAR: J. Degallaix, written in Matlab
- SIESTA: LMA, written in C

When spending millions to build a new optical design, you can never have too many independent simulators

- Cross-checking builds confidence in the simulation results

Decision to develop another code based on two main objectives:

- Total flexibility of optical configurations for the end user
- High speed simulations to accelerate understanding of physical system

No one FFT simulator existing on the “market” fulfills all these objectives

Finesse is a modal simulator which fits the bill → this is why it is so popular

However, the use of MSRC configuration in AdV Virgo makes it absolutely necessary to use FFT simulators for many design tasks

A brief history of FOG

FOG is the product of two and a half years of labor

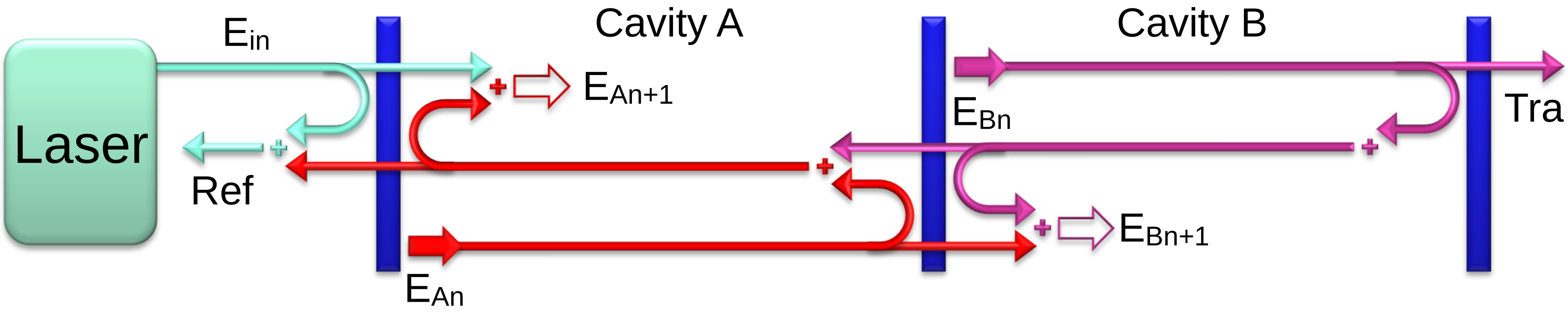
Nov 2009	Simulation of Virgo+ Input Mode Cleaner to understand losses (VIR-0398A-10) ➢ Written in Matlab using Oscar 1.0 source code and DarkF locking scheme
Feb 2011	Simulation of Virgo+ arm cavities: find CHRoCC working point (VIR-0059A-11) Beginning of intense design activity for MSRC for AdV Virgo Simulation of CITF for small angle wedges in MSRC
Sep 2011	Implementation of accelerated convergence used in DarkF to simulate full ITF
Oct 2011	Implementation of modular architecture to simulate “thick” wedged beamsplitter
Nov 2011	Rectangular grids and 1D estimator (J. Degallaix) to simulate large wedge angles
Mar 2012	Global acceleration for simulation of high finesse coupled cavities
Apr 2012	Adoption of MIST (‘Modal Interferometer Simulation Tool’ by Gabriele Vajente) user interface and Matlab program architecture.
Jun 2012	General release of FOG beta version

Thanks to developers of DarkF, Oscar and MIST for making all this possible !

FOG: a modular code

All FFT simulators work in much the same way

Take example of double cavity



- Laser supplies field E_{in}. Each cavity has one guess field E_{An} and E_{Bn}
- Carry out one iteration of propagations, transmissions and reflections.
- Determine guess fields E_{An+1} and E_{Bn+1} for next iteration
- Continue iterations until nth guess field and (n+1)th field are “identical”

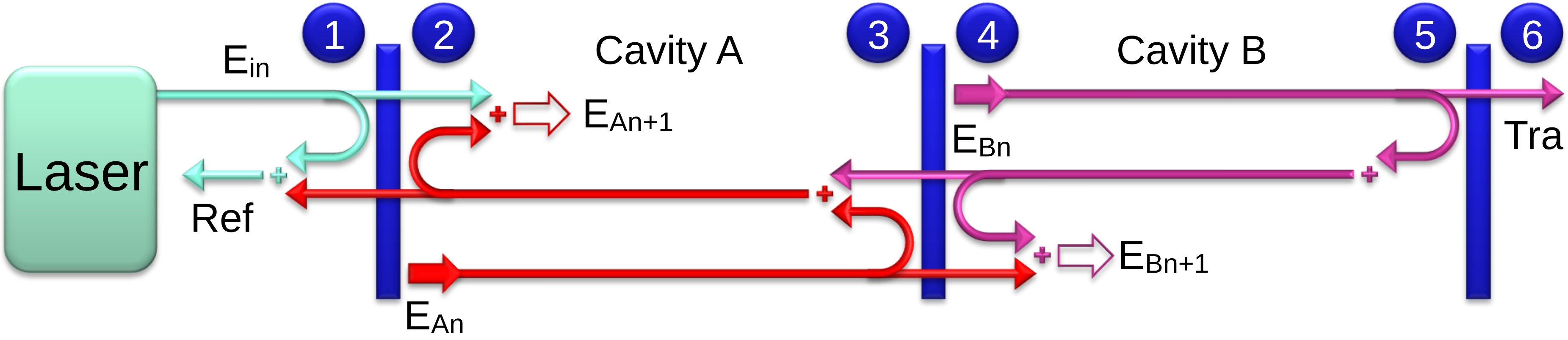
In all existing FFT codes these iterations are hard-coded

As optical configurations become more complicated coding becomes more difficult, potentially less optimized and more prone to bugs

FOG: a modular code

The modular approach is simple

Label each side of the mirror with a node number



Find how nodes connected to each other with propagation, transmission and reflection look-up-tables

0 means not applicable

Propagation = [0, 3, 2, 5, 4, 0]   Transmission= [2, 1, 4, 3, 6, 5]   Reflection= [1, 2, 3, 4, 5, 6]

Field at node 2 propagated to node 3 where it will transmit to node 4 and reflect to node 3

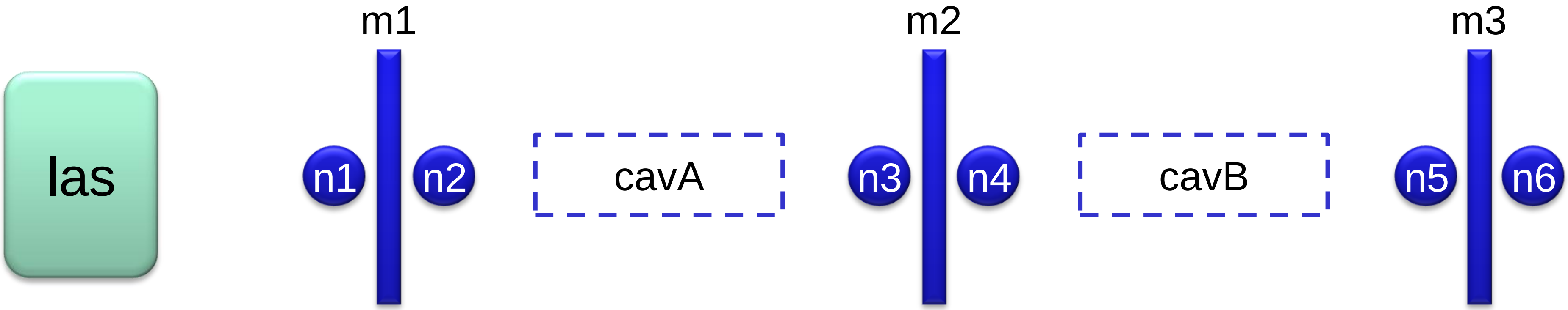
From look-up-tables algorithm can determine nodes having guess fields

Also determines correct sequence of operations to reproduce hard-coded equivalent

Arbitrarily complicated configurations possible with optimal efficiency & zero bugs

FOG: a modular code

Information entered by user using a MIST configuration file

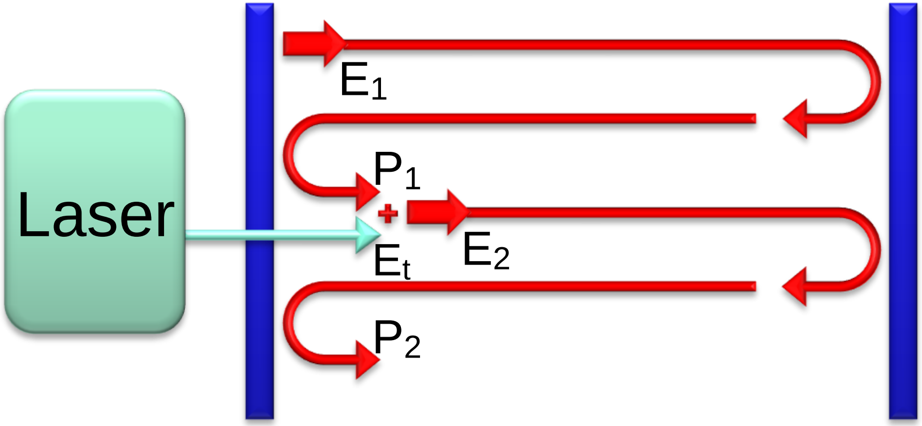


keywords	Element names	Node connections		Parameters	
laser	las	n1		P=1	
beamparam		n1		R=1431	W=0.049
mirror	m1	n1	n2	R=0.95	Rc=-1431
space	cavA	n2	n3	L=12	
mirror	m2	n3	n4	T=0.014	Rc=-1420
space	cavB	n4	n5	L=3000	
mirror	m3	n5	n6	T=1e-6	Rc=1683

FOG: an accelerated code

High finesse cavities in AdV interferometers → need accelerated convergence

Technique for single cavity described by P. Saha (Vol. 14, No. 9, P. 2195 J. Opt. Soc. Am. A)



$$D_1 = E_1 - P_1 \qquad D_2 = E_2 - P_2$$

$$\underbrace{\begin{bmatrix} \sum D_1^* D_1 & \sum D_1^* D_2 \\ \sum D_2^* D_1 & \sum D_2^* D_2 \end{bmatrix}}_{M} \underbrace{\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}}_{C} = \underbrace{\begin{pmatrix} \sum D_1^* E_t \\ \sum D_2^* E_t \end{pmatrix}}_{A}$$

$$E_1' = c_1 E_1 + c_2 E_2$$

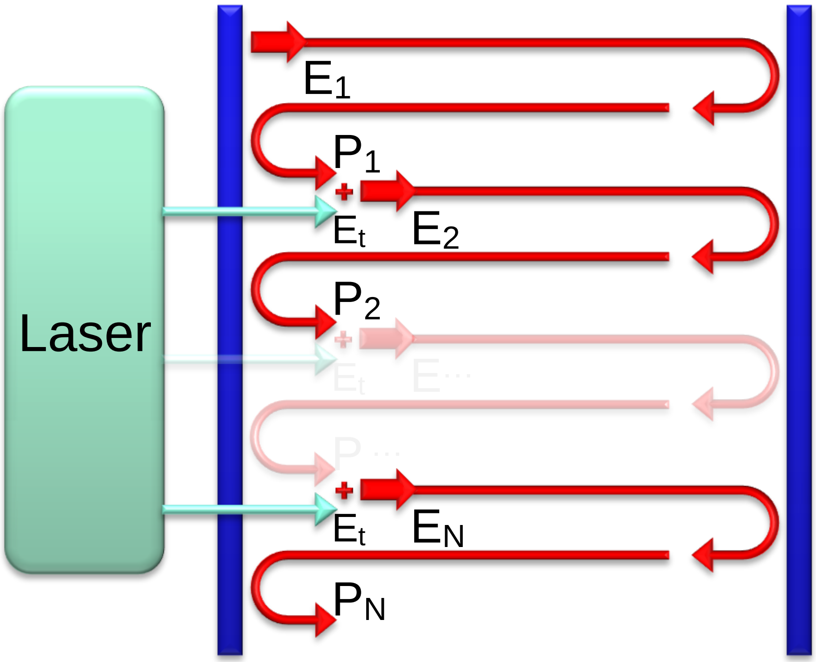
- Propagate round-trip with initial guess field E_1 field which gives P_1
- Construct second guess field E_2 and do second round-trip which gives P_2
- Construct matrices M & A and inverse problem to find coefficients → $C = M^{-1}A$
- Use coefficients and guess fields to determine E_1' best estimate of steady state field

This technique is used in SIS and DarkF with acceleration of orders of magnitude

FOG: an accelerated code

Smoothed Acceleration

Extension of technique is “smoothed acceleration” → N round-trips instead of 2



$$\boxed{D_1 = E_1 - P_1} \; , \; \boxed{D_2 = E_2 - P_2} \; \cdots \; \boxed{D_N = E_N - P_N}$$

$$\underbrace{\begin{bmatrix} \sum D_1^* D_1 & \sum D_1^* D_2 & \cdots & \sum D_1^* D_N \\ \sum D_2^* D_1 & \sum D_2^* D_2 & \cdots & \sum D_2^* D_N \\ \vdots & \vdots & \ddots & \vdots \\ \sum D_N^* D_1 & \sum D_N^* D_2 & \cdots & \sum D_N^* D_N \end{bmatrix}}_{M} \underbrace{\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix}}_{C} = \underbrace{\begin{pmatrix} \sum D_1^* E_t \\ \sum D_2^* E_t \\ \vdots \\ \sum D_N^* E_t \end{pmatrix}}_{A}$$

$$E_1' = c_1 E_1 + c_2 E_2 + \cdots + c_N E_N$$

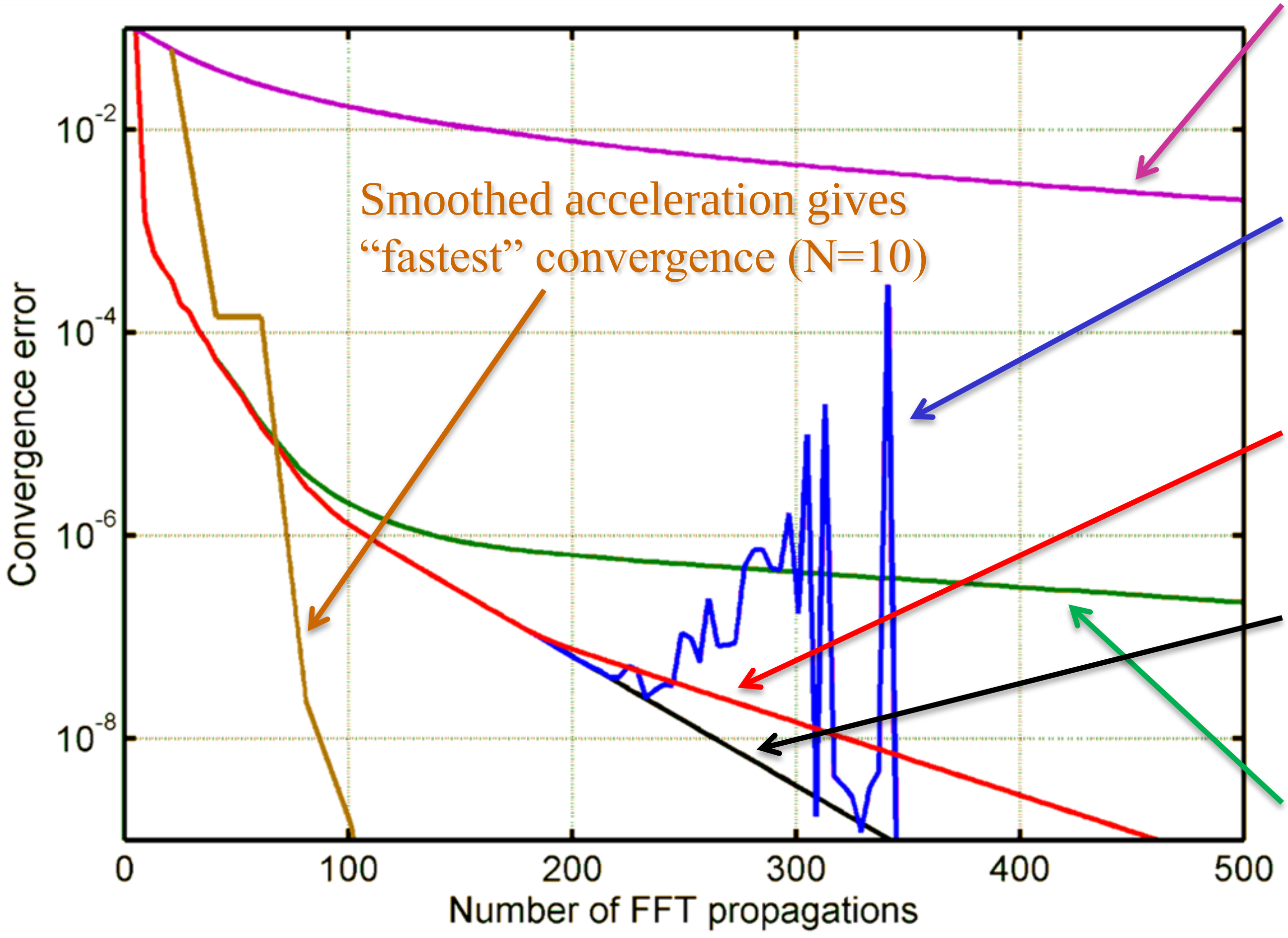
Better estimate of steady state field because it uses more “data points”

Reduces total number of FFT propagations by up to a factor 3

Increases stability of accelerated convergence

FOG: an accelerated code

Accelerated convergence is fast but not always stable



Smoothed acceleration gives “fastest” convergence (N=10)

Non-accelerated convergence very long in high finesse cavity

Matrix inversion can become unstable due to machine precision

Can use Singular Value Decomposition to remove singularities

Even better to use quad precision for matrix inversion

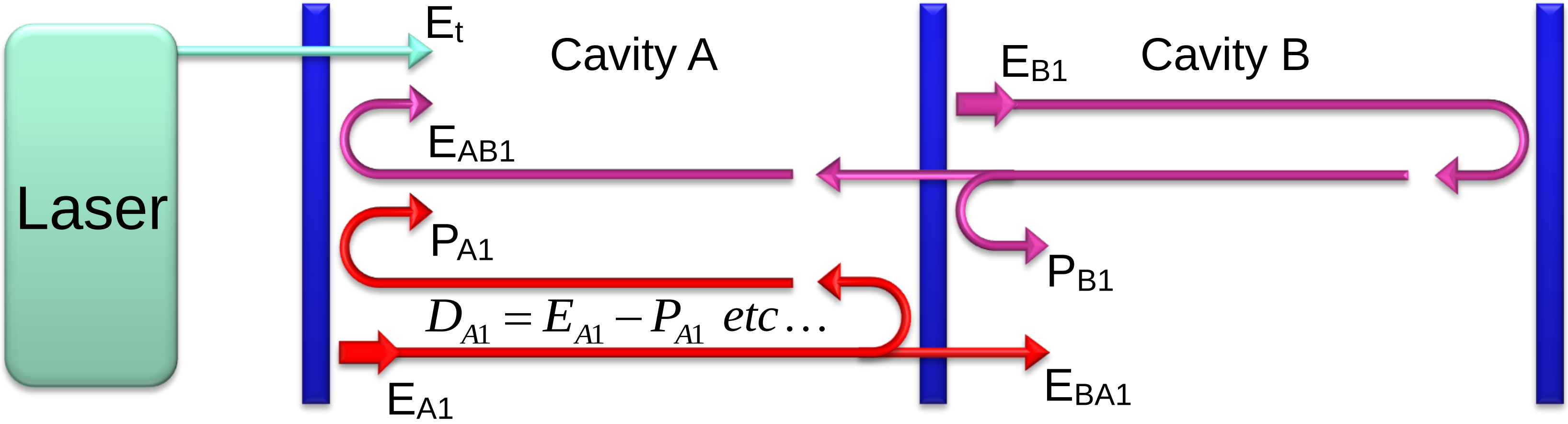
If light artificially reflects of FFT window border convergence is ruined

We still have a lot to learn about accelerated convergence

FOG: an accelerated code

Global Acceleration

To accelerate coupled cavities need to remember which cavity each field comes from

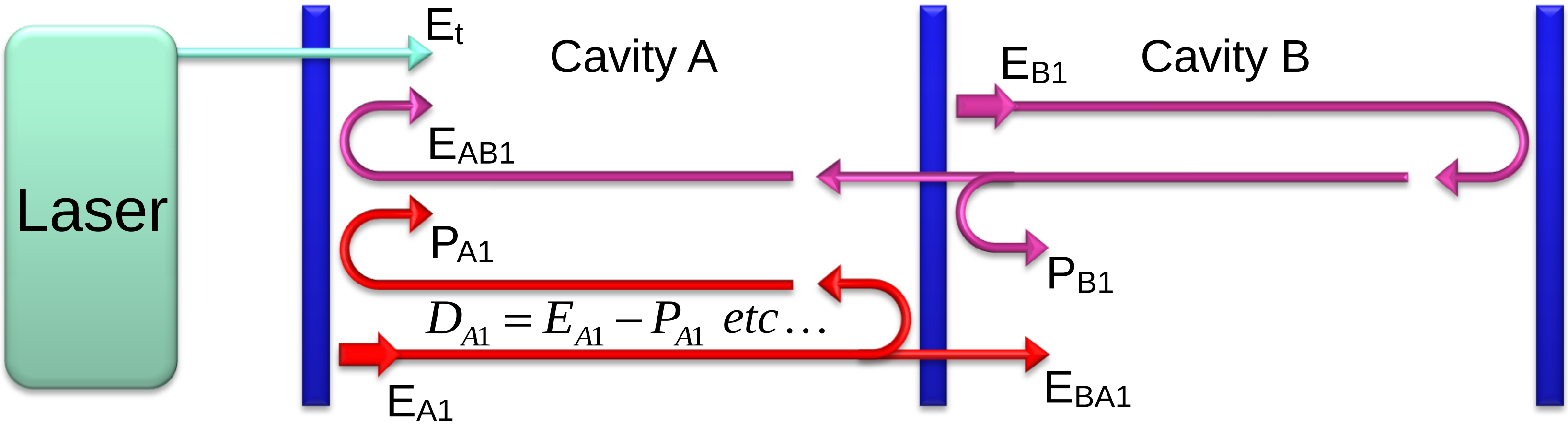


$D_{A1} = E_{A1} - P_{A1}$ etc…

FOG: an accelerated code

Global Acceleration

To accelerate coupled cavities need to remember which cavity each field comes from



$$D_{A1} = E_{A1} - P_{A1} \;\; etc\ldots$$

$$\begin{bmatrix} \sum D^*_{A1}D_{A1} - \sum E^*_{BA1}E_{BA1} & \sum D^*_{A1}D_{A2} - \sum E^*_{BA1}E_{BA2} & -\sum D^*_{A1}E_{AB1} + \sum E^*_{BA1}D_{B1} & -\sum D^*_{A1}E_{AB2} + \sum E^*_{BA1}D_{B2} \\ \sum D^*_{A2}D_{A1} - \sum E^*_{BA2}E_{BA1} & \sum D^*_{A2}D_{A2} - \sum E^*_{BA2}E_{BA2} & -\sum D^*_{A2}E_{AB1} + \sum E^*_{BA2}D_{B1} & -\sum D^*_{A2}E_{AB2} + \sum E^*_{BA2}D_{B2} \\ \sum E^*_{AB1}D_{A1} - \sum D^*_{B1}E_{BA1} & \sum E^*_{AB1}D_{A2} - \sum D^*_{B1}E_{BA2} & -\sum E^*_{AB1}E_{AB1} + \sum D^*_{B1}D_{B1} & -\sum E^*_{AB1}E_{AB2} + \sum D^*_{B1}D_{B2} \\ \sum E^*_{AB2}D_{A1} - \sum D^*_{B2}E_{BA1} & \sum D^*_{AB2}D_{A2} - \sum D^*_{B2}E_{BA2} & -\sum D^*_{AB2}E_{AB1} + \sum D^*_{B2}D_{B1} & -\sum D^*_{AB2}E_{AB2} + \sum D^*_{B2}D_{B2} \end{bmatrix} \begin{pmatrix} c_{A1} \\ c_{A2} \\ c_{B2} \\ c_{B2} \end{pmatrix} = \begin{pmatrix} \sum D^*_{A1}E_t \\ \sum D^*_{A2}E_t \\ \sum D^*_{B2}E_t \\ \sum D^*_{B2}E_t \end{pmatrix}$$

Ouch! Looks complicated, but simple pattern → can be applied to any configuration

Find new guess fields with coefficients as before $\longrightarrow$

$$E'_{A1} = c_{A1}E_{A1} + c_{A2}E_{A2}$$
$$E'_{B1} = c_{B1}E_{B1} + c_{B2}E_{B2}$$

Global acceleration important for coupled cavities both having high finesse

FOG: a rectangular code

Rectangular grid

“Rectangular” refers to spatial sampling frequency and not physical dimensions

Many simulations involve simulating tilted surfaces
➢ requires high spatial sampling frequency

Rectangular grid allows to increase sampling frequency just in direction of tilt

Memory usage and computational time increases proportionally with tilt and not proportionally with the square of the tilt

Quantitative results remain accurate using this method

1-D Estimator (first developed by J. Degallaix)

Extreme case of rectangular grid is a grid with 1xN points → very fast !

Gives qualitative results of any configuration & compatible with acceleration

Ideal for exploring new configurations before launching 2D simulations

Very useful tool and quantitative results are surprisingly accurate !

FOG: part of a simulation family

Modal Interferometer Simulation Tool (MIST) (G. Vajente)

MIST is an optical simulation tool based on Hermite-Gauss modal decomposition of fields

- ➢ Integrated in the MATLAB environment

Simulates arbitrary user-defined optical systems

- ➢ User defines configuration file with description of the optical system

```
# define some constants
const fmod 6272659.0
const midx 0.3
const c 299792458

# a constant declaration can use any valid MATLAB
# expressions, involving all the constants defined
# before it in the configuration
const LPRC c/(4*fmod)
const Larm 3000

const Ri 0.9
const Te 10e-6
const Rci 1420
const Rce 1683

# input laser and modulation
laser l nL P=1
modulator eom1 nL nM1 f=fmod m=midx type=pm

# one useless space
space s1 nM1 nI1 L=LPRC

# 3-km long cavity
mirror mI nI1 nI2 R=Ri Rc=-Rci
space cav nI2 nE1 L=Larm
mirror mE nE1 nE2 T=Te Rc=Rce

# this command tells MIST to use the mode of the
# cavity to define the gaussian beam parameter
cavity cav

# add some probes
probe Tra nE2 from=mE
probe Ref nI1 from=mI
probe Cav nI2 from=mI

power B7_DC nE2 from=mE
demod B7_AC nE2 from=mE f=fmod
```

Symbolic manipulation of field equations to provide fast handling of large number of modes

- ➢ Computations performed inside MATLAB environment

  → Benefit from all Matlab functionalities

Tool still in beta version. Release available soon with basic functionalities

- ➢ Computation of static fields, demodulated signals, lock
- ➢ Next functionalities: transfer functions, radiation pressure

Contact the author for more information (gabriele.vajente@pi.infn.it)

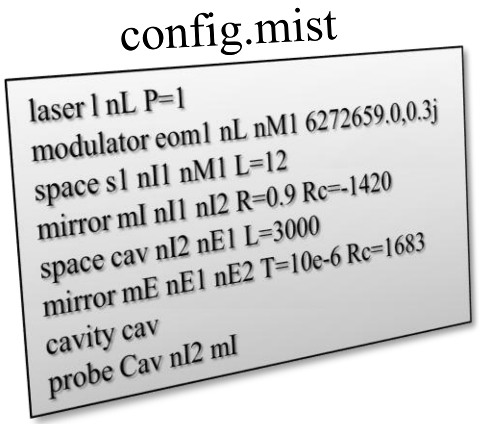
FOG: part of a simulation family

Modal and FFT simulators are complimentary

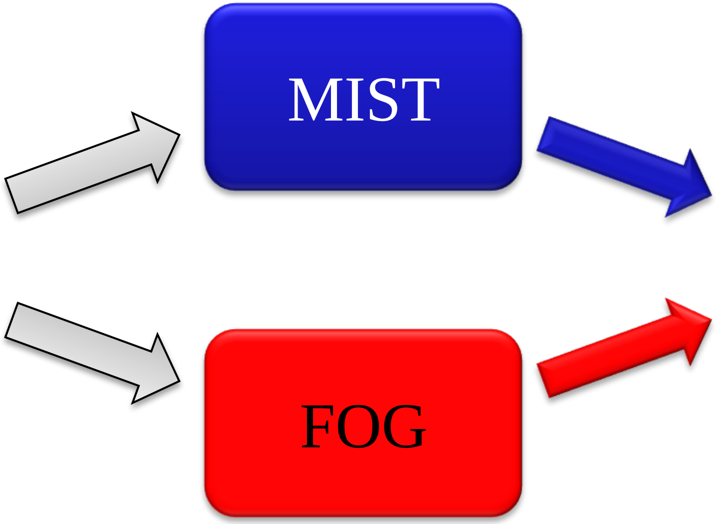
The end user needs to use both on a given configuration to get required results

FOG user interface and Matlab program architecture based on MIST

config.mist

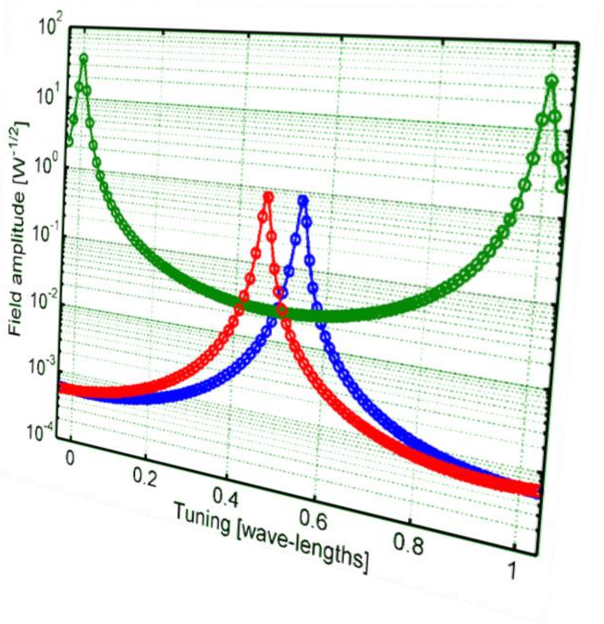


```
laser l nL P=1
modulator eom1 nL nM1 6272659.0,0.3j
space s1 nI1 nM1 L=12
mirror mI nI1 nI2 R=0.9 Rc=-1420
space cav nI2 nE1 L=3000
mirror mE nE1 nE2 T=10e-6 Rc=1683
cavity cav
probe Cav nI2 mI
```



MIST

FOG



MIST and FOG not only share same config file but also same philosophy

User can seamlessly pass between MIST and FOG getting the best out of Modal & FFT

FOG: designing AdV Virgo

FOG used in five different subsystems of AdV Virgo during design phase

Optical Simulation & Design

Simulate power recycled & dual recycled ITF with “real” mirror maps

- For 128x128 grid convergence in 13 seconds (3100 FFT propagations)

RoC requirements of arm cavity end mirror

Polishing requirements of recycling cavity mirrors

Tilted AR surfaces in recycling cavity (compensation plates & beamsplitter)

Injection

Polishing requirements of input mode cleaner mirrors

Study of coupled parasite cavity created by backscatter

Stray Light Control

Damage threshold requirements on arm cavity baffles

Recoupling of scattered light from baffles

Thermal Compensation System

Measure correcting performance of TCS actuation in recycling cavity

Develop strategy for using “Phase camera” as TCS sensor

Detection

Higher order mode filtering requirement of output mode cleaner

See “spare” slides for examples

FOG: release

First General Release of FOG → June 2012

Barebones Beta version to iron out bugs (if any)

MIST user interface and modular architecture

Basic optical elements → laser, eom, beamsplitter, mirror etc…

Rectangular grid & 1-D estimator

Developed and to be implemented in future releases

Smoothed global acceleration

Thick optical elements → compensation plates, wedged beam splitters

Backscattering beamsplitters (or tilted mirrors)

Adaptable FFT window size → to simulate efficiently stable cavities

To be developed

Perturbation of elements → audio sidebands

Transfer functions

Radiation pressure

Conclusion

Development of new FFT simulator: FOG

Fast Fourier Transform **O**ptical Simulation of **G**ravitational Wave Interferometers

FOG has a number of features setting it apart from other simulators:

- Modular structure end user can create any optical configuration
- Optimized accelerated convergence
  - ➢ Full dual recycled ITF simulated in seconds
- Rectangular grids for simulating tilted surfaces
- 1-D estimator for fast understanding of physical systems
- Full user interface compatibility with MIST modal simulator

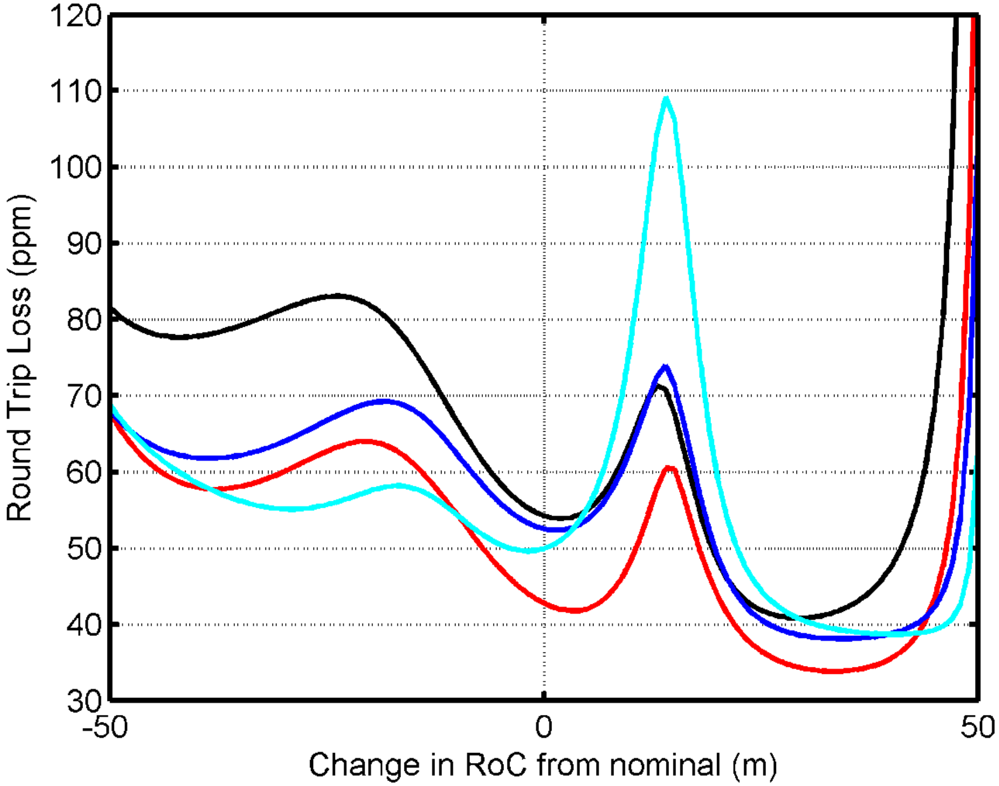
FOG used intensively during design phase of AdV Virgo

FOG designed to be easy, quick and flexible

➢ Hopefully will also become an important tool outside of the GW community

Spare: designing AdV Virgo with FOG

RoC requirements of arm cavity end mirror

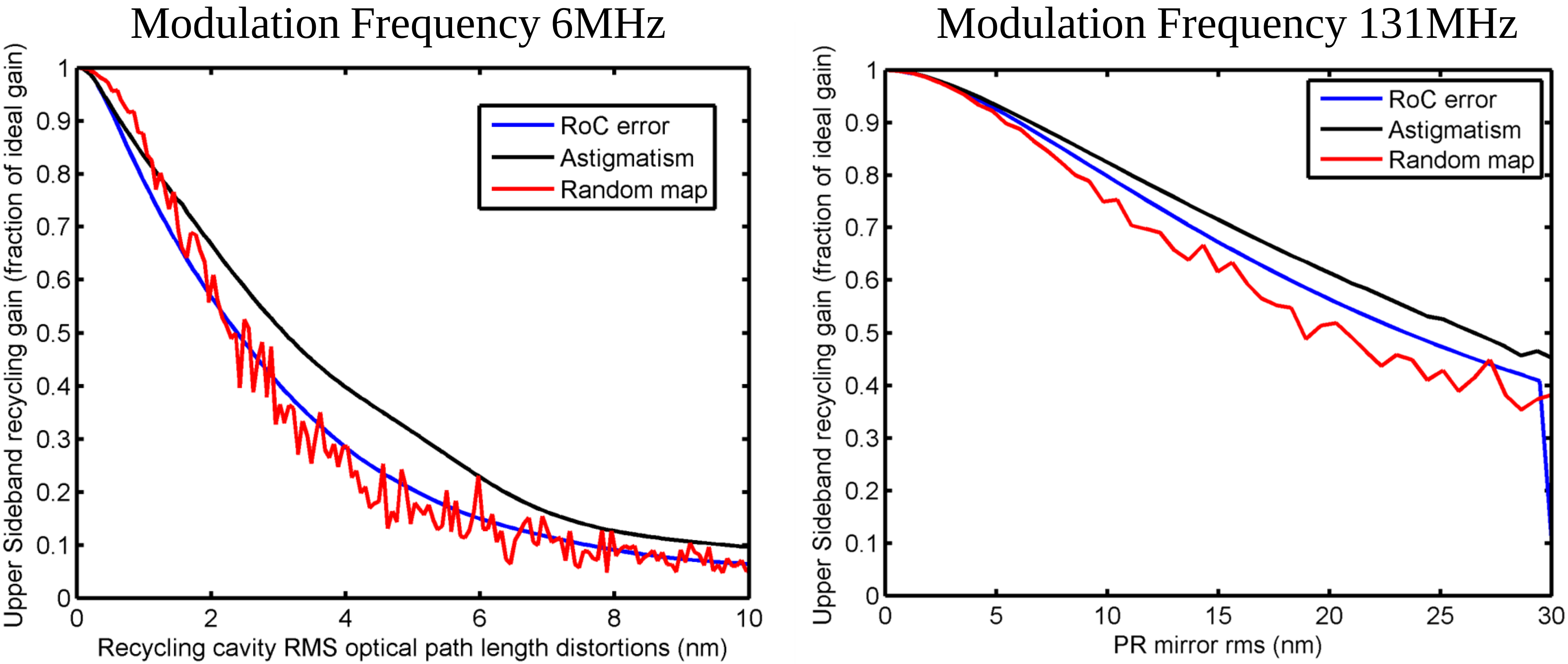


Look at RTL of arm cavity with “real” AdV mirror maps

Scan end mirror RoC to see effect of HOM resonances on RTL

Spare: designing AdV Virgo with FOG

Polishing requirements of recycling cavity mirrors



Modulation Frequency 6MHz

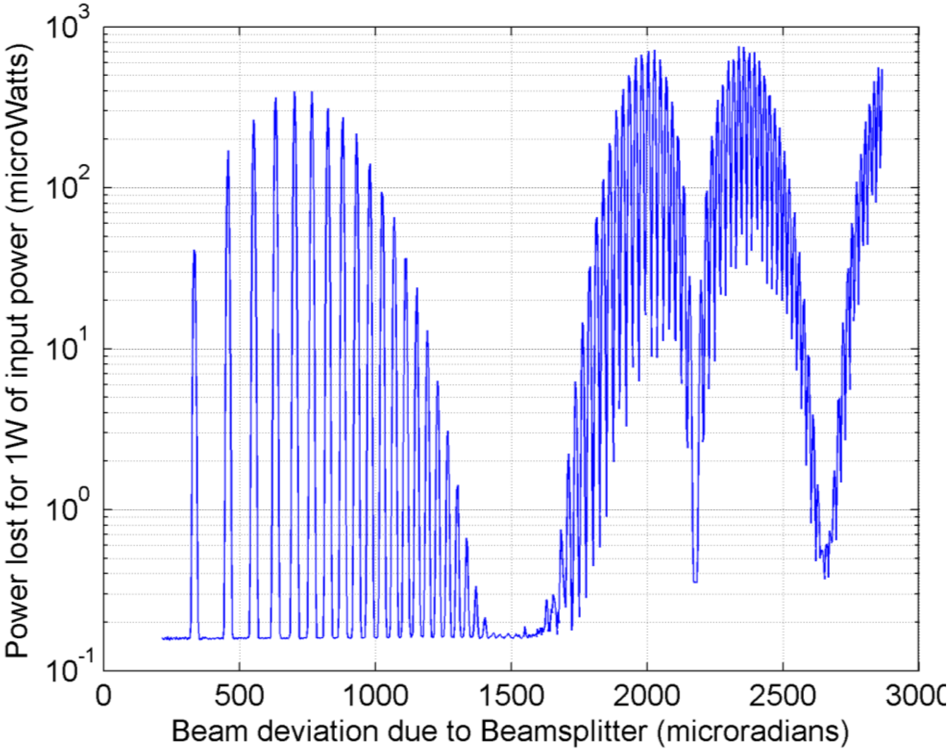
Modulation Frequency 131MHz

Study of recycling gain as a function of aberrations for two modulation frequencies

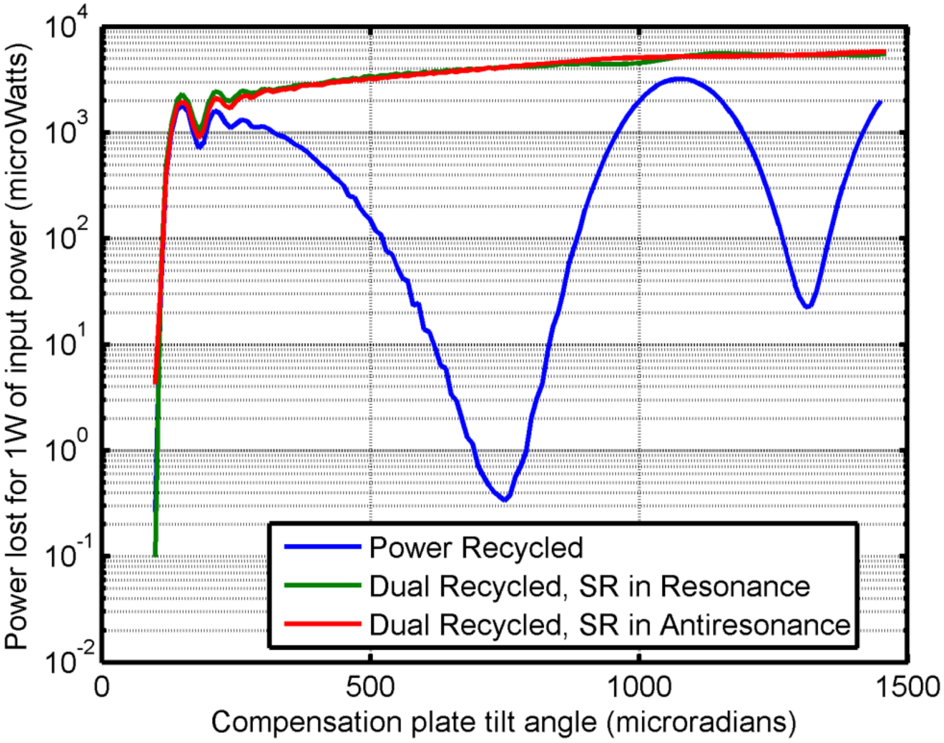
Spare: designing AdV Virgo with FOG

Tilted AR surfaces in recycling cavity

Wedged beamsplitter in power recycled CITF



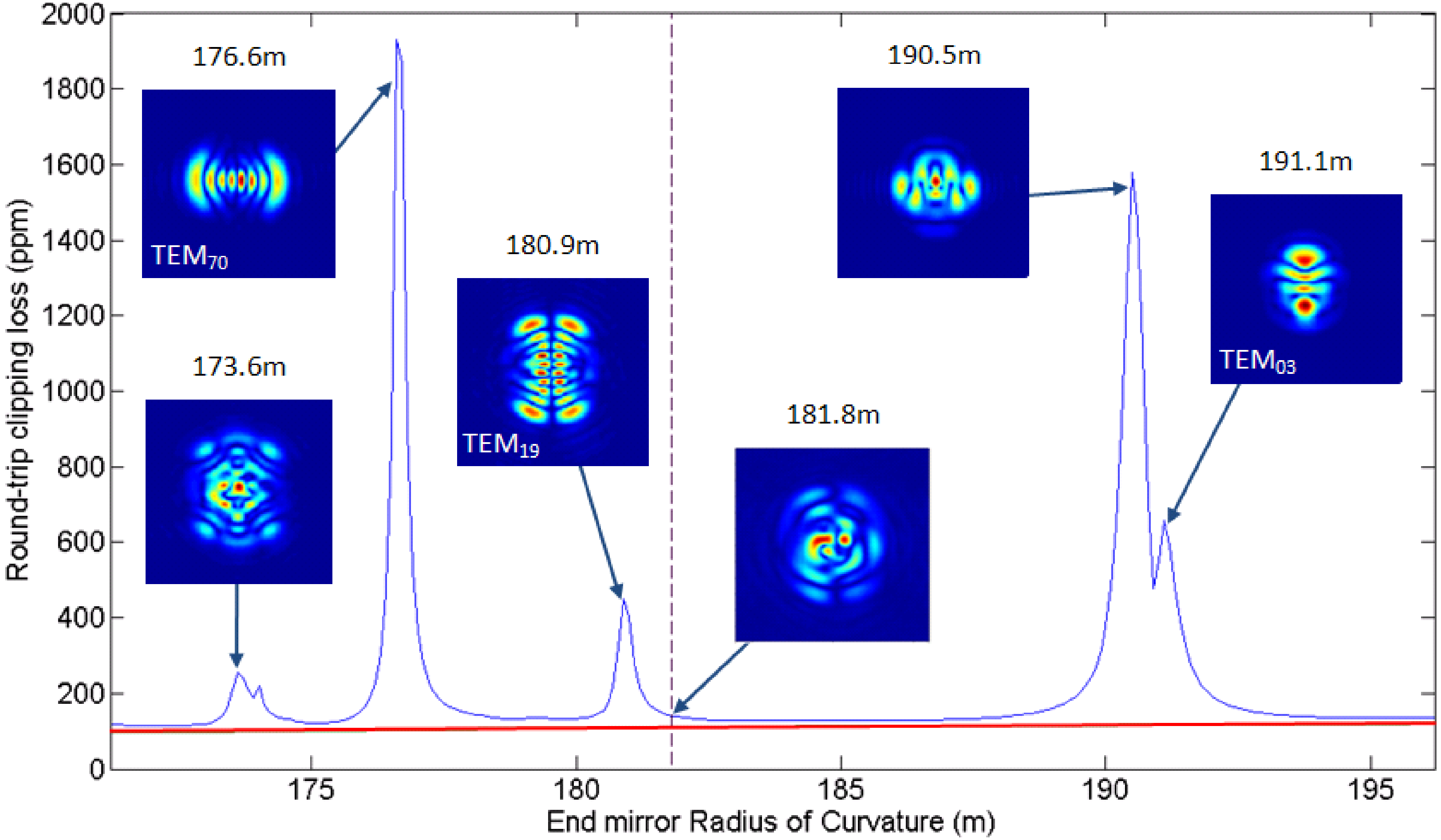
Compensation plate in CITF



Study clipping losses of fields generated by AR surfaces of beamsplitter and compensation as a function of tilt angle

Spare: designing AdV Virgo with FOG

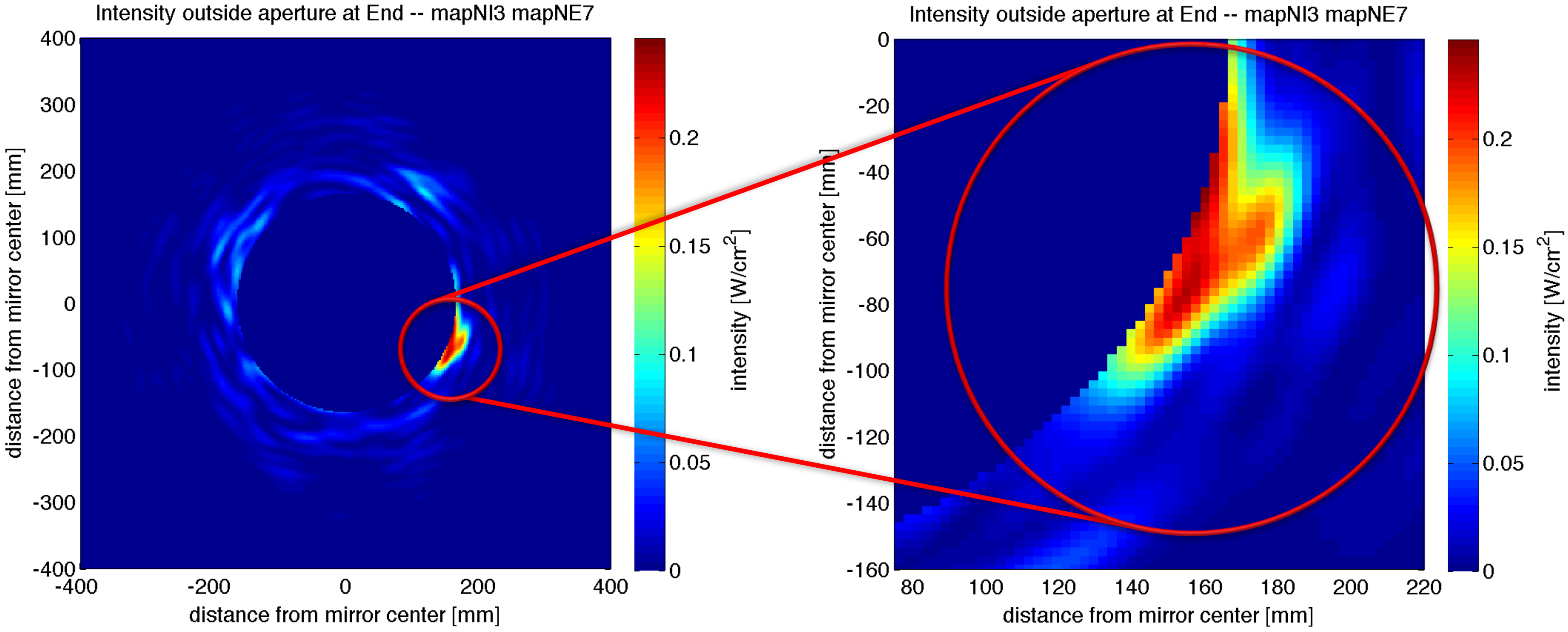
Polishing requirements of input mode cleaner mirrors



Look at RTL of IMC with “real” mirrors while scanning End mirror RoC

Spare: designing AdV Virgo with FOG

Damage threshold requirements on arm cavity baffles

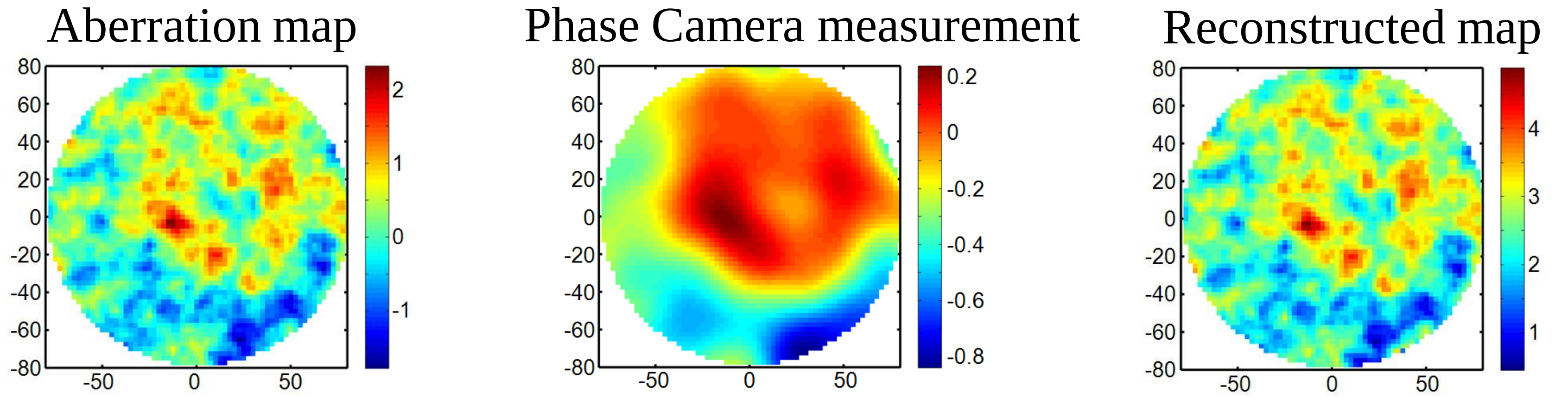


Study power density of scattered light on baffles around mirrors in arm cavity to make right choice of baffle material

Spare: designing AdV Virgo with FOG

Develop strategy for using “Phase camera” as TCS sensor

Common Aberrations → Measure sideband phase (relative to carrier) in RC

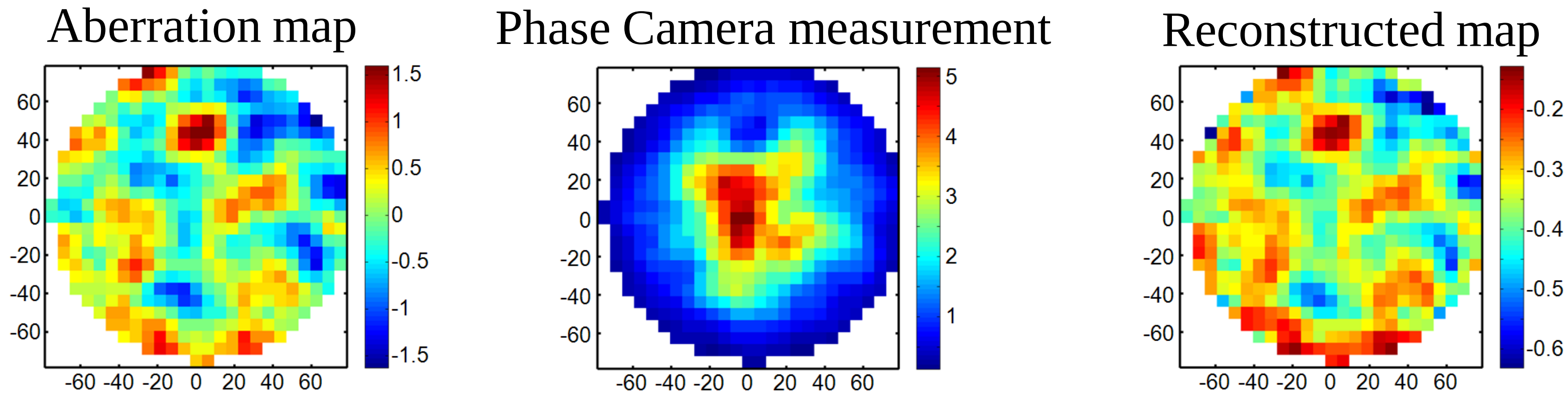


Aberration map

Phase Camera measurement

Reconstructed map

Differential Aberrations → Measure sideband amplitude on Dark Fringe



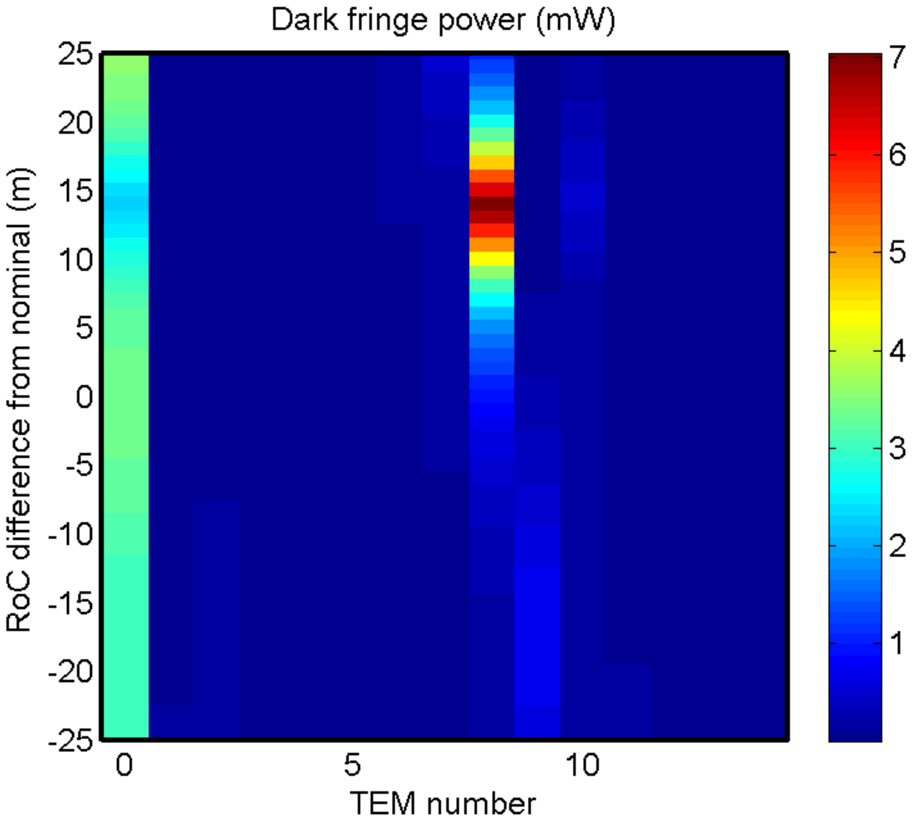
Aberration map

Phase Camera measurement

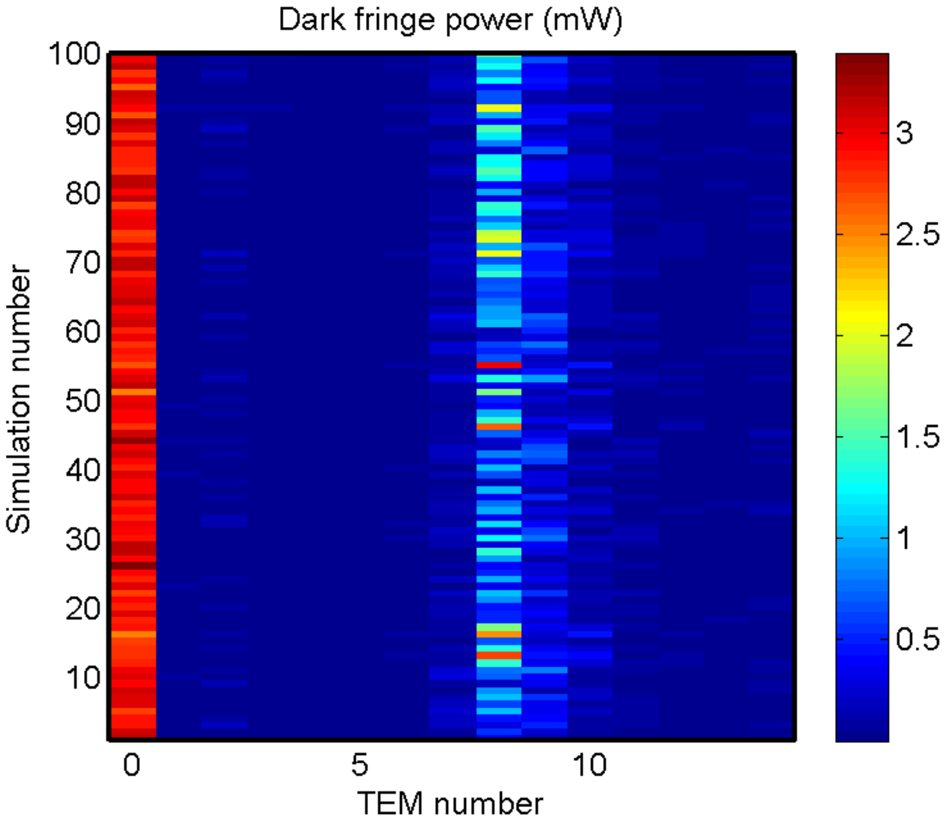
Reconstructed map

Spare: designing AdV Virgo with FOG

Higher order mode filtering requirement of output mode cleaner



Study HOM’s on dark fringe of ITF when changing RoC of end mirror with AdV Virgo maps



Statistical analysis of HOM’s on dark fringe of ITF with different random mirror maps