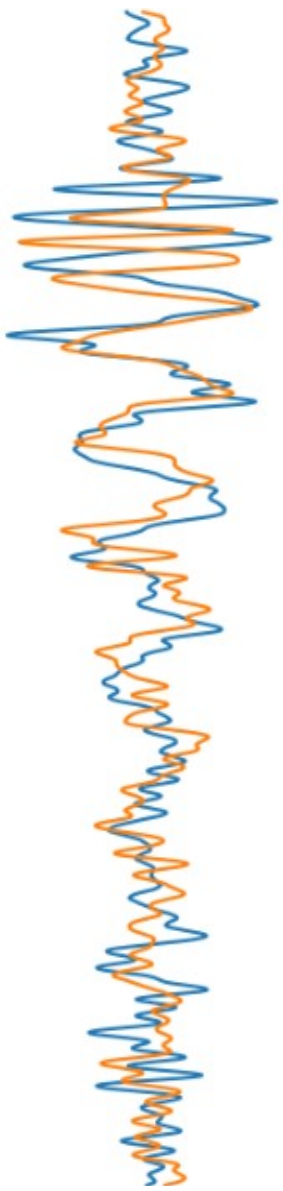
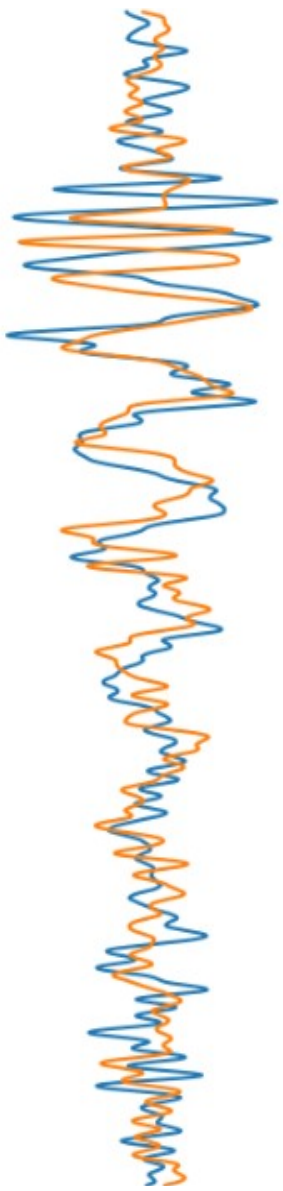


Introduction to FINESSE

Frequency domain interferometer modelling software

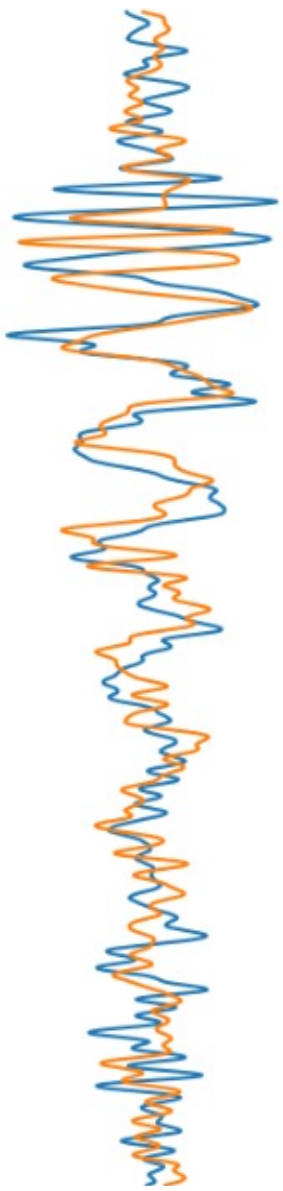
Dr Sean Leavey
Max Planck Institute for Gravitational Physics (Albert Einstein Institute)
2021-12-08

Max-Planck-Institut
für Gravitationsphysik
ALBERT-EINSTEIN-INSTITUT



The many types of simulation tool in the community

- FFT
  - OSCAR (J. Degallaix)
  - SIS (H. Yamamoto)
  - DarkF (M. Pichot)
- Gaussian optics / ray tracing
  - IFOcad (G. Heinzel et al.)
  - JamMT (A. Thüring / N. Lastzka)
- Frequency domain
  - FINESSE (A. Freise / D. Brown)
  - Optickle (M. Evans)
  - Phasor (L. McCuller)
  - MIST (G. Vajente)
- Time domain
  - E2E (H. Yamamoto)

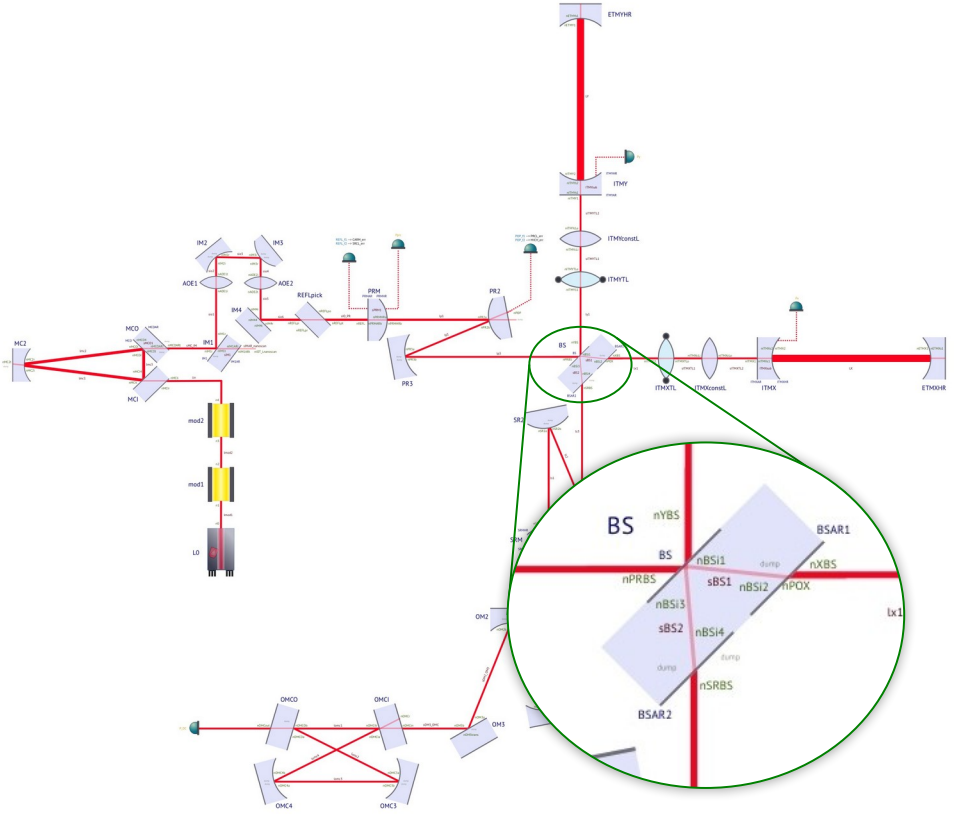


FINESSE

- Simulates in the **frequency domain**:
  - Steady state operation of interferometers
    - Power build-up
    - Error signals
    - Noise couplings
  - **Not** lock acquisition, nonlinear behaviour
- Simulates using the **modal** formulation of light:
  - Exact higher order modes (c.f. approximation via FFT grid)
  - But also supports modelling of apertures and mirror maps – by approximating them as a series of HOMs
- Originally developed at GEO600

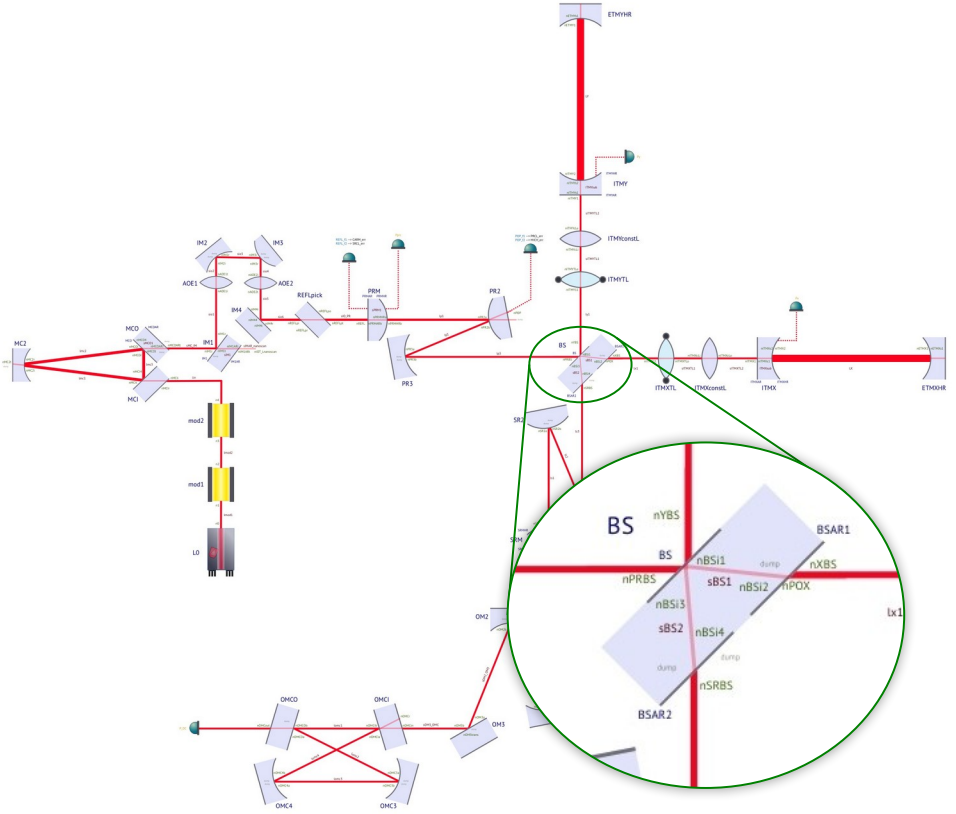
Frequency domain interferometer modelling

- Gives steady-state optical field everywhere in the interferometer for a given excitation
- Can use an admittance matrix to describe optical components and their couplings
- End up with series of linear equations
- FINESSE solves these numerically and plots the results



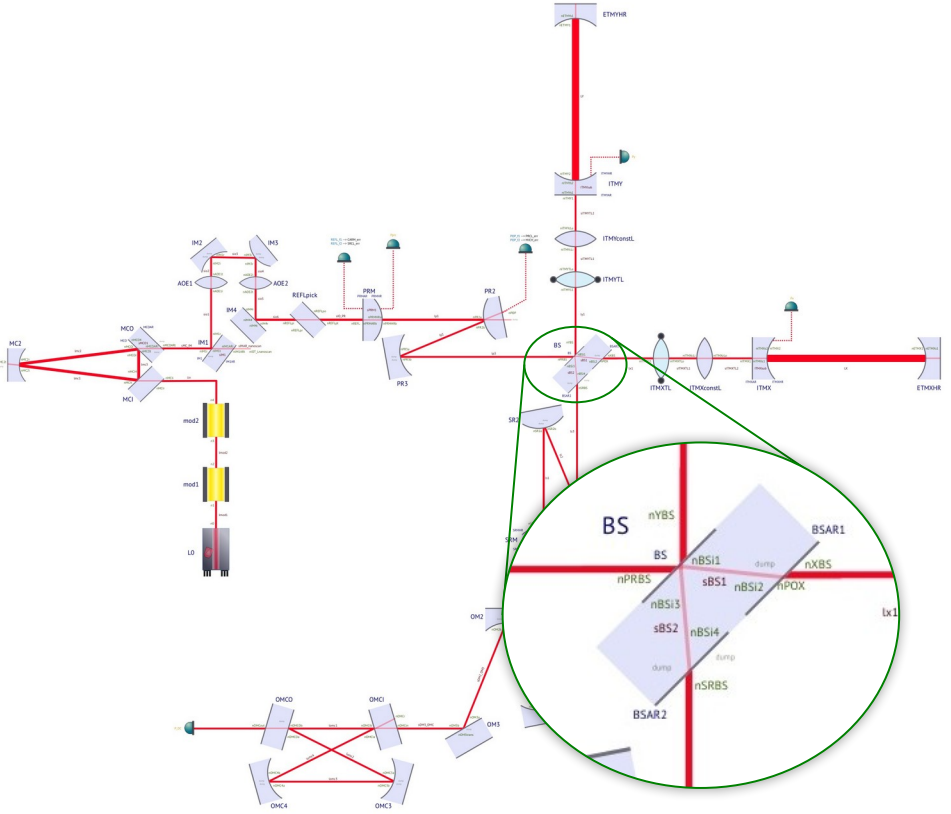
What FINESSE calculates for you

- Noise couplings for sensitivity calculations
- Error signals for design of control schemes (including modulation / demodulation)
- Control loop effects (“locks”)
- Optical transfer functions
- Higher order mode couplings (thermal distortion, misalignments, surface maps, etc.)



What FINESSE **doesn't** calculate for you

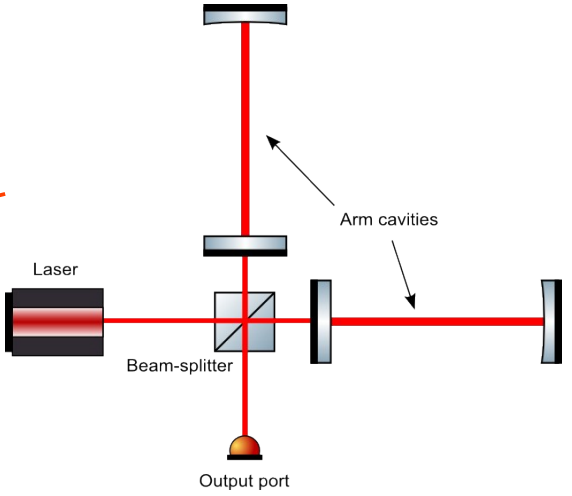
- Not so good for:
  - Scattered light simulations (use IFOcad or Zemax?)
  - Non-linear optics, non-static and non-quasi-static setups (use E2E?)
  - High mode orders (use an FFT tool)

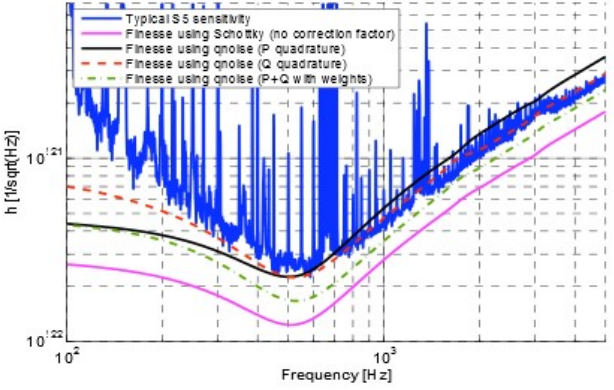


How do you use FINESSE?

```
%% INTERFEROMETER COMPONENTS

l L0 1 0 n1
s s0  1 n1 nbsp1
bs BSP 0.01 0.99 0 45 nbsp1 dump nbsp3 dump

s s01 1 nbsp3 n2

bs BS0 0.5 0.5 59.6 45 n2 n3 n4 n5     # Beam Splitter

const T_ITM 7e-3 # 7000ppm transmission from ET book
const T_ETM 0E-6 # 6ppm transmission from ET book

s sNin  1           n3 n6
m1 IMN $T_ITM 0 0   n6 n7
s sNarm 10000       n7 n8
m1 EMN $T_ETM 0 180 n8 dump

s sWin  1           n4 n9
m1 IMW $T_ITM 0 0   n9 n10
s sWarm 10000       n10 n11
m1 EMW $T_ETM 0 180 n11 dump
```

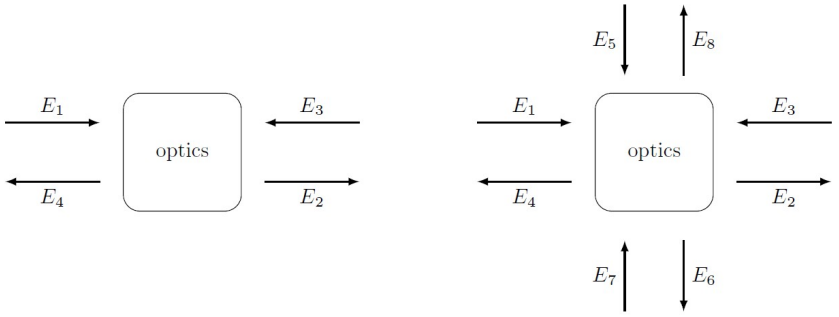
This is "KatScript" syntax



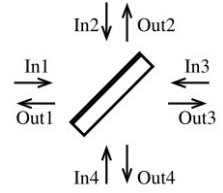


How does it work?

- FINESSE creates a nodal network of components
- Each component has a coupling matrix
- Interferometer matrix built from nodal network
- **Ax = B** style matrix equation solved for **x** via sparse inversion (**A** = input vector, **B** = output vector)
- **x** contains solved field amplitudes at each node

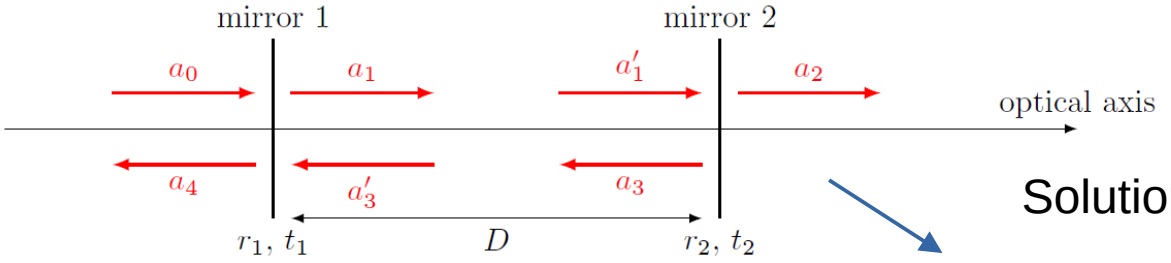


$$\begin{pmatrix} \text{Out1} \\ \text{Out2} \\ \text{Out3} \\ \text{Out4} \end{pmatrix} = \begin{pmatrix} 0 & bs_{21} & bs_{31} & 0 \\ bs_{12} & 0 & 0 & bs_{42} \\ bs_{13} & 0 & 0 & bs_{43} \\ 0 & bs_{24} & bs_{34} & 0 \end{pmatrix} \begin{pmatrix} \text{In1} \\ \text{In2} \\ \text{In3} \\ \text{In4} \end{pmatrix}$$



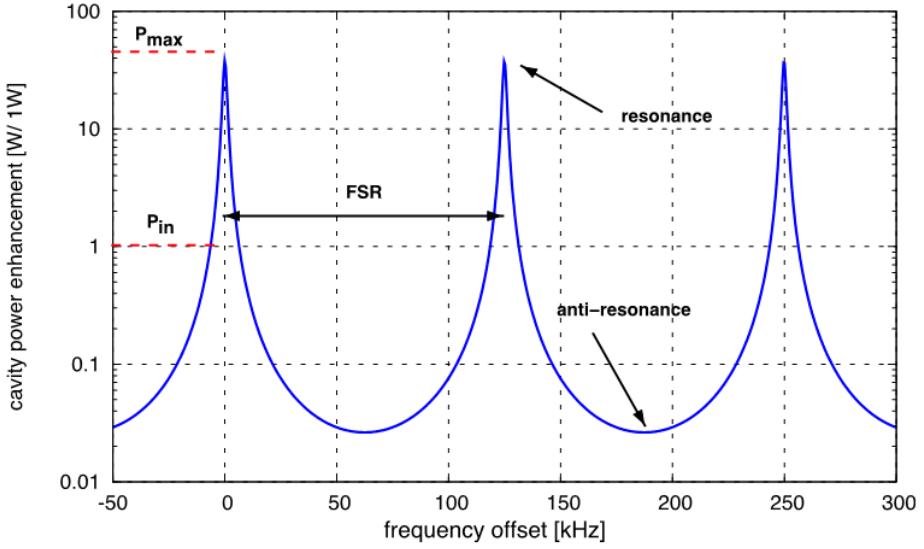
$$\begin{aligned} bs_{12} &= bs_{21} = r\ \exp\left(\mathrm{i}\,2\phi\omega/\omega_0 \cos\alpha\right), \\ bs_{13} &= bs_{31} = \mathrm{i}\,t, \\ bs_{24} &= bs_{42} = \mathrm{i}\,t, \\ bs_{34} &= bs_{43} = r\ \exp\left(-\mathrm{i}\,2\phi\omega/\omega_0 \cos\alpha\right), \end{aligned}$$

How does it work?



Solution: interferometer matrix

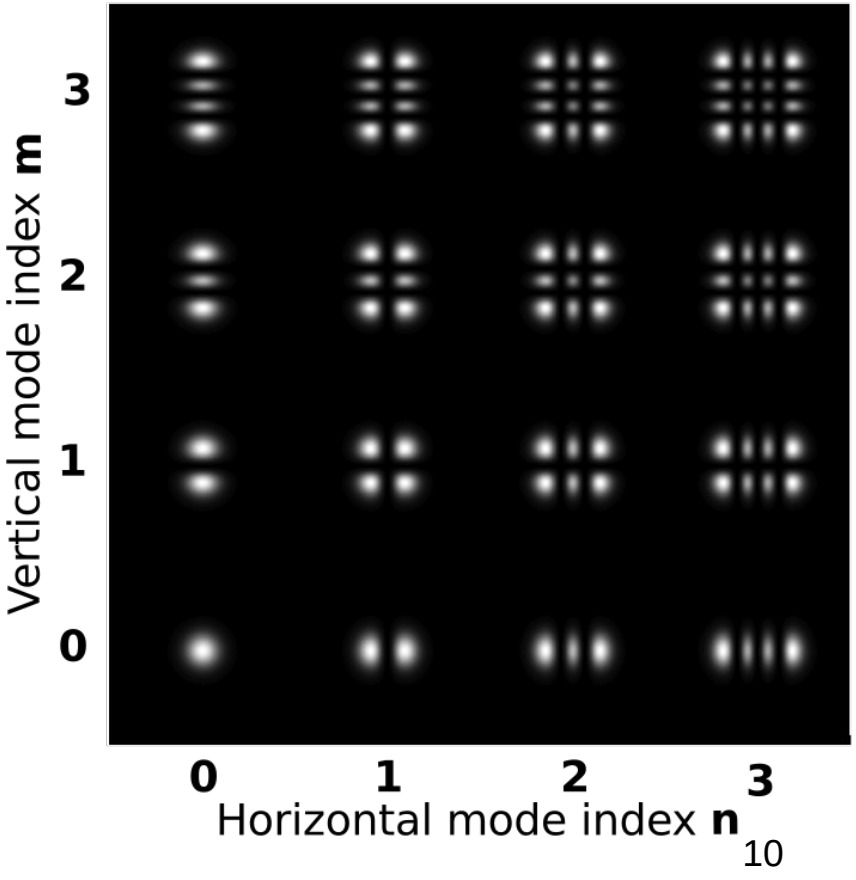
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mathrm{i}t_1 & 1 & 0 & -r_1 & 0 & 0 & 0 \\ -r_1 & 0 & 1 & -\mathrm{i}t_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -e^{-\mathrm{i}kD} \\ 0 & -e^{-\mathrm{i}kD} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mathrm{i}t_2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -r_2 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_4 \\ a_3' \\ a_1' \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} a_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



Post-processing computes powers, phases, demodulated signals (e.g. for Pound-Drever-Hall control) as a function of some variable

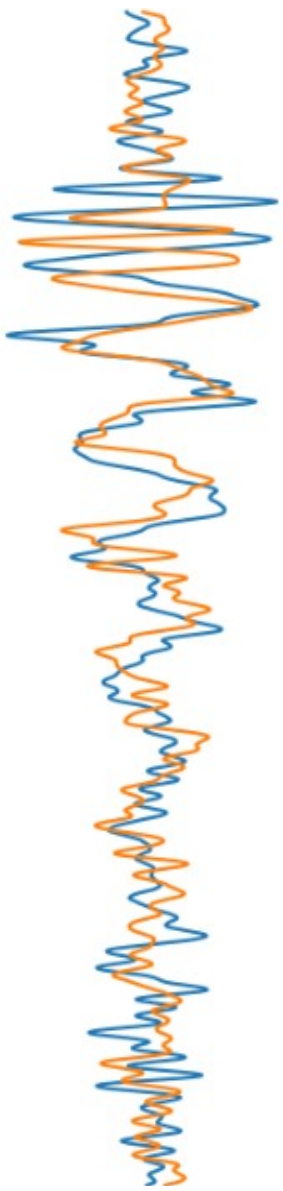
How does it work?

- FINESSE can also model transverse modes using Hermite- and Laguerre-Gaussian beams
- Misalignments and distortions can be represented as a series of higher order modes
- Simulation time dramatically increases with higher order modes – for certain studies an FFT model may be more appropriate



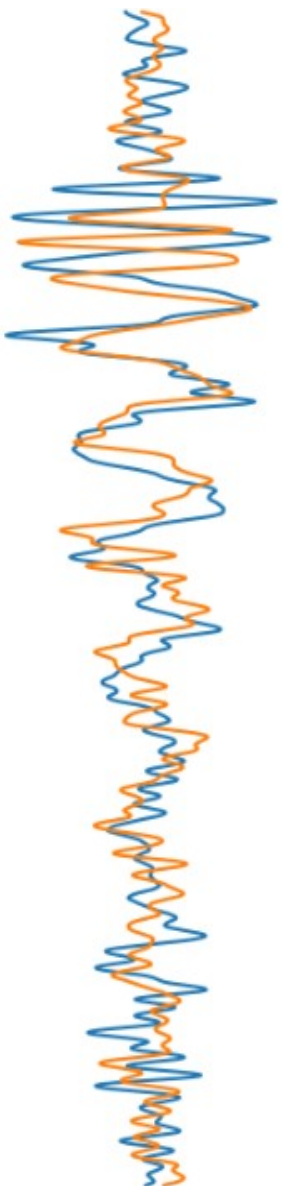
Pykat

- FINESSE was originally a stand-alone program, requiring a textual input file and producing a textual output file, e.g.:
  1) Write a `myfile.kat` file with KatScript syntax to make a specific plot
  2) Run `kat myfile.kat` in a console to produce (many) output files
  3) Display plot using e.g. gnuplot, MATLAB, Python, etc.
  4) Edit `myfile.kat` as appropriate, repeating steps 1-3
- This is fine for getting started, but a hassle for more complex tasks
- Enter **Pykat**: a Python library for interacting with FINESSE
  - Simplifies process of building complex simulations
  - Usable within a Jupyter notebook to display a sequence of plots
  - The vast majority of FINESSE users use it via Pykat these days



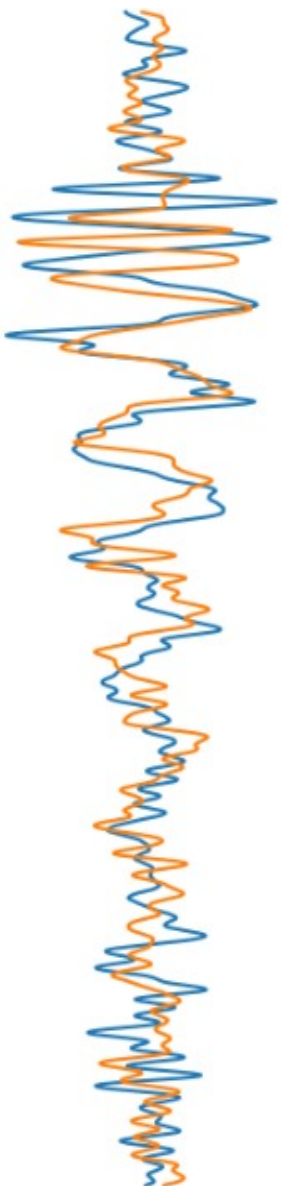
Typical Pykat workflows

- Use a **Jupyter notebook**:
  - Write KatScript syntax into a Python string then parse it using Pykat
  - Plot the result directly in the notebook
  - Edit the KatScript to meet your needs, rerun the cell
  - Create multiple clones of the Pykat “object” to compare models with different parameters
- Write **Python functions** to perform repetitive tasks:
  - e.g. optimisation of some parameter or finding the operating point of the interferometer
- Use a static KatScript file as a **reference model** (e.g. for Advanced LIGO), load it with Pykat to play with parameters



Brief timeline

- **1997**: Andreas Freise developed FINESSE as side project while doing PhD at GEO600
- **2006**: SimTools for using FINESSE with MATLAB developed
- **2012**: FINESSE code made open source
- **2013**: FINESSE v1 released
- **2014**: FINESSE v2 released (radiation pressure, rigorous quantum noise treatment)
- **2015**: Pykat released
- **2018-Present**: FINESSE v3 development ongoing



Getting started

- See http://www.gwoptics.org/finesse/#install for installation instructions
- See http://www.gwoptics.org/learn/ for more complete guide to using FINESSE for laser interferometry
- More specialised examples at https://logbooks.ifosim.org/pykat/
- Q&A for LIGO members: https://chat.ligo.org/ligo/channels/finesse
- Time to show some quick examples here...? (from https://git.ligo.org/finesse/finesse2_getting_started)