University of Glasgow

Science & Technology Facilities Council

Using Sim-Tools for the AEI-10m Finesse input file

S. Hild,

GEO-SIM meeting,

Hannover, August 2011

Overview

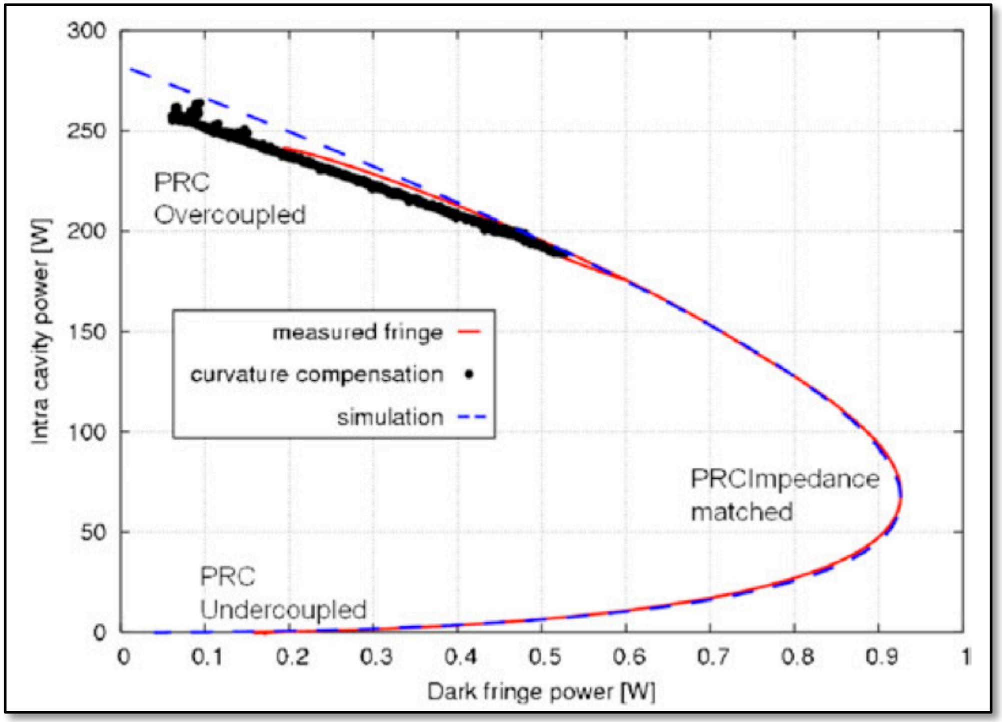
- Why are accurate simulations useful?
- An accurate input file is the key!
- Different demands on an input file: GEO vs AEI-10m.
- One file for global parameters.
- Doing a Finesse simulation with the Finesse tools.
- Documentation for the AEI-10m master input file.

Overview

- Why are accurate simulations useful?
- An accurate input file is the key!
- Different demands on an input file: GEO vs AEI-10m.
- One file for global parameters.
- Doing a Finesse simulation with the Finesse tools.
- Documentation for the AEI-10m master input file.

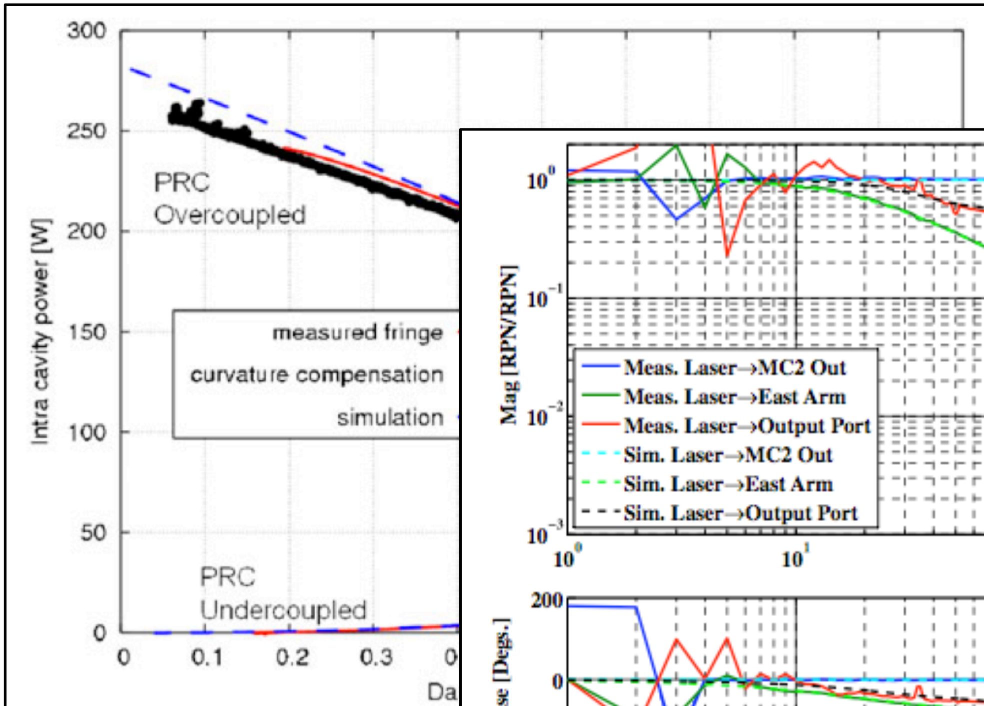
No rocket science --- just an example for a workflow that we found to be useful.

Why are accurate simulations useful?

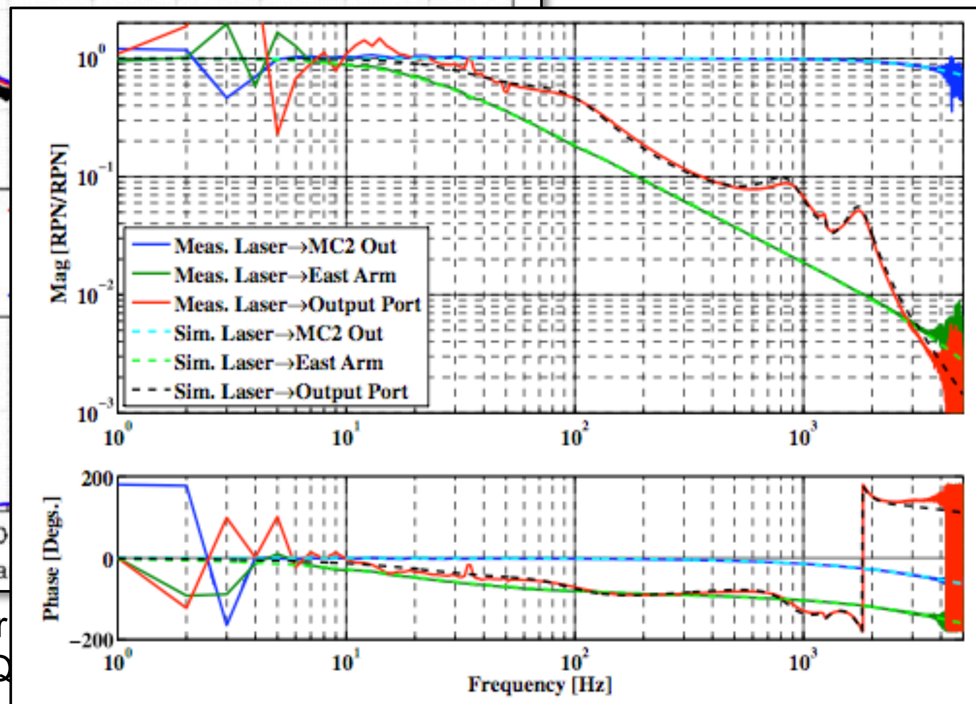


H. Lueck et al: Thermal correction of the radii of curvature of mirrors for GEO 600, CQG, **2004**, *21*, S985-S989

Why are accurate simulations useful?

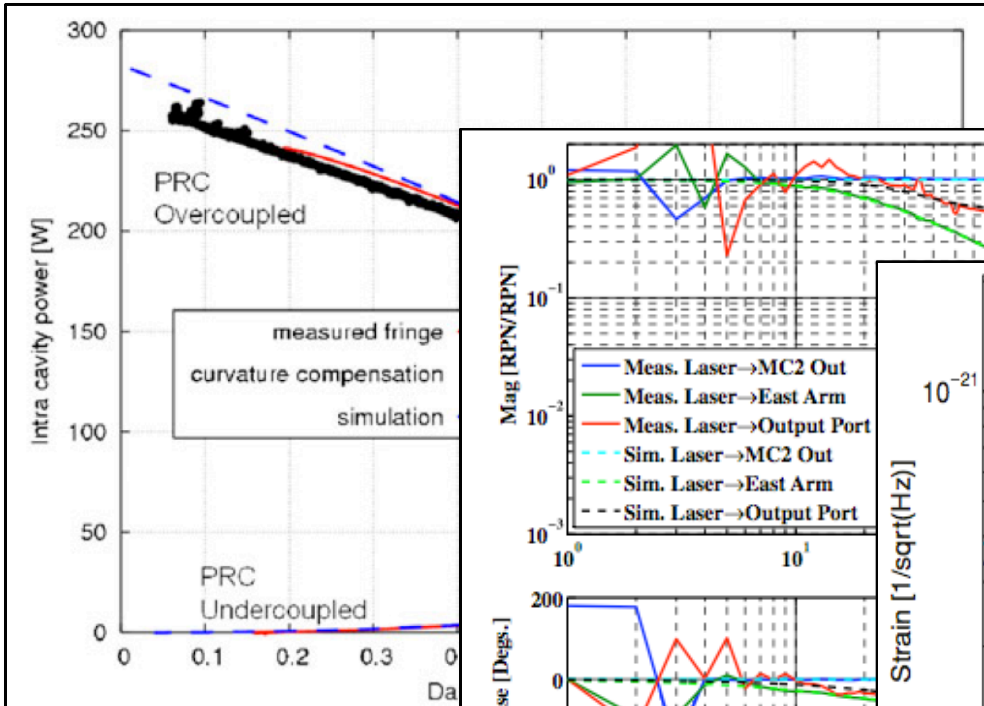


H. Lueck et al: Thermal cor
of mirrors for GEO 600, CQ

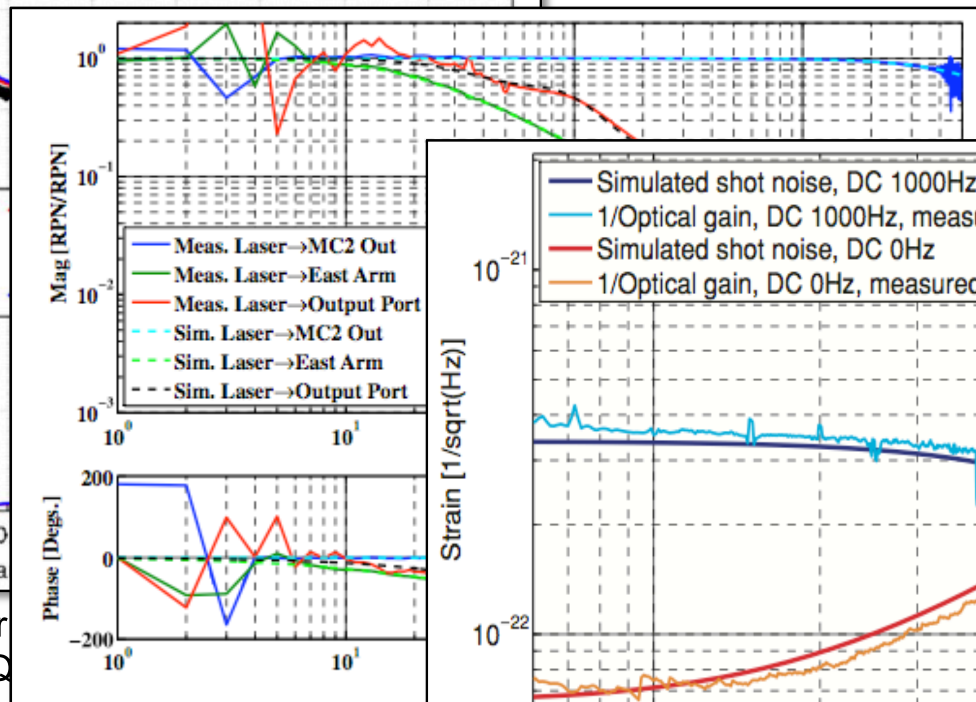


J.R. Smith et al: Measurement and simulation of laser power noise in GEO 600, *CQG*, **2008**, *25*, 035003

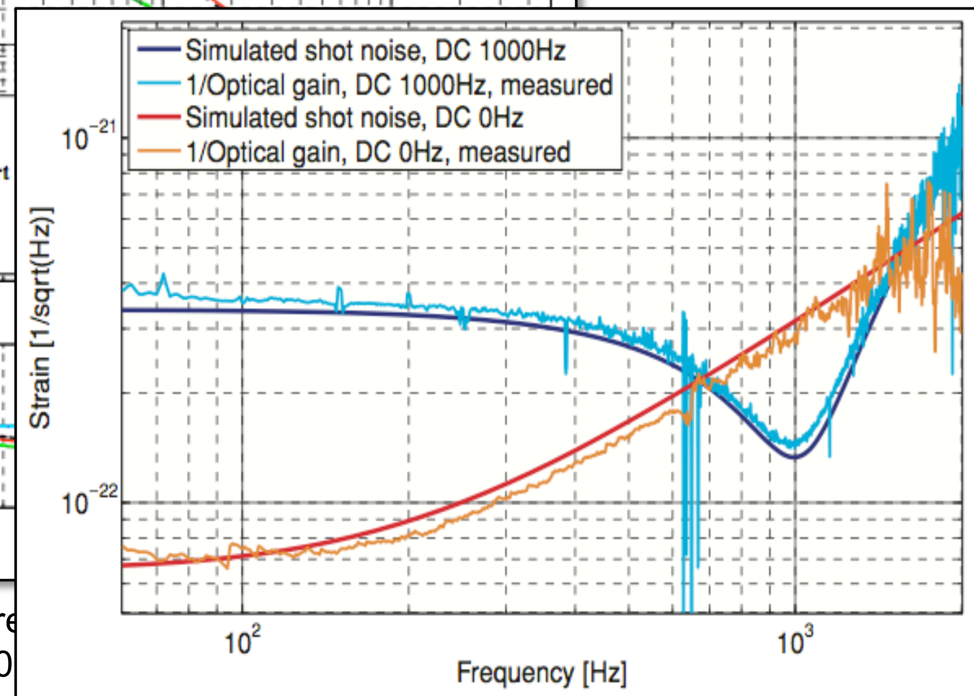
Why are accurate simulations useful?



H. Lueck et al: Thermal cor
of mirrors for GEO 600, CQ



J.R. Smith et al: Measure
power noise in GEO 600



S. Hild et al: DC-readout of a signal-recycled gravitational wave detector: CQG, **2009**, *26*, 055012

University of Glasgow

The GEO-Finesse file

- Originally started by Andreas more than 10 years ago.
- Survived a handful of ‘file-keepers’.
- A continuous process of improvements: More accurate parameters + more and more parameters included.
- Everything in this file is hard-coded.
- Every time the system is change (DC-readout, OMC, new MSR) someone implements this change in the file.
- System works pretty well for GEO.

```
#--------------------------------------------------------------------------
# geo600-main.kat $Rev: 42 $
# Andreas Freise (adf@star.sr.bham.ac.uk)
# $Date: 2007-09-11 16:23:08 +0100 (Tue, 11 Sep 2007) $
#
# Input File for FINESSE (www.rzg.mpg.de/~adf)
#
# Optical layout of GEO 600 with 'real' parameters .
#
# This file should NOT be used as is. It serves as a data
# container. For every simulation task only a subsystem
# should be copied out of this file. Please take care that
# you check the 'operating' point of the subsystem properly
# before doing any complex analysis!
#----------------------------------------------------------------
#
# History:
#
# 16.05.2007 by  Andreas Freise (adf@star.sr.bham.ac.uk)
#
# - changed comments on laser power (email from Josh)
# - corrected BS thermal lens to 9k according to formula
# - changed order of the cavity commands
#
# 12.12.2006 by  Andreas Freise (adf@star.sr.bham.ac.uk)
#
# - changed distribution of losses to be 130ppm on each
#   surface inside the DR MI
#
# 12.12.2006 by  Andreas Freise (adf@star.sr.bham.ac.uk)
#
# - changed MC mirror parameters according to labbook page 4027
# - changed laser power to be at (75deg ->) 4.82W (page 3984)
# - changed node names of MPR and MSR
# - changed mirror specs of MCN, MFN, MCE. MFE according to
#   labbook page 4028, results are exactly as stated there by Hartmut
# - not yet done: mode-matching, curvature, compensation check, ....
#
# 21.11.2006 by  Andreas Freise (adf@star.sr.bham.ac.uk)
#                Stefan Hild    (stefan.hild@aei.mpg.de)
#
#   Restarting GEO 600 file maintenance, see Labbook page 4011
#   Staring from a file called power_evl_curr_8_car.kat
#   (26.06.2006, labbook page 3656)
#
# - changed syntax to Finesse version 0.99.4
# - put Mode cleaners back in
# - updated modulation indices (see page 4011)
```

Why could a different system be useful for the AEI-10m?

- GEO is fairly mature and parameter change rather rarely.
- In contrast, for the AEI-10m nearly none of the parameters have so far values ‘set in stone’.
- At the current stage of the design you also often want to play with several different configurations to compare them.

- Therefore a Finesse input file with hard-wired parameters would be of very limited use.

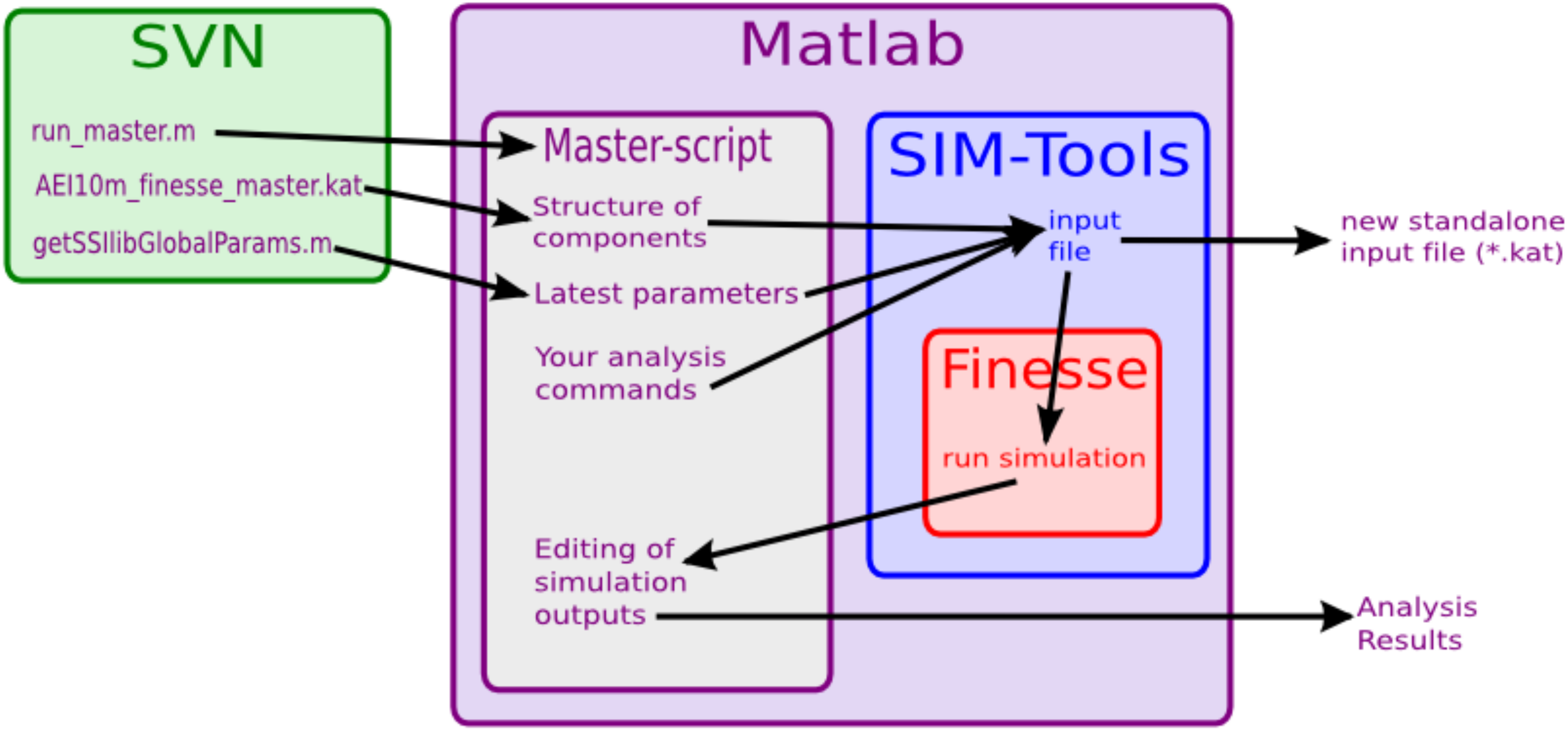
One global parameter file

- Christian started one Matlab file called *getSSIGlobalParms.m* which should serve as the one and only point where we collect optical parameters.
- All different simulations (Finesse, Optickle etc) can access the parameters from that file.
- Idea is that we only have to change a parameter in a single file.
- This way we ensure to always have the latest parameters.

```matlab
function [ parmStruct ] = getSSIGlobalParms()
%getSSIGlobal_Parms Returns global parameters for the SubSQL IFO
%
%    Usage: ps = getSSIGlobalParms()
%
%    Parameters can be accessed from the caller of the function using the
%    dot notation, e.g. x = ps.IM.R to store the IM power reflectivity in
%    the variable x.
%
%    All numbers, if not explicitly noted, taken from Kentaro's
%        "Conceptual design of an interferometer with a sub-SQL sensitivity ver. 2.0"
%
%
%     For non-pure Matlab-based usage keep an eye on potential TRUNCATION ERROR!
%     (e.g. for passing small numbers to Finesse via SimTools).
%     Matlab displays numbers at lower precision than they are handled
%     internally. It is advisable to enforce the desired precision e.g.
%     with a command in the style of
%                  num2str(RoC_IM_HR,'% 10.8f');
%     i.e. by using format specifiers to export the constant RoC_IM_HR to
%     a .kat file via SimTools or similar applications.
%
% THIS FILE SHOULD ONLY BE EDITED BY CHRISTIAN GRAEF AND STEFAN HILD



    % Fundamental laser wave length
    parmStruct.lambda0 = 1064e-9;

    % Laser input power
    parmStruct.Pin = 5.5; %%% This number needs TBC !!!

    % Refractive index FS
    parmStruct.nFusedSilica = 1.4496309898590634;

    % Thickness of main suspension substrates
    parmStruct.SubstrateThickness = 0.0245;

    % Radius of main suspension substrates
    parmStruct.SubstrateRadius = 0.0243;

    % optimal spot radius on IFO mirrors, a/2.5 to keep diffraction loss
    % at a tolerable level of ~ 2.25ppm
    parmStruct.w_optim = parmStruct.SubstrateRadius / 2.5;

    % optical pathlength in IEM substrate
```

This file is available for everyone and can be found in the following svn repository:
https://arran.physics.gla.ac.uk/svn/AEI-PT- locking/trunk/Matlab/SSILib/

How is the Finesse input file created?



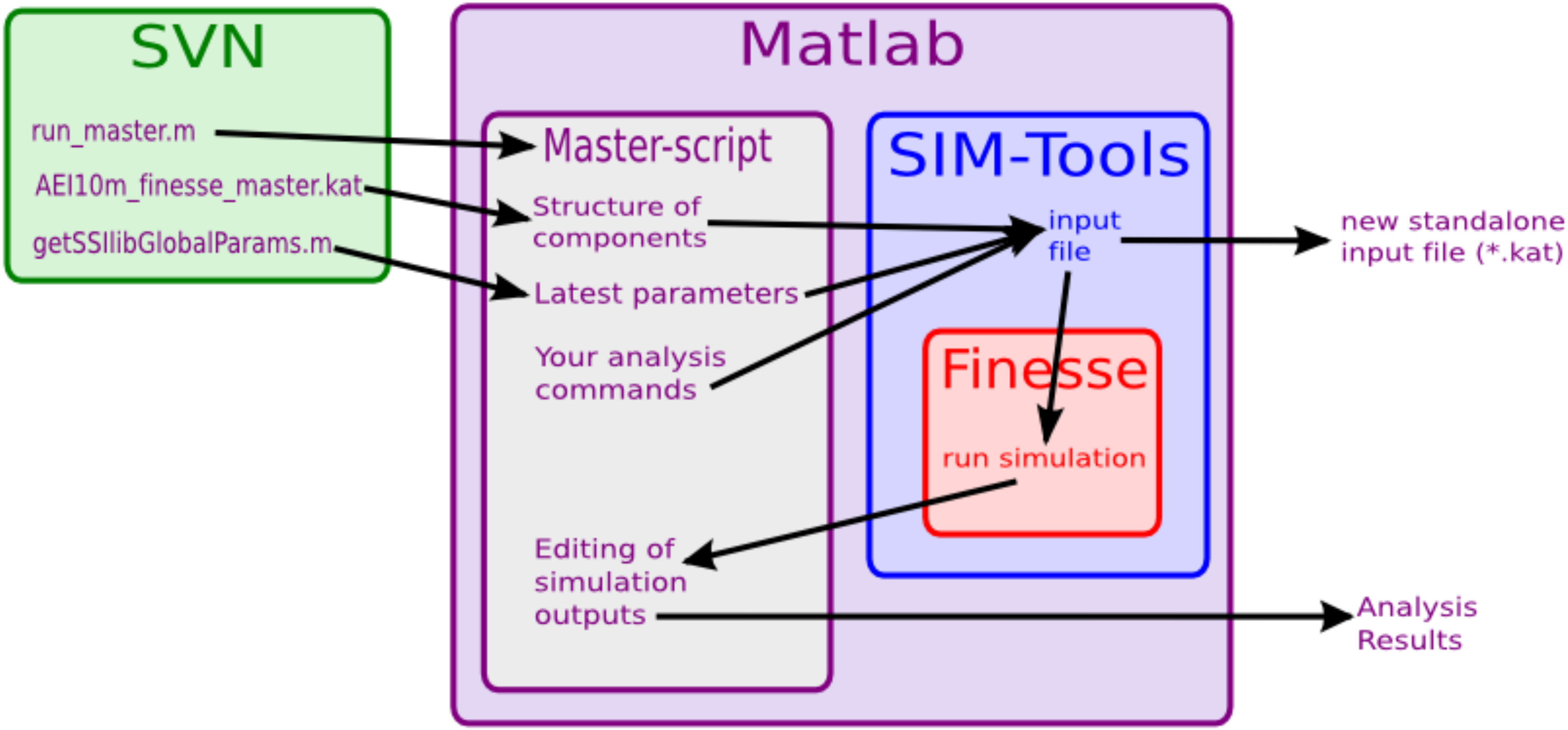
Structure file

- Finesse input file with all the components included: AEI10m_finesse_master.kat
- However this file is static and does not include any values for the parameters or commands.
- Please not FT_commands are included to properly read this into Matlab via Finesse_tools.

This file can be found in the following svn repository: https://arran.physics.gla.ac.uk/svn/AEI-PT- locking/ trunk/Finesse/Stefan/Finesse_master

```
%
s lx $lx nbs3 nIMx1                                        % length betwee
%
% IMx
bs2 IMxAR $RIMxAR $LIMxAR $IMxARphi 0 nIMx1 nPDMx nIMxi1 dump
s sIMx $sIM $nsilica nIMxi1 nIMxi2
m1 IMx $TIMx $LIMx $IMxphi nIMxi2 nIMx2
attr IMxAR Rc $RCIMxAR
attr IMx Rc $RCIMx
%
s LX $LX1 nIMx2 nEMx1
%
% EMx
m1 EMx $TEMx $LEMx $EMxphi nEMx1 nEMxi1
s sEMx  $sEM $nsilica nEMxi1 nEMxi2
m2 EMxAR $REMxAR $LEMxAR $EMxARphi nEMxi2 nEMxo
attr EMx Rc $RCEMx
%
cav x_FP IMx nIMx2 EMx nEMx1
%
%
%%%%%%%%%%%%%%%%%%%%%%%%% Y Arm %%%%%%%%%%%%%%%%
%
s ly $ly nbs2 nIMy1                                         % length betw
%
% IMy
bs2 IMyAR $RIMyAR $LIMyAR $IMyARphi 0 nIMy1 nPDMy nIMyi1 dump
s sIMy $sIM $nsilica nIMyi1 nIMyi2
m1 IMy $TIMy $LIMy $IMyphi nIMyi2 nIMy2
attr IMyAR Rc $RCIMyAR
attr IMy Rc $RCIMy
%
s LY $LY1 nIMy2 nEMy1
%
% EMy
m1 EMy $TEMy $LEMy $EMyphi nEMy1 nEMyi1
s sEMy  $sEM $nsilica nEMyi1 nEMyi2
m2 EMyAR $REMyAR  $LEMyAR $EMyARphi nEMyi2 nEMyo
attr EMy Rc $RCEMy
%
cav y_FP IMy nIMy2 EMy nEMy1
%
%
%%% FTend
```

How is the Finesse input file created?



Example of one full simulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This file is the Finesse run script for the AEI-10m Sub-SQL interferometer
% in the configuration without Khalili cavities, i.e. a simple Michelson
% with FP arm cavities. The skelton of the finesse input file is read in
% from 'AEI10m_finesse_master_3dof_14062011.kat' which is available at:
% https://arran.physics.gla.ac.uk/svn/AEI-PT-locking/trunk/Finesse/Stefan/Finesse_master/
% Most of the constants are then read in from the unique parameter file
% getSSIGlobalParams.m which is available form:
% https://arran.physics.gla.ac.uk/svn/AEI-PT-locking/trunk/Matlab/SSIlib/
% Please note that you need the Sim-Tools package for these operations.
%
% 20.06.2011 S. Hild (stefan.hild@ligo.org) and C. Graef (cristian.graef@aei.mpg.de)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

ps = getSSIGlobalParms;                      % read in parameters from unique constant file


FT=FT_init_Finesse('name','kat','path','~/Software/Finesse0997/bin/')

%% Read parameters from the intial Finesse input file
blocks=FT_read_blocks_from_file('AEI10m_finesse_master_3dof_14062011.kat');


%% Create new block 'commands1' as a container for the 'dynamic' properties
constants = FT_create_new_block('constants');


%% Commands to be merged to the kat-file dynamically
constants = FT_add_line_to_block(constants, '% ------------------');
constants = FT_add_line_to_block(constants, '% Substrates');
constants = FT_add_line_to_block(constants, '% ------------------');
constants = FT_add_line_to_block(constants, strcat('const nsilica', {' '}, num2str(ps.nFusedSilica, '% 16.14f')));
constants = FT_add_line_to_block(constants, strcat('const sIM', {' '}, num2str(ps.SubstrateThickness, '% 16.14f')));
constants = FT_add_line_to_block(constants, strcat('const sEM', {' '}, num2str(ps.SubstrateThickness, '% 16.14f')));
constants = FT_add_line_to_block(constants, strcat('const sBS', {' '}, num2str(ps.SubstrateThickness_BS, '% 16.14f')));
constants = FT_add_line_to_block(constants, '% ');
```

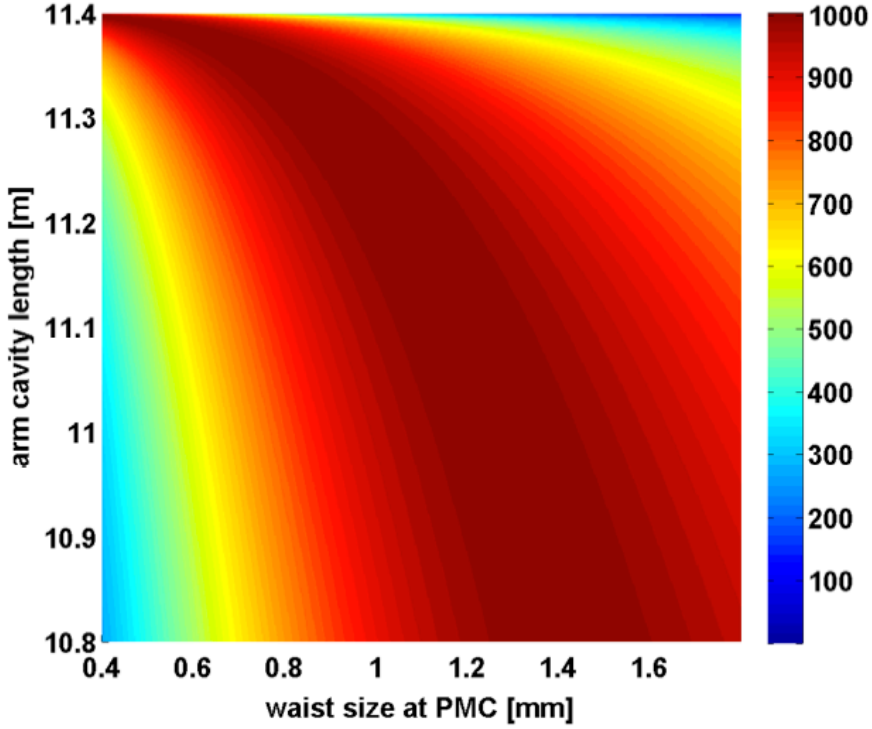
Example of one full simulation

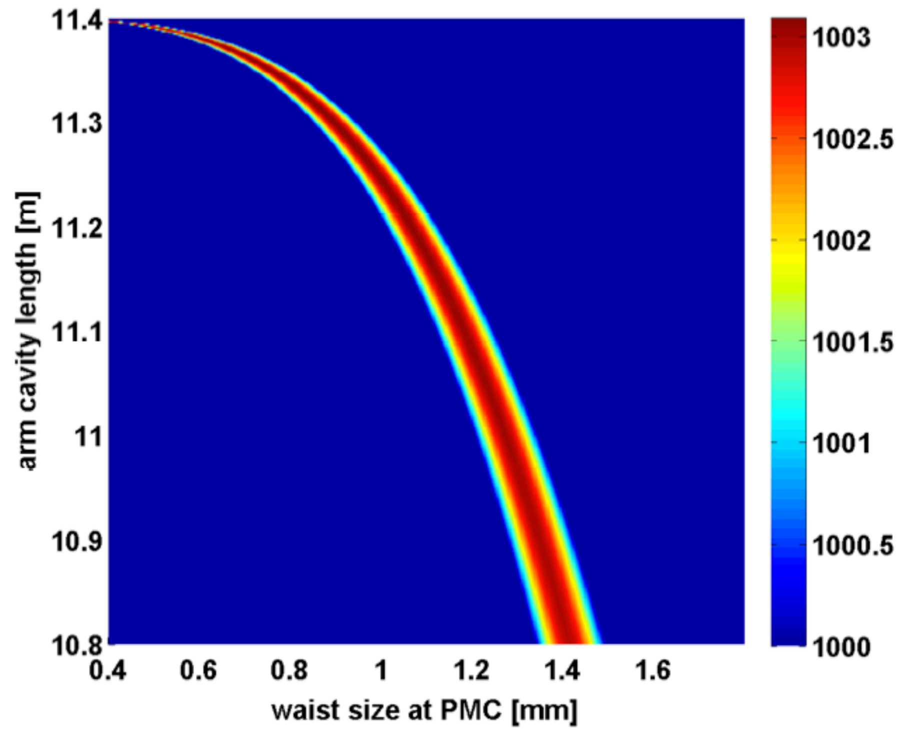
```
constants = FT_add_line_to_block(constants, '% ');

w_scan = 400e-6:2e-6:1800e-6;

for i=1:length(w_scan);

%% Create new block 'commands1' as a container for the 'dynamic' properties
commands1 = FT_create_new_block('commands1');


%% Commands to be merged to the kat-file dynamically

commands1 = FT_add_line_to_block(constants, strcat('const i1w', {' '}, num2str(w_scan(i), '% 16.14f')));


commands1 = FT_add_line_to_block(commands1, 'maxtem 3');

commands1 = FT_add_line_to_block(commands1, 'xaxis LX 1 lin 10.8 11.4 400');
commands1 = FT_add_line_to_block(commands1, 'put LY 1 $x1');

%commands1 = FT_add_line_to_block(commands1, 'x2axis beam_in w0 lin 420e-6 1000e-6 20');

commands1 = FT_add_line_to_block(commands1, 'pd PDs_DC nPDs');
commands1 = FT_add_line_to_block(commands1, 'pd PDa_DC nPDa');

commands1 = FT_add_line_to_block(commands1, 'pd PDx_DC nEMx1');
commands1 = FT_add_line_to_block(commands1, 'pd PDy_DC nEMy1');

%commands1 = FT_add_line_to_block(commands1, 'bp analyser x w nIMx1*');

commands1 = FT_add_line_to_block(commands1, 'yaxis abs');
commands1 = FT_add_line_to_block(commands1, 'retrace');

commands1 = FT_add_line_to_block(commands1, 'gnuterm aqua');
```

Example of one full simulation

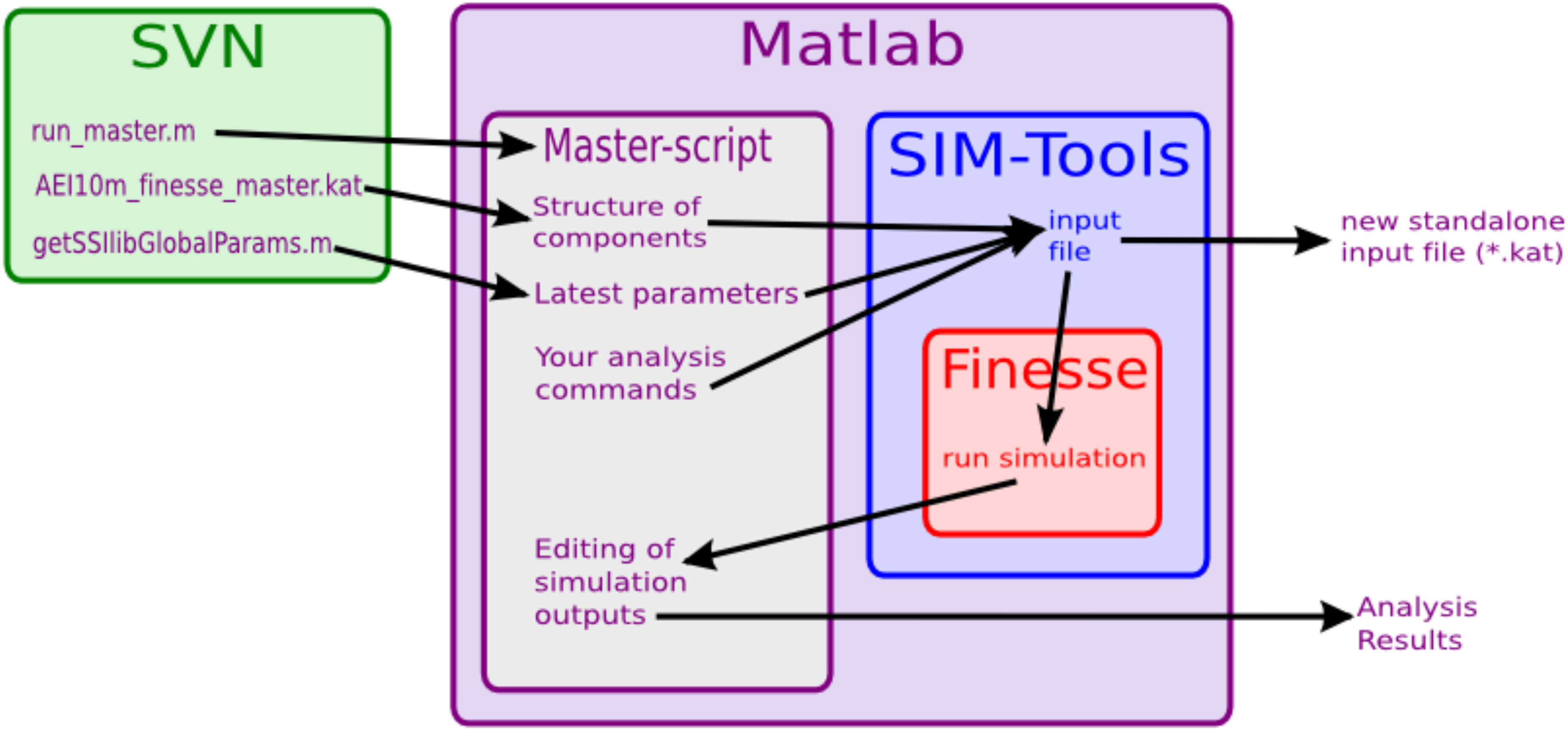
```
%% Collect blocks necessary for this task from definition kat-file
outbl1(1) = FT_copy_block(blocks, 'AEI10m');
outbl1(2) = constants;
outbl1(3) = commands1;

%% Write the new kat file
FT_write_blocks_into_file(outbl1,'test_master.kat');
Run1=FT_create_new_run();
Run1.filename='test_master.kat';
Run1=FT_run_kat_simulation(FT,Run1);

Py(:,i)=Run1.data(:,5);
Px(:,i)=Run1.data(:,4);
Pa(:,i)=Run1.data(:,3);
Ps(:,i)=Run1.data(:,2);


%bp(i) = Run1.data(1,6);
i
end

%% Create some error signal plots
 figure;
 subplot(2,1,1)
 plot(Run1.data(:,1), Run1.data(:,2),Run1.data(:,1), Run1.data(:,3));
 hold;
 grid on
 xlabel('BS detuning (deg)');
 yaxis(0,0.5)
 ylabel('Light power [W]');
 legend('PDs','PDa');

  subplot(2,1,2)
 plot(Run1.data(:,1), Run1.data(:,4),Run1.data(:,1), Run1.data(:,5));
 hold;
 grid on
 xlabel('BS detuning (deg)');
 ylabel('Light power [W]');
 legend('x arm','y arm');
```

Example of one full simulation: The Result





How is the Finesse input file created?



Documentation of the Finesse master file

Documentation of the Finesse input file of the AEI-10m Sub-SQL interferometer

Stefan Hild, Christian Gräf and Sabina Huttner

July 14, 2011

Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut)

University of Glasgow

SUPA, School of Physics and Astronomy, The University of Glasgow, Glasgow, G12 8QQ, United Kingdom
Max–Planck–Institut für Gravitationsphysik and Leibniz Universität Hannover, D-30167 Hannover, Germany

Contents

- Continuously growing file documenting the sanity checks simulations carried out.
- Doc also available on svn

Documentation of the Finesse master file

3 Modematching into the FP-Michelson

3.1 Initial Mode Matching Set-up

The initial mode matching set-up refers to the configuration in which the arm cavities feature extremely large beam sizes at the main test masses, i.e. $w_1 = 9.72$ m. A schematic overview of this configuration is shown in Figure 3. The arm cavity mode features equal beam sizes on the input (IM) and end test masses (EM), which leads to the cavity waist position being exactly centered in the arm cavities. The mode matching is then done as follows: From the waist (i1w) within the pre-mode-cleaner (PMC) the beam diverges over the distance S1 (of the order of 10 m, which is the distance between the central and the far tables) and is the collimated by a mirror (Mcol) and send back over S2 towards the beam splitter. As the beam passing the beam splitter is not diverging, no unwanted astigmatism is introduced. The final step of the mode matching is then achieved by shaping the rear surface of the input test masses (IMxAR) to focus the beam into the arm cavities.

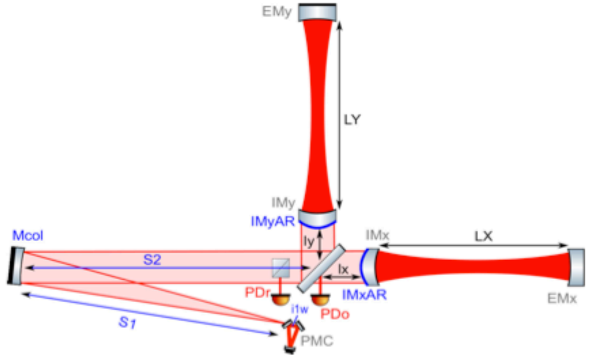


Figure 3: Simplified layout of the AEI-10m Sub-SQL interferometer as currently set up in the Finesse input file. The components relevant for the mode matching into the arm cavities are indicated by blue colour.

One of the keypoints of this configuration is that the radius of curvature (RoC) of the collimating mirror (Mcol) is twice the distance between the PMC waist and Mcol, in order to have a completely collimated beam in the central interferometer. Having a collimated beam in the central interferometer is a main requirement for achieving optimal mode matching into both arm cavities when using a macroscopic Schnupp asymmetry. The key parameters of this mode matching telescope are given as follows:

```
const i1w 420u
const S1 12.0
const RC_Mcol 24.0
const S2 12.0
```

```
const lx 0.400
const ly 0.600
const RCIMxAR -1.77600
const RCIMyAR -1.77600
const LX 11.395300
const LY 11.395300
const RCIMx -5.7
const RCIMy -5.7
const RCEMx 5.7
const RCEMy 5.7
```

Figure 4 shows the same plot as in Figure 2, but this time with proper gaussian beams. This simulation was done with *maxtem = 3* and using the *cav* command. As one can see using the realistic mode matching only very slightly decreases the intracavity powers by each 0.2 W from 1003.1 to 1002.9 W and from 1002.9 to 1002.7 W for the y and x-arm cavities, respectively. So, this seems to be a mode matching configuration we can in principle go forward with. The 0.2 W difference can potentially originate from imperfections: Firstly the beam between the collimating mirror and the input mirrors can of course not be perfectly collimated, but will still have a tiny, though measurable divergence. Secondly, we only use a certain number of digits for our input parameters.

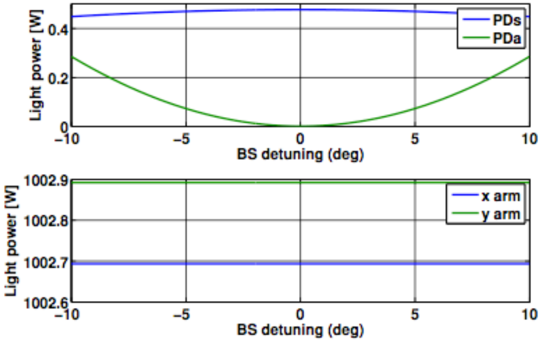


Figure 4: Optical powers achieved with the nominal mode matching configuration as shown in Figure 3 and the parameter values stated above.

The automatically generated finesse file used for this analysis `test_master_22062011a.kat` can be found in the svn for reference.

Documentation of the Finesse master file

3 Modematching into the FP-Michelson

3.1 Initial Mode Matching Set-up

The initial mode matching set-up refers to the configuration in which the arm cavities feature extremely large beam sizes at the main test masses, i.e. $w_1 = 9.72\,$m. A schematic overview of this configuration is shown in Figure 3. The arm cavity mode features equal beam sizes on the input (IM) and end test masses (EM), which leads to the cavity waist position being exactly centered in the arm cavities. The mode matching is then done as follows: From the waist (i1w) within the pre-mode-cleaner (PMC) the beam diverges over the distance S1 (of the order of 10 m, which is the distance between the central and the far tables) and is the collimated by a mirror (Mcol) and send back over S2 towards the beam splitter. As the beam passing the beam splitter is not diverging, no unwanted astigmatism is introduced. The final step of the mode matching is then achieved by shaping the rear surface of the input test masses (IMxAR) to focus the beam into the arm cavities.

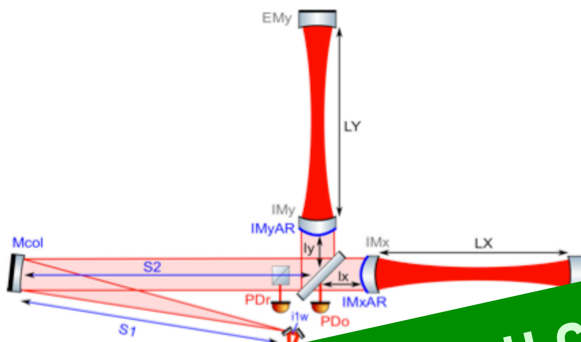


Figure 3: Simpli…

…ature (RoC) of the …MC waist and Mcol, in order to …interferometer. Having a collimated beam …main requirement for achieving optimal mode matching …when using a macroscopic Schnupp asymmetry. The key parameters …matching telescope are given as follows:

```
const i1w 420u
const S1 12.0
const RC_Mcol 24.0
const S2 12.0
```

```
const lx 0.400
const ly 0.600
const RCIMxAR -1.77600
const RCIMyAR -1.77600
const LX 11.395300
const LY 11.395300
const RCIMx -5.7
const RCIMy -5.7
const RCEMx 5.7
const RCEMy 5.7
```

Figure 4 shows the same plot as in Figure 2, but this time with proper gaussian beams. This simulation was done with *maxtem = 3* and using the *cav* command. As one can see using the realistic mode matching only very slightly decreases the intracavity powers by each 0.2 W from 1003.1 to 1002.9 W and from 1002.9 to 1002.7 W for the y and x-arm cavities, respectively. So, this seems to be a mode matching configuration we can in principle go forward with. The 0.2 W difference can potentially originate from imperfections: Firstly the beam between the collimating mirror and the input mirrors can of course not be perfectly collimated, but will still have a tiny, though measurable divergence. Secondly, we only use a certain number of digits for our input parameters.

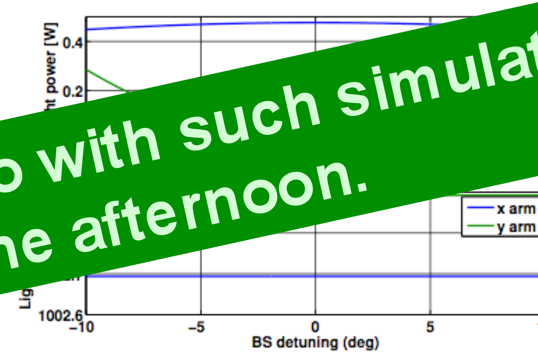


Figure 4: Optical powers achieved with the nominal mode matching configuration as shown in Figure 3 and the parameter values stated above.

The automatically generated finesse file used for this analysis `test_master_22062011a.kat` can be found in the svn for reference.

For the nice physics you can do with such simulations, Please see Christian's talk in the afternoon.

Summary

- Finesse Tools are very useful.
- AEI-10m master finesse file automatically uses the latest parameter set.
- Good documentation, via auto-vreated stand-alone input files plus the document on the svn.

- Everybody is welcome to use this stuff …
- If you need an account for the svn, please just write me an email …