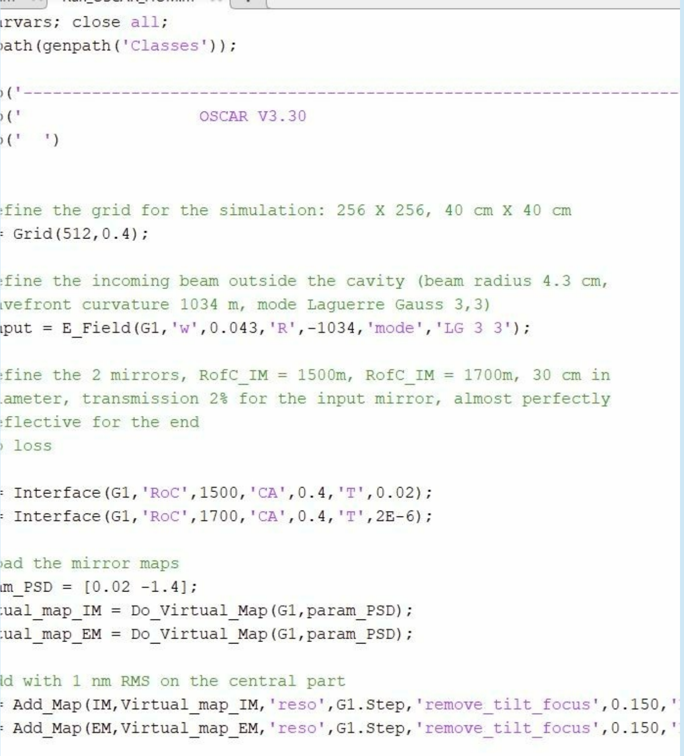
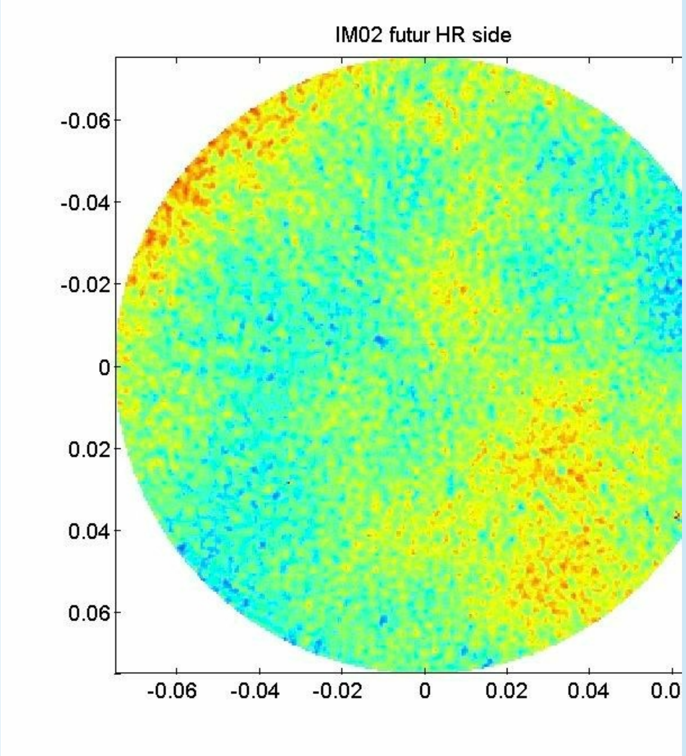
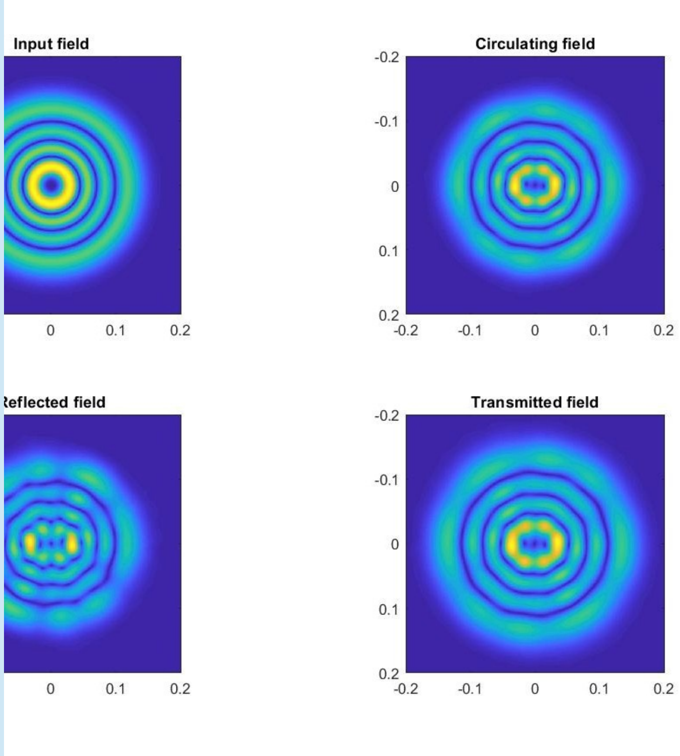






OSCAR presentation

J. Degallaix

Interferometer Simulation Workshop – June 2023

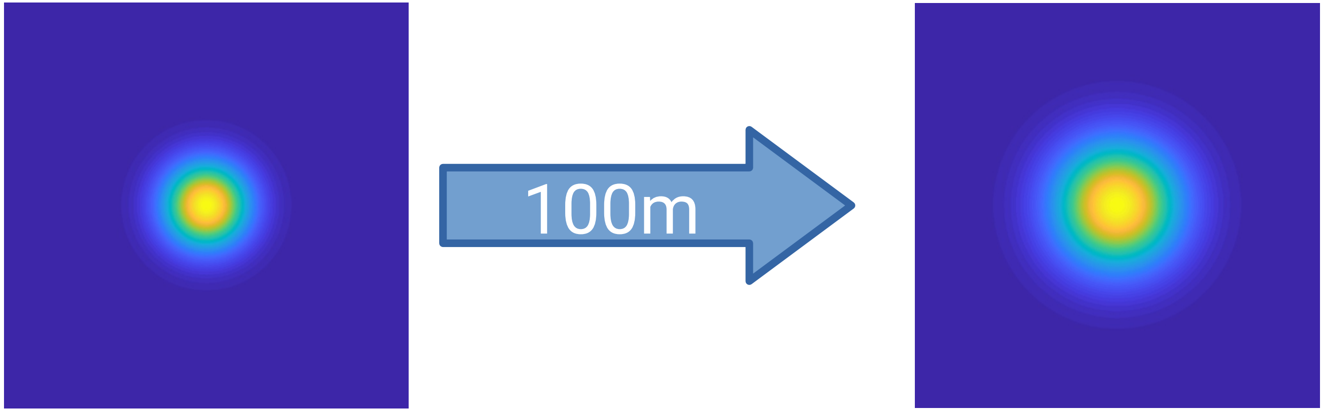
OSCAR is

- an optical simulation code
- written in Matlab
- used to simulate optical cavities with imperfections
- calculate the steady state electrical fields

I
Principle

The foundation : propagation of arbitrary fields

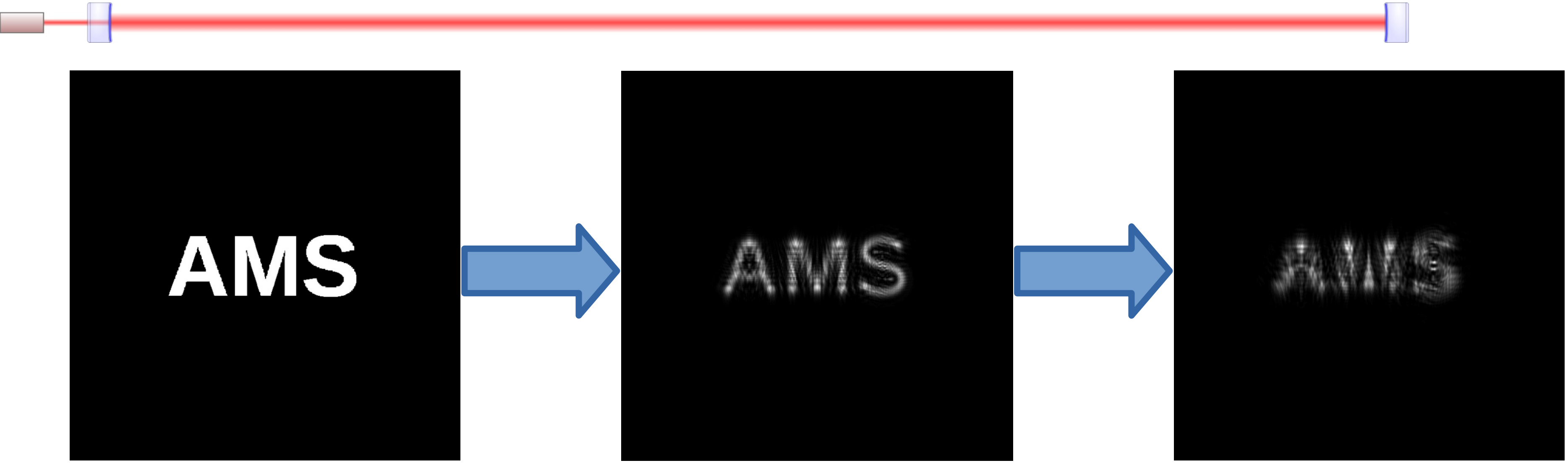
- numerically based on 2D FFT



Electric field =
2D matrix of complex numbers

- worked under the paraxial approximation
- very easy to implement new propagation algorithms

Independent of any complete basis



- includes automatically in the propagation the correct :
  - beam size
  - complex wavefront curvature
  - (Gouy) propagation phase

Adding arbitrary distortions

- lenses, mirrors, distortions are given as wavefront distortions

$$E_o(x,y) = E_i(x,y)\exp\left(-jk\Delta OPL(x,y)\right)$$

ΔOPL = Optical Path Length in meter

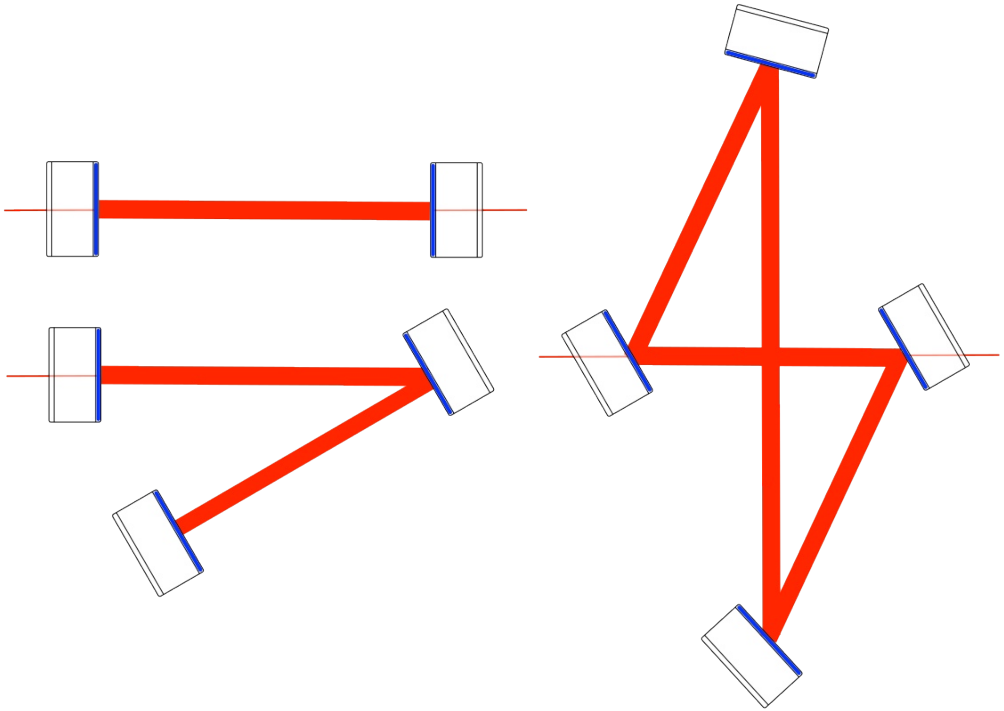
- Example for a lens:

$$\Delta OPL(x,y) = 2\left(RofC - \sqrt{RofC^2 - (x^2+y^2)}\right)$$

- Can also define arbitrary apertures

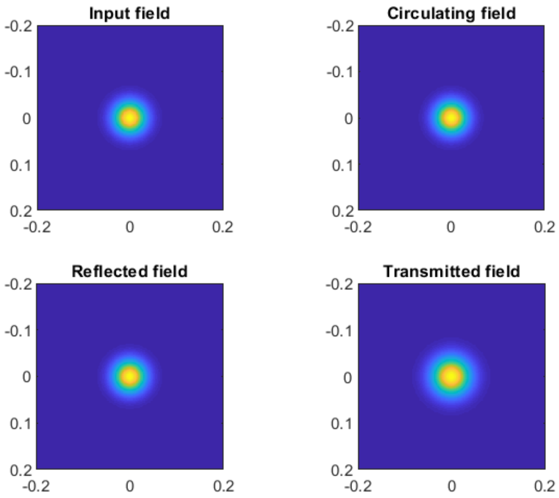
Optical cavities

- can create cavities with n mirrors $n \geq 2$
- resonant or not
- stable of not
- coupled cavities
- with imperfections
  - arbitrary mirror shape
  - misaligned
  - mismatched



What kind of results to expect

- steady state fields (reflected, transmitted power
- beam shape
- optical mode spacing
- error signals
- clipping loss




```
---------- For the carrier ---------------
  Power in the input beam:   0.995009    [W]
  Circulating power:            282.095       [W]
  Transmitted power:         0.00141048       [W]
  Reflected power:             0.993599       [W]
  Round trip losses:         1.55851e-05      [ppm]

---------- For the sidebands 1, frequency:    10 [MHz] ---------------
 for the lower and upper sidebands respectively
 Power in the input beam:    0.00248131      0.00248131       [W]
 Circulating power:          4.90482e-05     4.90483e-05      [W]
 Transmitted power:          2.45241e-10     2.45241e-10      [W]
 Reflected power:            0.00248131      0.00248131       [W]
```

II
A typical program and examples

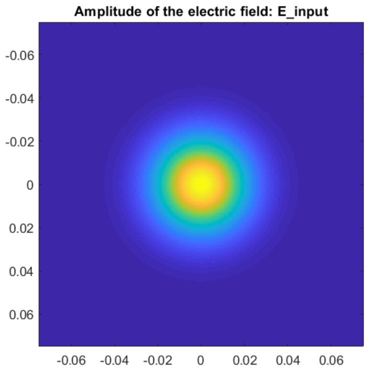
Practically, the starting point

First you must define a grid size for the simulation
- number of points
- physical size

```
% Define the grid for the simulation: 256 X 256, 15 cm X 15 cm
G1 = Grid(256,0.15);

% Define the incoming beam outside the input mirror (beam radius 2 cm), at
% the waist

E_input = E_Field(G1,'w0',0.022);
%E_input = Add_Sidebands(E_input,'Mod_freq',3.4E6,'Mod_depth',0.2);
```



E_Plot(E_input)

Then define the input beam

Then add surfaces to define mirrors

Interface = surface separating 2 media of different refractive indexes

```
% Define the 2 mirrors, one flat and the other with a RoC of 2400m, 10 cm in diameter, transmission 2% and 0.1%,
% no loss

IM = Interface(G1,'RoC',inf,'CA',0.1,'T',0.02);
EM = Interface(G1,'RoC',2400,'CA',0.1,'T',0.001);
```

Radius of curvature (in meter)

Aperture of the diameter (in meter)

Transmission (in power)

Refractive index assumed to be fused silica by default, second surface flat

It is all we need to define the cavity

```
% Use the 2 previous Interfaces and the input beam to defing a cavity 1000
% meter long
C1 = Cavity1(IM,EM,1000,E_input);
```

For 2 mirrors cavity

Input and end surfaces

Cavity length

Input beam

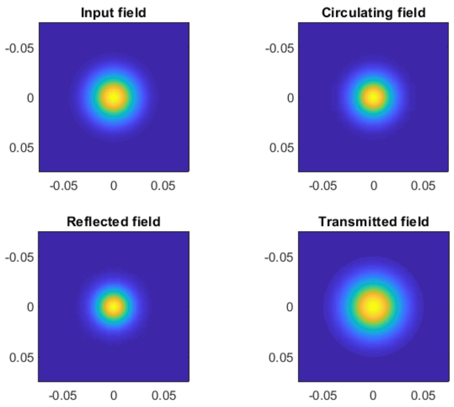
Calculate the resonance conditions and the steady state

```
% Calculate the resonance length
C1 = Cavity_Resonance_Phase(C1);

% Display the circulating power, reflected and transmitted powers
C2 = Calculate_Fields_AC(C1);
C2.Display_Results('display',false);
```

Results:

```
Found the phase for resonance in cavity: C1
  Power in the input beam:          1        [W]
  Circulating power:             153.098        [W]
  Transmitted power:            0.152881        [W]
  Reflected power:              0.590598        [W]
  Round trip losses:             1675.58        [ppm]
```



So the complete program is 8 lines

```
G1 = Grid(256,0.15);
E_input = E_Field(G1,'w0',0.022);
IM = Interface(G1,'RoC',inf,'CA',0.1,'T',0.02);
EM = Interface(G1,'RoC',2400,'CA',0.1,'T',0.001);
C1 = Cavity1(IM,EM,1000,E_input);
C1 = Cavity_Resonance_Phase(C1);
C2 = Calculate_Fields_AC(C1);
C2.Display_Results;
```

All the results are directly accessible and other calculations can be derived

A look at all the variables for the cavity

```
>> C2

C2 =

  Cavity1 with properties:

                 I_input: [1×1 Interface]
                   I_end: [1×1 Interface]
                  Length: 1000
                Laser_in: [1×1 E_Field]
    Laser_start_on_input: 0
              Run_on_GPU: 0
         Resonance_phase: 0.7978 - 0.6029i
   Cavity_scan_all_field: []
       Cavity_scan_param: [1000 500 2.0000e-09 1000]
      Cavity_phase_param: 200
           Cavity_scan_R: []
          Cavity_scan_RZ: []
           Cavity_EM_mat: []
         Propagation_mat: [1×1 Prop_operator]
              Field_circ: [1×1 E_Field]
               Field_ref: [1×1 E_Field]
             Field_trans: [1×1 E_Field]
         Field_reso_guess: [1×1 E_Field]
                Loss_RTL: 0.0017
```

C2.Resonance_phase = C2.Resonance_phase * exp(1i*pi)

E_Plot(C2.Field_trans)

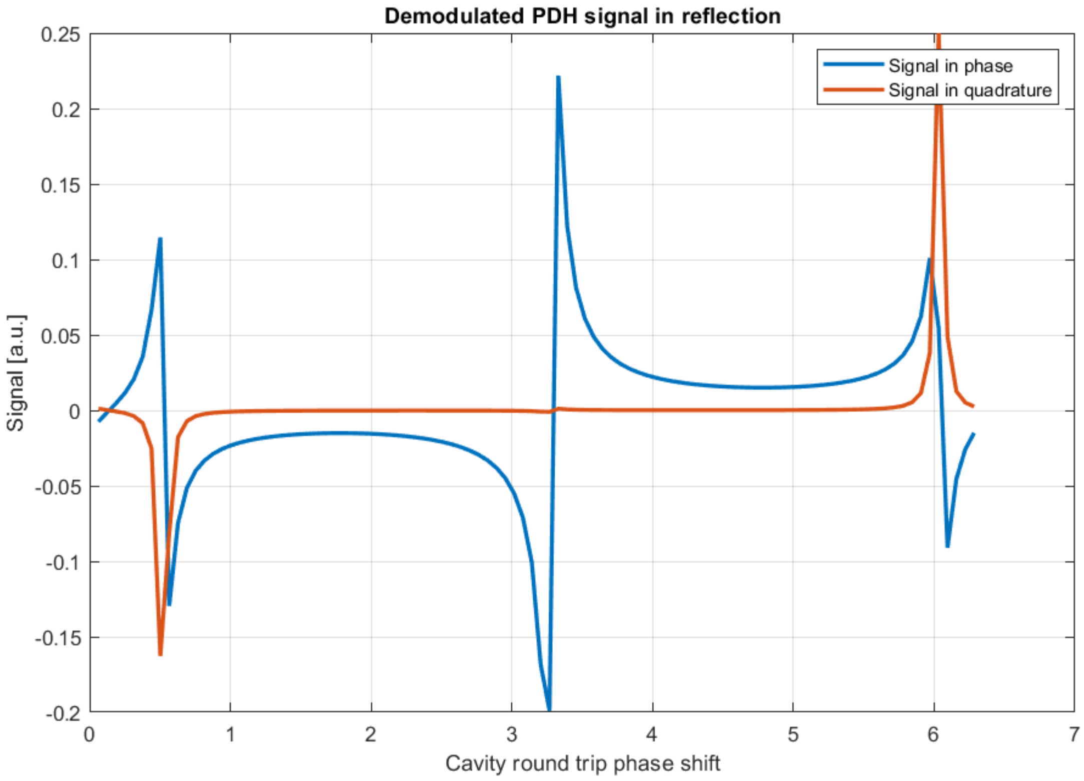
Calculate_Power(C2.Field_ref)

Live examples

- calculate cavity eigen modes
- cavity FSR scan
- PDH signal

Live examples

- calculate cavity eigen modes
- cavity FSR scan
- PDH signal



III
Best practices

« So, I think we have some issues in our code »

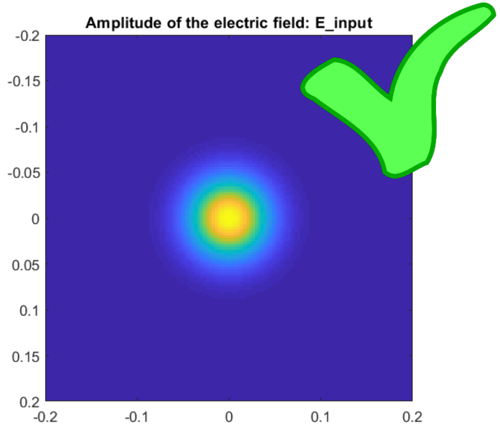
- a complex system requires complex simulations
- think before coding, understand the advantages/benefits of various packages
- ABCD matrix are underrated
- read the documentation, look at the examples

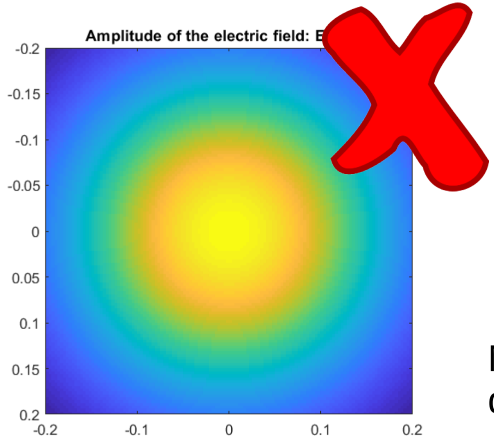
Choose the right grid size

```
% Define the grid for the simulation: 256 X 256, 15 cm X 15 cm
G1 = Grid(256,0.15);
```

Lateral size for the simulation

- the grid must contains your all your beams and so be larger





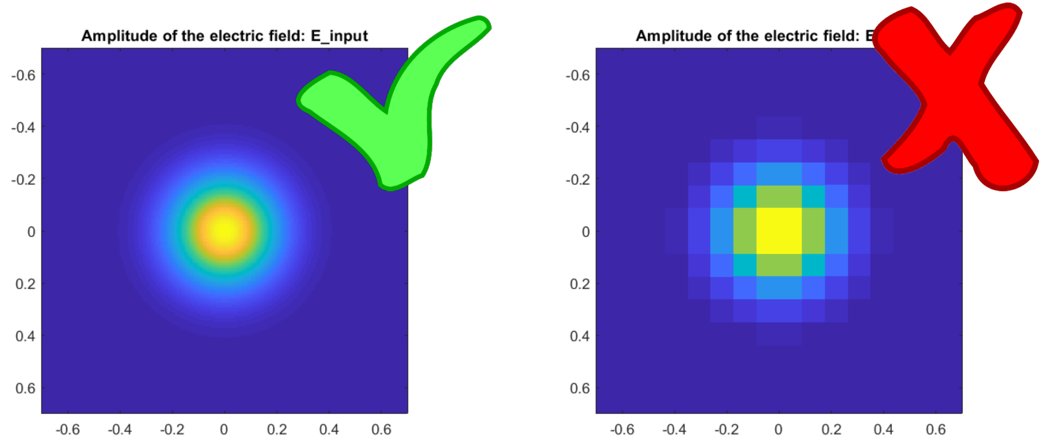
Must be valid at all the distance in your simulations

- if clipping is relevant, the grid must be larger than the optic apertures

Choose the right number of points

```
% Define the grid for the simulation: 256 X 256, 15 cm X 15 cm
G1 = Grid(256,0.15);
```

Electric field defined as 256×256 matrix



- should be cautious in case of very focused beams
- overall simulations pretty tolerant to low resolution

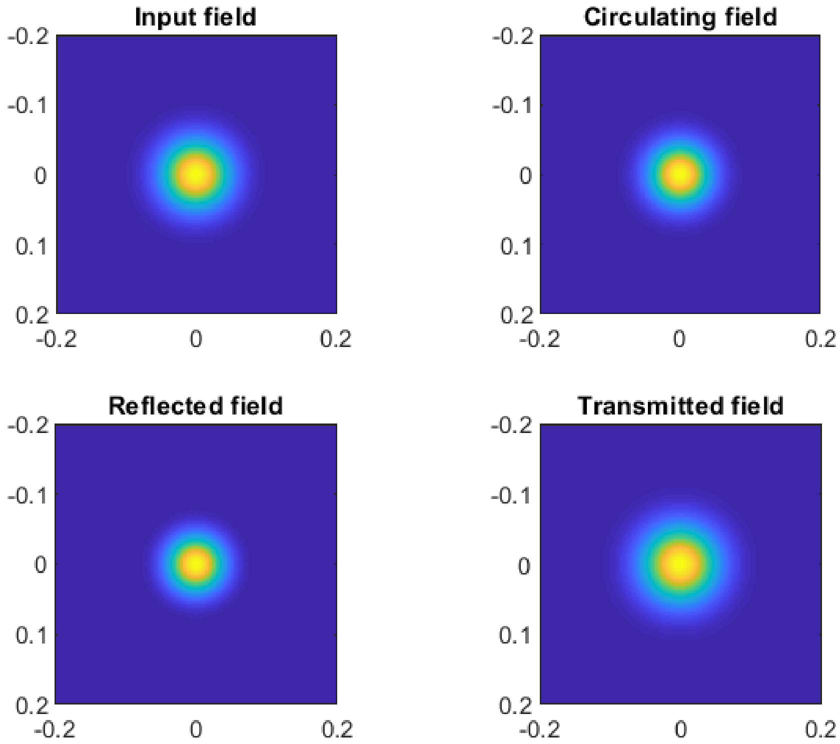
Choose the right number of points

Example: impact of the number of points

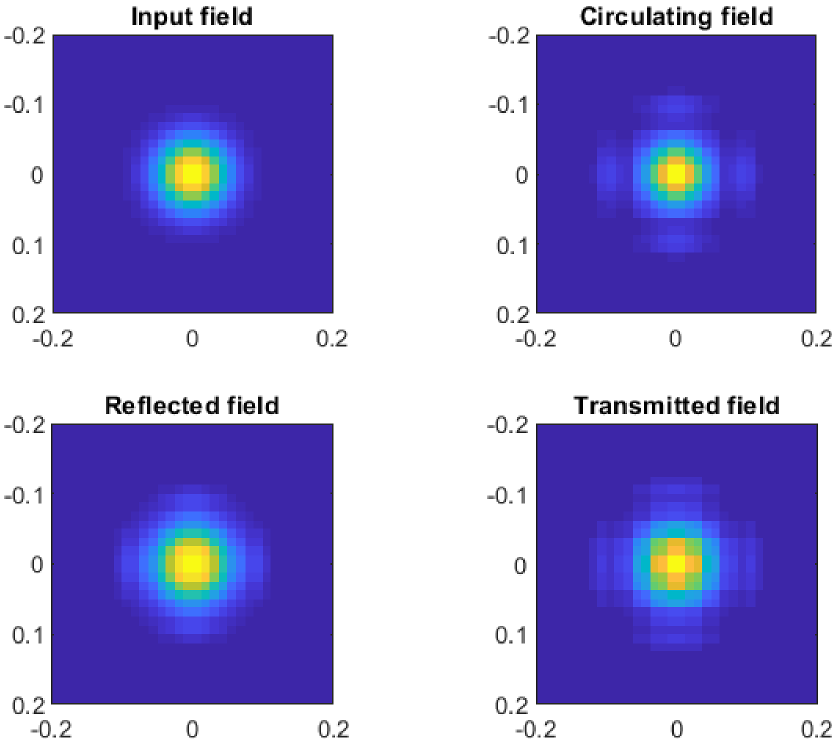
Number of points	Circulating power [W]	Round Trip Loss [ppm]	Running time [s]
512 × 512	277.9	58.2	14.3
256 × 256	277.9	58.3	4.5
128 × 128	277.9	58.7	1.8
64 × 64	278.2	51.3	1.5
32 × 32	10.2	57000	1.5

Choose the right number of points

256 × 256

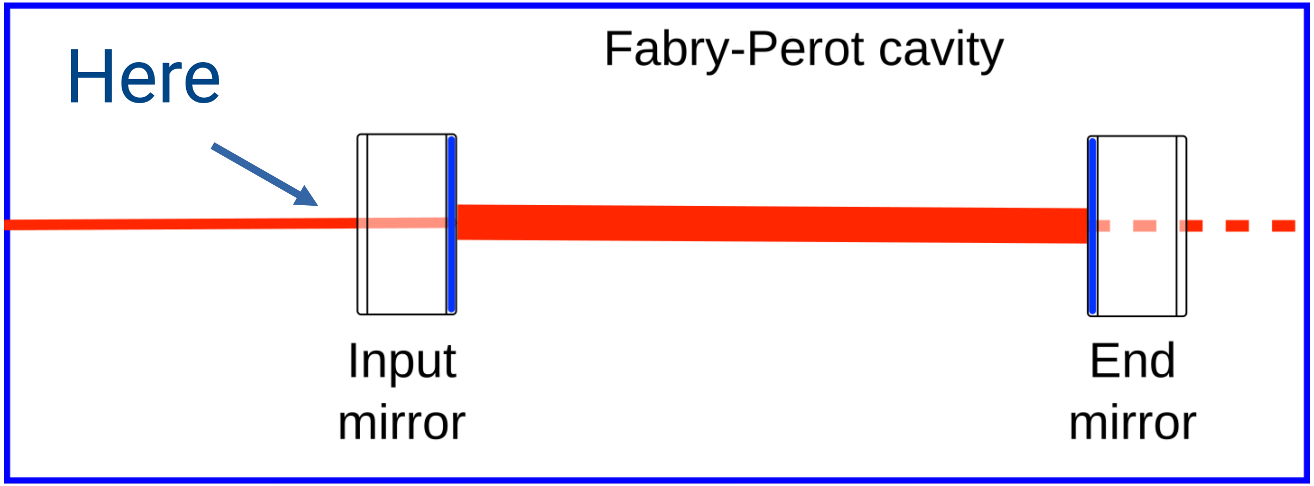


32 × 32



Mode matching

OSCAR input beam is defined outside of the cavity



If the input mirror is curved it will act a lens which must be taken into account.

Mode matching

The optimal mode matched beam can be calculated directly by OSCAR:

```
>> Arm_cavity.Check_Stability
----------------- For the input mirror -----------------
RofC fitted (m): 1420
Tilt horizontal (nrad): 8.19545e-09
Tilt vertical (nrad): 8.20032e-09
Flatness RMS (nm): 6.89495e-05,

----------------- For the end mirror -----------------
RofC fitted (m): 1733
Tilt horizontal (nrad): -1.67918e-09
Tilt vertical (nrad): -1.67952e-09
Flatness RMS (nm): 7.42283e-05,

The g-factor of the cavity is: 0.81348
The beam waist size in the cavity: 0.0106786
distance with the ITM: 1335.09
Consecutive optical mode separation [1/FSR]: 0.14215

Beam radius on ITM: 0.0436694
Beam radius on ETM: 0.0538732

 Cavity finesse: 445.491
 Cavity gain: 283.51
Mode matched input beam parameters:
Beam radius [m]: 0.0436694          Wavefront curvature [m]: -979.31
```

Parameters of the input beam

III
New features in version 3.30 (and 3.31)

Adding GPU calculation capability

Work done by Nils Melchert.

For the details procedure:

GPU implementation of FSR simulations: Performance improvements and limitations

Nils Melchert[a], Jérôme Degallaix[b], Lennart Hinz[a], and Eduard Reithmeier[a]

[a]Leibniz Universität Hannover, Institute of Measurement and Automatic Control, An der Universität 1, Garbsen, Germany
[b]Laboratoire des Matériaux Avancés - IP2I, CNRS/IN2P3, Université de Lyon, 69100 Villeurbanne, France

https://doi.org/10.1117/12.2633434

Example of speed increase on FSR scan

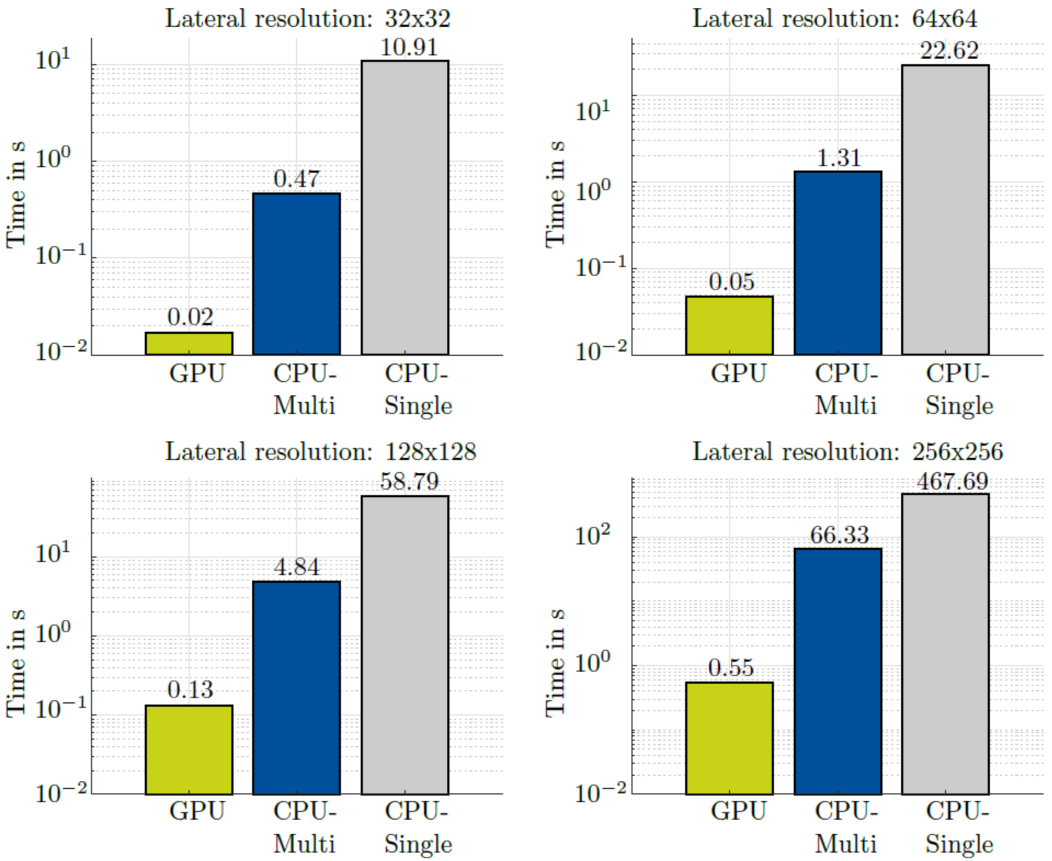


Figure 3: Evaluation of the simulation runtime depending on the lateral resolution of the E-field

Optimal mode matching directly

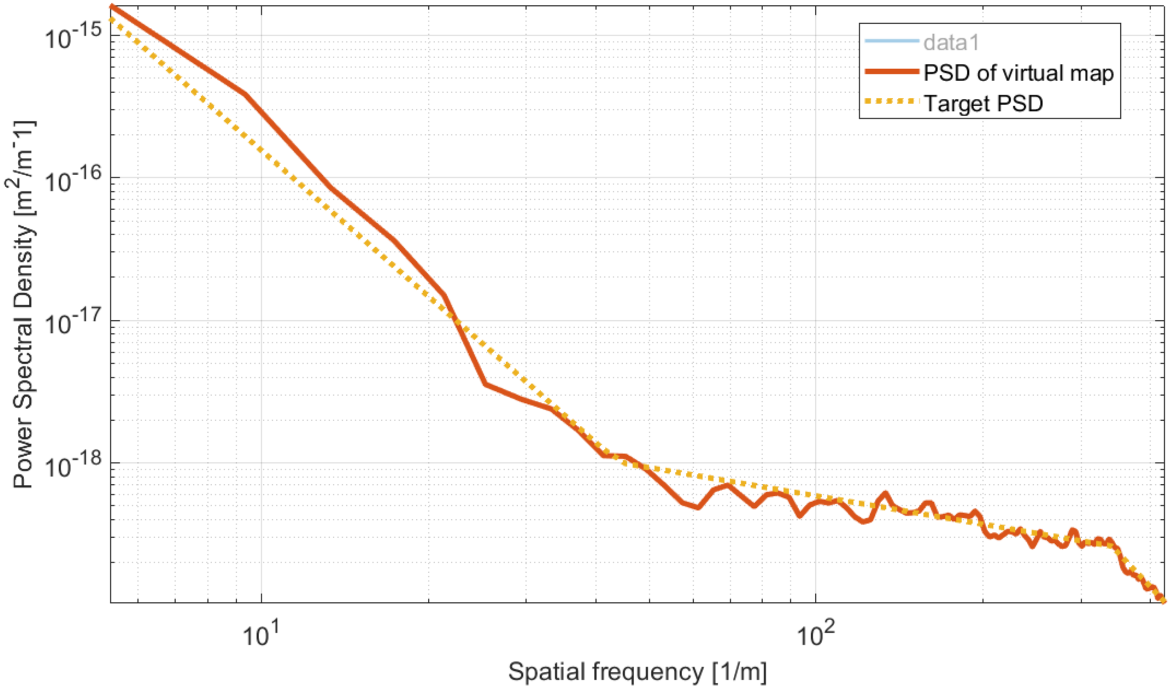
New option to define the beam:

```
E_input = E_Field(G1,'Optimal_MM',true);
```

So the input beam parameters are defined after the cavity is declared

More rigourous surface PSD calculation

Re-wrote the code to calculate the PSD and create virtual maps from a given PSD.



III
Some final words

What to expect next ?

- create SB signals in the cavity
- stable recycling cavities
- adding birefringence and polarisation effects

How to contribute ?

- every new features comes from a need
- share the functions you developed
- can also create a new branch from git : https://github.com/Jerome-LMA/oscar

A new logbook to share news / tips

For this workshop, a new logbook about OSCAR:
https://logbooks.ifosim.org/oscar/