GRAY

EC quasi-optical ray-tracing code for ECRH&ECCD calculations in tokamaks

***L.Figini**, D. Farina*
IFP-CNR, EURATOM-ENEA-CNR Assoc.
Milano,Italy

Physics model

Propagation of EC Gaussian beams is dealt with in the framework of the **compex eikonal** approximation

- Gaussian beam: $\boxed{E(\mathbf{x}) = E_0(\mathbf{x}) \exp\left[-\left(\frac{\xi_w^2}{w_\xi^2}+\frac{\eta_w^2}{w_\eta^2}\right)\right] \exp\left[-ik_0\left(\bar{z} + \frac{\xi_R^2}{2R_\xi}+\frac{\eta_R^2}{2R_\eta}\right)\right]}$
- wave equation solution of the form: $\boxed{E(\mathbf{x},t) = \mathbf{e}(\mathbf{x})E_0(\mathbf{x})\exp\left[-ik_0S(\mathbf{x})+i\omega t\right]}$
  $S(x)$ complex eikonal function $S = S_R + i\,S_I$ $\boxed{\bar{\mathbf{k}} = \mathbf{k}+i\mathbf{k}' = k_0\left(\nabla S_R + i\nabla S_I\right)}$

  beam propagation beam shape
- λ wavelength, w beam width, L system dimension: $\lambda << w << L$
- asympotic expansion wave equation -> QO cold dispersion relation has additional terms **(grad S_I)** with respect to geometric optics (GO)
  - $\Rightarrow$ coupled ray equations describing *diffraction effects*

Absorption coefficient computed solving either the **fully** of the **weakly relativistic** EC dispersion relation

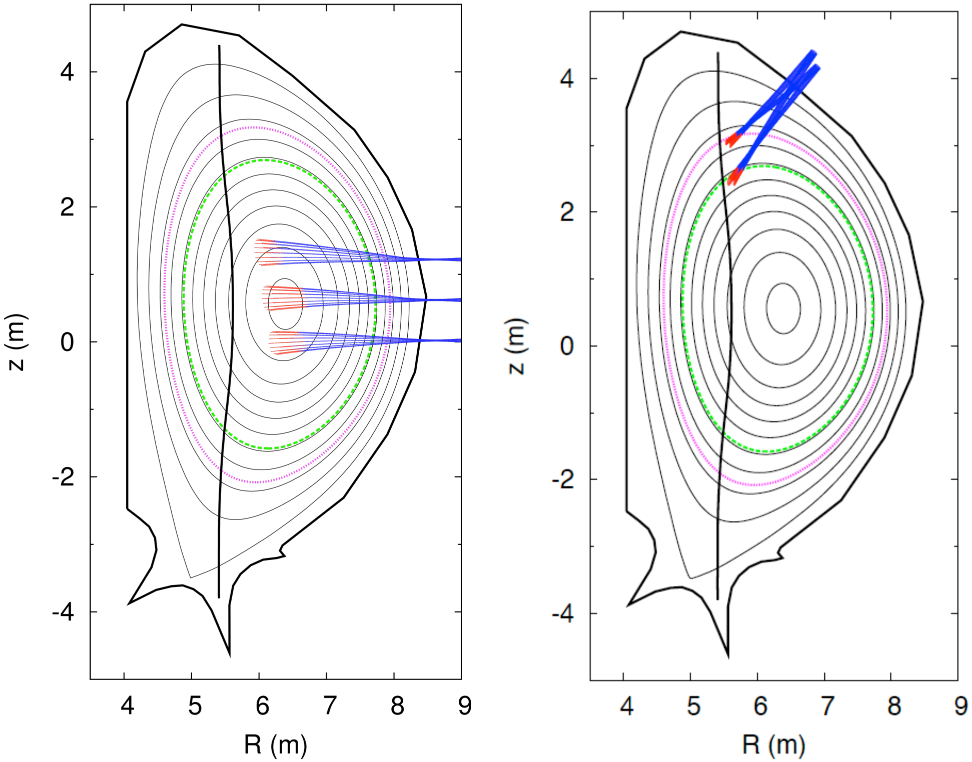
CD efficiency computed using the adjoint method (***Cohen** model or **momentum conservation** model from Marushenko, not yet in ITM version)*

Standalone version

Fortran 77 code. Input files:

- Magnetic equilibrium description (G EQDSK format)
- n_e, T_e, Z_{eff} profiles
- Code parameters (including initial conditions for beamtracing) and options

ITER

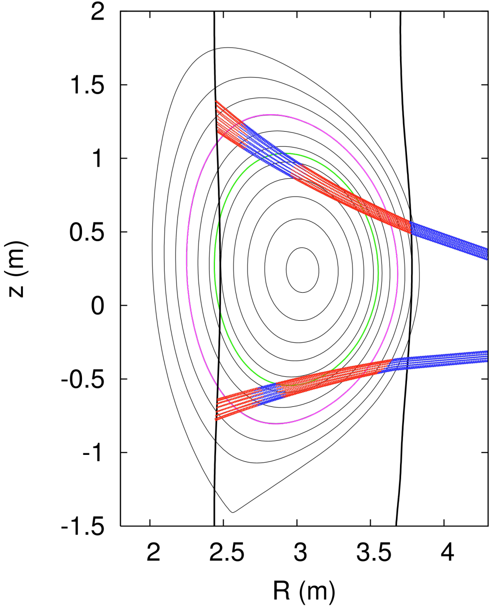


EC Eq. launcher

EC Upper launcher

JET

E4J project



D. Farina, L. Figini, NF **50** (2010) 095007
G. Ramponi, et al., NF **48** (2008) 054012

GRAY on the ITM Gateway

On the ITM Gateway:

- subroutine communicating with the calling routine through CPOs (4.08a)
- Input CPOs:
  - **equilibrium**
  - **coreprof**
  - **antennas**
- Output CPO:
  - **waves**
- code **parameters** stored in XML format, and passed to GRAY in a `codeparam` structure
- **Kepler actor** is now available

```
interface
  subroutine gray(equilibrium,coreprof,antenna,wave,codeparam)
  use euitm_schemas

  type(type_equilibrium), pointer :: equilibrium(:)
  type(type_coreprof), pointer :: coreprof(:)
  type(type_antennas), pointer :: antenna(:)
  type(type_waves), pointer :: wave(:)
  type(type_param) :: codeparam
  end subroutine gray
end interface
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="./input_gray.xsl" charset="ISO-8859-
<parameters>
  <code_specific>
    <i_warm> 2 </i_warm>  <!-- 0=no absorption, 1=weakly rel, 2=fully rel w
asympt expansion, 3=fully rel w/ num integration -->
    <i_larm> 5 </i_larm>  <!-- order of larmor expansion -->
    <i_eccd> 1 </i_eccd>  <!-- 0=no, 1=yes ECCD calculation -->
    <i_grad> 0 </i_grad>  <!-- 0=raytracing, 1=gaussian beam -->
    <n_rad_ray> 11  </n_rad_ray>  <!-- radial # of rays -->
    <n_ang_ray> 16 </n_ang_ray>  <!-- angular # of rays -->
    <rho_max> 1.0 </rho_max>  <!-- beam truncation radius -->
    <ds> 0.1 </ds>  <!-- beamtracing step (cm) -->
    <n_steps> 8000 </n_steps>  <!-- max # of beamtracing steps -->
  </code_specific>
  <output_control>
    <i_rho> 1 </i_rho>  <!-- profiles w/ constant 0=dpsi, 1=drhop -->
    <n_proj_ec> 501 </n_proj_ec>  <!-- points in EC profiles -->
    <i_step_proj> 100 </i_step_proj>  <!-- beam shape data density -->
    <i_step_ray> 5 </i_step_ray>  <!-- ray data density -->
  </output_control>
  <data_adjust>
    <i_x_pos> 0 </i_x_pos>  <!-- -1=lower, 1=upper, 0=no X-point -->
    <psi_ne_bnd> 1.05 </psi_ne_bnd> <!-- density boundary -->
    <spline_psi> 0.0001 </spline_psi> <!-- psi smoothing coeff -->
  </data_adjust>
</parameters>
```

will be removed if equilibrium%eqgeometry%boundarytype and equilibrium%eqgeometry%xpts are available

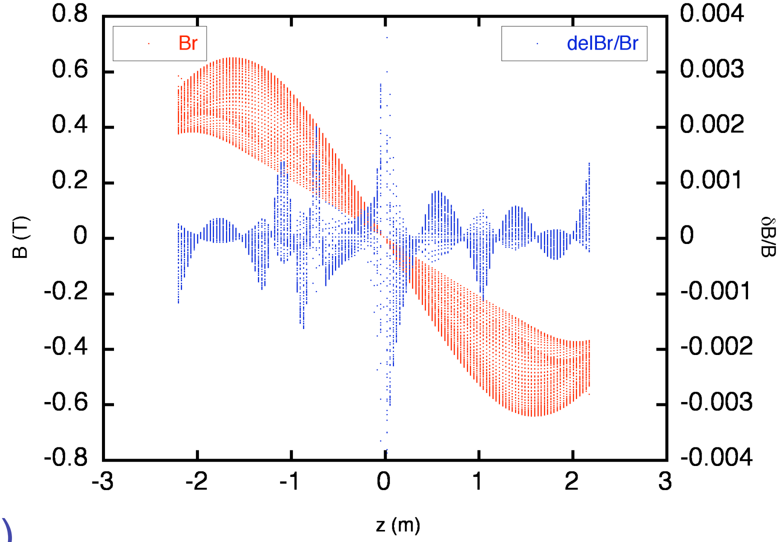
Equilibrium CPO:
B components

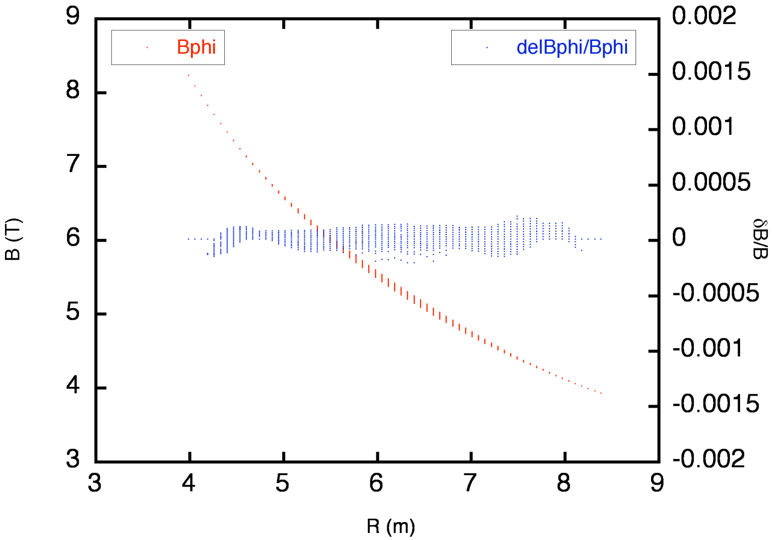
B components are **computed internally** from F_{dia} and ψ derivatives:

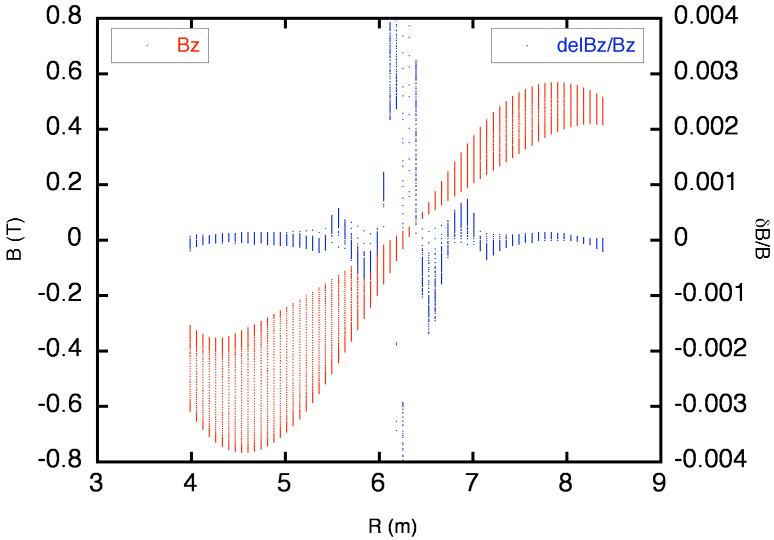
$$B_\phi = \frac{F_{\text{dia}}}{R}, \quad B_R = -\frac{1}{R}\frac{\mathrm{d}\psi}{\mathrm{d}z}, \quad B_z = \frac{1}{R}\frac{\mathrm{d}\psi}{\mathrm{d}R}$$

Dierckx routines used for F_{dia} and ψ interp., and for ψ derivatives calculation

Test on shot 5/67 (Helena + equilibrium_augmenter)







Results coincide with values in `equilibrium%profiles_2d%bphi, %br, %bz`

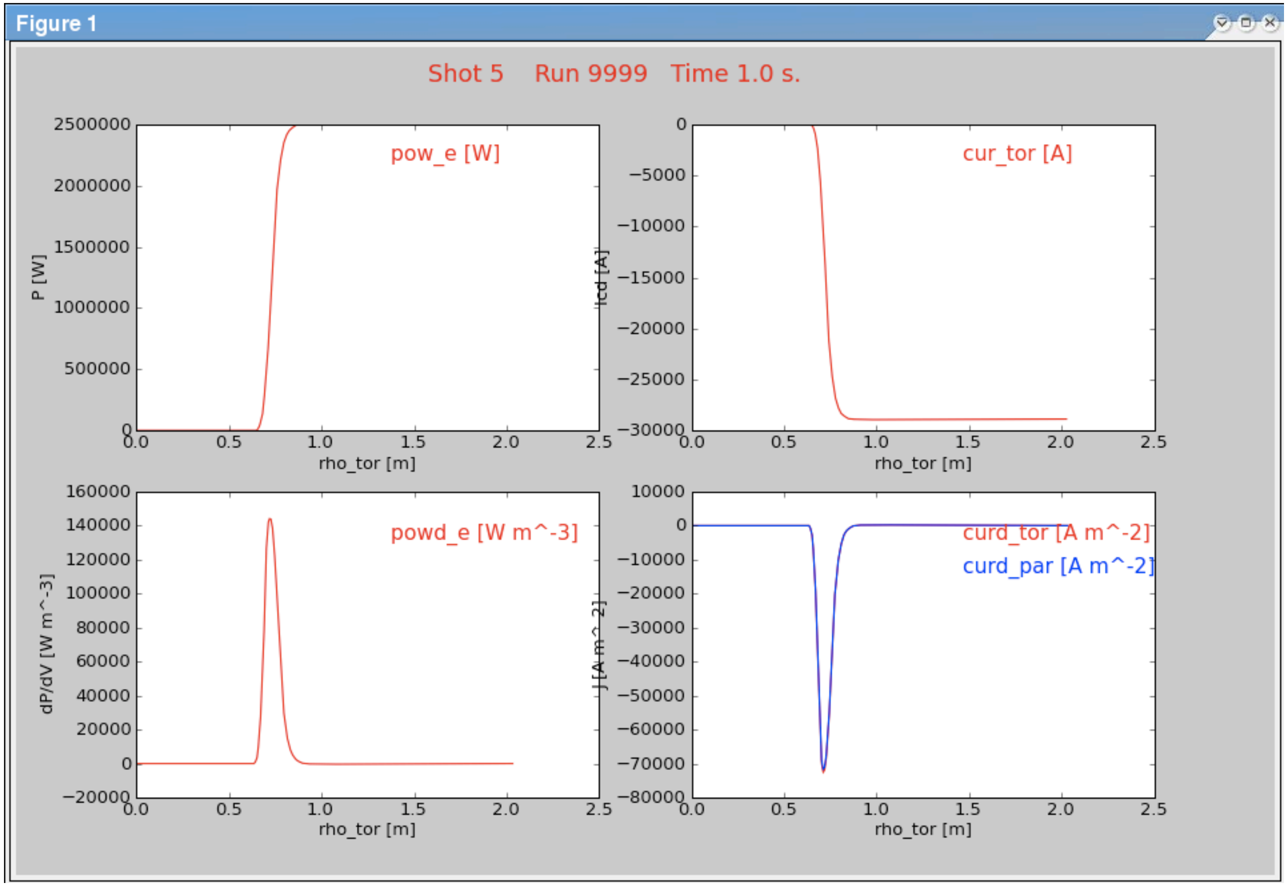
Equilibrium CPO:
extrapolation

- 2 types of mesh for `equilibrium%profiles_2d`
    - **Rectangular** coordinates (R, z).
      Usually it extends outside LCFS, favourite choice for ray-tracing.
    - **Straight field lines** coordinates (ψ, χ).
      More natural for fixed boundary equilibrium solvers, but it does not provide information outside LCFS.

- GRAY needs a **smooth transition** from vacuum to plasma volume
    - ψ values is needed also **outside LCFS**
    - The type of mesh is guessed from psi values along `grid%dim1` and emptiness of `profiles_2d%position%r`, `%z`
    - If straight field coordinates, grid points are considered as **scattered**, and an **extrapolating spline** is computed (Dierckx)
        - no constraints, same approach as equilibrium_augmenter?
    - Tested successfully on cases 5/64 (ψ, χ), 5/67 (R, z)

Waves CPO: current status

- Power density (dP/dV) and EC driven current profiles (J_{tor}, $J_{||}$) written in `waves` CPO (together with the corresponding integrated quantities P, I_{tor})

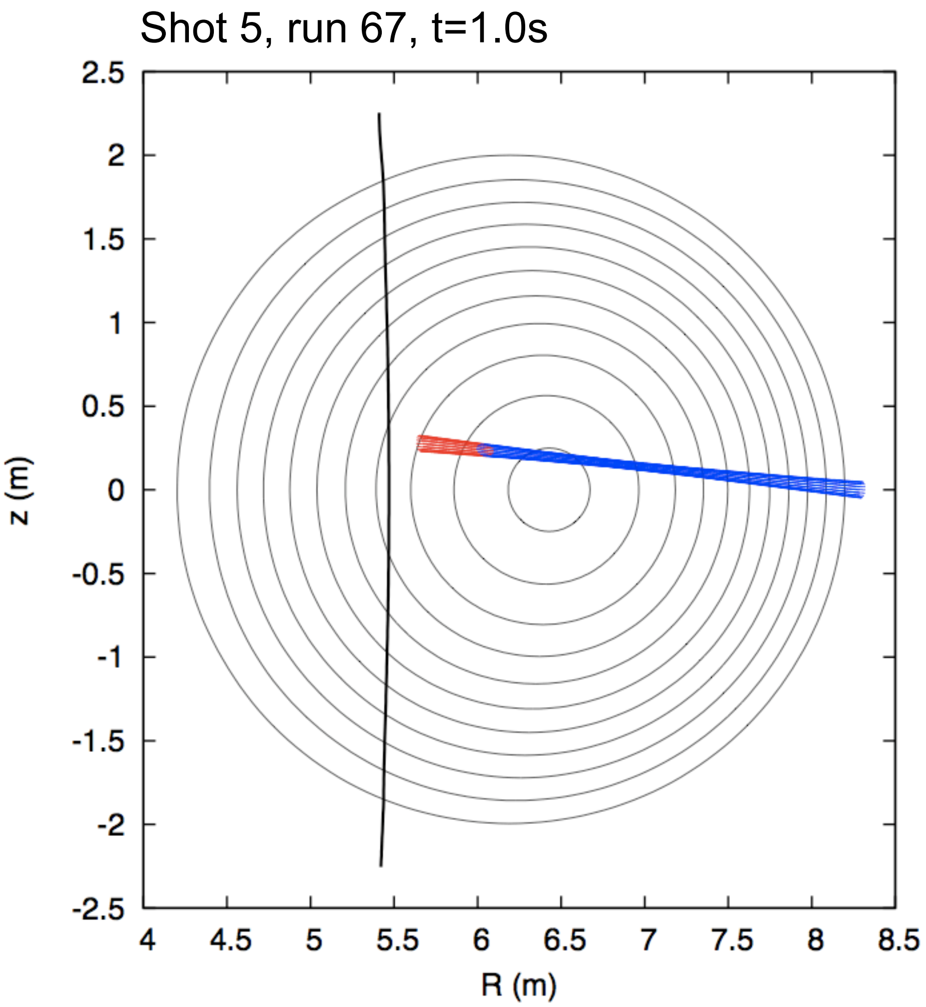


- A module should be developed to write `coresource` CPO from `waves`

Waves CPO: still to be done

Detailed beamtracing data still written on disk only

- **Beam trajectory**
- Evolution along the beam of
  - **Power**
  - **Wave vector**
  - **Polarization**



Antennas CPO

An `antennas` CPO is needed to provide the **initial conditions** for the beamtracing

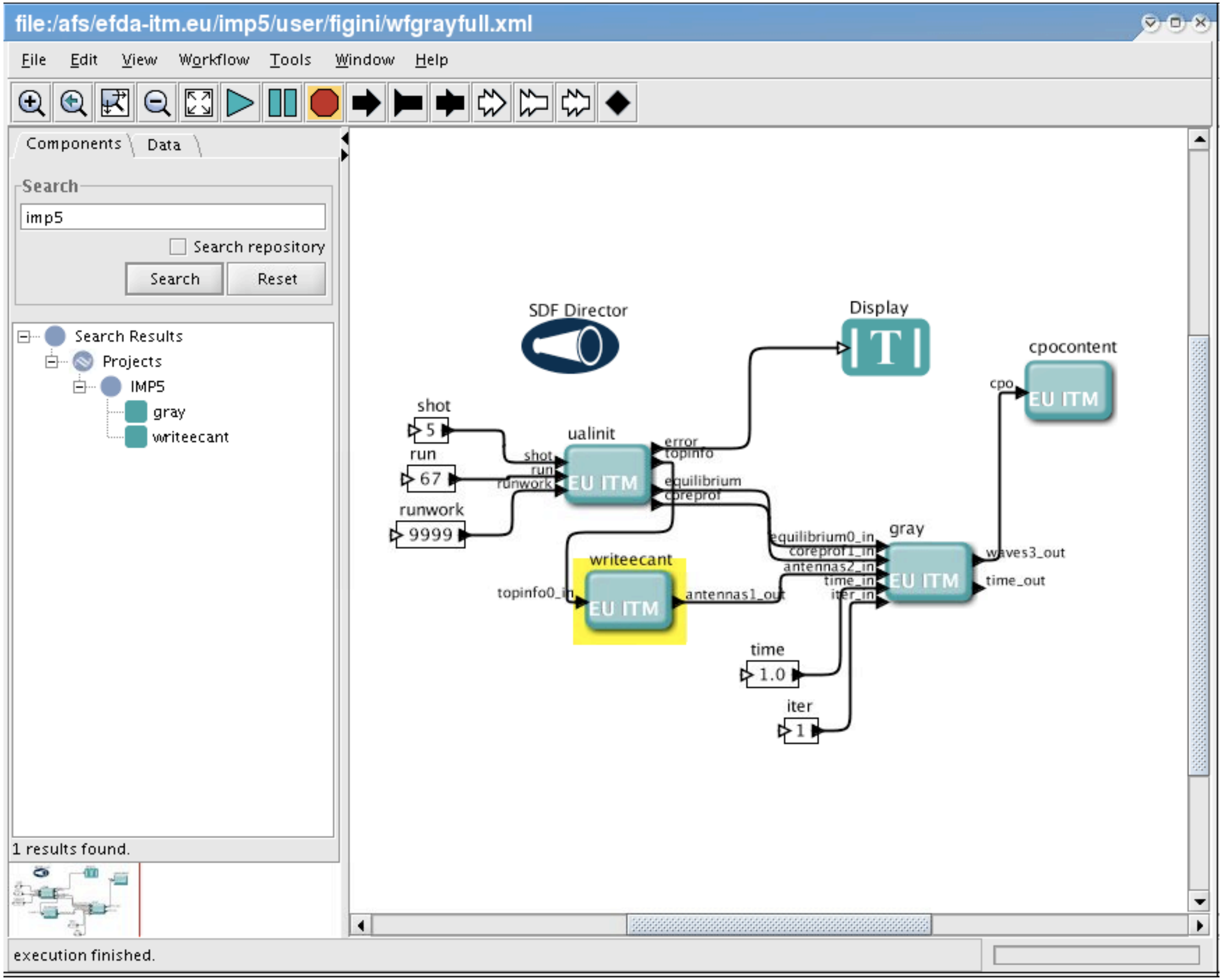
A complete case with an `antennas` CPO is not yet available

```
! initial wave power (MW)
  p0mw = antenna%antenna_ec%power%value(iant)*1.e-6_dp
! wave frequency (Hz)
  fhz = antenna%antenna_ec%frequency(iant)
! wave mode: sox=-1/+1 -> OM/XM
  sox = -dble(antenna%antenna_ec%mode(iant))
! beam shape data
  wxt  = antenna%antenna_ec%beam%spot%waist(iant,1)*1.e2_dp
  wyt  = antenna%antenna_ec%beam%spot%waist(iant,2)*1.e2_dp
  phiw = antenna%antenna_ec%beam%spot%angle(iant)/cvdr
  rcixt = antenna%antenna_ec%beam%phaseellipse%invcurvrad(iant,1)*1.
  rciyt = antenna%antenna_ec%beam%phaseellipse%invcurvrad(iant,2)*1.
  phir  = antenna%antenna_ec%beam%phaseellipse%angle(iant)/cvdr
! antenna position and launching angles
  phi0 = antenna%antenna_ec%position%phi(iant)
  r00 = antenna%antenna_ec%position%r(iant)*1.e2_dp
  x00 = r00*cos(phi0)
  y00 = r00*sin(phi0)
  z00 = antenna%antenna_ec%position%z(iant)*1.e2_dp
  alfac = antenna%antenna_ec%launchangles%alpha(iant)/cvdr
  betac = antenna%antenna_ec%launchangles%beta(iant)/cvdr
```

- The module **write_ec_ant** has been built to fill the `antenna_ec` part of an `antenna` CPO
- Values used to fill the CPO are read from XML file
- A **Kepler actor** has been generated (no CPOs in, `antennas` CPO out)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="./input_gray.xsl" charset="ISO-
<parameters>
  <mirror>
    <alpha> -5 </alpha>  <!-- poloidal angle (deg) -->
    <beta> 20 </beta>  <!-- toroidal angle (deg) -->
    <x0> 830 </x0>
    <y0> 0 </y0>
    <z0> 34.5 </z0>  <!-- mirror coordinates (deg) -->
</mirror>
  <beam>
    <frequency> 170 </frequency>  <!-- (GHz) -->
    <power> 2.5 </power>  <!-- (MW) -->
    <i_mode> 1 </i_mode>  <!-- 1=OM, 2=XM -->
    <w0_x> 2 </w0_x>
    <w0_y> 2 </w0_y>  <!-- waist dimensions (cm) -->
    <dist_w0_x> 160 </dist_w0_x>
    <dist_w0_y> 160 </dist_w0_y>  <!-- mirror-waist distance (cm), >0
plasma -->
    <spot_rot> 0 </spot_rot>  <!-- spot ellipse rotation (deg) -->
    <phase_rot> 0 </phase_rot>  <!-- phase ellipse rotation (deg) -->
  </beam>
</parameters>
```

Kepler workflow



file:/afs/efda-itm.eu/imp5/user/figini/wfgrayfull.xml

Search: imp5

☐ Search repository

1 results found.

execution finished.

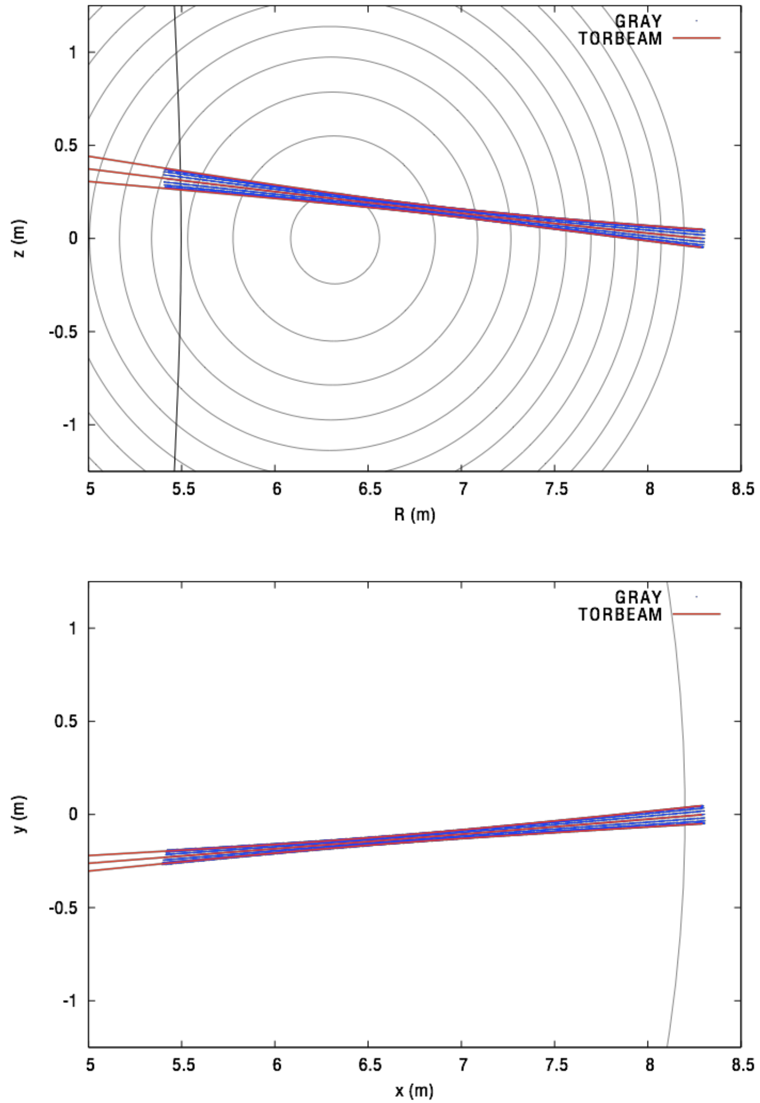
Benchmarking

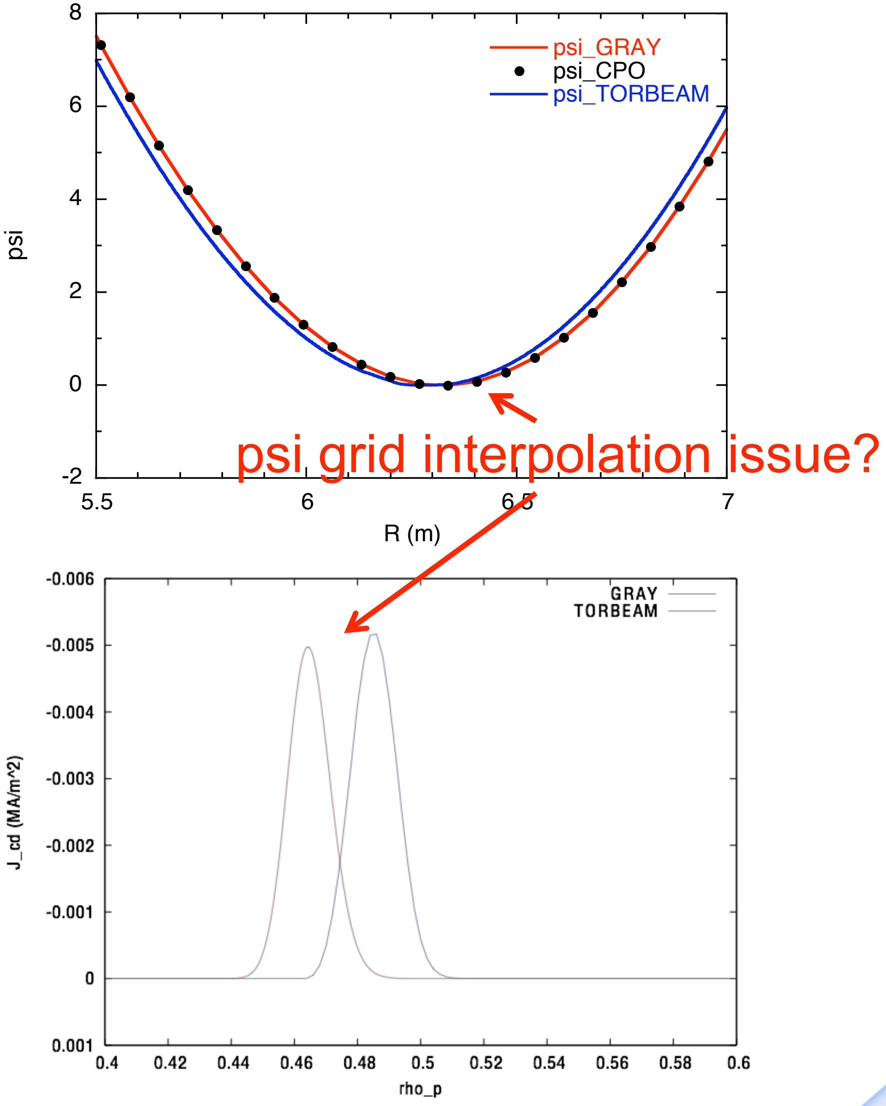
- ITM version of GRAY has been benchmarked against the original F77 version
  - Test case: ITER standard scenario 2
    1. Data read from file and passed to the F77 version
    2. Data read from file, used to fill the corresponding CPO structures and passed to the ITM version
  - Results match

- Benchmark has started against TORBEAM beamtracing code
  - Test case: `test` machine, shot 5, run 67
  - Work in progress

GRAY vs TORBEAM:
preliminary results

Shot 5, run 67, t=0.0s





psi grid interpolation issue?

Future work

- Fill **`waves` CPO** with all the available data (→ `coresource` CPO)
- Replace physical constants values with `ITM_constants` module
- Use **flux averaged quantities** from `equilibrium` CPO, when available (no more need of internal computation)

- Test code robustness
  - try other shots and machines
- Continue **benchmarking** with TORBEAM
  - investigate profiles displacement
  - compare approaches to vacuum-plasma transition
- Benchmark with TORAY-FOM and TRAVIS when ready
- Include momentum conservation method (from Maruschenko) for CD computation also in ITM version

- Use SVN for revision control
- Write code **documentation**

Physics model

Propagation of EC Gaussian beams is dealt with in the framework of the **compex eikonal** approximation

- Gaussian beam: $\boxed{E(\mathbf{x}) = E_0(\mathbf{x}) \exp\left[-\left(\frac{\xi_w^2}{w_\xi^2} + \frac{\eta_w^2}{w_\eta^2}\right)\right] \exp\left[-ik_0\left(\bar{z} + \frac{\xi_R^2}{2R_\xi} + \frac{\eta_R^2}{2R_\eta}\right)\right]}$
- wave equation solution of the form: $\boxed{E(\mathbf{x},t) = \mathbf{e}(\mathbf{x})E_0(\mathbf{x}) \exp\left[-ik_0 S(\mathbf{x}) + i\omega t\right]}$

  $S(x)$ complex eikonal function $S = S_R + i\, S_I$ $\boxed{\bar{\mathbf{k}} = \mathbf{k} + i\mathbf{k}' = k_0\left(\nabla S_R + i\nabla S_I\right)}$

  beam propagation beam shape
- λ wavelength, w beam width, L system dimension: $\lambda << w << L$
- *asympotic expansion wave equation -> dispersion relation:*

$$\boxed{D(\mathbf{x},\bar{\mathbf{k}},\omega) = D(\mathbf{x},\mathbf{k},\omega) + i\mathbf{k}' \cdot \frac{\partial D}{\partial \mathbf{k}} - \frac{1}{2}\mathbf{k}'\mathbf{k}' : \frac{\partial^2 D}{\partial \mathbf{k}\partial \mathbf{k}} = 0}$$

cold dispersion relation

$$\boxed{D(\mathbf{x},\mathbf{k},\omega) = N^2 - N_s^2(\mathbf{x},N_{\|},\omega) = 0}$$

N_s : local cold refractive index OM/XM solution of the Appleton-Hartree dispersion relation

Quasi-optical ray equations

D. Farina, Fusion Sci. Techn. **52** (2007) 154.

Real and **imaginary** part of the QO dispersion relation:

$$\boxed{D_R(\mathbf{x},\mathbf{k},\omega) = N^2 - N_s^2(\mathbf{x},N_\parallel,\omega) - |\nabla S_I|^2 + \frac{1}{2}\nabla S_I \nabla S_I : \frac{\partial^2 N_s^2}{\partial \mathbf{N}\partial \mathbf{N}} = 0}$$

$$\boxed{D_I(\mathbf{x},\mathbf{k},\omega) = \nabla S_I \cdot \frac{\partial D}{\partial \mathbf{k}} = 0}$$

additional terms with respect to geometric optics (GO) approximation:

⇒ *diffraction effects*

QO ray equations *at dominant order in δ*

$$\boxed{\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} &= \left.\frac{\partial D_R/\partial \mathbf{k}}{|\partial D_R/\partial \mathbf{k}|}\right|_{D_R=0} \\ \frac{\mathrm{d}\mathbf{k}}{\mathrm{d}s} &= -\left.\frac{\partial D_R/\partial \mathbf{x}}{|\partial D_R/\partial \mathbf{k}|}\right|_{D_R=0} \\ \frac{\partial D_R}{\partial \mathbf{k}} &\cdot \nabla S_I = 0 \end{aligned}}$$

formally equal to GO ray eqs.

with D_R depending also on ∇S_I

⇒ ray eqs are coupled together

partial differential eq.

coupled to ray eqs.

ECRH&CD

D. Farina, Fusion Sci. Techn. **52** (2007) 154.

- **EC power evolution along the ray**

$$\frac{\mathrm{d}P}{\mathrm{d}s} = -\alpha P, \quad \alpha = -2 \left.\frac{\mathrm{Im}\left(N^2_{\perp w}\right)}{|\partial D_R/\partial \mathbf{k}|}\right|_{D_R=0}$$

absorption coefficient α computed solving either the **fully** or the **weakly relativistic** EC dispersion relation

- **EC driven current along the ray**

$$\frac{\mathrm{d}I_{\mathrm{cd}}}{\mathrm{d}s} = -\mathcal{R}(s)\frac{1}{2\pi}\langle\frac{B_\phi}{R}\rangle\frac{1}{\langle B\rangle}\frac{\mathrm{d}P}{\mathrm{d}s}$$

Current drive efficiency computed using the adjoint method

$$\mathcal{R} \equiv \frac{\langle J_\parallel\rangle}{\langle p_d\rangle} = -\frac{e}{mc^2}\frac{B}{\langle\mathcal{H}B\rangle}\frac{\langle\int \mathrm{d}\mathbf{u}\,\mathbf{S}_w\cdot\frac{\partial\chi}{\partial\mathbf{u}}\rangle}{\langle\int \mathrm{d}\mathbf{u}\,\frac{\mathbf{u}}{\gamma}\cdot\mathbf{S}_w\rangle}$$

X reponse function: Cohen model & momentum conservation model (from Marushenko, not yet in the ITM version)