

FOCUS documentation

(version 1.2)

Introduction

“Fast Monte Carlo approach to Coherence of Undulator Sources” (FOCUS) is a GPU-based computer code for calculation of the transverse coherence properties of undulator X-ray radiation generated by ultra-relativistic electrons in an undulator device.

The code relies on scaled dimensionless quantities and analytic expressions for the emitted electric fields obtained with a Fourier optics approach to classical relativistic electrodynamics. In particular, the slowly varying amplitude of the electric field in the space-frequency domain is obtained by solving paraxial Maxwell equations in free space with the help of a parabolic Green’s function, in addition with the resonance approximation. Disregarding optical surface imperfections, the method can be adapted to imaging geometries as well, simply by looking at conjugate planes with the proper magnification factor.

FOCUS is written in C++, accelerated with CUDA to harness the compute capabilities of modern GPUs. The main aim of the code is to enable fast and accurate evaluation of the transverse coherence properties of undulator X-ray radiation as a function of the electron beam parameters.

FOCUS is freely available from the official homepage at the website of the Instrumental Optics Group of the Physics Department at Università degli Studi di Milano¹. The source code is also freely available from the GitHub page of the corresponding author².

¹ <https://instrumentaloptics.fisica.unimi.it/focus/>

² <https://github.com/MirkoSiano/FOCUS>

Prerequisites (required dependencies)

FOCUS natively runs on GPUs via CUDA, therefore a platform with a CUDA-capable NVIDIA graphic card is required. The CUDA Toolkit from NVIDIA must also be installed. Detailed installation instructions for the CUDA Toolkit on Windows platforms can be found on the official NVIDIA website³. For Linux users, it is preferred to refer to the official documentation from the package manager of the installed distribution.

³ <https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html>

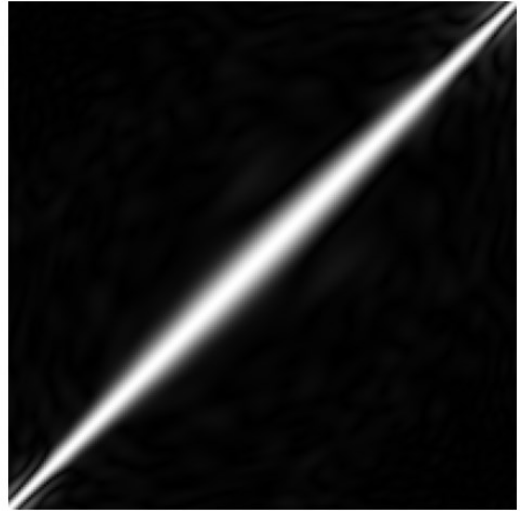
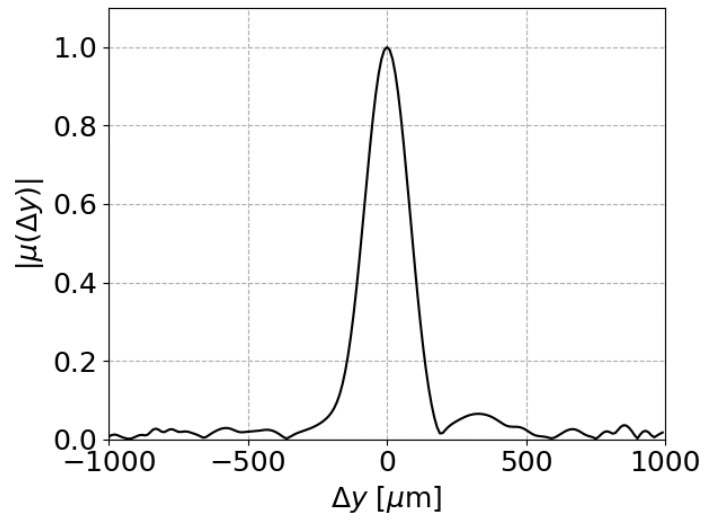
Running pre-compiled executables (Windows only)

Pre-compiled executables are provided on the FOCUS homepage as a zipped archive. The content of the archive, with a brief description of each item, is listed in the following table.

<i>FOCUS_Example01.exe</i>	It computes the 1D profiles (either horizontal or vertical) of the Spectral Degree of Coherence, sampling the electron beam phase space by an internal method relying on random number generators. It can be run directly from a command prompt by typing: <i>"FOCUS_Example01.exe"</i>
<i>FOCUS_Example02.exe</i>	It computes the 1D profiles (either horizontal or vertical) of the Spectral Degree of Coherence, reading the electron beam phase space from a user-supplied file. It can be run directly from a command prompt by typing: <i>"FOCUS_Example02.exe name_of_phase_space_file.txt"</i>
<i>FOCUS_Example03.exe</i>	It computes the 2D maps of the Spectral Degree of Coherence, sampling the electron beam phase space by an internal method relying on random number generators. It can be run directly from a command prompt by typing: <i>"FOCUS_Example03.exe"</i>
<i>FOCUS_Example04.exe</i>	It computes the 2D maps of the Spectral Degree of Coherence, reading the electron beam phase space from a user-supplied file. It can be run directly from a command prompt by typing: <i>"FOCUS_Example04.exe name_of_phase_space_file.txt"</i>
<i>FOCUS_Example05.exe</i>	It computes the transverse coherence length along the specified axis, sampling the electron beam phase space by an internal method relying on random number generators. It can be run directly from a command prompt by typing: <i>"FOCUS_Example05.exe"</i>
<i>__configElectronBeam.txt</i>	Configuration file for the electron beam (see dedicated section for details).
<i>__configUndulator.txt</i>	Configuration file for the undulator device (see dedicated section for details).
<i>__configObserver.txt</i>	Configuration file for the observer (see dedicated section for details).
<i>phaseSpace_example_1e4_part.txt</i>	Example of user-supplied phase space file, with 10^4 electrons.

Note on the external phase space file formatting. *FOCUS* assumes that the external phase space file supplied by the user to *FOCUS_Example02.exe* and *FOCUS_Example04.exe* is formatted as different rows with 5 entries. Each row corresponds to a different electron, and the 5 entries describe the horizontal position, horizontal deflection angle, vertical position, vertical deflection angle, and relative energy deviation of the electron, respectively. Positions and angles are expressed in m and rad, respectively. Notice that an internal conversion to μm and μrad is automatically performed during the simulation.

As references for users, running *FOCUS_Example02.exe* and *FOCUS_Example04.exe* with the supplied *phaseSpace_example_1e4_part.txt* and default values for the configuration files (see dedicated section) should produce the following results:



Compile the source code (Windows and Linux)

The full source code of *FOCUS* can be found on GitHub. To compile it, follow the instructions reported below.

First, make sure to satisfy all prerequisites. Then, clone or download the *FOCUS* Git repository onto a local machine.

On Windows. Download and install the *make* utility from the *GnuWin* project homepage⁴. Then open a command prompt inside the *FOCUS* folder and type “*make all*”. This will compile the *FOCUS* class, the *FOCUS* kernels, and the source code of all examples, linking them into executable files. To run the examples, simply write the name of the corresponding executable in the command prompt.

On Linux. Install the *make* utility from the package manager of the installed distribution. Open the *makefile* provided within the *FOCUS* folder with a text editor. Change the suffix “.obj” in all object files into “.o”, and the suffix “.exe” in all executable files into “.x”, and save. Then open a terminal inside the *FOCUS* folder and type “*make all*”. This will compile the *FOCUS* class, the *FOCUS* kernels, and the source code of all examples, linking them into executable files. To run the examples, simply write the name of the corresponding executable in the terminal.

⁴ <http://gnuwin32.sourceforge.net/packages/make.htm>

How to use the configuration files

All parameters required to run a *FOCUS* simulation are contained in three configuration files:

- `__configElectronBeam.txt`, with all parameters strictly related to the electron beam
- `__configUndulator.txt`, with all parameters strictly related to the undulator device
- `__configObserver.txt`, with all parameters strictly related to the observer

Do not modify the name of these files, as they are automatically searched for by the *FOCUS* class.

Each file has two columns. The first column reports text labels for the input quantities, with an indication of units where needed. The second column is where the user can change and set the values of the corresponding input parameters.

Do not change any of the label in the first column nor the order of the parameters, as this would result in unwanted and unpredictable behaviors.

A detailed list of the parameters in each configuration file is provided hereafter, alongside with default values describing to the NCD-SWEET beamline at ALBA⁵⁶.

`__configElectronBeam.txt`

- `SIGMAX_[um]`: horizontal beam size (rms) at the undulator center, in μm . Default values is 130 μm .
- `SIGMAXP_[urad]`: horizontal beam divergence (rms) at the undulator center, in μrad . Default value is 48 μrad .
- `SIGMAY_[um]`: vertical beam size (rms) at the undulator center, in μm . Default values is 6 μm .
- `SIGMAYP_[urad]`: vertical beam divergence (rms) at the undulator center, in μrad . Default value is 5 μrad .
- `ENERGY_SPREAD`: energy spread of the electron beam. Default value is $1.05 \cdot 10^{-3}$.
- `NMC`: number of electrons in the beam. Default value is 10^6 .

`__configUndulator.txt`

- `GAMMA`: Lorentz relativistic factor (nominal or average value) of the electron beam. Default value is roughly 6000.
- `K`: undulator strength parameters. Default value is roughly 1.56.
- `PERIOD_NUM`: undulator number of periods. Default values is 92.
- `PERIOD_LENGTH_[mm]`: undulator period length, in mm. Default value is 21.6.
- `HARMONIC`: harmonic number of the undulator radiation. Default value is 7.

`__configObserver.txt`

- `MESH_SIZE`: mesh size, in pixels. Default value is 200.
- `PIXEL_SIZE_[um]`: pixel size (i.e. mesh resolution), in μm . Default value is 10.0.
- `Z_[m]`: distance of the observation plane from the undulator center, in m. Default value is 33 m.
- `LAMBDA_OBS_[nm]`: observed radiation wavelength, in nm. Notice that it can be different from the wavelength of the undulator harmonic (detuned conditions). To ensure perfect resonance with the undulator harmonic, insert a negative value. Default value is -0.1 nm.
- `PLANE`: choose between horizontal or vertical coherence. For one-dimensional coherence profiles, only HOR and VER values are allowed. HOR will compute the transverse coherence properties along the horizontal direction, while VER will compute the transverse coherence properties along the vertical direction. For two-dimensional coherence maps, three values are allowed: HOR, VER and 2D.

⁵ <https://www.cells.es/en/beamlines/bl11-ncd>

⁶ <https://journals.aps.org/prab/pdf/10.1103/PhysRevAccelBeams.25.052801>

HOR will compute the two-dimensional horizontal coherence map as a function of x_1 and x_2 , VER will compute the two-dimensional vertical coherence map as a function of y_1 and y_2 , while 2D will compute the two-dimensional coherence map as a function of Δx and Δy . Default value is VER.

- REFERENCE_POINT: choose the reference point that is fixed. Only OBS_POINT and MID_POINT values are allowed. OBS_POINT keeps \vec{x}_1 fixed, and only \vec{x}_2 is moved. This corresponds to a Young double-pinhole experiment where the first aperture is fixed, and the second one changes position. MID_POINT keeps the midpoint $(\vec{x}_1 + \vec{x}_2)/2$ fixed, and \vec{x}_1 and \vec{x}_2 are moved symmetrically with respect to it. This corresponds to a Young double-pinhole experiment where the two apertures are moved symmetrically with respect to a reference point. For two-dimensional coherence maps with either PLANE = HOR or PLANE = VER, this quantity has no effects and is neglected. Default value is OBS_POINT.
- XO_[um]: for one-dimensional coherence profiles, horizontal coordinate of the reference point, in μm . For two-dimensional coherence maps with PLANE=HOR, this quantity is interpreted as the vertical coordinate of x_1 , while in the case of PLANE=VER it represents the horizontal coordinate of y_1 . Default value is 0 μm .
- YO_[um]: for one-dimensional coherence profiles, vertical coordinate of the reference point, in μm . For two-dimensional coherence maps with PLANE=HOR, this quantity is interpreted as the vertical coordinate of x_2 , while in the case of PLANE=VER it represents the horizontal coordinate of y_2 . Default value is 0 μm .