

# An open source GUI for collaborative cloud-based X-ray optics modeling

D.L. Bruhwiler<sup>&</sup> M. Rakitin<sup>#</sup> P. Moeller<sup>&,%</sup> R. Nagler<sup>&</sup> O. Chubar<sup>#</sup>



## Software for Optical Simulations (SOS) Workshop

4 October 2016 – Trieste

# Vision

- *develop open source software, useful for:*
  - *X-ray beamline designers & computational physicists*
    - *full access to capabilities & complexity, with minimal restrictions*
    - *flexible, publication quality graphics*
    - *a subset of them require a development environment*
  - *light source facility users*
    - *require a graphical user interface (GUI)*
    - *ease of use, intuitive*
    - *ability to understand X-ray optics of specific beamline*
- *software must be sustainable over the long-term*
  - *all components must be maintainable*
  - *easy for the X-ray optics community to contribute*

# Capabilities Matrix

Capabilities	Software	Comments
electron beam dynamics	SRW, ...	more 3D magnetic field sources and elements
synchrotron emission	SRW, ...	improved solvers
SASE FEL	GENESIS, database...	import laser pulse data (2D and 3D)
geometric optics	SHADOW, ...	interoperability with SRW
physical optics	SRW, ...	improve reliability, speed
partial coherence	SRW, SHADOW...	improve reliability, speed
X-ray optical elements	SRW, SHADOW...	increasing diversity & complexity
element imperfections	SRW, SHADOW...	benchmark different methods
material properties	database	automated access to community resources
Python scripting	SRW, SHADOW...	how much commonality?
file formats	SRW, SHADOW...	configuration, binary data – commonality?
GUI	SRW, SHADOW...	cloud computing
<u>Jupyter (IPython) notebooks</u>	SRW, ...	WPG effort at EXFEL
simulate experiments	SRW, ...	material interactions, diagnostics
optimization & design	SRW, ...	inverse algorithms, nonlinear optimizers

# Open Source Cloud Computing for Science

- *The browser is the GUI*
  - *HTML5 technologies, especially JavaScript*
  - *simulation code runs on a server, supercomputer, etc.*
- *Seamless legacy*
  - *export SRW python script, or IPython Notebook from browser*
  - *GUI will always help users, never restrict what they can do*
- *Containerized computing*
  - *open source technology: Docker*
    - *Docker enables rapid cloud deployment*
      - *no overhead on Linux*
    - *headless VMs on Mac OS and Windows with low overhead*
    - *eliminates pain of code installation, cross-platform development*
  - *archival, reproducibility, instantaneous collaboration*
    - *user input files, output files, etc. are saved in the container*
    - *share the container with a collaborator, students, etc.*

SRW in the cloud: <https://beta.sirepo.com>

Sirepo - Radiasoft

<https://beta.sirepo.com>

Sirepo

Sirepo brings computational science to the cloud.  
Develop, run and share your HPC simulations.

Explore the world of particle accelerators and electromagnetic radiation:

- Young's Double Slit – classic experiment from the history of physics
- Compact Storage Ring – use magnets to confine a beam of electrons
- Laser-Plasma Wakefield – create ultra-high accelerating electric fields

Sirepo is FREE and Open Source [View the GitHub project](#)

Sirepo is ready to support the world of scientific computing.  
Today, we support the following HPC physics codes on our cloud servers:

### Synchrotron Radiation Workshop (SRW)

SRW computes synchrotron radiation from relativistic electrons in arbitrary magnetic fields and propagates the radiation wavefronts through optical beamlines. SRW is open source and is primarily supported by Oleg Chubar of NSLS-II at Brookhaven National Laboratory.

### Elegant

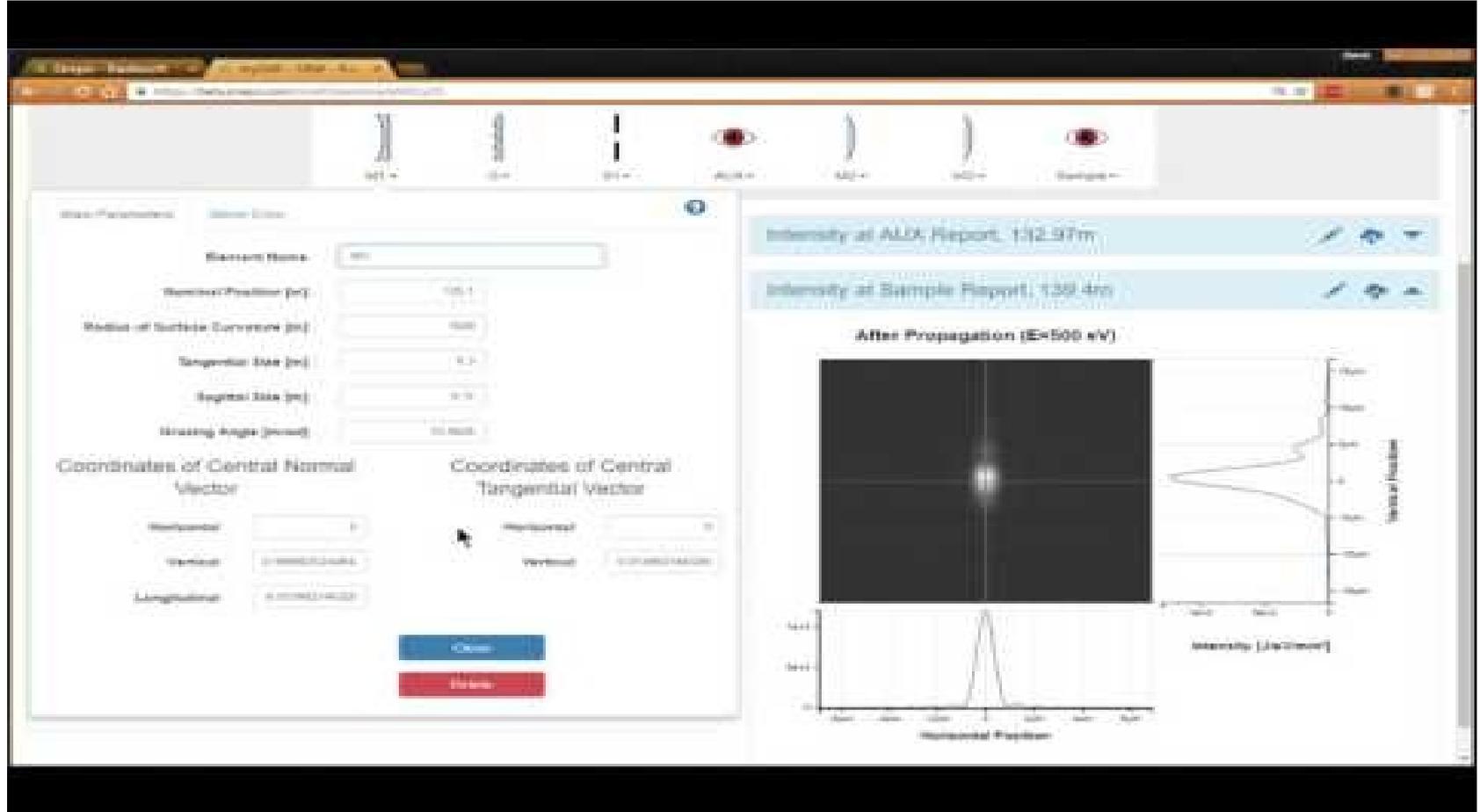
Elegant simulates charged particle accelerators with a wide range of features, including support for nonlinear optimization and design. Elegant is freely available and open source.

### Warp

Warp is a particle-in-cell (PIC) code designed to simulate high-intensity charged particle beams and plasmas in both the electrostatic and electromagnetic regimes, with a wide variety of integrated physics models and diagnostics. At present, Sirepo supports a small subset of Warp's capabilities. Warp is open source and is part of the Berkeley Lab Accelerator Simulation Toolkit.

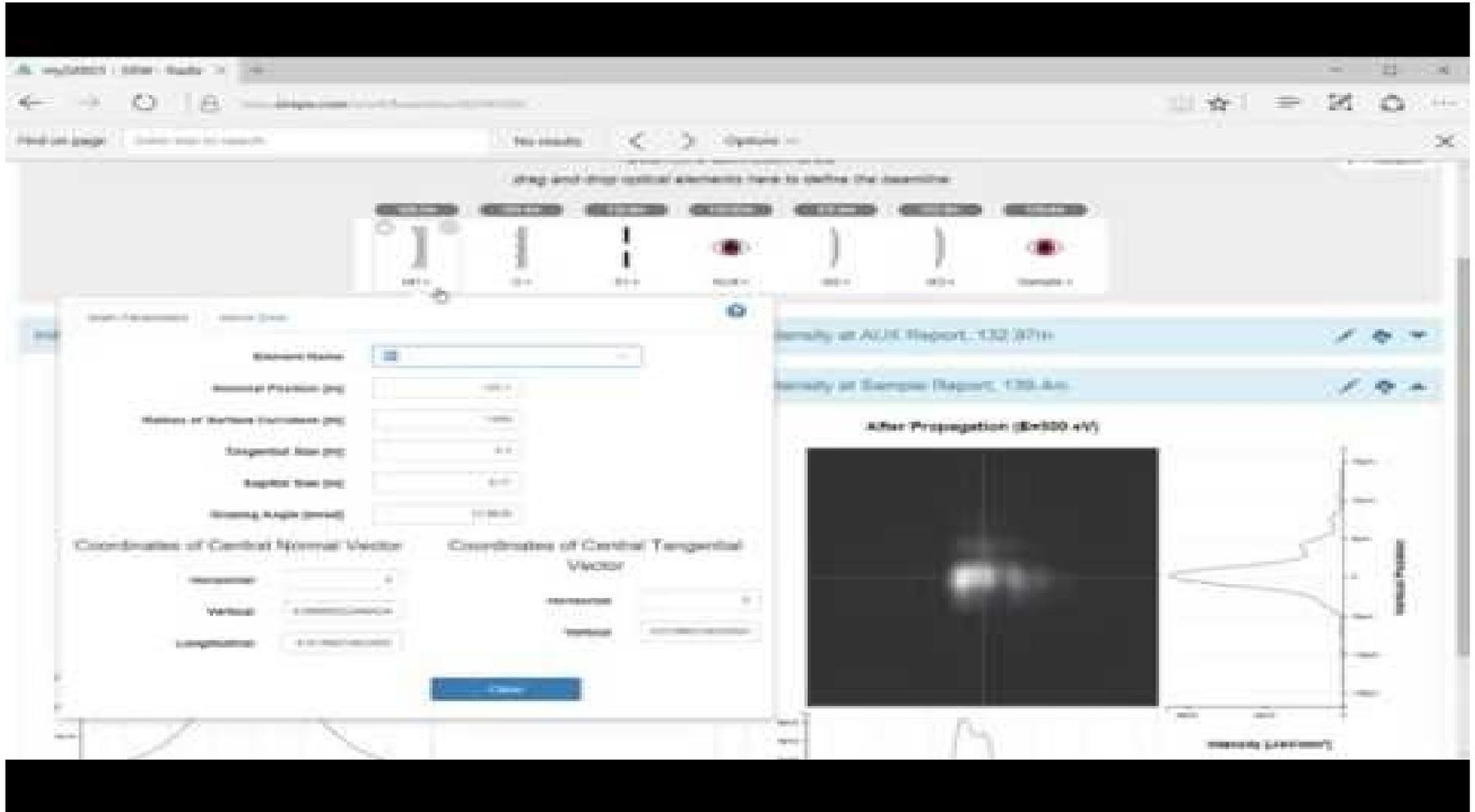
<https://www.youtube.com/watch?v=1hhivULQwOM>

# Sharing your SRW simulation, Part 1:



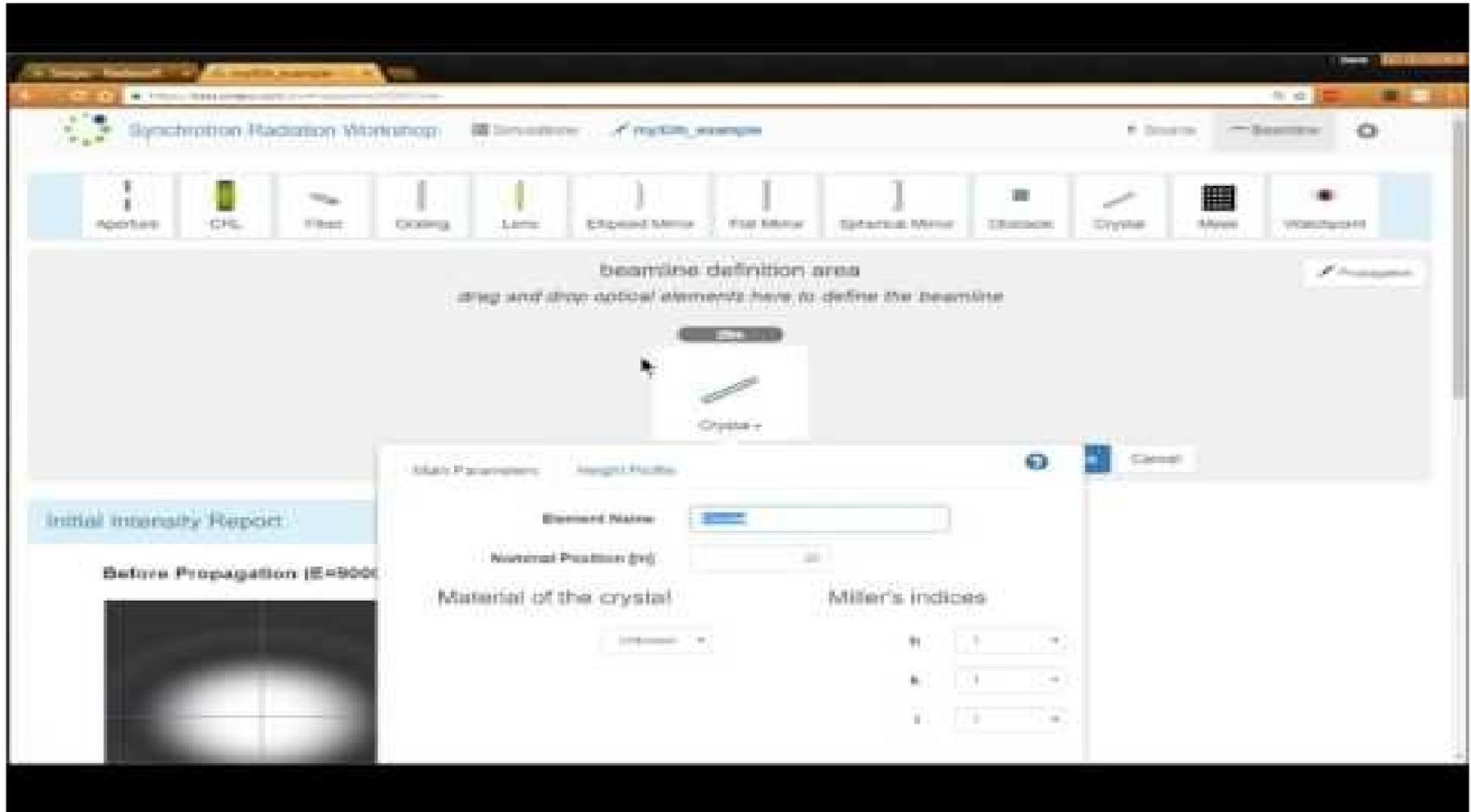
<https://www.youtube.com/watch?v=loa7TZ3PkGc>

# Sharing your SRW simulation, Part 2:



<https://www.youtube.com/watch?v=DJBCrmB7Kc0>

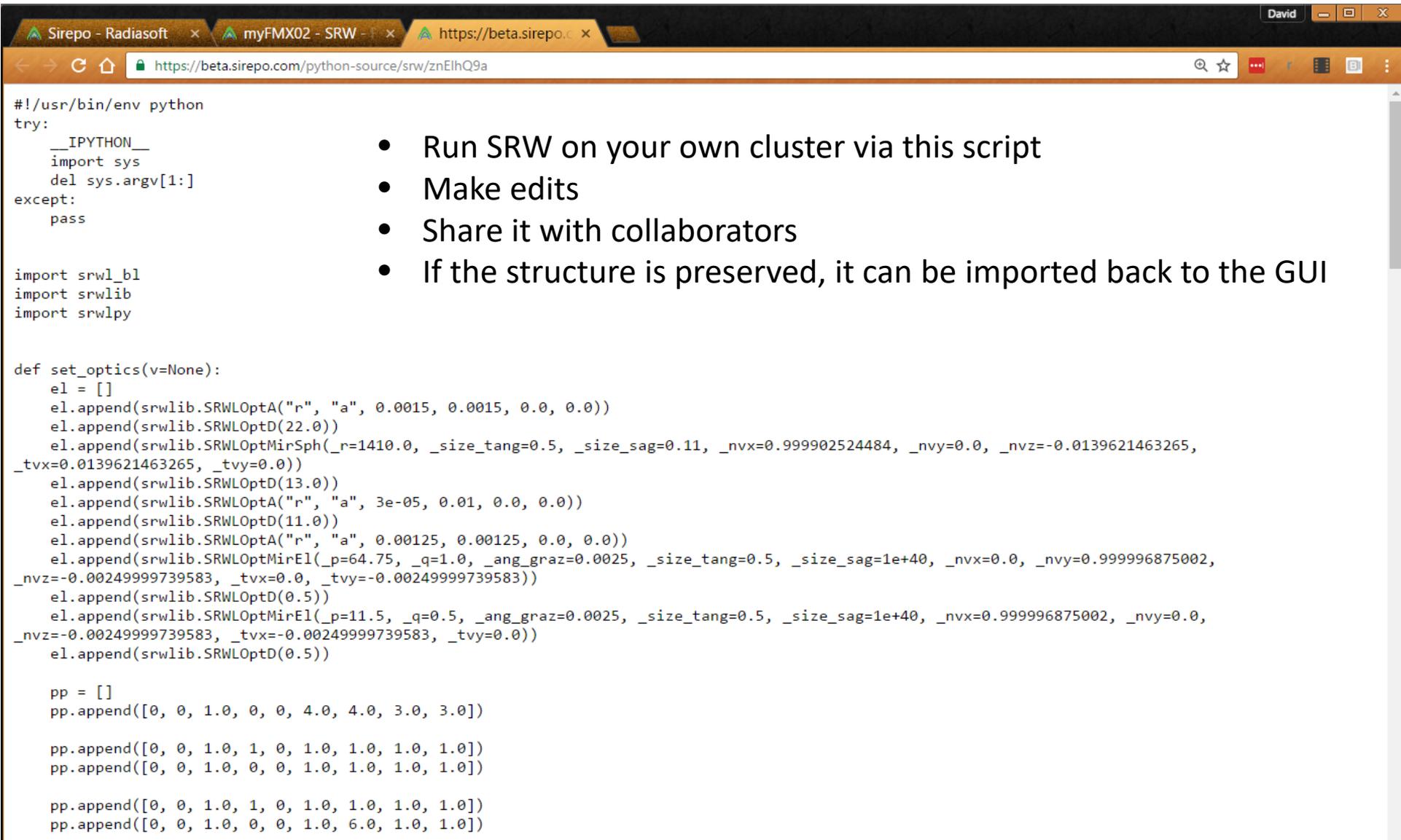
# Automated access of X0h data server at ANL



<https://www.youtube.com/watch?v=r6JEAUXfJol>



# Experts can download Python script



#!/usr/bin/env python  
try:  
    \_\_IPYTHON\_\_  
    import sys  
    del sys.argv[1:]  
except:  
    pass

import srwl\_b1  
import srwlib  
import srwlpv

def set\_optics(v=None):  
    el = []  
    el.append(srwlib.SRWLOptA("r", "a", 0.0015, 0.0015, 0.0, 0.0))  
    el.append(srwlib.SRWLOptD(22.0))  
    el.append(srwlib.SRWLOptMirSph(\_r=1410.0, \_size\_tang=0.5, \_size\_sag=0.11, \_nvx=0.999902524484, \_nvy=0.0, \_nvz=-0.0139621463265, \_tvx=0.0139621463265, \_tvy=0.0))  
    el.append(srwlib.SRWLOptD(13.0))  
    el.append(srwlib.SRWLOptA("r", "a", 3e-05, 0.01, 0.0, 0.0))  
    el.append(srwlib.SRWLOptD(11.0))  
    el.append(srwlib.SRWLOptA("r", "a", 0.00125, 0.00125, 0.0, 0.0))  
    el.append(srwlib.SRWLOptMirEl(\_p=64.75, \_q=1.0, \_ang\_graz=0.0025, \_size\_tang=0.5, \_size\_sag=1e+40, \_nvx=0.0, \_nvy=0.999996875002, \_nvz=-0.00249999739583, \_tvx=0.0, \_tvy=-0.00249999739583))  
    el.append(srwlib.SRWLOptD(0.5))  
    el.append(srwlib.SRWLOptMirEl(\_p=11.5, \_q=0.5, \_ang\_graz=0.0025, \_size\_tang=0.5, \_size\_sag=1e+40, \_nvx=0.999996875002, \_nvy=0.0, \_nvz=-0.00249999739583, \_tvx=-0.00249999739583, \_tvy=0.0))  
    el.append(srwlib.SRWLOptD(0.5))

    pp = []  
    pp.append([0, 0, 1.0, 0, 0, 4.0, 4.0, 3.0, 3.0])

    pp.append([0, 0, 1.0, 1, 0, 1.0, 1.0, 1.0, 1.0])  
    pp.append([0, 0, 1.0, 0, 0, 1.0, 1.0, 1.0, 1.0])

    pp.append([0, 0, 1.0, 1, 0, 1.0, 1.0, 1.0, 1.0])  
    pp.append([0, 0, 1.0, 0, 0, 1.0, 6.0, 1.0, 1.0])

- Run SRW on your own cluster via this script
- Make edits
- Share it with collaborators
- If the structure is preserved, it can be imported back to the GUI

You can export and import JSON files  
– enables sharing

```
Sirepo - Radiasoft x myFMX02 - SRW - x https://beta.sirepo.com x
https://beta.sirepo.com/simulation/srw/znElhQ9a/1
{
  "models": {
    "beamline": [
      {
        "horizontalOffset": 0,
        "horizontalSize": 1.5,
        "id": 14,
        "position": 20,
        "shape": "r",
        "title": "Aperture",
        "type": "aperture",
        "verticalOffset": 0,
        "verticalSize": 1.5
      },
      {
        "grazingAngle": 13.962599999998195,
        "heightAmplification": 1,
        "heightProfileFile": null,
        "id": 13,
        "normalVectorX": 0.9999025244842406,
        "normalVectorY": 0,
        "normalVectorZ": -0.013962146326506367,
        "orientation": "x",
        "position": 42,
        "radius": 1410,
        "sagittalSize": 0.11,
        "tangentialSize": 0.5,
        "tangentialVectorX": 0.013962146326506367,
        "tangentialVectorY": 0,
        "title": "HFM",
        "type": "sphericalMirror"
      },
      {
        "horizontalOffset": 0,
        "horizontalSize": 0.03,
        "id": 3,
        "position": 55,
        "shape": "r",
        "title": "SSA",
        "type": "aperture",
        "verticalOffset": 0,
        "verticalSize": 10
      }
    ]
  }
}
```

# Open source: <https://github.com/radiasoft>

The screenshot shows a web browser window with three tabs: 'Sirepo - Radasoft', 'SRW - Radasoft', and 'Young's Double Slit'. The active tab is the GitHub repository page for 'radiasoft / sirepo'. The browser address bar shows the URL 'https://github.com/radiasoft/sirepo/wiki/Young's-Double-Slit-Experiment-examples'. The repository page header includes the repository name 'radiasoft / sirepo', navigation links for 'Code', 'Issues 184', 'Pull requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area displays the title 'Young's Double Slit Experiment examples' with an 'Edit' button and a 'New Page' button. Below the title, it states 'Maksim Rakitin edited this page on Mar 28 · 5 revisions'. The main text of the page describes the 'Young's Double Slit Experiment' and lists its components. A sidebar on the right shows a list of pages under the heading 'Pages 12', including 'Home', 'Backup SRW Sirepo simulations', 'Dev NewReportExample', 'Diffraction by an aperture example', 'Dynamical access of crystal data and optical constants from external servers', 'ManualAlphaTest', 'SRW CRL', 'SRW Crystal', 'SRW Electron Beam', and 'SRW Fiber'.

radiasoft / sirepo

Unwatch 7 Star 4 Fork 4

Code Issues 184 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

## Young's Double Slit Experiment examples

Maksim Rakitin edited this page on Mar 28 · 5 revisions

### Young's Double Slit Experiment

The [example](#) demonstrates classical [Young's interference experiment](#) first performed by Young in 1801.

The optical scheme consists of the following elements:

- a lens with focal length  $F=0.5$  m located at 1 m from a Gaussian beam source (green laser, 535 nm);
- a double slit (emulated by a 10 by 0.5 mm rectangular aperture and a 10 by 0.3 mm rectangular obstacle) located just after the lens;
- a watch point showing the intensity of the beam just after propagation through the aperture;
- a watch point showing the intensity in the far field at 1 m from the aperture (Fresnel number  $F=0.3$ ).

Another [example](#) replicates the optical scheme of the above example, but a focusing lens is removed.

Pages 12

- Home
- Backup SRW Sirepo simulations
- Dev NewReportExample
- Diffraction by an aperture example
- Dynamical access of crystal data and optical constants from external servers
- ManualAlphaTest
- SRW CRL
- SRW Crystal
- SRW Electron Beam
- SRW Fiber

Many helpful discussions with and feedback from the following scientists:

Gerard Andonian

Alexey Buzmakov

Carsten Fortmann-Grote

Mark Glass

Jacek Krzywinski

Luca Rebuffi

Ruben Reininger

Liubov Samoylova

Manuel Sanchez del Rio

Timur Shaftan

Xianbo Shi

Lin Zhang

## Acknowledgments

Please try our beta implementation of SRW in the cloud: <https://beta.sirepo.com>

Please send all feedback to David Bruhwiler [bruhwiler@radiasoft.net](mailto:bruhwiler@radiasoft.net)

YouTube videos can be found here: <http://radiasoft.net/youtube>

This work is supported by the US DOE Office of Science,  
Office of Basic Energy Sciences under Award # DE-SC0011237.