

Proposal for a x-ray optical surfaces metrology database

Draft: Document for discussion at the MEADOW meeting

<http://www.elettra.trieste.it/Conferences/2013/MEADOW/>

Contents

Introduction	1
Name.....	2
Database contents	2
Database format	3
Repository (server).....	3
Access and Analysis tools.....	3
Documents	3
Implementation steps.....	4
Alpha test.....	4

Introduction

This is a draft document may serve as a starting point for discussion. The goal to build an open-source database of x-ray mirrors metrology to be used with SHADOW or any other ray-tracing and wave propagation codes that can simulate the effect of the surface errors on the performance of a beamline.

This draft is accompanied by a test repository and software to access and process the data. The proposal is that any of synchrotron facilities send a few mirror metrology files and we store then in a common repository. The submissions should be done by the end of 2013, the test version would be checked during the beginning of 2014, and a paper for presenting the project would be written and submitted by Spring 2014.

For the first test version we selected the easier and simpler solution and integrated it in the test database. These items are **marked in red** in this document.

Name

We should find a name for the database and software, e.g., **dabam (database for metrology)**, metDB (metrology database), xmdb (x-ray mirror database), etc.

Database contents

The entries in the data base must contain raw data (mandatory), metadata (minimum information of what is in the database, mandatory) and possibly analyzed data (results of rms slope, rms roughness, PSD, etc.)

- Raw data for 1D profiles.
 - **Measured slope x,y'** or Measured profile x,y
 - Measured set of
 - profiles x,y1,y2,...yN (same abscissas) or x1,y1,x2,y2,...xN,yN (different abscissas)
 - slopes x,y1', y2', ..., yN' or x1,y1',x2,y2',...xN,yN'
 - prepare the possibility for 2D data for the future
- Metadata (and **suggested name for keyword in first implementation**)
 - Type of file used (**FILE_FORMAT**)
 - Number of header lines in the raw data file (**FILE_HEADER_LINES**)
 - Units used (**X1_FACTOR, Y2_FACTOR**)
 - Year of fabrication of mirror (**YEAR_FABRICATION**)
 - Type of mirror (toroidal, sagittal cylinder, elliptical, etc.) (**SURFACE_SHAPE**)
 - Mirror function (white beam mirror, collimating mirror, focusing mirror) (**FUNCTION**)
 - Physical size in SI units (m): length (**LENGTH**) x width (**WIDTH**) x thickness (**THICK**)
 - Optical size in SI units (m): (**LENGTH_OPTICAL**)
 - Substrate (**SUBSTRATE**)
 - Coating (**COATING**)
 - Facility (**FACILITY**) and instrument of measurement (**INSTRUMENT**)
 - Other information:
 - polishing method (**POLISHING**)
 - Environment (clamped, gravity direction, cooling system, etc.) (**ENVIRONMENT**)
 - measurement setup (date, counter weight, direction, averaging, etc.) (**SCAN_DATE, ...**)
 - Plot information: titles (**PLOT_TITLE_X1, PLOT_TITLE_Y1**)
- Analysed data
 - Detrended/corrected profile/slope
 - Info from the analysis
 - Applied operations
 - RMS slope error (**CALC_SLOPE_RMS**) or RMS height (**CALC_HEIGHT_RMS**). Usually in SI units, otherwise use **CALC_SLOPE_RMS_FACTOR** and **CALC_HEIGHT_RMS_FACTOR** to retrieve the conversion (multiplicative) factor.
 - Optional: PSD and $\text{Sqrt}(\text{Integral}(\text{PSD}(f)))$

Database format

Possible data formats for data

- Raw, individual (copy original files, multiple formats)
- ASCII common format for multiple files
- Entries for multiple files: names or numbers. Criteria of assignation.
- Single file containing all data (e.g., DABAX/SPEC file)
- Binary formats (e.g., HDF5) to be prepared for future 2D data
- A real database: sqlite (e.g., <https://github.com/XraySpectroscopy/XrayDB>)

Possible format for metadata

- Keywords in a separate file (sharing name with *.txt extension)
- Keywords in a header (in a common ASCII)
- Keywords stored in database
- Define a dictionary of keywords.

Repository (server)

- Facility web/ftp site (e.g., <http://ftp.esrf.eu/pub/scisoft/DabamFiles/>)
- Code repository for local cloning (e.g., github.com)
- Dedicated site

Access and Analysis tools

- Off line (python code `dabam.py`, etc.)
- On line (web interface, submission form,...)
- Post processing (ray tracing, scattering calculations)

Documents

- This proposal
- Database documentation
- Minutes of Discussions
- Standards
- Publications
- Wiki?
- Mailing list to be used : <https://lists.bnl.gov/mailman/listinfo/xrayoptics-l>

Implementation steps

- Initial discussions. (APS July 2013, Trieste Oct. 2013)
- Test version available (see next paragraph)
- Starting data (6-10 profiles per participant/facility by end of 2013)
- Initial repository (example test repository: <http://ftp.esrf.eu/pub/scisoft/DabamFiles/>)
- Preliminary tools (example python code: dabam.py)
- Tests
- Continuity?
- Publication draft March 2014

Alpha test

A test version of an incipient database has been created. Data files for four mirrors (plane 1, plane 2, spherical and elliptical, respectively) used for the "Data analysis round-robin" were kindly made available by Amparo Vivo. They have been named `dabam-N.dat` ($N=1,\dots,4$) and placed in the repository <http://ftp.esrf.eu/pub/scisoft/DabamFiles/>

A python program `dabam.py` has been created and can be downloaded from the same repository. This programs access one of the remote files in the database and performs a series of operations (linear detrending, histogram, compute profile, PSDs, integral of PSDs and text info) and write these outputs in local files. The syntax is:

```
python dabam.py N
```

where N is the entry number in the database (now from 1 to 4) and you can add a number of options using command arguments starting by `-` or `--`. To get help of all options enter:

```
python dabam.py -h
```

or, equivalently:

```
python dabam.py --help
```

For example, to retrieve and analyze entry 3 (spherical mirror) we do:

```
python dabam.py 3
```

The results are:

```
Administrator: C:\Windows\system32\cmd.exe
C:\TMP>c:\Python27\python.exe dabam.py 1
File tmpProfile.dat written to disk. Cols: coordinate (m), height (m).
File tmpPSD.dat written to disk. Cols: freq (m^-1), psd_prof (m^3), psd_slope (
ad^3), cdf(psd_prof), cdf(psd_slope).

----- profile results -----
Data File: dabam-1.dat
Metadata File: dabam-1.txt
Linear fit coefficients: array([ 5.46078094e-05, -2.49791840e-12])
Radius of curvature [m] : 18312.399111233135

Slope error s_RMS [micro rad]: 0.48651846141972949
                             from PSD: 0.32980875952807326
Shape error h_RMS [nm]: 43.836191037681218
                             from PSD: 24.724035615907269
PU of height profile (before detrend) [nm]: 9895.724000000002
PU of height profile (after detrend) [nm]: 133.99860964579383

-----

C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
C:\TMP>
```

In Unix you can loop over the whole database by doing:

```
for i in {1..4} ; do python dabam.py $i ; done
```