cpymad intro

R. De Maria

cpymad

cpymad is a Cython binding to MAD-X for giving full control and access to a MAD-X interpreter within the Python environment.

- https://github.com/hibtc/cpymad
- http://hibtc.github.io/cpymad/

Developed mainly by Thomas Gläßle [thomas@coldfix.de]

Status:

- Features complete, maintenance mode
- Registered maintainers R. De Maria, T. Persson (ABP), Y. Dutheil (ABT)

Example

https://cernbox.cern.ch/index.php/s/SAXZtSGCri8oKp3

LHC Optics with `cpymad`

Documentation http://hibtc.github.io/cpymad/getting-started.html

```
from cpymad.madx import Madx
```

```
#inout=[]; madx = Madx(stdout=inout.append,command_log=inout.append)
madx = Madx()
madx.options.echo=False; madx.options.warn=False
madx.call("/eos/project/a/abpdata/lhc/optics/runII/2018/lhc_as-built.seq")
madx.call("/eos/project/a/abpdata/lhc/optics/runII/2018/PROTON/opticsfile.1")
madx.call("/eos/project/a/abpdata/lhc/optics/runII/2018/toolkit/macro.madx")
```

```
++++++++++++++++++++++++++++++++++++++++++++
+     MAD-X 5.05.01  (64 bit, Linux)       +
+ Support: mad@cern.ch, http://cern.ch/mad +
+ Release   date: 2019.06.07               +
+ Execution date: 2019.10.09 15:27:48      +
++++++++++++++++++++++++++++++++++++++++++++
```

```
madx.options.echo=False;madx.options.warn=True;
madx.exec_("mk_beam(450)")
```

```
++++++ info: nrj redefined

True
```

```
madx.use(sequence="lhcb1")
twiss=madx.twiss()
```

```
enter Twiss module
++++++ info: Zero value of SIGT replaced by 1.
++++++ info: Zero value of SIGT replaced by 1.

iteration:   1 error:   1.584040E-03 deltap:   0.000000E+00
orbit:  -5.596222E-04  1.204436E-06  1.988554E-06  1.611390E-04  0.000000E+0
0  0.000000E+00

iteration:   2 error:   2.548180E-05 deltap:   0.000000E+00
orbit:  -5.500010E-04 -1.594733E-10 -6.203780E-11  1.600002E-04  0.000000E+0
0  0.000000E+00

iteration:   3 error:   1.606903E-09 deltap:   0.000000E+00
orbit:  -5.500000E-04  3.345911E-11 -4.522176E-12  1.600000E-04  0.000000E+0
0  0.000000E+00

++++++ table: summ

            length             orbit5               alfa            gammatr
       26658.8832                 -0   0.0003478742677       53.61531322

                q1                dq1            betxmax               dxmax
       62.30999986        1.990949377        588.9482118        2.788411704

             dxrms             xcomax             xcorms                 q2
       1.500137515      0.01248750637   0.0008589570887       60.32000005

               dq2            betymax               dymax             dyrms
       1.963587272        640.9198581       0.4003798825      0.1003069822

             ycomax             ycorms              deltap            synch_1
   0.01301332164   0.0009498690428                   0                  0

            synch_2            synch_3            synch_4            synch_5
                  0                  0                  0                  0

             nflips
                  0
```

Code structure

- libmadx.so: standard MadX compiled by cmake into a share library (not included in the source tree)
- clibmadx.pxd: Cython defintiions to expose MadX c structures and functions
- libmadx.pyx: Cython code to compile a Python module that load libmadx.so library and give simple access to internal structures
- madx.py: High level class that gives “pythonic” control to MadX. By default madx.py spawn a Python process that loads libmadx.
- _rpc.py, stream.py, types.py, util.py: provides tools to support the structure

Code distribution

- Installation: pip install cpymad:
  - Uses precompiled wheels which include a compiled of libmadx.so.
  - Try to compile from source when wheel not available in the systems
- Recent version is LCG, available in SWAN
- One can download madx sources and cpymad sources to use it with the latest madx versions

Main features

- Call external file, input madx command as string
- Read-write access to variables and deferred expressions
- Read-write access to elements
- Execute commands
- Read access to sequence nodes including errors and attached beams
- Read access to table column as numpy array
- Loop over variable names, element names, table names
- Automatic support of new commands and elements through introspections

Maintenance tasks

- Test cpymad with new madx releases
  - TODO: Including cpymad test suite in madx workflow
- Produce new wheels and update to PyPI new releases
- Announce new releases to LCG managers
- Follow-up issues

Further developments

- Improve testing between madx-cpymad to avoid potential regressions
- Improve performance-critical code for big machines
- Improve Mac installation support