First experience using MAD-NG with the PS lattice

A. Huschauer, L. Deniau

MAD-NG review, 20 January 2021

Introduction

- **Significant discrepancies between MAD-X/PTC and MAD-NG initially observed (and reported in section meeting in October)**
  - **Understood and solved** by adapting the number of slices used in MAD-NG

- **Using flat bottom LHC optics including**
  - SBENDs with dipole, quadrupole and sextupole components
  - thin MULTIPOLEs with quadrupole, sextupole and octupole components
  - individual short quadrupoles
  - ➢ closed orbit at zero

Introduction

- **Typically using PTC for PS optics for several reasons**
  - benchmarking done several years ago defining proper setup of the PTC universe
  - Often working with **time=false** to take relativistic β into account
  - search for stable **fixed points at x/x’ ≠ 0** (MTE)
  - matching of higher-order components using **nonlin**-option (normal forms)
  - interface to PyORBIT via **flat files**
  - **slice_magnets** option to obtain Twiss functions at integration steps

MAD-X vs. MAD-NG input

MAD-X

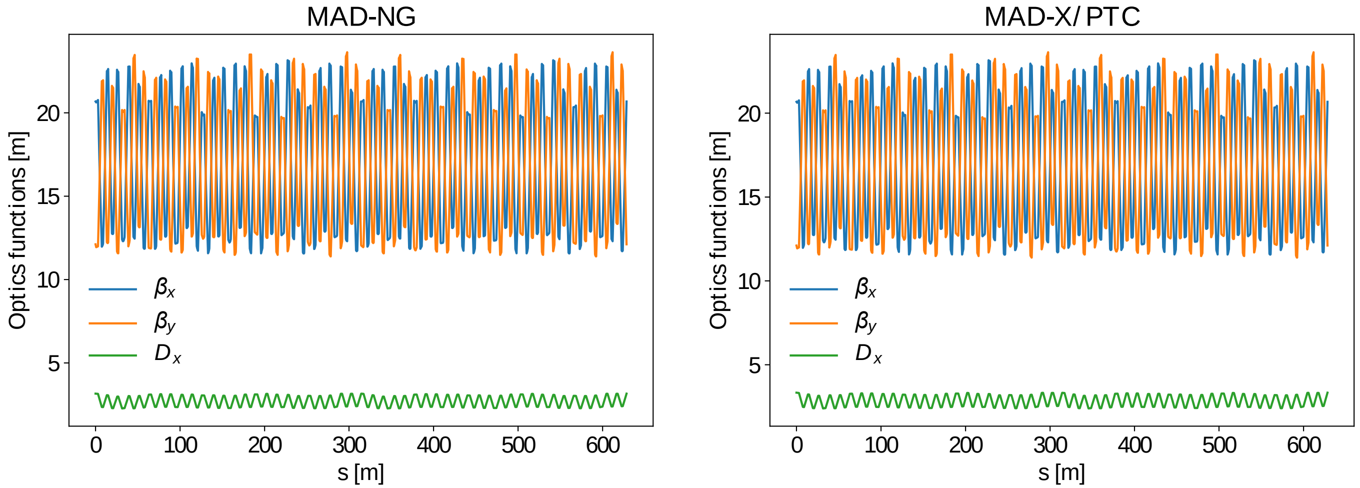
```
BEAM, particle=proton, pc=2.794987;
BRHO      := beam->brho;

call, file="ps_mu.seq";
call, file="ps_ss.seq";
call, file="ps_fb_lhc.str";

select, flag=ptc_twiss, clear;select, flag=ptc_twiss,
column={name,keyword,s,x,px,beta11,alfa11,beta22,alfa22,
disp1,disp2,mu1,mu2,l,angle,k1l,k2l,k3l,hkick,vkick};

use, sequence=PS;
ptc_create_universe;
ptc_create_layout, time=true, exact=true, model=2, method=6, nst=3;
ptc_twiss, closed_orbit, icase=56, no=2,file="PS_twiss_ptc.tfs";
ptc_end;
```

MAD-NG

```
local beam, survey, twiss in MADlocal
psbeam = beam 'psbeam' { particle="proton", pc=2.794987 }
MADX.BEAM =  psbeam
MADX.BRHO =\ psbeam.brho -- brho = 9.323073097 ;

local reload = false
MADX:load("ps_unset_vars.mad”    , reload)
MADX:load("ps_mu.mad"            , reload)
MADX:load("ps_ss.mad"            , reload)
MADX:load("ps_fb_lhc_str.mad"    , reload)

local ps in MADX
ps.beam = psbeam -- attach beam

local tws = twiss {sequence=ps, method=6, nslice=3, chrom=true}

-- add strengths to table
local melmcol in MAD.gphys
melmcol(tws, {'angle', 'tilt',  'k0l' , 'k1l' , 'k2l' , 'k3l' , 'k4l' ,
'k5l' , 'k6l', 'k0sl', 'k1sl', 'k2sl', 'k3sl', 'k4sl', 'k5sl', 'k6sl',
'ksl', 'hkick', 'vkick' })

-- write table to file
tws:write("PS_twiss_madng.tfs", {'name','kind','s',
'x','px','beta11','alfa11','beta22','alfa22','dx','dpx','mu1','mu2',
'l','angle','k0l','k1l','k2l','k3l','hkick','vkick'})
```

Twiss comparison

- **PTC and MAD-NG compute identical optics (not even numerical differences)**
  - Obviously crucial to use the same setup for the Twiss environment



  - Difference in Q' due to different ways of computing it: normal forms (MAD-X/PTC) vs. finite differences (MAD-NG)

	Qx	Qy	Qx'	Qy'	Time [s] (avg. over 5 executions)
MAD-X/PTC	0.21	0.245	0.7669541	-3.0285795	4.60
MAD-NG	6.21	6.245	0.78048442	-3.0211840	3.38

Conclusion and outlook

- **Initial tests with the PS lattice allowed to understand and resolve differences between MAD-X/PTC and MAD-NG**
  - Conversion of scripts
  - PS lattice description using sub-sequences which required modifications to the sequence parser
  - Matching and tracking studies still to be done for benchmarking

- **PTC physics very well reproduced with MAD-NG**
  - For this simple example MAD-NG showed faster performance than MAD-X
  - Further testing certainly needed, but looks promising for low-energy machines