

New Matching Mode in MadX

Riccardo de Maria

CERN AB-ABP-LOC

MADX Meeting, 11 April 2006

New Match Mode in MadX

The new mode is based on the idea that:

- ▶ the constraints can be user defined expressions,
- ▶ the calculations for evaluating these expressions can be defined via macros.

In this way, with a little more effort for the user, everything that can be calculated via macros and expressions can be used for matching:

- ▶ solution for linear or non linear system of equations,
- ▶ chromatic and non linear effects.

Root of a polynomial

```
option, -info;  
x=-1.5;  
  
match, use_macro ;  
  vary,name=x;  
  m1: macro= { y=(x-3.5)*(x+2)*(x-4); };  
  constraint,expr= y=0;  
  jacobian,tolerance=1.e-24;  
endmatch;
```

Matching non linear chromaticity with PTC

```
chrom: macro={ ptc_create_universe;
  ptc_create_layout,model=2,method=6,nst=1;
  select_ptc_normal, q1=0, q2=0, dq1=1, dq2=1;
  select_ptc_normal, dq1=2, dq2=2, dq1=3, dq2=3;
  ptc_normal,closed_orbit,normal,icase=5,no=4;
  ptc_end; };
qx :=table(normal_results,value,1); q1x :=table(normal_results,value,3);
q2x :=table(normal_results,value,5); q3x :=table(normal_results,value,7);
qy :=table(normal_results,value,2); q1y :=table(normal_results,value,4);
q2y :=table(normal_results,value,6); q3y :=table(normal_results,value,8);
match, use_macro ;
  vary, name=ksd1.a12b1; vary, name=ksd1.a23b1; vary, name=ksd1.a34b1; vary, name=ksd1.a45b1;
  vary, name=ksd1.a56b1; vary, name=ksd1.a67b1; vary, name=ksd1.a78b1; vary, name=ksd1.a81b1;
  vary, name=ksd2.a12b1; vary, name=ksd2.a23b1; vary, name=ksd2.a34b1; vary, name=ksd2.a45b1;
  vary, name=ksd2.a56b1; vary, name=ksd2.a67b1; vary, name=ksd2.a78b1; vary, name=ksd2.a81b1;
  vary, name=ksf1.a12b1; vary, name=ksf1.a23b1; vary, name=ksf1.a34b1; vary, name=ksf1.a45b1;
  vary, name=ksf1.a56b1; vary, name=ksf1.a67b1; vary, name=ksf1.a78b1; vary, name=ksf1.a81b1;
  vary, name=ksf2.a12b1; vary, name=ksf2.a23b1; vary, name=ksf2.a34b1; vary, name=ksf2.a45b1;
  vary, name=ksf2.a56b1; vary, name=ksf2.a67b1; vary, name=ksf2.a78b1; vary, name=ksf2.a81b1;
  use_macro,name=chrom;
  constraint,expr= q1x=2 ;
  constraint,expr= q1y=2 ;
  constraint,expr= abs(q2x)<100 ;
  constraint,expr= abs(q2y)<100 ;
  constraint,expr= abs(q3x)<1000 ;
  constraint,expr= abs(q3y)<1000 ;
  jacobian;
endmatch;
```

Matching non linear chromaticity without PTC

```
use, period=lhcb1;
match, use_macro ;
  vary, name=ksd1.a12b1; vary, name=ksd1.a23b1;
  vary, name=ksd1.a34b1; vary, name=ksd1.a45b1;
  vary, name=ksd1.a56b1; vary, name=ksd1.a67b1;
  vary, name=ksd1.a78b1; vary, name=ksd1.a81b1;
  vary, name=ksd2.a12b1; vary, name=ksd2.a23b1;
  vary, name=ksd2.a34b1; vary, name=ksd2.a45b1;
  vary, name=ksd2.a56b1; vary, name=ksd2.a67b1;
  vary, name=ksd2.a78b1; vary, name=ksd2.a81b1;
  vary, name=ksf1.a12b1; vary, name=ksf1.a23b1;
  vary, name=ksf1.a34b1; vary, name=ksf1.a45b1;
  vary, name=ksf1.a56b1; vary, name=ksf1.a67b1;
  vary, name=ksf1.a78b1; vary, name=ksf1.a81b1;
  vary, name=ksf2.a12b1; vary, name=ksf2.a23b1;
  vary, name=ksf2.a34b1; vary, name=ksf2.a45b1;
  vary, name=ksf2.a56b1; vary, name=ksf2.a67b1;
  vary, name=ksf2.a78b1; vary, name=ksf2.a81b1;
c1: macro={
  twiss;
  dq1=table(summ,dq1);
};
constraint,expr= dq1=2;
c2: macro={
  twiss,deltap=1E-4;
  dq1m=table(summ,dq1);
  twiss,deltap=-1E-4;
  dq1p=table(summ,dq1);
};
constraint,expr= (dq1m-2*dq1+dq1p)/2E-4<100;
jacobian;
endmatch;
```

Conclusions

The code is ready but still not fully debugged. I might have introduced new bugs in the normal match (which is still available for backward compatibility).

Suggestion and users are very welcome.

Missing features: constrain several elements using same command (i.e. `constraint, range= ...`) is not straightforward to implement.