PTC-ORBIT notes

Alexander Molodozhentsev (KEK)

Etienne Forest (KEK)

start: 24.02.2012

VERSION 2 (March 19, 2012)

'WHAT and HOW' notes

PTC is basically the s-based integrator (with some possibility to use a first-order time-based integrator).

… the best manual is the code itself …

PTC in a tracking mode pushes particles though a link list of integration nodes. For the purpose of ORBIT only, these integration nodes are grouped into "Nodes".

In the presence of space charge (or for other reasons) it may be more important to observe the beam at many locations and therefore have a large number of integration steps. For this reason, it is sometime preferable to use a low order method with many steps rather than a high order method with fewer steps. (Although it must be said that Forest allows the use of so-called "odd" methods to look inside the steps of a higher order method. This is possible for methods that do no contain "negative" propagators.).

To be able to use the PTC-ORBIT (MPI) code it is necessary to make the following steps:

(1) create the machine FLAT file, which contains information about all PTC elements of the machine including parameters for the integration procedure;

(2) prepare the ORBIT script, which should use:
- FLAT file of the machine
- Time-tables for individual PTC elements
- PTC scripts to get basic parameters of the machine (like betatron tunes, chromaticity and Twiss parameters)
- PTC script to energize the lattice with the time-dependent elements
- File (FINAL_SETTING.txt) with the proper setting of INITIAL time for the tracking
- 6D particle distribution (generated by ORBIT or read from the external file)
- Space charge module with proper parameters for the space-charge simulations for each space-charge node
- Aperture definitions in required locations around the machine
- Tracking procedure, including the turn-by-turn ORBIT diagnostic

(3) post-processing to analyze the obtained results.

PART 1

PTC FLAT FILE preparation …

exact_model = TRUE or FALSE

→ **split** the elements of the lattice into integration nodes using one of PTC's **integration method**

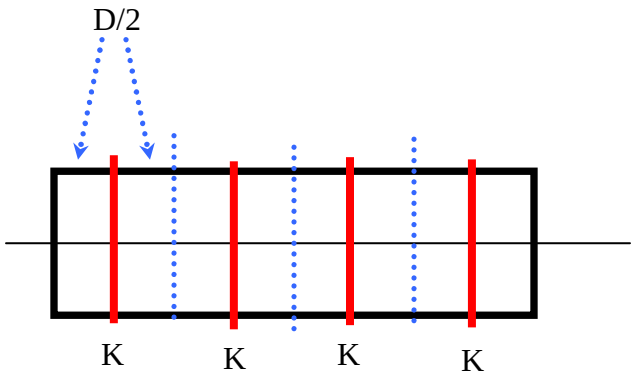
PTC integration methods:

PTC generally attempts to integrate the maps of each magnet using an explicit symplectic integrator.
PTC has six integration methods.

Drift-Kick-Drift (D-K-D) case → model = 1
method 2 → the naïve second-order method, which has one kick per integration step,
method 4 → the Ruth-Neri-Yoshida fourth-order method, which has three kicks per integration step
method 6 → the Yoshida sixth-order method, which has seven kicks per integration step

Higher order integration schemes are based on the fundamental second order method [PTC report, Yoshida's paper, Forest's book or review].

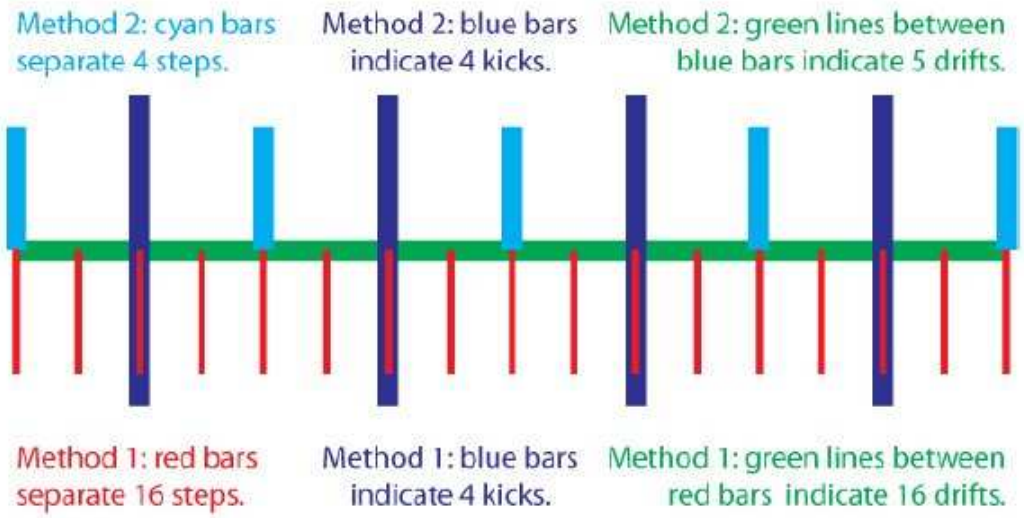
Example:
method 2, 4 steps (4 kicks)



Additional possibilities:

→ the exact_model=TRUE and the exact_model=FALSE, the ‘D-K-D’ algorithm can be switched into method 1 if and only if the original method was the method 2 (the ‘second-order’ integrator). The method 1 in this case is simply a re-splitting the drifts. Obviously the results should be the same to machine precision. Larger number of steps for this method 1 does not change the accuracy.

That is the illustration of the D-K-D case for the integration methods 1 and 2.



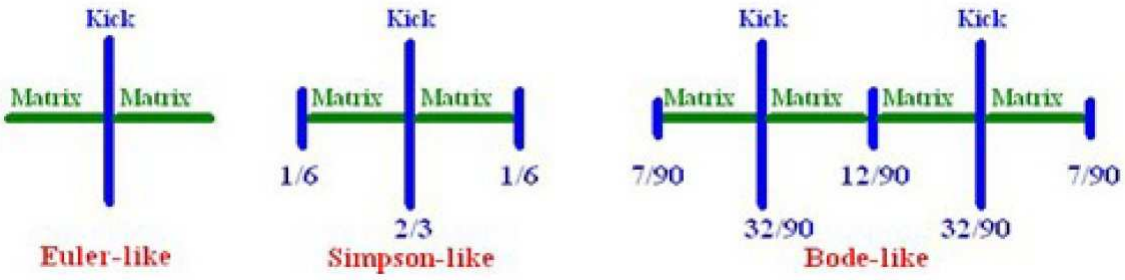
The blue bars and green lines represent the original D-K-D case of the integration method 2. For this case there are 4 kicks, 5 drifts and 4 steps, separated by the cyan bars. The new method 1 will simple split the drifts, keeping the same number of kicks. There are now 16 drifts.

Matrix-Kick-Matrix (M-K-M) case → model = 2

PTC has also a 'Matrix-Kick-Matrix' method where the matrix of the PTC element is the energy independent one. The 'delta' dependence is buried in the kick itself. The integration methods in this case always split the integration step using matrices of equal length.

Exact_model=false

The method of integration 2,4 and 6 in the case of M-K-M.



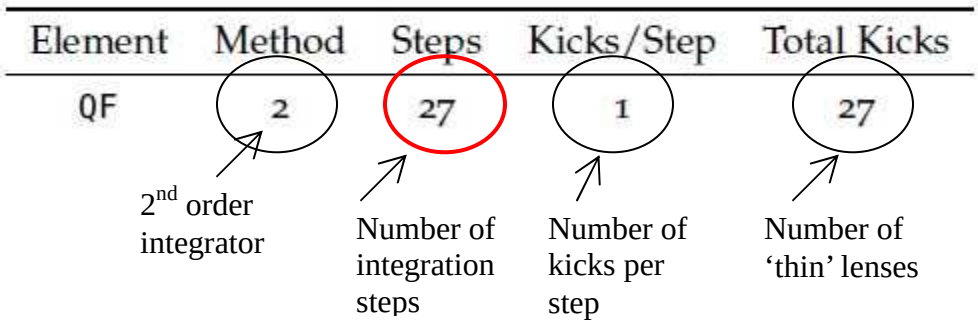
The Exact_model=true only works with the straight elements. Bends should be presented as the D-K-D case (method 2).

Splitting of elements

THIN > 0 → Splitting Quadrupole magnets

The optional parameter THIN describes an approximate integrated quadrupole strength for which a single integration node (or thin lens) in the body of an element should be used.
The optional parameter LIMITS(1:2) can be used to control the splitting algorithm to switch the second-order integrator to the forth-order or six-order integrator.

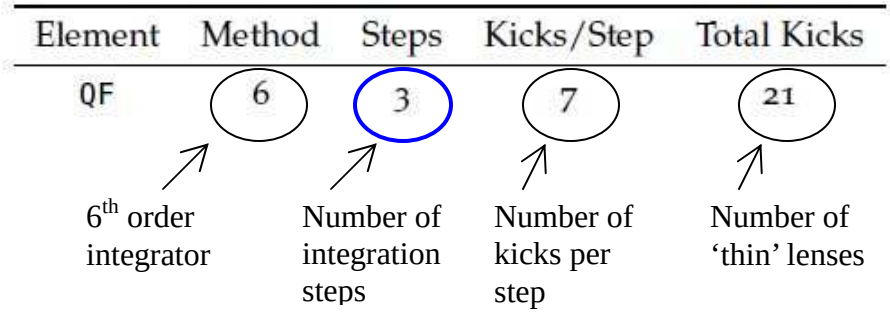
Let's use THIN=0.01 (for the case of the QF integrated quadrupole strength of $K_Q L$= 0.2793096).



In this example for the QF element will use 27 identical 'thin' lenses. There is not efficient if the number of steps must be large. PTC can change the splitting algorithm
By using PTC abilities in the case of large total number kicks one can change the integration method from the second order till the six order method re-arranging the kicks along the PTC element (by using non-identical kicks).

The parameter LIMITS determines when PTC should choose the second-order, forth-order or sixth-order integration method.
The array LIMITS(2) is used for this purpose:
LIMITS(1) > KL/THIN → 2nd order integration method
LIMITS(2) > KL/THIN > LIMITS(1) → 4th order integration method
KL/THIN > LIMITS(2) → 6th order integration method

For the previous example, if the parameter LIMITS is defined the integration method for the quadrupole magnet can be changed. Let us use LIMITS(1)=8 and LIMITS(2)=24 for example. As the result, PTC will use the following integration methods:



PTC will use in this case for the QF element the integration method 6, which results in three integration steps with seven kicks per step: a total amount of kicks for the element is 21 kicks. (Remark: The sixth order integrator has nine second order maps. Yoshida found more effective solutions using just seven second order maps).

Enforcing second-order integrator : if LIMITS(1:2)=10000 … any big number …the integrator will remain a second order integrator for any reasonable value of THIN).

THIN < 0 → Splitting Sextupole magnets (for the version: PTC-vFeb-2012)

To split the sextupole magnets one can use the similar way, used for the splitting procedure implemented into PTC. To activate this procedure the 'THIN' parameter should be negative. In addition it's necessary to define the maximum radius 'sexr0' at which the sextupole field nonlinearity will be taken into account. In this case the integrated quadrupole strength is $KL = |K_S * sexr0| * L$
Additional parameter is XBEND = -1 (no effects for the bends … see below).

Bending magnets (XBEND parameter)

The optional argument XBEND can be used to affect the bends (for the EXACT_MODEL=TRUE or FALSE). Usually XBEND parameter is of order 1.e-4 to 1.e-10 (depending on the bending angle of the dipole magnet). The original purpose of XBEND was to insure, without changing bn(1), that the ideal SBEND (exact=true) sends the origin into itself. Otherwise closed orbit distortions, resulting in quadrupole feed-downs, can occur.

EVEN option

There are three possibilities for the EVEN optional argument of the splitting procedure, implemented into PTC:

EVEN = TRUE → enforces an even split,
EVEN = FALSE → enforces an odd split,
EVEN is not in the call statement: an even or odd split is acceptable.

EVEN=true is important if the motion must be observed at the center of a magnet.

LMAX

This parameter defines the approximate minimum length between collective effects (the space-charge nodes in the case of the tracking with the space charge effects, by using the PTC-ORBIT combined code […]). The parameter 'fuzzylmax' will define the accuracy for the 'lmax' around the machine.

Using LMAX, PTC will try to group the 'integration nodes' of the node layout of PTC into groups whose length does not exceed LMAX (± fuzzylmax). Since the node layout has already been sliced by PTC.

RF cavity cannot be divided/combined by LMAX. It's always one piece PTC element.

Example:

```
# … MADX-PTC command
…
ptc_create_universe;
…
use, period=RING;
ptc_create_layout, model=2, method=2, nst=5, exact;
```

In this example the machine (RING) description will be created inside the PTC universe using the following:

→ *EXACT=TRUE*
→ *MATRIX-KICK-MATRIX propagation method (model=2)*
→ *2nd ORDER integrator (method=2)*
→ *5 integration steps for the PTC element (NST=5)*

By using this command in MADX-PTC

```
ptc_create_layout,model=1,method=6,nst=10;
```

new machine description will be created:

→ *EXACT=FALSE*
→ *DRIFT-KICK-DRIFT propagation method (model=1)*
→ *6th ORDER integrator (method=6)*
→ *10 integration steps for the PTC element (NST=10)*

The machine description with full set of the information about the integration procedure for each PTC element of the machine can be saved in the FLAT file.

… part of the PTC script, used for the splitting procedure

…

```
CUTTING ALGORITHM
2
LIMIT FOR CUTTING
10000 10000

thin lens =1
lmax
1.25d0
fuzzylmax
0.10
THIN LENS
0.01
THIN LENS
-0.01
0.10 -1
```

…

→ *… force to use the 2nd order integrator ('cut' algorithm including DRIFTS)*
→ *… set the limit for the number of the integration steps to use the 4th and 6th order integrators*

→ *… RESTART cutting procedure from scratch*
→ *… set the 'LMAX' parameter*

→ *… set the 'fuzzylmax' parameter*

→ *… define the splitting procedure for QUDRUPOLES*

→ *… define the splitting procedure for SEXTUPOLES*

PTC splitting script should be inside of any MADX(PTC) matching procedure

…..

```
MATCH, USE_MACRO;
  VARY, NAME = dkf,   STEP = .1;
  VARY, NAME = dkd,   STEP = .1;
  VARY, NAME = dkd3,  STEP = .1;
  VARY, NAME = dkd14, STEP = .1;
  VARY, NAME = betin, STEP = .1;
  VARY, NAME = alfin, STEP = .01;
  VARY, NAME = xin,   STEP = .001;
  HorMatch: MACRO =
   {
      USE, PERIOD = PSBRing; ! whole PSB ring
      ptc_create_universe;
         ptc_create_layout, model=2, method=2, nst=5,exact=true;
         ptc_setswitch, debuglevel=0, nocavity=false, fringe=true,
              exact_mis=true, time=true, totalpath=true;
         ptc_script, file="thin4.PTC";
         PTC_TWISS, table=twiss, closed_orbit, icase=5, no=2, rmatrix,
              BETX=betin, ALFX=alfin, X=xin, BETY=4.5, ALFY=0.0, BETZ=1.0;
      ptc_end;
   }
!  CONSTRAINT, weight=1e6, EXPR = betin > 0.05; ! safety contraint to avoid beta < 0
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, BETA11) = betin;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, ALFA11) = alfin;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, X)      = xin;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, XP)     =  0.;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, RE33)   = -1.;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, RE34)   =  0.;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, RE43)   =  0.;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, RE44)   = -1.;
   CONSTRAINT, EXPR=Table(twiss, PSBRing$end, MU1)    = QH;
!  CONSTRAINT, EXPR=Table(twiss, PSBRing$end, MU2)    = QV;
   LMDIF, TOLERANCE = 1.0E-15;
ENDMATCH;
```

Splitting of the QUADRUPOLE magnets (PTC)

 # case-1

→ part of the 'splitting' PTC script

…….

```
CUTTING ALGORITHM
2
LIMIT FOR CUTTING
10000 10000
thin lens =1
lmax
1.25d0
fuzzylmax
0.10

THIN LENS
0.01
```

……..

→ part of the PTC FLAT file

QFR08005A:
L = 1.76m, b_2 = 9.855×10^{-2} [m^{-2}] → $K_Q L$ = 0.173448 [m^{-1}]

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ELEMENT $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         37 QFR08005A                QFR08005A:1
  1.76000000000000      F F  L,PERMFRINGE,MIS
 0.500000000000000       0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 NO ELECTRIC ELEMENT INFORMATION
 NO_SOLENOID_PRESENT   0.000000000000000E+000
MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART
 F         2         17          2  EXACT METHOD NST NMUL
```

2nd order integrator

Number of Steps

Exact=False

 # case-2

→ part of the 'splitting' PTC script

…….

```
CUTTING ALGORITHM
2
LIMIT FOR CUTTING
10000 10000
thin lens -1
lmax
1.25d0
fuzzylmax
0.10

THIN LENS
0.001
```

→ part of the PTC FLAT file

QFR08005A:
L = 1.76m, $b_2 = 9.855\times10^{-2}$ [m^{-2}] → $K_QL = 0.173448$ [m^{-1}]

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ELEMENT $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         37 QFR08005A                QFR08005A:1
  1.76000000000000      F F  L,PERMFRINGE,MIS
 0.500000000000000       0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 NO ELECTRIC ELEMENT INFORMATION
 NO_SOLENOID_PRESENT   0.000000000000000E+000
MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART
F          2      173          2  EXACT METHOD NST NMUL
```

Number of Steps

2^{nd} order integrator

 # case-3

→ part of the 'splitting' PTC script

```
CUTTING ALGORITHM
2
LIMIT FOR CUTTING
10 20
thin lens =1
lmax
1.25d0
fuzzylmax
0.10

THIN LENS
0.01
```

→ part of the PTC FLAT file

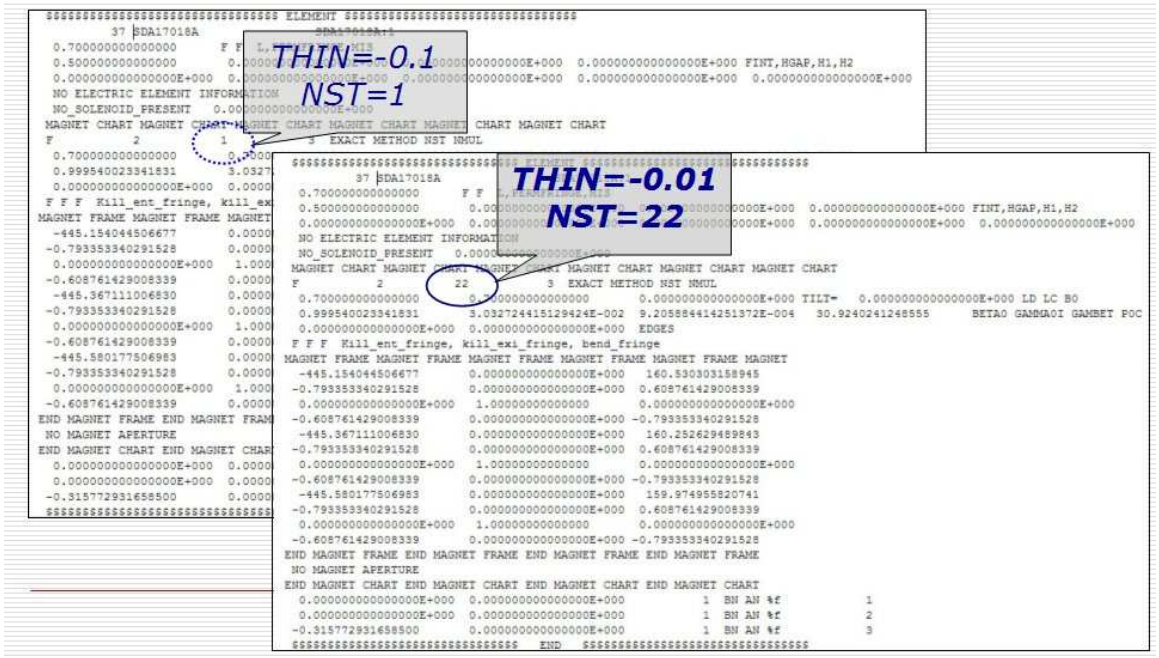
QFR08005A:
L = 1.76m, $b_2 = 9.855\times10^{-2}$ [m^{-2}] → $K_QL = 0.173448$ [m^{-1}]

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ELEMENT $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         37 QFR08005A                QFR08005A:1
  1.76000000000000      F F  L,PERMFRINGE,MIS
 0.500000000000000       0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 NO ELECTRIC ELEMENT INFORMATION
 NO_SOLENOID_PRESENT   0.000000000000000E+000
MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART MAGNET CHART
F          6          24           2  EXACT METHOD NST NMUL
```

Number of Steps

6^{th} order integrator

Example:
→ result of the splitting procedure for the SEXTUPOLE magnets



WAY to study the most effective cutting algorithm for some lattice …

EXAMPLE … from Etienne Forest (7.03.2012)

The most efficient cutting with a symplectic integrator involves the following set of procedures on a bare (ideal) lattice:

1) Do a tentative cut
2) Fit the parameters of the lattice
3) See if what you get is sufficient for your studies
4) Iterate 1,2,3 until happy
5) The lattice is now fixed, hopefully for ever. Avoid any recutting if possible. Proceed with serious studies with errors, space charge, etc….

Here I want to illustrate the effect of XBEND and FITBEND. It is not a good example in the sense that all methods work…. But it shows how to use these things.

PSR MAD-X lattice to generate an example

```
TITLE "PSR Ring"


assign, echo="PSR_MADX.out";
option,echo,rbarc=FALSE;

        D1M: DRIFT, L:= 2.28646-0.1;
        D1: DRIFT, L:= 2.28646;
        D2: DRIFT, L:= 0.45-0.1;
        SD1: SEXTUPOLE, L:= 0.1,K2:=0.0;  !-362.438255924030*2;
        SD2: SEXTUPOLE, L:= 0.1,K2:=0.0;  !157.133255882913*2;
        QD  :  QUADRUPOLE, L:=0.5, K1:= 0.558619275157041;
        QF  :  QUADRUPOLE, L:=0.5, K1:=-0.394319488346147;
        B   :  RBEND,  L:=2.54948, ANGLE:= 0.628318530717959, E1:=0.,
E2:=0.;

        CELL:    LINE=( D1M,SD1,QD,D2,SD2,B,D2,SD2,QF,D1);
        PSR:     LINE=( 10*CELL );


eg :=  1.73527443433878; !  ... in GeV

beam, particle = proton, energy = eg;


use, period=PSR;
ptc_open_gino, command=opengino;
ptc_create_universe;
```

```
BEAM, particle=proton, energy=eg, EX=54E-9, sige=5e-4; !,radiate
use, period=PSR;

!  this cutting will give us inefficiently  high accuracy
ptc_create_layout,model=1,method=6,nst=40,exact;

ptc_script, file="FIT_BEND_b_05_5.txt";

ptc_open_gino, command=mini;
ptc_open_gino, command=closegino;
ptc_end;
stop;
```

The above lattice is the PSR of Los Alamos. It is a convenient lattice since EXACT=true is essential.

We would like to investigate the issue if XBEND. To emphasize the effect of a poor cutting of the bends, we turned the drifts into sextupoles to magnify any feed-down from a poor closed orbit. The bends we use contain NO quadrupole focusing. This means that a cutting based on focusing strength alone will not be efficient for the bends. Of course, we use the drift -kick-drift split with exact=true.

Properties of the highly cut lattice

K_ptc_sex= (`0.591959408211528,-0.470470912360793`)

Tunes = 0.25410281000000 0.25543771000000
Chromaticities = 0.68912018098376E-14 -0.50712177421113E-14
2nd order chromaticities = 1.3420643310293 0.94957128618840
Amplitude Tune Shift = 0.65474574641022 -1.2847546841043 -0.21556413619884

In bend, bn(1)= `0.246449680216342`

Next let us look at the following script called inside MAD-X
Using `ptc_script, file="FIT_BEND_b_05_5.txt";`

```
select layout
1
+nocavity
ALLOCATE BETA
FILL BETA
1 1
THINLENS=1
 THIN LENS
 .2
!FITBENDS      !!! bend not fitted and no xbend
```

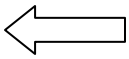
```
set families
4
 1 qf
 2 1
 1 qd
 2 2
 1 Sd1
 3 3
 1 Sd2
 3 4
FITTUNECHROMATICITY
1.e-8
0.25410281 0.25543771 0 0
FILL BETA
2 1
return
```

Results for:

THIN LENS

.02

FITBENDS



METHOD 2 100 100
METHOD 4 0 0
METHOD 6 0 0
number of Slices 100
Total NST 102
Total NST due to Bend Closed Orbit 0
Total NST due to Sextupoles 0

closed orbit
-1.259794197770755E-015 -1.624651800670882E-016

Tunes 0.254102810000000 0.255437710000000
<DBETA/BETA> = 3.082149609892620E-002
MAXIMUM OF DBETA/BETA = 5.180394645189255E-002
<DETA/ETA> = 0.108647545151797

In bend, bn(1)= 0.244651878992296
Properties of cut lattice
K_ptc_sex= (0.556319667416927, -0.416905725992572)

Tunes = 0.25410281000000 0.25543771000000
Chromaticities = 0.70678992921411E-15 0.16079470889621E-14
2nd order chromaticities = 1.9020303770233 0.88628557228841
Amplitude Tune Shift = 0.56270312685264 -0.93760015330086 -0.10751699977080

Results:

THIN LENS
.02
!FITBENDS no fitbend and no xbend

closed orbit
1.128165680521308E-002 1.191425531929293E-003

METHOD 2 70 70
METHOD 4 30 270
METHOD 6 0 0
number of Slices 340
Total NST 162
Total NST due to Bend Closed Orbit 0
Total NST due to Sextupoles 0

<DBETA/BETA> = 2.406302580966122E-003
MAXIMUM OF DBETA/BETA = 4.761499157228347E-003
<DETA/ETA> = 7.928361895842942E-003

Tunes = 0.25410281000000 0.25543771000000
Chromaticities = 0.42407395752847E-15 -0.25091042487101E-14
2nd order chromaticities = 1.4522860619982 0.99682945655324
Amplitude Tune Shift = 0.67770920090281 -1.2965366789141 -0.21727200041451

Same as above `turned on`

```
THIN LENS
 -.2
 .0  1.E-6
!FITBENDS     no fitbend using xbend=1.e-6
```

closed orbit
2.491143705443199E-004 2.552264830504116E-005

METHOD 2 90 90
METHOD 4 10 150 ⇐
METHOD 6 0 0
number of Slices 240
Total NST 142
Total NST due to Bend Closed Orbit 170
Total NST due to Sextupoles 0

Tunes 0.254102810000001 0.255437710000000
<DBETA/BETA> = 1.540609936485017E-002
MAXIMUM OF DBETA/BETA = 2.225226287246854E-002
<DETA/ETA> = 6.807325990412687E-003

Tunes = 0.25410281000000 0.25543771000000
Chromaticities = 0.35339496460706E-14 0.41877303305936E-14
2nd order chromaticities = 2.0294221707359 1.2321830779883
Amplitude Tune Shift = 0.75435240816005 -1.2127166383901 -0.15728230356128

Results:

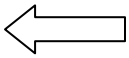
THIN LENS

.2

!FITBENDS no fitbend and no xbend

closed orbit
-0.235096546788369 -2.302434526186372E-002

METHOD 2 100 100
METHOD 4 0 0
METHOD 6 0 0
number of Slices 100
Total NST 102
Total NST due to Bend Closed Orbit 0
Total NST due to Sextupoles 0



<DBETA/BETA> = 6.199294623215892E-002
MAXIMUM OF DBETA/BETA = 0.113568889889915
<DETA/ETA> = 6.647929195195945E-002

Tunes = 0.25410281000000 0.25543771000000
Chromaticities = 0.35339496460706E-16 0.22970672699459E-15
2nd order chromaticities = 2.4035858199180 0.95593829901721
Amplitude Tune Shift = 0.69819531868702 -0.97075789688654
0.47858261086598E-01

RF cavity description by MADX(PTC)

For the PTC tracking the RFCAVITY element of MADX should be used will the following attribute: **no_cavity_totalpath**. In this case the particle is kept on the crest of RF voltage. LAG=0 for the case below the gamma-transition. To be sure that the RF parameters are set correctly**, it is necessary to check the single particle motion in the longitudinal phase plane.**

Example:

…

```
  BR.C02  : RFCAVITY, VOLT:= .008, HARMON:=1, L:=1.0, LAG:=0.0,
no_cavity_totalpath;
```

PART 2

TO RUN PTC-ORBIT code …

… it is necessary to perform the following preparations:

- create the FLAT file of the machine by using MADX-PTC code
- create the Time-tables for individual PTC elements to use the time-dependent elements of the machine *(it could be ANY elements of the machine lattice !*)
- use in the ORBIT scripts the existing PTC **script to energize** the lattice with the time-dependent elements
- use the 'FINAL_SETTING.txt' file with the proper setting of INITIAL time for the particle tracking, including the time dependence of the PTC elements
- create the 6D particle distribution (**generated** by ORBIT or **read** from the external file)
- include the Space charge module in the ORBIT script with the proper parameters for the space-charge simulations for each space-charge node
- include the Aperture definitions in the ORBIT script in required locations around the machine
- Tracking procedure, including the turn-by-turn ORBIT diagnostic
- OFF-LINE post-processing to analyze the obtained results

In the case of the combined PTC-ORBIT […] code the single/multi-particle tracking including the collective effects, implemented in the ORBIT code [..], can be made. The MPI version of the code allows to use multi-processor computer to minimize the simulation time. PTC in this combine code tracks particles through the PTC element, which could contain one or a few machine elements, depending on the 'LMAX' parameter, which define the length of the PTC element for the tracking procedure.

To create the required FLAT file **one can use just the PTC code alone** including very complicated machine layouts, machine errors (misalignment errors) and fitting of linear-nonlinear machine parameters. For people how knows the MADX code, MADX can be used to perform the machine description and to make required matching of the machine properties for different machine sets.

- **Create the FLAT file of the machine by using the MADX-PTC code**

The machine's FLAT file should contain the information about the machine model (exact=true or false), the integration and splitting methods.

→ ***… look above to get information about all these possibilities.***

- **Create the Time-tables for individual PTC elements to use the time-dependent elements of the machine**

The time-table can be created for ANY PTC elements, including different kind of magnets and RF cavities.

<u>Time-table for PTC magnets</u> (Example #2.1)

1st line	6 1 1
2nd	1
3rd	0.0 0.019056563971362087 0.0
4th	0.0002146667 0.017613039090306887 0.0
5th	0.0004146667 0.01472598932819656 0.0
6th	0.0006146667 0.010189196844880219 0.0
7th	0.0008146667 0.005033750841111647 0.0
8th	0.001014667 -0.00032791300280766105 0.0

1st line:

'6' → number of lines in the time-table with data, representing the time-variation of the field components of the PTC element
'1' → time unit in the table [sec]
'1' → number of field components of the PTC element to be changed according to the table

2nd line:

'1' → 'field index' <u>according to the PTC notation</u> to be changed by the time-table (*'1'* – dipole → $B_0/(B\rho)$; *'2'* - quadrupole, *'3'* – sextupole and so on … up to '20' for PTC … limited by the internal array)

from the 3rd line … up to the 8th line:

Time in units, defined above in this table| NORMAL component | SKEW component.

Time-table for RF cavities (Example #2.2)

1st line
2nd
3rd
4th

```
 2  1  1  0  2
 1  2
0.000 1.400 -0.024 0.0  0.0  0.0
0.010 1.400 -0.024 0.0  0.0  0.0
```

1st line:
'2' → number of lines in the time-table with data, representing the time-variation of the RF system
'1' → time unit in the table [sec]
'1' → percentage of the RF voltage, used by beam itself
'0' → 'ct' value for PTC
'2' → number of harmonics

2nd line:
'1 2' → number of harmonic (using the MADX-PTC setting as the basis)

from the 3rd line:
RF time table: Time, Wkin [GeV], V1rf [MeV], Δφ1 [rad], V2rf [MeV], Δφ2[rad]

REMARK

… CHECK the RF setting in the FLAT file (frequency should be correct!)

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ ELEMENT $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
         34 DUMMY_CAVITY              DUMMY_CAVITY:1
  1.00000000000000      F F  L,PERMFRINGE,MIS
 0.500000000000000       0.000000000000000E+000  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 CAVITY INFORMATION
 2.400000000000000E-002  3061380.72      0.000000000000000E+000  0.00000
```

The frequency in the flat file should corresponds to the required value, defind by the kinetic energy of the particle, the machine circumference and the fundamental harmonic number. The frequency is in Hz.

… to avoid any confusion please check the longitudinal particle motion

by using the single particle before you start the multi-particle tracking !!!

Energize the machine

By using the external tables one can change the setting of the machine elements as a function of time (for example, to create the time variation of the chicane magnets to vary the closed orbit distortion during the injection or extraction processes). Using the time-tables for some machine elements one can change the machine properties complitily (it means that the machine properties at the MADX(PTC) level would not be the same after the machine setting has been modified by NEW values, taken from the time-tables at an appropriate time). To set the table values correctly for the proper time it is necessary to use 'time' from the 'FINAL_SETTINGS.TXT' file. By using this file it's possible to start the tracking procedure, taken the setting of the machine elements at the proper time.

Example below illustrates the required file to energize the lattice for the case when the time dependent elements are introduced. The element of the machine (BR.BSW1) is changed by using the time table 'BS_ramp.TXT' with the scaling factor '1'. The RF system of the machine could have a few RF cavities with an individual table.

Example #2.3 (to energize the machine by using 'time' tables)

```
select layout                    <-- … use the existing layout '1'
1                                <-- … use TIME instead of S
+time                            <-- … use CAVITY during tracking
+cavity                          <-- … use 'RAMPING' of the machine elements
SET ORBIT RAMPING
set orbit state
SET ORBIT RESTORE
 f f           !!!  keeps the layout upgraded (no return to original)
ramp                                                  <-- 'Ramping' table
  BR.BSW1      "./Tables/BS_ramp.TXT"      1.000
initialize cavity
1          ! number of cavities                       <-- 'RF system' table
BR.C02,  "./Tables/RF_DoubleHarm.dat"
INITIAL TIME IN MY UNITS                              … set initial time from
 ! set the the RIGHT energy for the actual time      'FINAL_SETTINGS.TXT' file
energize ORBIT lattice
set orbit acceleration           <-- … set acceleration according to the RF table
return
```

Example #2.4 for the 'FINAL_SETTINGS.TXT' file

1st line	0.0 1.000000000000000E-003 0 f
2nd line	1
3rd line	0.000000000000000E+000

The 'FINAL_SETTINGS.TXT' file contains the following information, which is required by the PTC to define the correct setting of the machine's elements for the multi-particle tracking procedure with the time-dependent elements.

The 1st line contains:
(1) the initial time for the tracking (0.0)
(2) the time units (1.00000E-003) → [msec]
(3) the initial number of turns (0, if NO tracking has been done yet)
(4) the flag to use the 'time-patch' option of PTC (development version):
→ 'f' … FALSE … NO time patch … works without any additional 'time-patch' file
→ 't' … TRUE … to use the 'time-patch' file, generated by PTC standalone separately … this version is not the public one … under development.

The 2nd line contains:
- number of the RF cavities in the machine

The 3rd line contains:
- 'ct' value at the beginning of the tracking procedure (0.0 if NO tracking has been done yet).

The special ORBIT script can be created to manage the continuous tracking procedure by using the tracking procedure with some limited tracking time (like a few thousand turns). In this case the 'FINAL_SETTINGS.TXT' file becomes extremely important to set the correct initial time for tracking at the beginning of each run.

Example #2.5 to show the way to prepare the ORBIT script for the machine study by the single or/and milti-particle tracking (a part of the required script)

```
…..

//+++++++++++++++++++++++++++
// (1)  initialize PTC
//+++++++++++++++++++++++++++

if(nRank==0){
 cerr << " Read FLAT file " << "\n";
}

   InitPTC("PTC_ORBIT_FLAT.TXT");


ReadPTCscript("chrom.ptc");


// ORBIT_Nodes inform

showNodes(cout);


if(nRank==0){
 cerr << " Read FLAT file ... DONE! " << "\n";
}


////////////////////////////////////////////////////////
// (2)  PTC scripts for tracking WITH time-dependent elements
////////////////////////////////////////////////////////

   ReadPTCscript("PTC_RAMP_RFTable.ptc");


if(nRank==0){
 cerr << " Initialize the machine ... DONE! " << "\n";
}

   ReadPTCscript("twiss_script.ptc");

…
```

nRank==0 … to suppress the output if the processor index is not 0 (used for the multi-processor case)

READ the flat-file, prepared by PTC separately (look at Part 1)

Use the PTC script to get inform about the betatron tunes and chromaticity for the basic lattice

ORBIT command to print the machine layout as the set of the PTC elements (look at Part 1)

Use the PTC script to set the machine elements by using the time-tables at the proper time

Use the PTC script to produce the TWISS table by PTC

REFERENCES:

[1]

[2]

[3]

[4]