Section 6: Nonlinear Transverse Motion

- **Taylor Maps**
  - Calculating Taylor maps for Problem 8.1 and Problem 8.7
  - Building the Map tracker
- **Nonlinear magnetic field**
- **Simulation**
  - Demo
  - Examples

Calculating a Taylor Map

Problem 8.1 and Problem 8.7 (eq_fodo_tune.adxf, lattice: fullcell)

shell.map(Args() << Arg("order", 2) << Arg("print", "./map.out"));

monomial exponents

R_{11} R_{12} R_{16} T_{122}

:VTps : size = 6 (dimension = 6 order = 2)

	.3012167032873e-18	2.2290537236164e-20	0.0000000000000e+00	0.0000000000000e+00	-5.4327582421323e-21	0.0000000000000e+00	0	0	0	0	0	0
1	1.5658620810311e-01	-2.9573156020218e-01	0.0000000000000e+00	0.0000000000000e+00	-2.9209626540159e-03	0.0000000000000e+00	1	0	0	0	0	0
2	3.2985345181454e-	620810311e-01	0.0000000000000e+00	0.0000000000000e+00	-8.3304608623055e-03	0.0000000000000e+00	0	1	0	0	0	0
3	0	000000000e+00	1.5660179505890e-01	-1.3909957211795e+00	0.0000000000000e+00	0.0000000000000e+00	0	0	1	0	0	0
4	0	000000000e+00	7.0127884862014e-01	1.5660179505890e-01	0.0000000000000e+00	0.0000000000000e+00	0	0	0	1	0	0
5	0	00000e+00	0.0000000000000e+00	0.0000000000000e+00	0.0000000000000e+00	1.0000000000000e+00	0	0	0	0	1	0
6	8.3304608623055e-03	2.9209626540159e-03	0.				0	0	0	0	0	1
7	-9.2647109051344e+01	-3.9349374072313e-01	0				00	2	0	0	0	0
8	9.0521499493200e+01	-2.2444547191656e+00	(				00	1	1	0	0	0
9	0.0000000000000e+00	0.0000000000000e+00	-1				+00	1	0	1	0	0
10	0.0000000000000e+00	0.0000000000000e+00	-				e+00	1	0	0	1	0
11	0.0000000000000e+00	0.0000000000000e+00	(				+00	1	0	0	0	1
12	1.8307115021842e	730422688e-01	(				+00	1	0	0	0	1
13	1.2908172716425e	689639097e+02					e+00	0	2	0	0	0
14	0.0000000000000e+00	0.0000000000000e+00	-				e+00	0	1	1	0	0
15	0.0000000000000e+00	0.0000000000000e+00	-				e+00	0	1	0	1	0
16	0.0000000000000e+00	0.0000000000000e+00	(				+00	0	1	0	0	1
17	-4.1953077695525e+00	2.2715350863065e-02	(				+00	0	1	0	0	1
18	-2.4263203323705e+02	-1.1717666771136e+02					e+00	0	0	2	0	0
19	-4.2582470178324e+02	-1.4208159893621e+02					+00	0	0	1	1	0
20	0.0000000000000e+00	0.0000000000000e+00	(				+00	0	0	1	0	1
21	0.0000000000000e+00	0.0000000000000e+00					+00	0	0	1	0	1
22	-1.2910934788799e+02	-2.9086497633423e+01					+00	0	0	0	2	0
23	0.0000000000000e+00	0.0000000000000e+00	(				+00	0	0	0	1	1
24	0.0000000000000e+00	0.0000000000000e+00	1.9215223453260e-01	1.4031887538360e+00	-0.0000000000000e+00	0.0000000000000e+00	0	0	0	1	0	1
25	0.0000000000000e+00	0.0000000000000e+00	0.0000000000000e+00	0.0000000000000e+00	-0.0000000000000e+00	0.0000000000000e+00	0	0	0	0	2	0
26	0.0000000000000e+00	0.0000000000000e+00	0.0000000000000e+00	0.0000000000000e+00	-0.0000000000000e+00	0.0000000000000e+00	0	0	0	0	1	1
27	-1.0069979895763e-02	2.5246021644063e-03	0.0000000000000e+00	0.0000000000000e+00	-5.3434514658347e-03	0.0000000000000e+00	0	0	0	0	0	2

$$
\begin{aligned}
T_{111} &= -S_1\, l(1+ql) - S_2\, l(1-ql)^2, \\
T_{122} &= -S_2 l^3, \\
T_{166} &= -2\Delta\theta\, l(1+ql) - \frac{1}{4} S_2 \Delta\theta^2\, l^3, \\
T_{112} &= -2S_2\, l^2(1-ql), \\
T_{116} &= 4q^2 l^2 - S_2 \Delta\theta\, l(1-ql), \\
T_{126} &= -2ql^2 - S_2 \Delta\theta\, l^3, \qquad (8.47) \\
T_{211} &= -S_2(1-3q^2l^2+2q^4l^4), \\
T_{222} &= -S_2\, l^2(1-ql) - 2S_1\, l^2(1+ql)^2, \\
T_{266} &= -\Delta\theta(2-2ql-3q^2l^2) - \frac{1}{4} S_2\, l^2 \Delta\theta^2(1-ql) - 2S_1\, l^2 \Delta\theta^2 (1+\frac{1}{2}ql)^2, \\
T_{212} &= -2S_2\, l(1-ql)^2 - 2S_1\, l(1-2q^2l^2)(1+ql), \\
T_{216} &= 4q^2 l(1-\frac{3}{2}ql) - S_2 \Delta\theta\, l(1-ql)^2 - 2S_1\, l \Delta\theta(1+\frac{1}{2}ql)(1-2q^2l^2), \\
T_{226} &= 4q^2 l^2 - S_2\, l^2 \Delta\theta(1-ql) - 4S_1\, l^2 \Delta\theta(1+ql)(1+\frac{1}{2}ql).
\end{aligned}
$$

Map Tracker Class

./src/MapTracker.hh

```
class MapTracker : public TEAPOT::BasicTracker {

public:

  …

  /** Calculates a sector map */
  void setLatticeElements(const UAL::AcceleratorNode& lattice, int i0, int i1,
                  const UAL::AttributeSet& beamAttributes);

  /** Propagates bunch */
  void propagate(UAL::Probe& probe);

protected:

  /** map order */
  int m_order;

  /** sector length */
  double m_l;

  /** Taylor map */
  PacTMap m_map;
```

The setLatticeElements method of the Map Tracker

./src/MapTracker.cc

```
void UAL::USPAS::MapTracker::setLatticeElements(const UAL::AcceleratorNode& sequence,
                                                int is0, int is1, const UAL::AttributeSet& attSet)
{
  TEAPOT::BasicTracker::setLatticeElements(sequence, is0, is1, attSet);

  const PacLattice& lattice     = (PacLattice&) sequence;
  PAC::BeamAttributes ba = (PAC::BeamAttributes&) attSet;

  …
  // Propagate the sector map

  PacTMap sectorMap(6);
  int oldMltOrder = sectorMap.mltOrder(); // truncation

  sectorMap.mltOrder(m_order);
  s_teapot.trackMap(sectorMap, ba, is0, is1);

  sectorMap.mltOrder(oldMltOrder);

  setMap(sectorMap);
}
```

Index of the last sector element + 1

Index of the first sector element

Nonlinear Magnetic Field

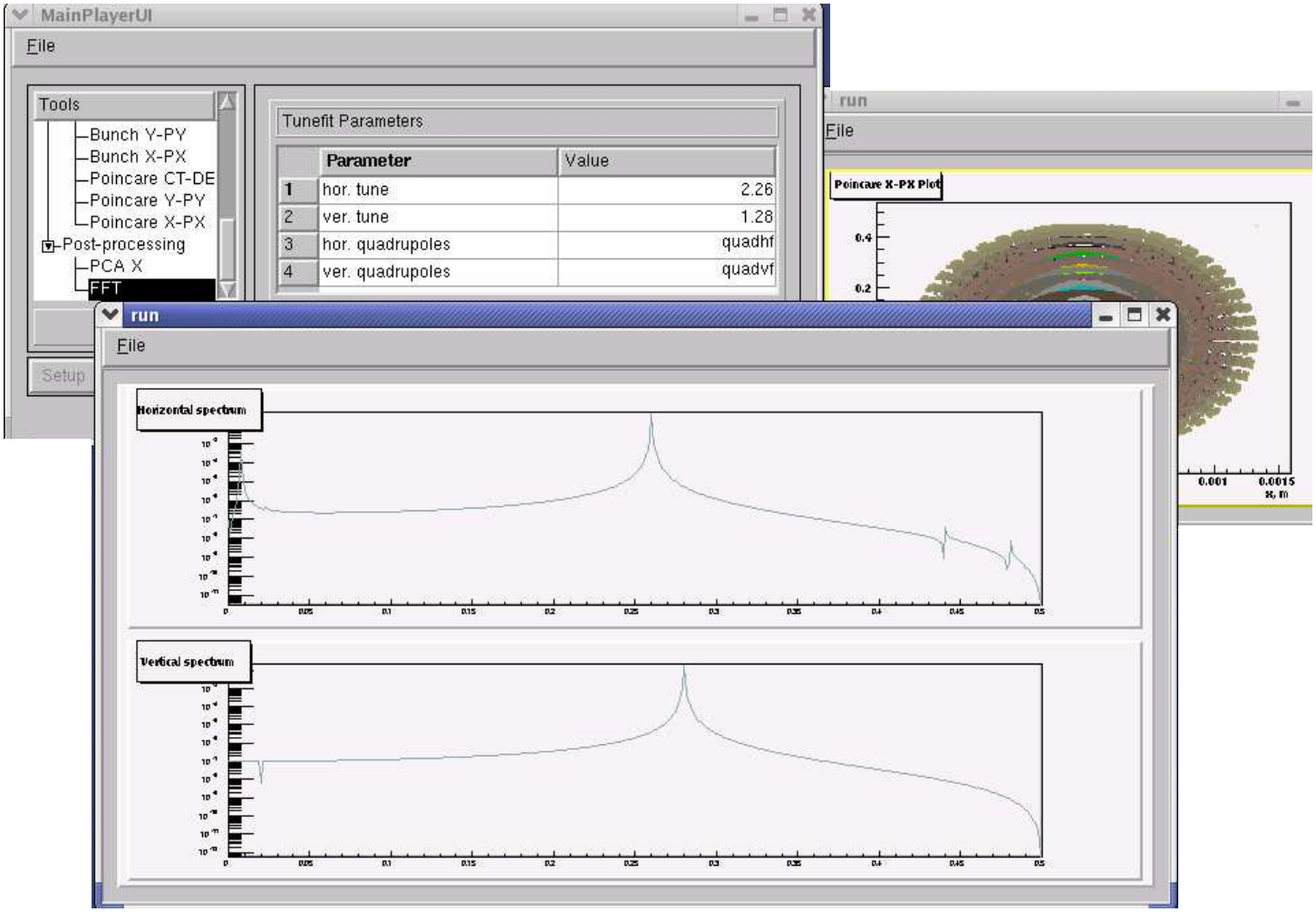
/lattices/general_fodo.mfield.adxf

```
<elements>
  …
  <kicker    name="kickerp" />
   <multipole name="mlt1" >
       <mfield a ="0.0 0.0 0.00" b="0.0 0.0 20.0" />
  </multipole>
   <quadrupole name="
   <quadrupole name="
  …
</elements>
<!--  -->
<sectors>
 …
 <sector name="ring"   line ="mbegin kickerp mlt1
                              fullcell fullcell fullcell fullcell fullcell
                              fullcell fullcell fullcell fullcell fullcell
                              mend"/>
</sectors>
</adxf>
```

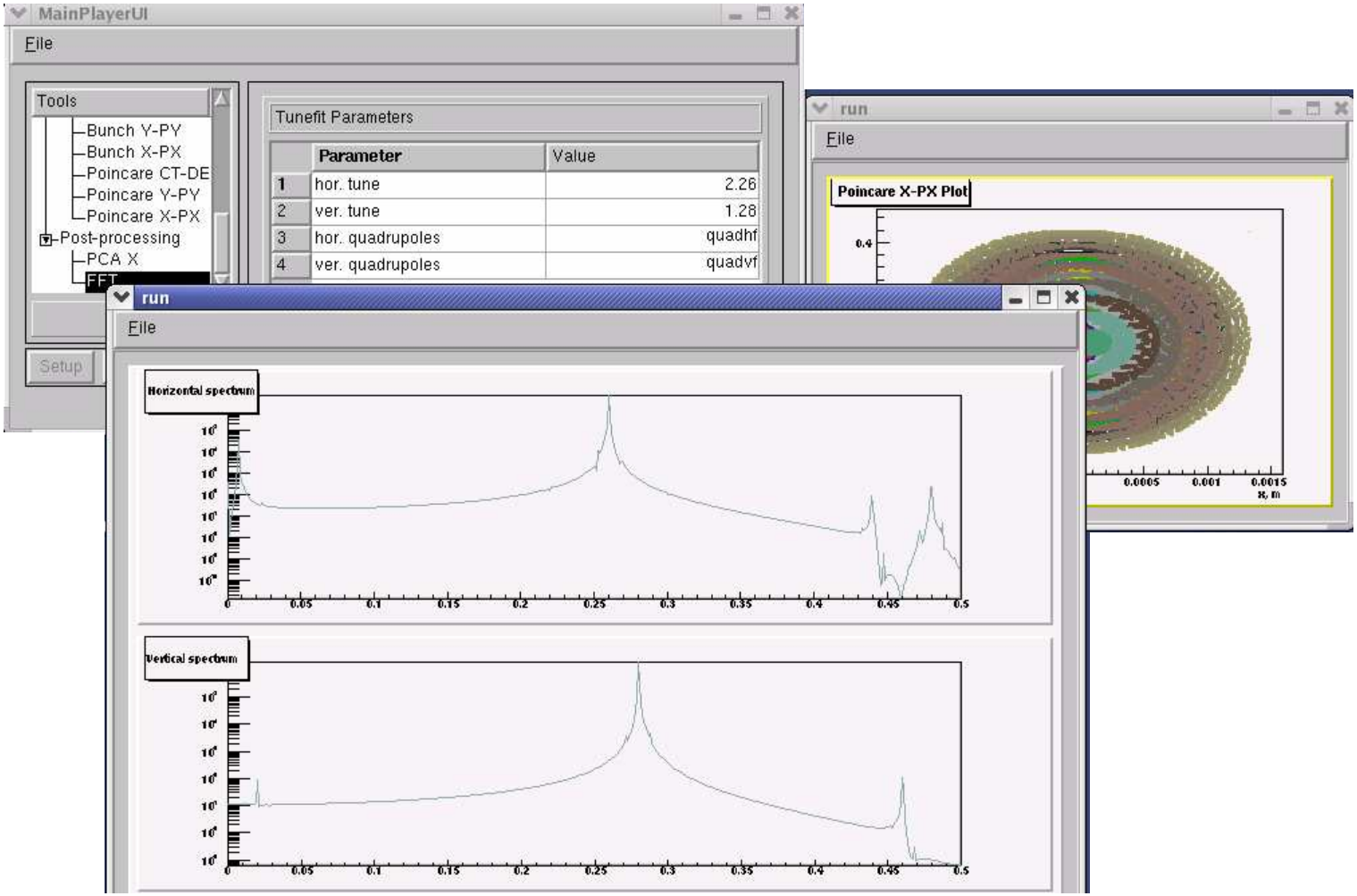
In ADXF, as in most lattice descriptions, magnetic field deviations are expressed as *multipole series*. For a bending magnet the magnetic field, as well as the effect of transverse positioning deviations $(\Delta x, \Delta y)$, are expressed as a (complex) series:

$$(B_y + iB_x)l_m = B_0 l_m \sum_{n=0}^{M} (b_n + ia_n)\big((x - \Delta x) + i(y - \Delta y)\big)^n. \qquad (8.13)$$

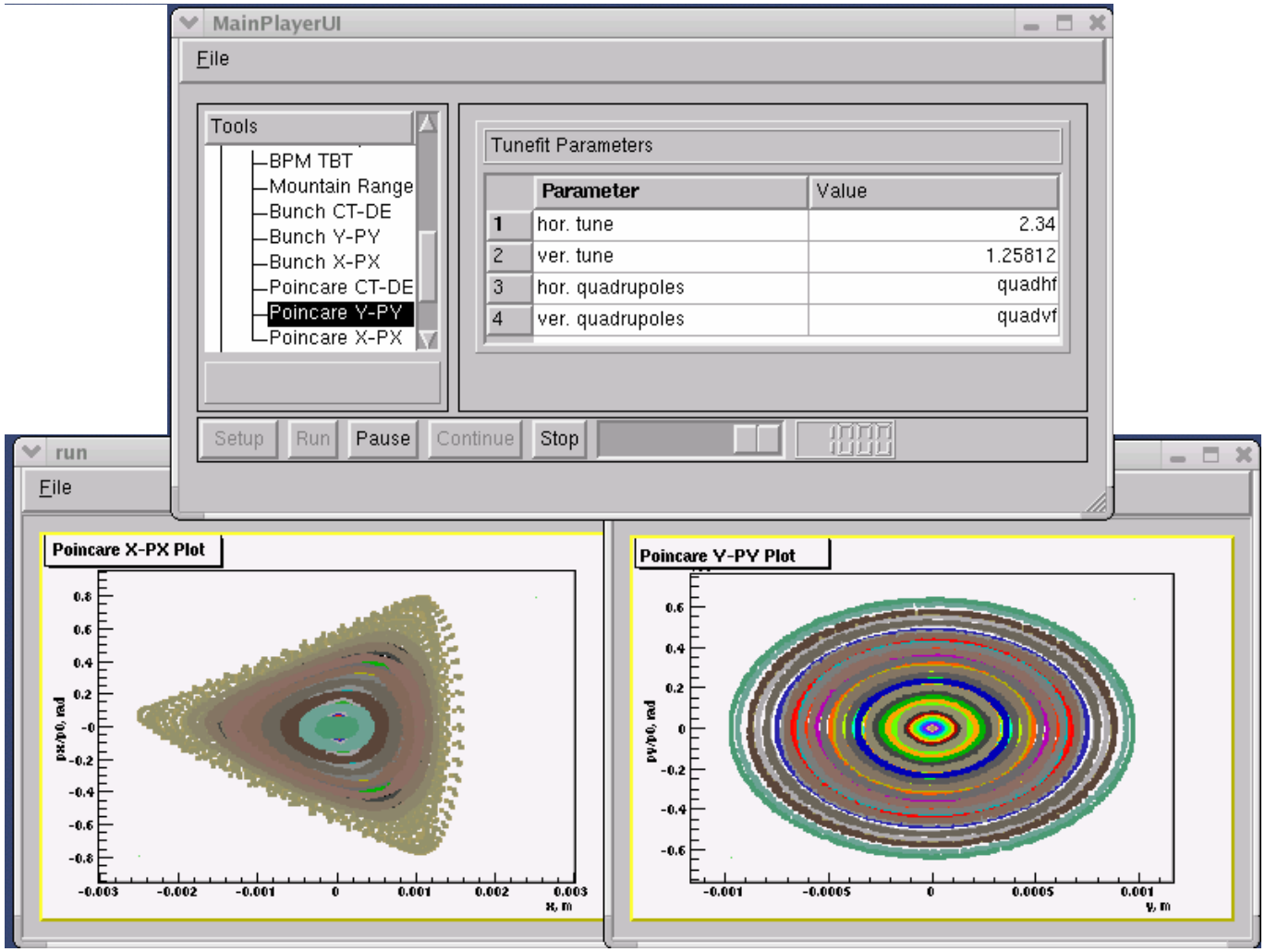
FFT Spectra of the general_fodo.adxf lattice



FFT Spectra of the general_fodo.mfield.adxf lattice



Poincare Plots near the 3rd order resonance



Simulation Examples

- **S 6.1**: Calculate one-turn maps used in Problem 8.1 and Problem 8.7
- **S 6.2:** Vary magnet multipole coefficients in the general_fodo.mfield.adxf file, run the nonlinear application with the Teapot element-by-element tracking, and identify picks in FFT spectra.
- **S 6.3**: Investigate the beam motion near the 3rd order resonance.
- **S 6.4:** Investigate the beam motion near the 4th order resonance.
- **S 6.5:** Repeat the 6.2-6.3 applications with the Map Tracker.