

v.16.1.19  
01/13/22

Coordinate assignment is important in the case of  
"absolute path" envelope  
Because dipole angle is applied in the X-Z plane

See 3D-envelope (deviations from the central axis)  
[http://lise.nsl.msu.edu/15/15\\_23\\_envelope3D.pdf](http://lise.nsl.msu.edu/15/15_23_envelope3D.pdf)

Monte Carlo calculation of fragment transmission

What isotope transmission to calculate?  
 One fragment of interest. Chose manually here  
 Group of isotopes already calculated by the Distribution method (Ncalc = 0)  
 List of isotopes from file to produce inside target  
 Input ions rays from file emitted from target

2-D  **2-D & 3-D**  ?

A: 48, Element: Ca, Z: 20  
 Stable  
 Charge states: 20+ PS1A  
 Reaction mechanism: Projectile Fragmentation

MC transmission options  
 "Distribution" calculation

Monte Carlo calculation (2D&3D-plots)

X-coordinate After BLOCK: Drift IC2 as Y  
 X mm  
 X (T) mrad  
 Y mm  
 Y (P) mrad  
 dP/P %  
 Radial f(X,Y) mm  
 Angle f(X,Y) mrad  
 Energy MeV/u  
 TKE MeV  
 Momentum MeV/c  
 Brho T\*m  
 Erho MJ/C  
 **Envelope** m  
 Energy Loss MeV  
 Range mm  
 Energy Deposition MeV/mm/particle  
 Time of flight ns  
 Length m  
 Stripper: Start  
 Scint1: Stop  
 Velocity: Velocity\_Z [cm/ns]  
 Ion parameter (M, Z, q...): A (mass number)

Y-coordinate After BLOCK: Drift IC2 as X  
 **X** mm  
 X (T) mrad  
 Y mm  
 Y (P) mrad  
 dP/P %  
 Radial f(X,Y) mm  
 Angle f(X,Y) mrad  
 Energy MeV/u  
 TKE MeV  
 Momentum MeV/c  
 Brho T\*m  
 Erho MJ/C  
 Envelope m  
 Energy Loss MeV  
 Range mm  
 Energy Deposition MeV/mm/particle  
 Time of flight ns  
 Length m  
 Stripper: Start  
 Scint1: Stop  
 Velocity: Velocity [cm/ns]  
 Ion parameter (M, Z, q...): Z-q

Z-coordinate After BLOCK: Drift IC2 as X  
 X mm  
 X (T) mrad  
 **Y** mm  
 Y (P) mrad  
 dP/P %  
 Radial f(X,Y) mm  
 Angle f(X,Y) mrad  
 Energy MeV/u  
 TKE MeV  
 Momentum MeV/c  
 Brho T\*m  
 Erho MJ/C  
 Envelope m  
 Energy Loss MeV  
 Range mm  
 Energy Deposition MeV/mm/particle  
 Time of flight ns  
 Length m  
 Stripper: Start  
 Scint1: Stop  
 Velocity: Velocity [cm/ns]  
 Ion parameter (M, Z, q...): A-2q

Gate 1: no gate  
 Gate 2: no gate  
 Gate 3: no gate  
 Gate 4: no gate

**ENVELOPE**

3D Monte Carlo Envelope

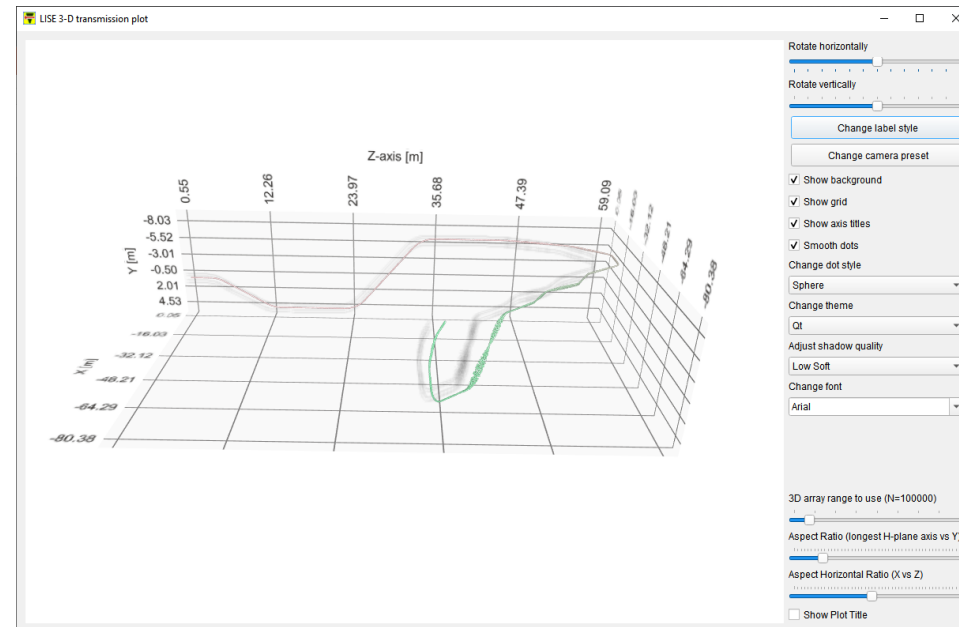
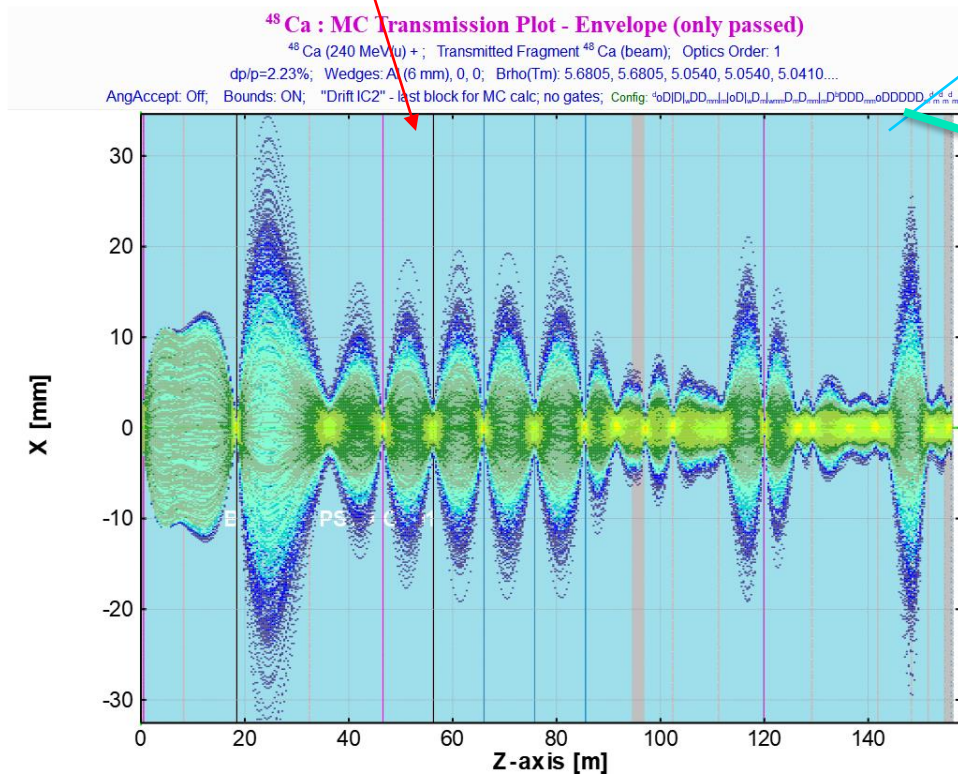
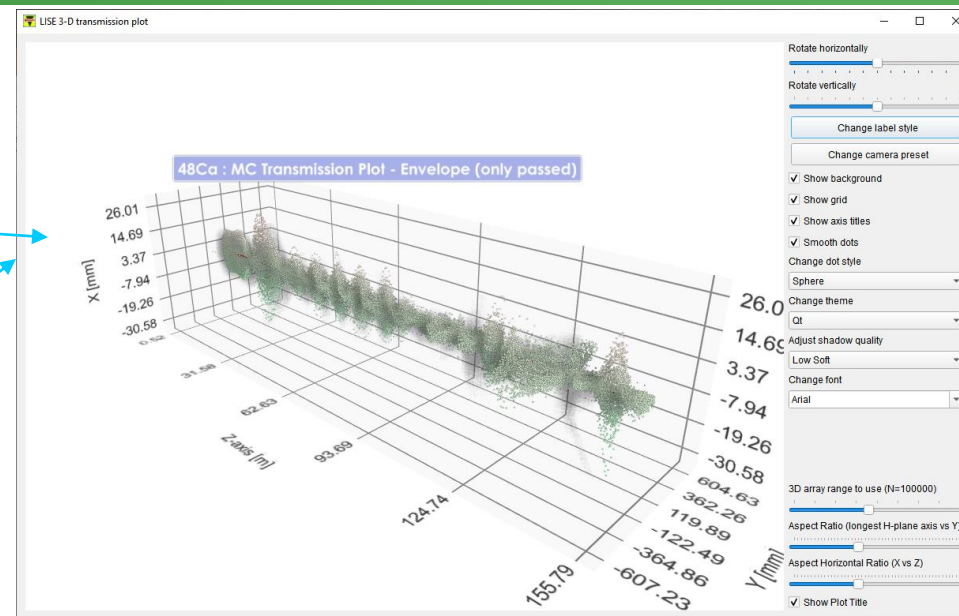
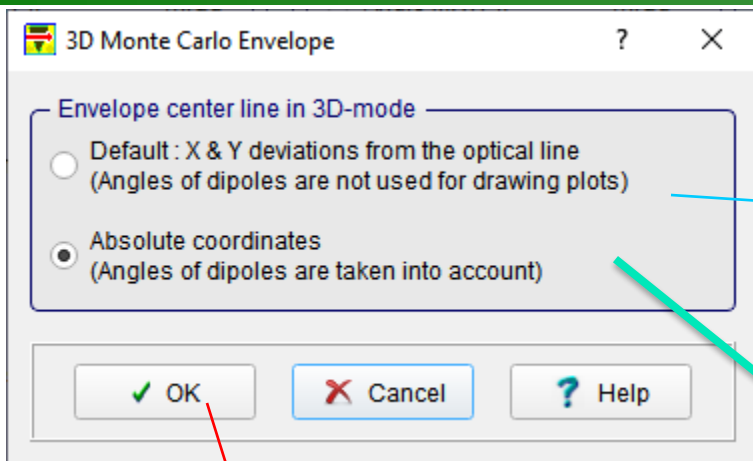
Envelope center line in 3D-mode

Default: X & Y deviations from the optical line (Angles of dipoles are not used for drawing plots)

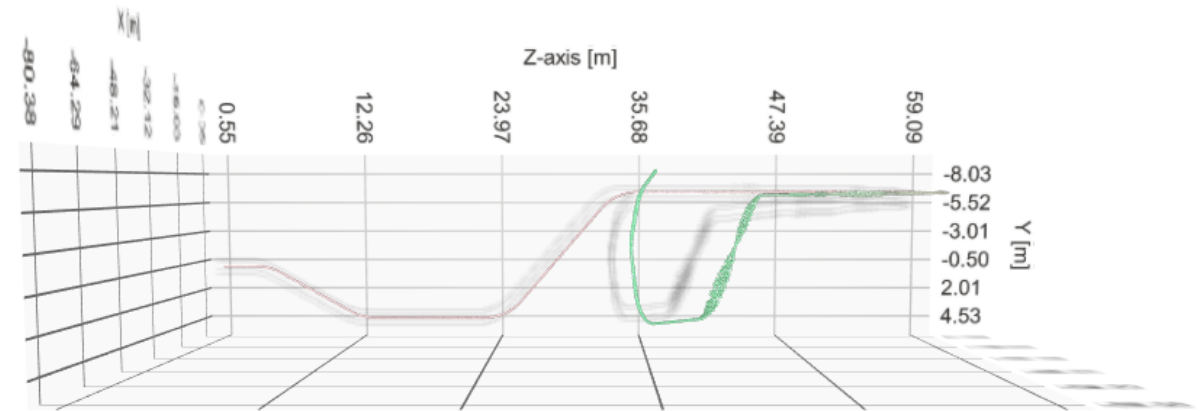
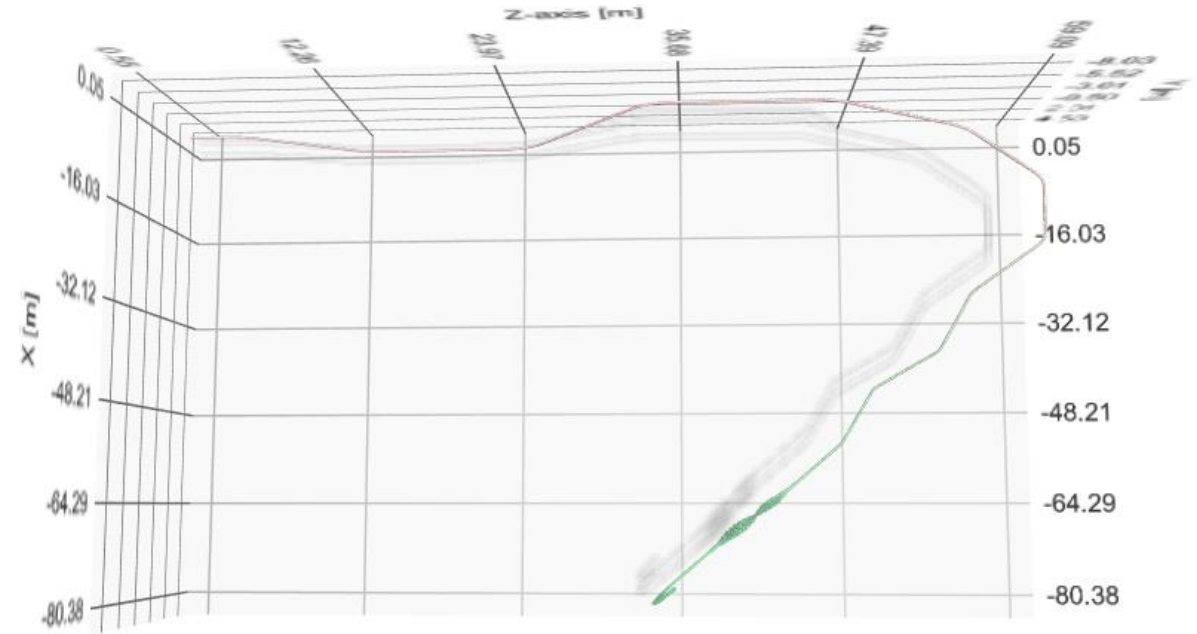
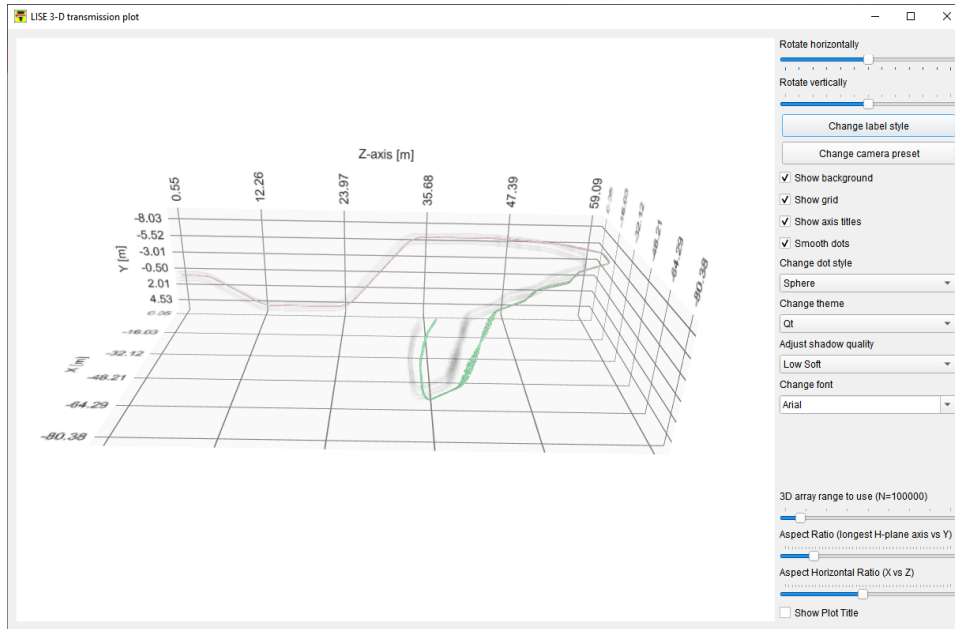
**Absolute coordinates** (Angles of dipoles are taken into account) **new**

OK Cancel Help

# 3D envelope plots



# 3D envelope “absolute path” → ARIS+S800





```

typedef QList<Vector3>      arrayVector3;
typedef QListIterator<Vector3> aV3_iter;

typedef QList<double>      arrayDouble;
typedef QListIterator<double> aD_iter;
//=====

class AbsolutePath3D
{
public:
    AbsolutePath3D();
    ~AbsolutePath3D();

    void init_base(distribution *init);
    void makePath();

    arrayVector3 Location;
    arrayVector3 Direction;
    arrayVector3 Propagation;

    arrayDouble Distance;
    distribution2 *base;

    Vector3 transform2Absolute(Vector3 &mc);

```

```

//=====
AbsolutePath3D::AbsolutePath3D() : Dimension()
{
    base = nullptr;
//-----
    Location.append(Vector3(0,0,0));
    Propagation.append(Vector3(0,1,0));
    Direction.append(Vector3(0,0,1));
    Distance.append(0);
}
//=====

//=====
Vector3 AbsolutePath3D::transform2Absolute(Vector3 &mc)
{
    int i=0;

    aV3_iter iterL(Location); iterL.toFront();
    aV3_iter iterD(Direction); iterD.toFront();
    aV3_iter iterP(Propagation); iterP.toFront();

    while(i<Dimension-1) {...}

    Vector3 location = (Vector3&)(iterL.next());
    Vector3 direction = (Vector3&)(iterD.next());
    Vector3 propagation = (Vector3&)(iterP.next());

    MATRIX m(3,3);
    Transformation3D(m, direction, propagation);

    Vector3 mc0(mc.x(),mc.y(),0);
    MATRIX v1 = m * mc0;
    Vector3 mc1(v1.a[0][0], v1.a[1][0], v1.a[2][0]);
    //mc1.printDebugVector("transform mc1-a");
    mc1 += location;
    //mc1.printDebugVector("transform mc1-b");
    V3buffer = mc1;

    return V3buffer;
}
//=====

```

The new direction after dipole rotation is based on [Rodrigues's rotation formula](#) assuming the X-Z plane is the dipole rotation plane

Vector3 from the "3D regular" envelope (X & Y deviations from the central optical axis)

=====

## Based on Q3DScatter class

### 1. Absolute3D path : improving performance

Currently is not effective for visualization transmission issues

- manual axis range
- maximum zoom properties

### 2. Absolute3D path : scheme with multipole objects

- a. manual drawing by points using the current LISE 3D algorithm  
(user lines in Q3DScatter ?)
- b. using custom objects implementation technology

=====

## "OpenGL" [QtGui]

QVector3D : manual drawing (includes lines)

- Take into account the A1900 beam dump visualization source

**3D envelope “central line” → A1900 (“special” beam)**

**EXAMPLES**

# 3D envelope “central line” → A1900 (“special” beam)

Emittance [#1]

Beam CARD (sigma, semi-axis, half-width...)

1D - shape (Distribution method)

2D mode

2D - shape (Monte Carlo method)

Correlated with

dimension mm  cm

respect to axis

dX 0 mm

dT 5 mrad

dY 0 mm

dP 0 mrad

dT 0.286 degree

dP 0 degree

1. X mm 0.3 Gaussian

2. T mrad 10 Rectangle uniform  Ellipse hollow  D

3. Y mm 0.3 Gaussian

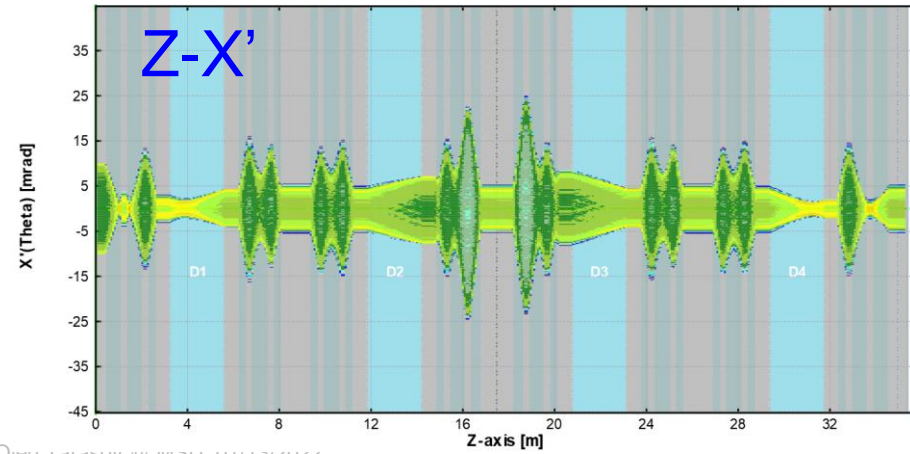
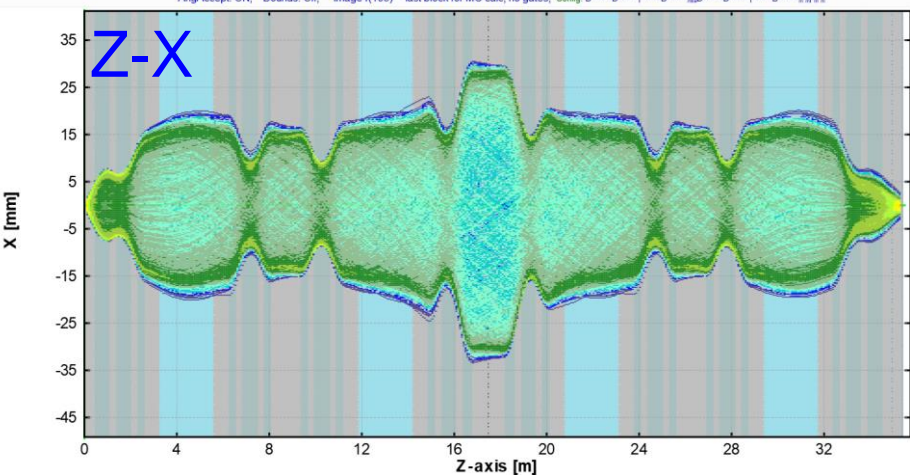
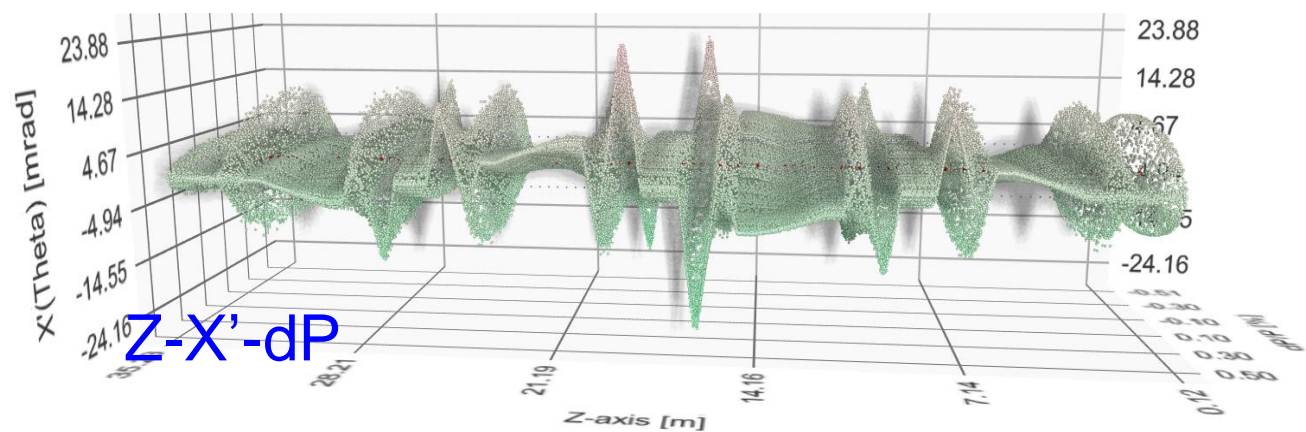
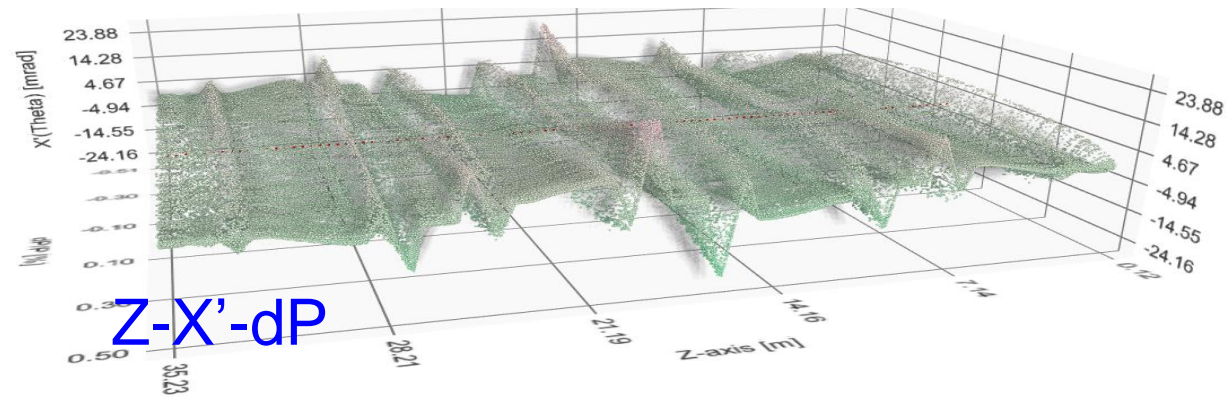
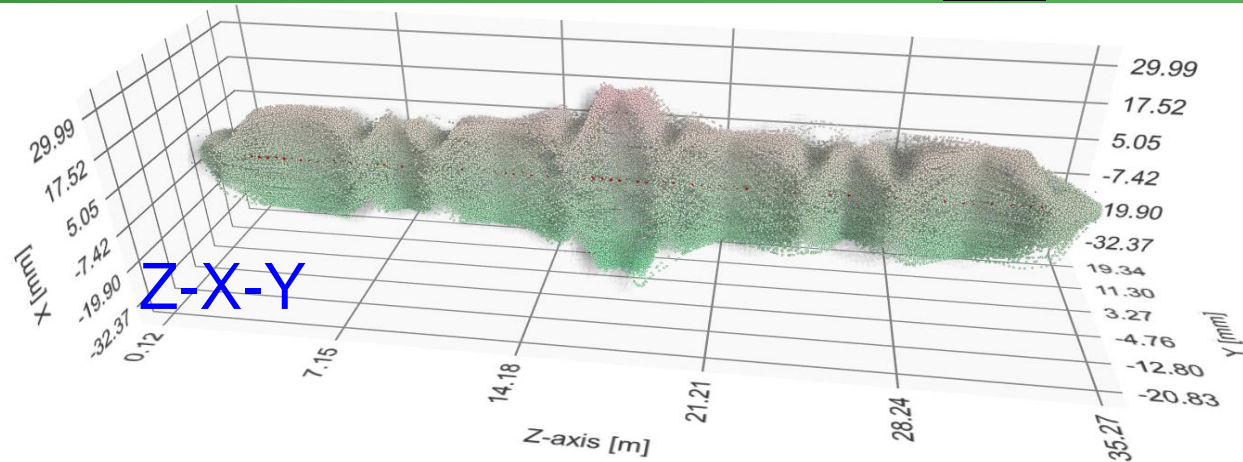
4. P mrad 1 Gaussian

5. L mm 0 Gaussian

6. D % 1 Rectangle uniform  Ellipse hollow  T

dp=5.07%; Wedges: 0; Bho(Tm): 4.2341, 4.2341, 4.2341, 4.2340, 4.2340...

AngAccept: ON; Bounds: Off; "image(105)" - last block for MC calc; no gates; Conf:

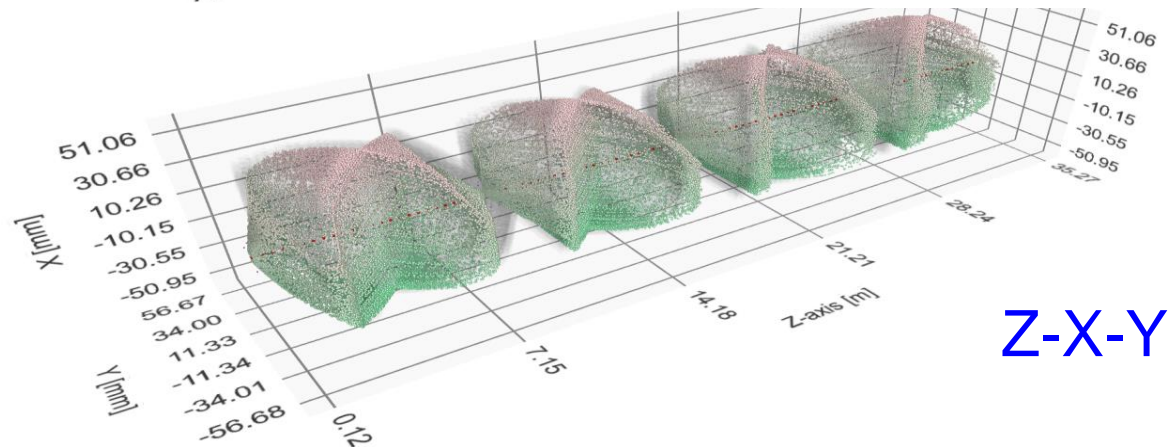
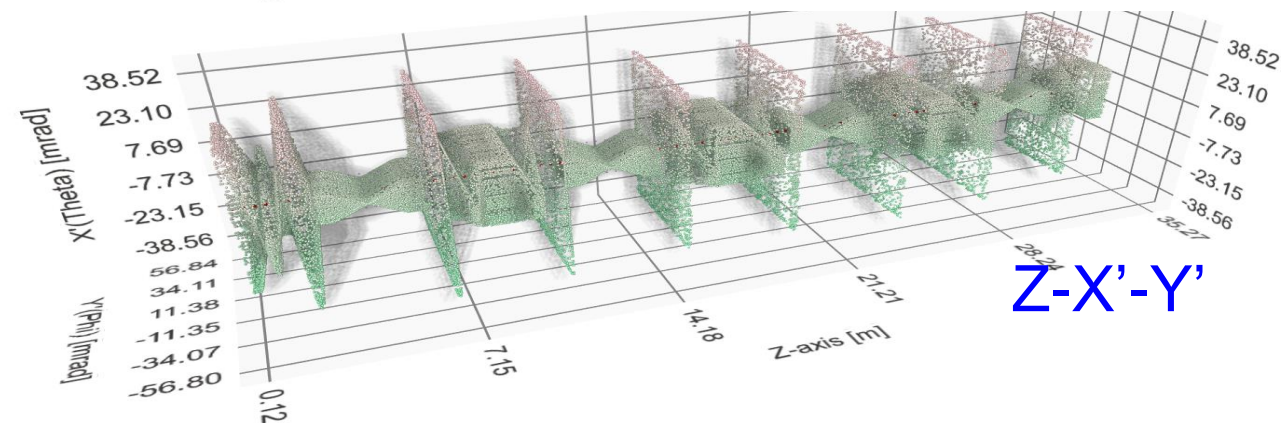
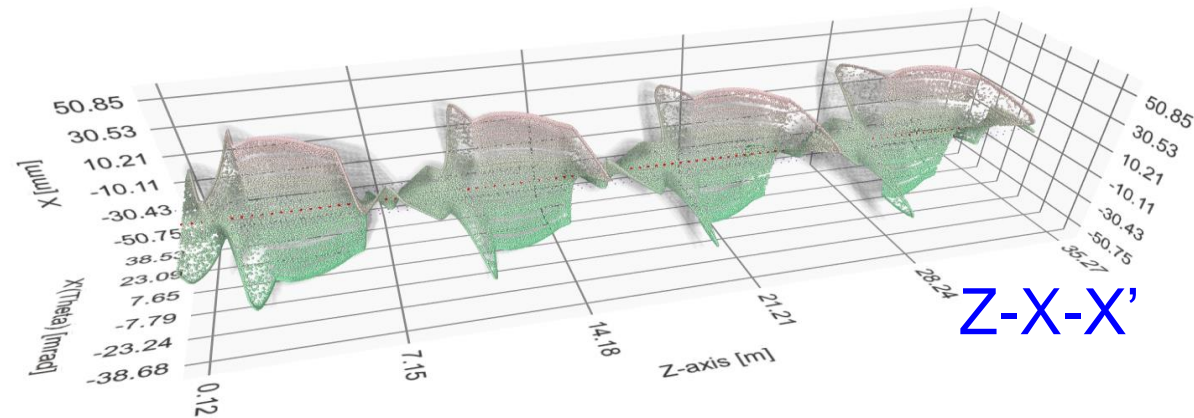
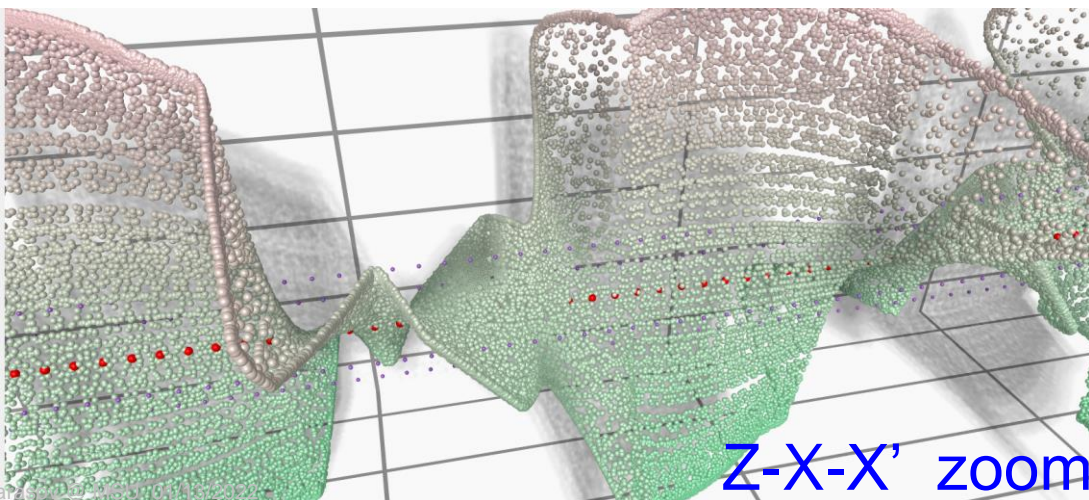
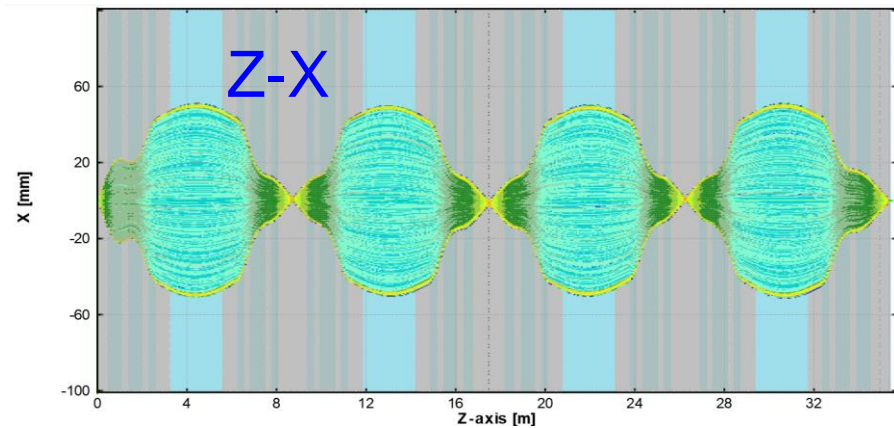


# 3D envelope “central line” → A1900 (“special” beam)

Emittance [#1]					
	Beam CARD (sigma, semi-axis, half-width...)	1D - shape (Distribution method)	2D mode	2D - shape (Monte Carlo method)	Correlated with
1. X	mm 0.1	Gaussian	<input type="checkbox"/>		
2. T	mrad 30	Rectangle uniform	<input checked="" type="checkbox"/>	Rectangle hollc	P
3. Y	mm 0.1	Gaussian	<input type="checkbox"/>		
4. P	mrad 20	Rectangle uniform	<input checked="" type="checkbox"/>	Rectangle hollc	T
5. L	mm 0	Gaussian	<input type="checkbox"/>		
6. D	% 0.01	Rectangle uniform	<input type="checkbox"/>		

dimension	
mm	<input checked="" type="radio"/>
cm	<input type="radio"/>
respect to axis	
dX	0 mm
dT	0 mrad
dY	0 mm
dP	0 mrad
dL	0 degree
dD	0 degree

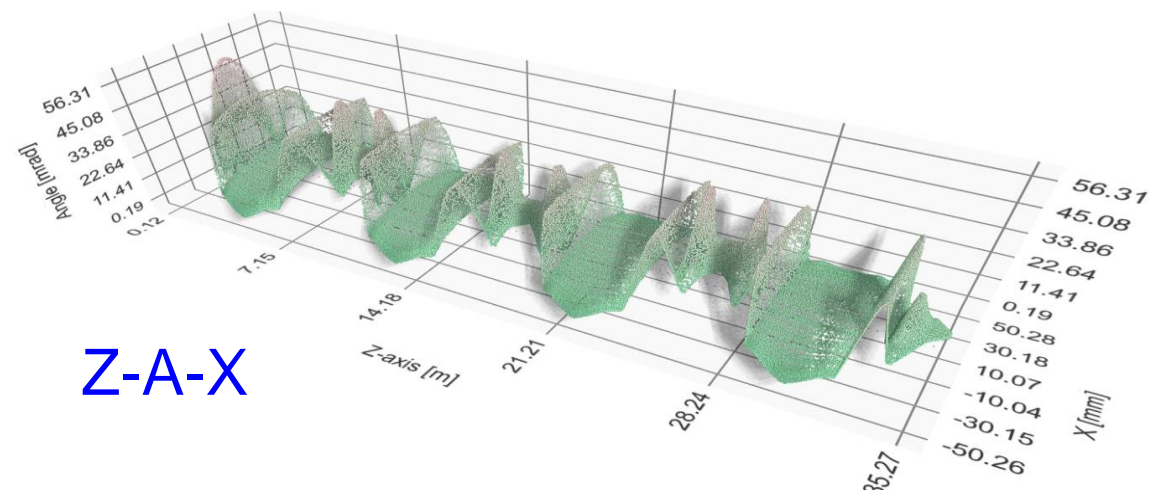


# 3D envelope “central line” → A1900 (“special” beam)

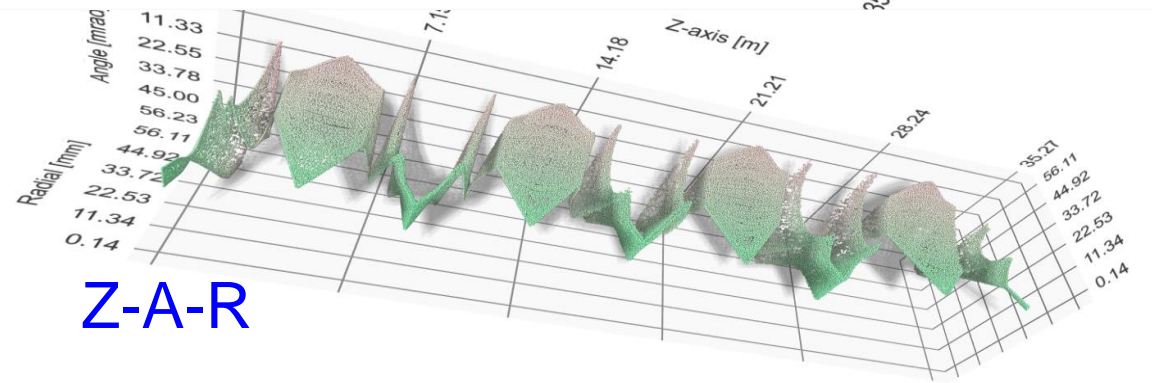
Emittance [#1]						dimension	
	Beam CARD (sigma, semi-axis, half-width...)	1D - shape (Distribution method)	2D mode	2D - shape (Monte Carlo method)	Correlated with	mm	cm
1. X	mm 0.1	Rectangle uniform	<input type="checkbox"/>			<input checked="" type="radio"/>	<input type="radio"/>
2. T	mrad 30	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	P	<input type="radio"/>	<input type="radio"/>
3. Y	mm 0.1	Rectangle uniform	<input type="checkbox"/>			<input checked="" type="radio"/>	<input type="radio"/>
4. P	mrad 20	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	T	<input type="radio"/>	<input type="radio"/>
5. L	mm 0	Gaussian	<input type="checkbox"/>			<input type="radio"/>	<input type="radio"/>
6. D	% 0.01	Rectangle uniform	<input type="checkbox"/>			<input type="radio"/>	<input type="radio"/>

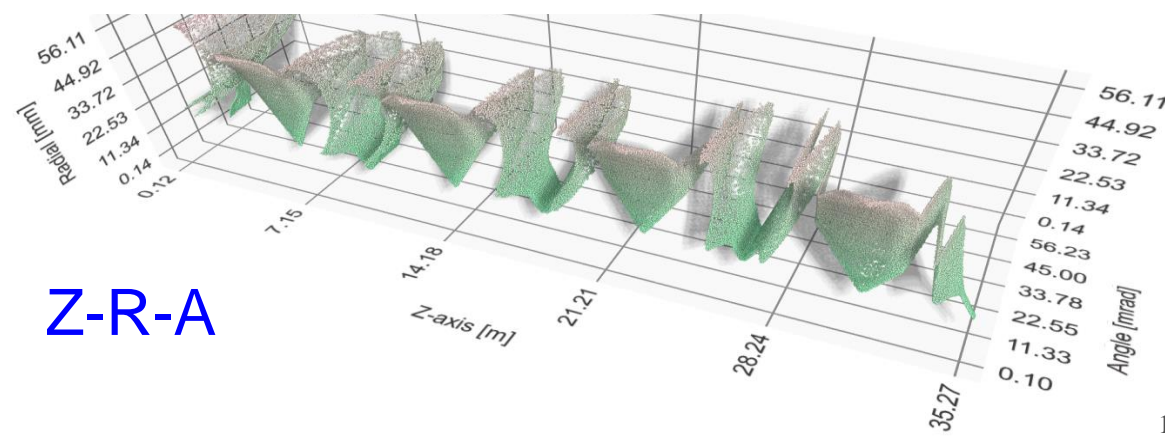
respect to axis		
dX	0	mm
dT	0	mrad
dY	0	mm
dP	0	mrad
dL	0	degree
dD	0	degree



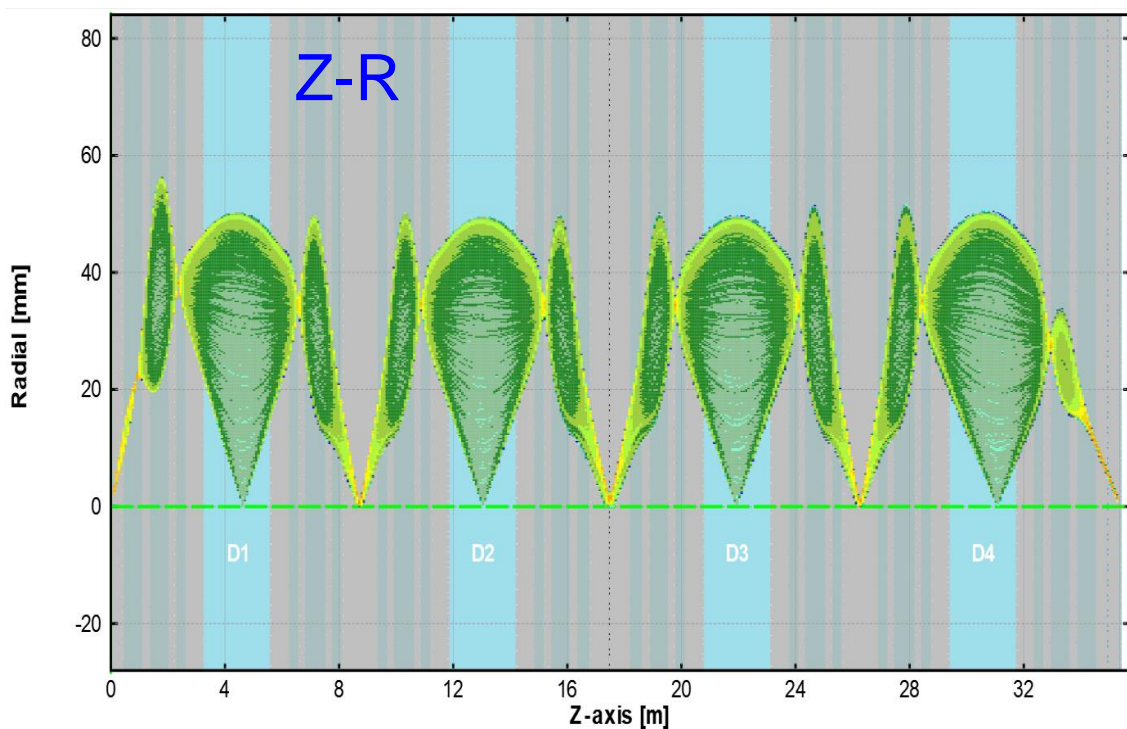
Z-A-X



Z-A-R



Z-R-A



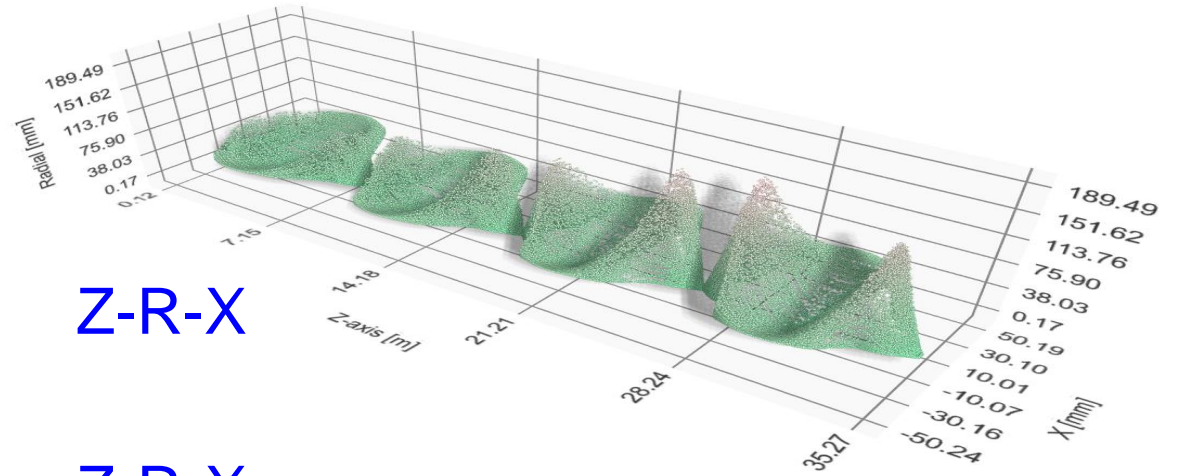
Z-R

# 3D envelope “central line” → A1900 (“special” beam)

Emittance [#1]	Beam CARD (sigma, semi-axis, half-width...)	1D - shape (Distribution method)	2D mode	2D - shape (Monte Carlo method)	Correlated with
1. X	mm 0.1	Rectangle uniform	<input type="checkbox"/>		
2. T	mrad 30	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	Y
3. Y	mm 7	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	T
4. P	mrad 20	Rectangle uniform	<input type="checkbox"/>		
5. L	mm 0	Gaussian	<input type="checkbox"/>		
6. D	% 0.01	Rectangle uniform	<input type="checkbox"/>		

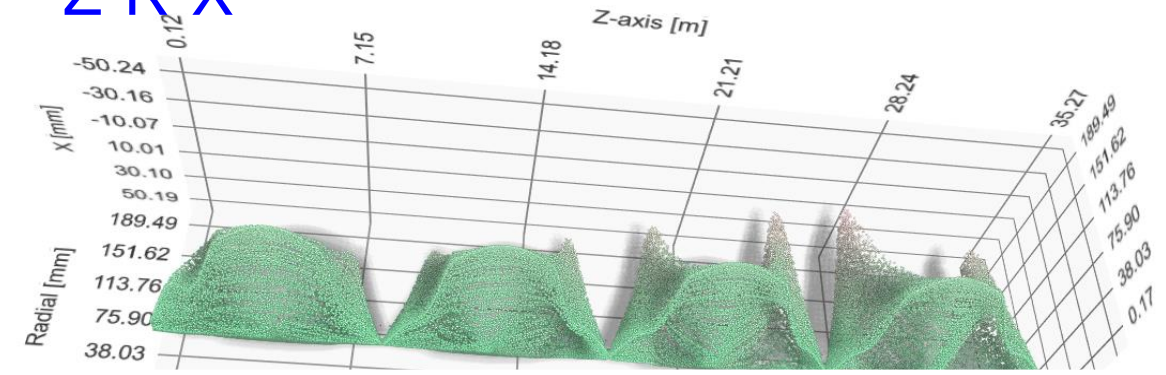
  

dimension	mm	cm
dX	0	mm
dT	0	mrad
dY	0	mm
dP	0	mrad
dT	0	degree
dP	0	degree

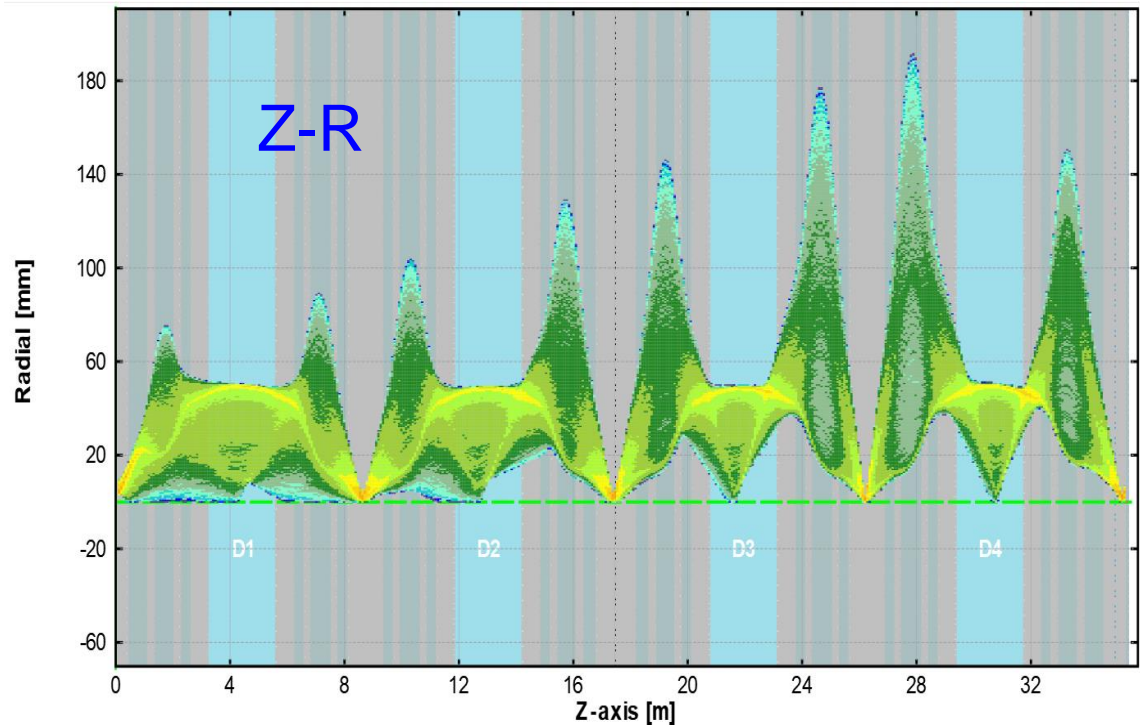
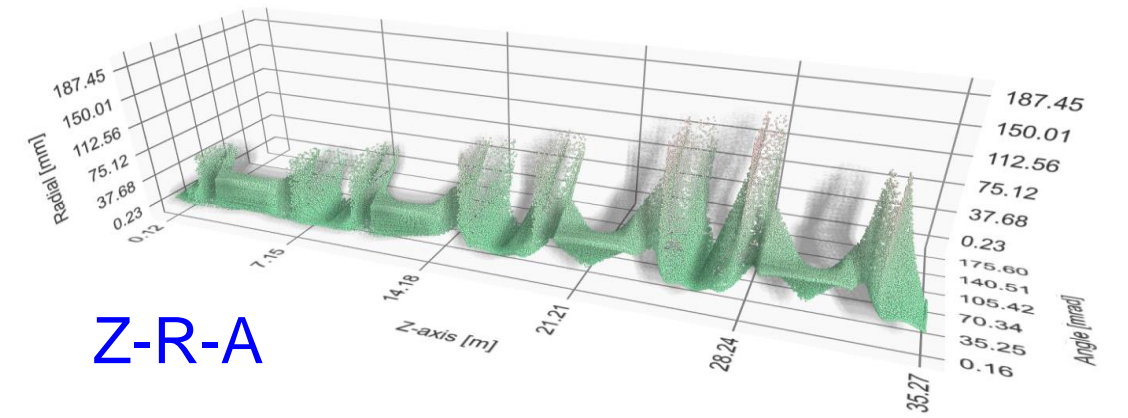


Z-R-X

Z-R-X



Z-R-A



# 3D envelope “central line” → A1900 (“special” beam)

Emittance [#1]	Beam CARD (sigma, semi-axis, half-width...)	1D - shape (Distribution method)	2D mode	2D - shape (Monte Carlo method)	Correlated with
1. X	mm 7	Rectangle uniform	<input checked="" type="checkbox"/>	Rectangle hollc	Y
2. T	0.1	Rectangle uniform	<input type="checkbox"/>		
3. Y	mm 7	Rectangle uniform	<input checked="" type="checkbox"/>	Rectangle hollc	X
4. P	0.1	Rectangle uniform	<input type="checkbox"/>		
5. L	0	Gaussian	<input type="checkbox"/>		
6. D	0.01	Rectangle uniform	<input type="checkbox"/>		

dimension: mm  cm

respect to axis:

dX 0 mm

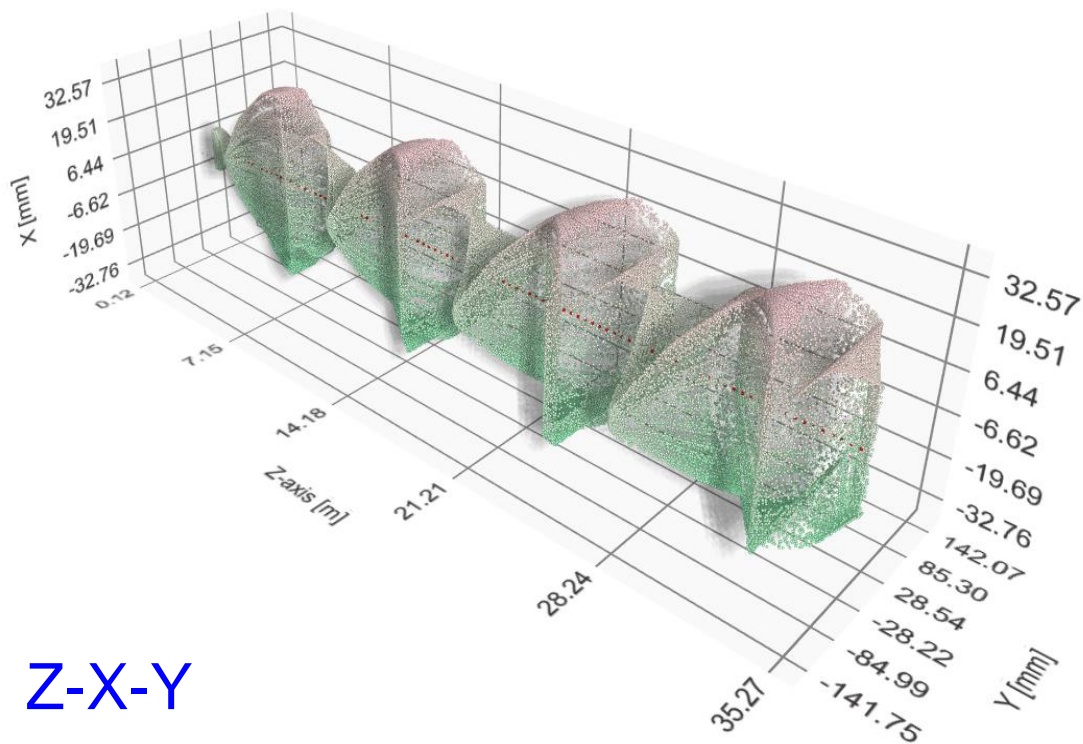
dT 0 mrad

dY 0 mm

dP 0 mrad

dT 0 degree

dP 0 degree



Emittance [#1]	Beam CARD (sigma, semi-axis, half-width...)	1D - shape (Distribution method)	2D mode	2D - shape (Monte Carlo method)	Correlated with
1. X	mm 0.1	Rectangle uniform	<input type="checkbox"/>		
2. T	30	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	D
3. Y	7	Rectangle uniform	<input type="checkbox"/>		
4. P	0.1	Rectangle uniform	<input type="checkbox"/>		
5. L	0	Gaussian	<input type="checkbox"/>		
6. D	2	Rectangle uniform	<input checked="" type="checkbox"/>	Ellipse hollow	T

dimension: mm  cm

respect to axis:

dX 0 mm

dT 0 mrad

dY 0 mm

dP 0 mrad

dT 0 degree

dP 0 degree

