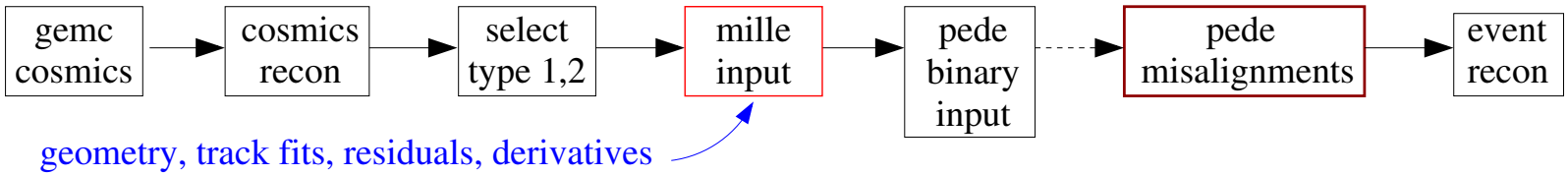
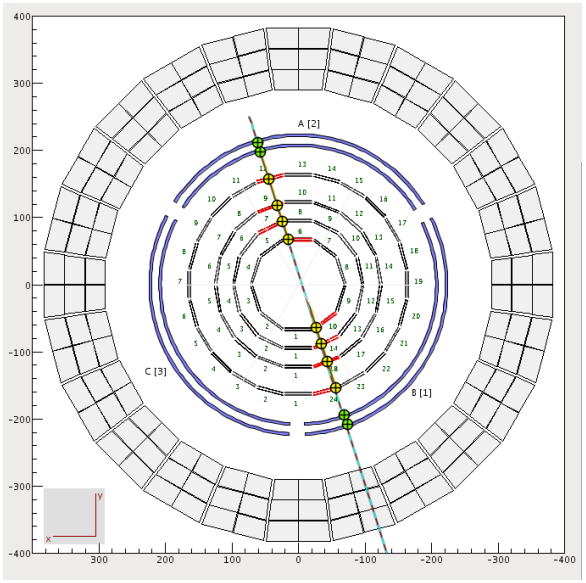
SVT Track-Based Alignment - Type 1 and 2 tests

1. Sequence of steps for track-based alignment with `millepede`.



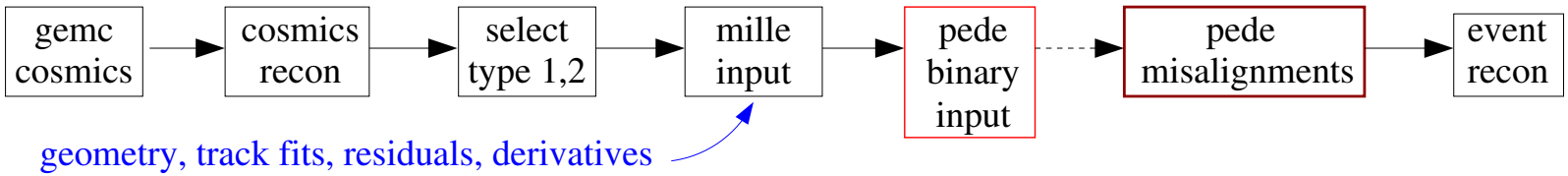
2. Extend code for all SVT topologies - generalized algorithms.
    1. Type 1 and Type 2 comparisons and visualizations validate code.
    2. beta version of Mille input code done.
    3. Writing pede binary input code now.

layer	sector	VZ (mm)	GG(mm)	JG (mm)
1	6	-0.1101	-0.1101	-0.1101
1	10	0.1940	0.1940	0.1940
2	10	0.1083	0.1083	0.1083
2	6	-0.0965	-0.0965	-0.0965
3	14	0.1717	0.1717	0.1717
3	7	-0.1631	-0.1631	-0.1631
4	7	-0.1596	-0.1596	-0.1596
4	14	0.1624	0.1624	0.1624
5	18	0.1665	0.1665	0.1665
5	9	-0.0984	-0.0984	-0.0984
6	9	-0.1023	-0.1023	-0.1023
6	18	0.1824	0.1824	0.1824
7	24	0.1686	0.1686	0.1686
7	12	-0.0845	-0.0845	-0.0845
8	12	-0.0925	-0.0925	-0.0925
8	24	0.2271	0.2271	0.2271



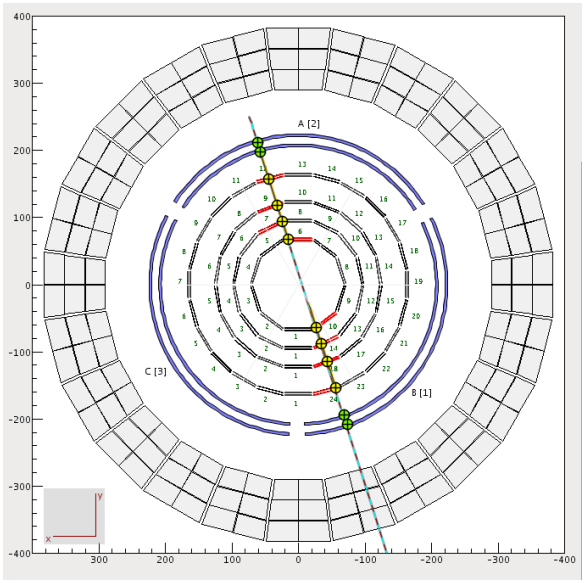
SVT Track-Based Alignment - Type 1 and 2 tests

1. Sequence of steps for track-based alignment with `millepede`.



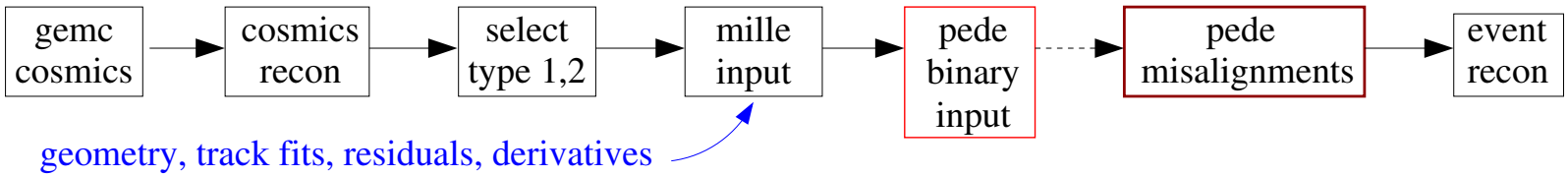
2. Extend code for all SVT topologies - generalized algorithms.
    1. Type 1 and Type 2 comparisons and visualizations validate code.
    2. beta version of Mille input code done.
    3. Writing pede binary input code now.

layer	sector	VZ (mm)	GG(mm)	JG (mm)
1	6	-0.1101	-0.1101	-0.1101
1	10	0.1940	0.1940	0.1940
2	10	0.1083	0.1083	0.1083
2	6	-0.0965	-0.0965	-0.0965
3	14	0.1717	0.1717	0.1717
3	7	-0.1631	-0.1631	-0.1631
4	7	-0.1596	-0.1596	-0.1596
4	14	0.1624	0.1624	0.1624
5	18	0.1665	0.1665	0.1665
5	9	-0.0984	-0.0984	-0.0984
6	9	-0.1023	-0.1023	-0.1023
6	18	0.1824	0.1824	0.1824
7	24	0.1686	0.1686	0.1686
7	12	-0.0845	-0.0845	-0.0845
8	12	-0.0925	-0.0925	-0.0925
8	24	0.2271	0.2271	0.2271



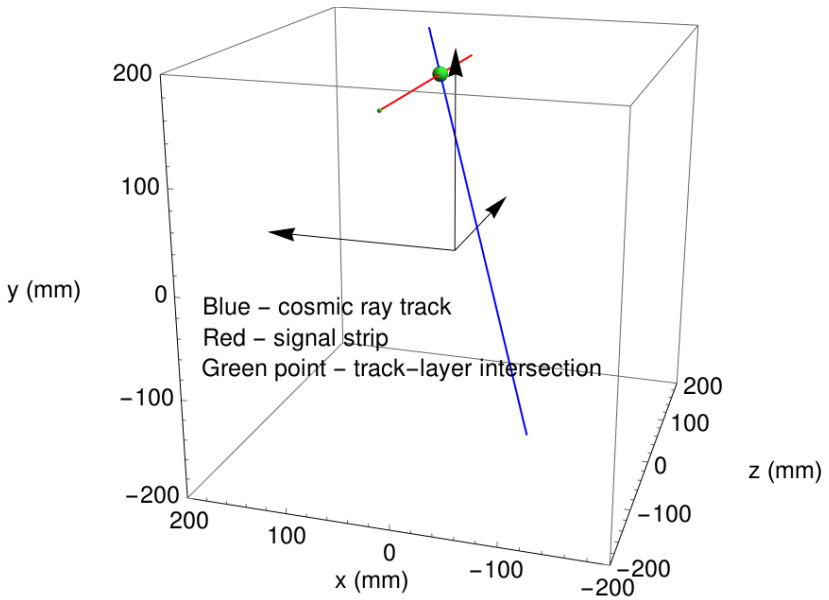
SVT Track-Based Alignment - Type 1 and 2 tests

1. Sequence of steps for track-based alignment with `millepede`.



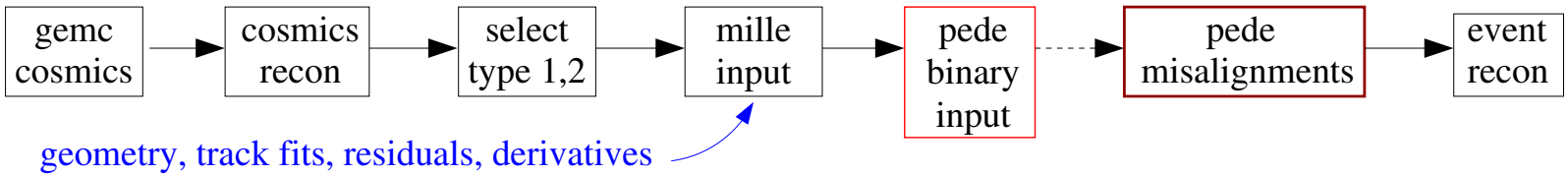
2. Extend code for all SVT topologies - generalized algorithms.
    1. Type 1 and Type 2 comparisons and visualizations validate code.
    2. beta version of Mille input code done.
    3. Writing pede binary input code now.

layer	sector	VZ (mm)	GG(mm)	JG (mm)
1	6	-0.1101	-0.1101	-0.1101
1	10	0.1940	0.1940	0.1940
2	10	0.1083	0.1083	0.1083
2	6	-0.0965	-0.0965	-0.0965
3	14	0.1717	0.1717	0.1717
3	7	-0.1631	-0.1631	-0.1631
4	7	-0.1596	-0.1596	-0.1596
4	14	0.1624	0.1624	0.1624
5	18	0.1665	0.1665	0.1665
5	9	-0.0984	-0.0984	-0.0984
6	9	-0.1023	-0.1023	-0.1023
6	18	0.1824	0.1824	0.1824
7	24	0.1686	0.1686	0.1686
7	12	-0.0845	-0.0845	-0.0845
8	12	-0.0925	-0.0925	-0.0925
8	24	0.2271	0.2271	0.2271



SVT Track-Based Alignment - Type 1 and 2 tests

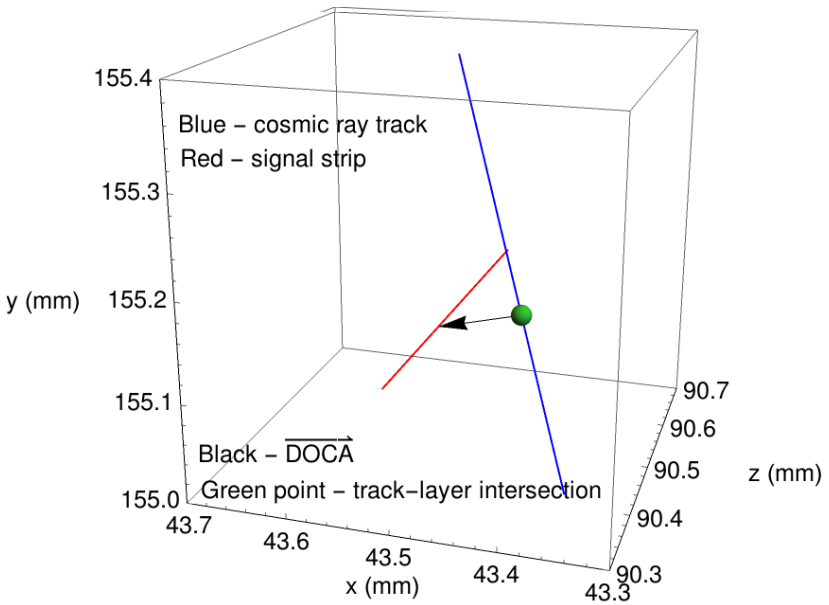
1. Sequence of steps for track-based alignment with `millepede`.



2. Extend code for all SVT topologies - generalized algorithms.

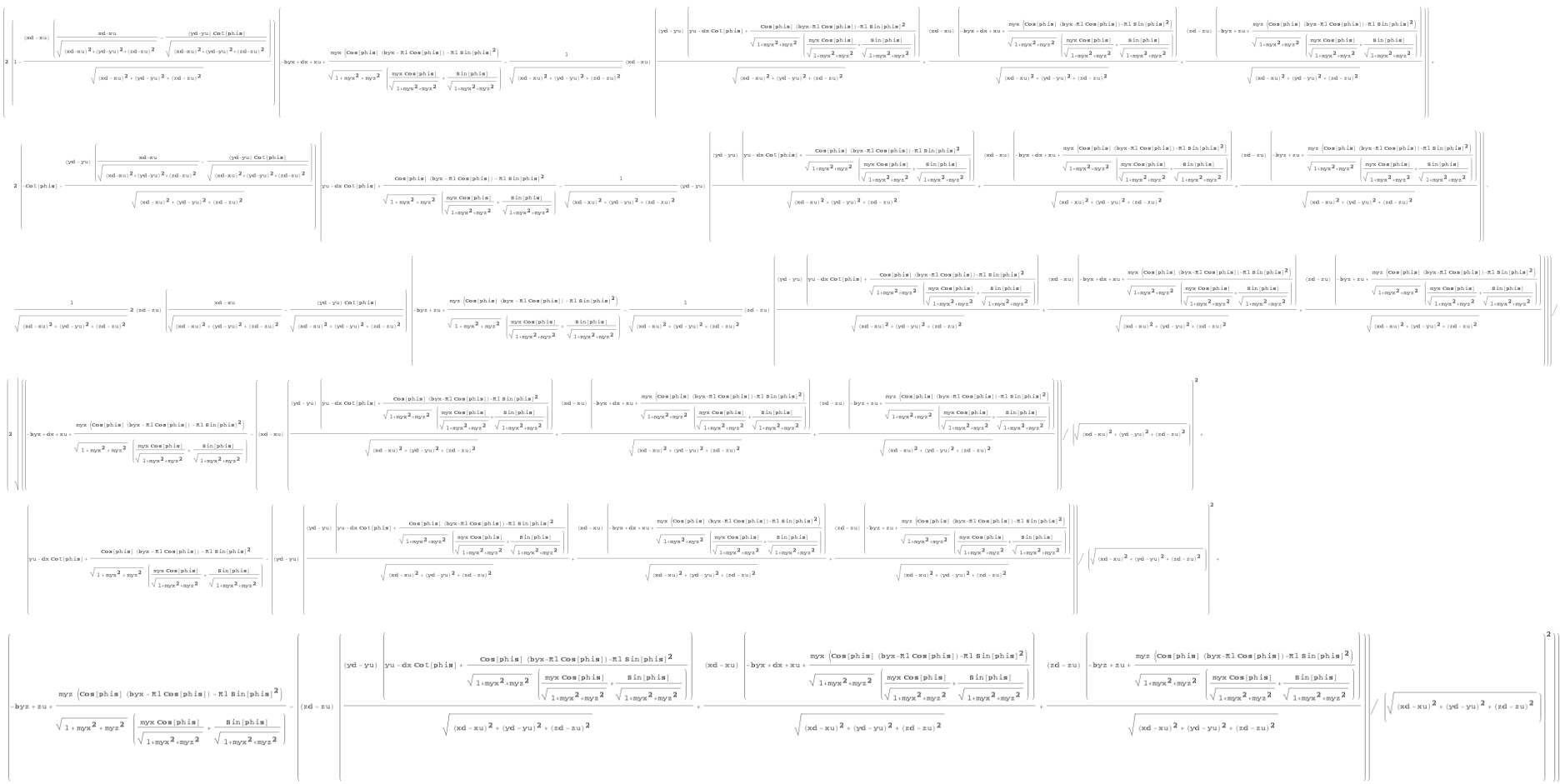
    1. Type 1 and Type 2 comparisons and visualizations validate code.
    2. beta version of Mille input code done.
    3. Writing pede binary input code now.

layer	sector	VZ (mm)	GG(mm)	JG (mm)
1	6	-0.1101	-0.1101	-0.1101
1	10	0.1940	0.1940	0.1940
2	10	0.1083	0.1083	0.1083
2	6	-0.0965	-0.0965	-0.0965
3	14	0.1717	0.1717	0.1717
3	7	-0.1631	-0.1631	-0.1631
4	7	-0.1596	-0.1596	-0.1596
4	14	0.1624	0.1624	0.1624
5	18	0.1665	0.1665	0.1665
5	9	-0.0984	-0.0984	-0.0984
6	9	-0.1023	-0.1023	-0.1023
6	18	0.1824	0.1824	0.1824
7	24	0.1686	0.1686	0.1686
7	12	-0.0845	-0.0845	-0.0845
8	12	-0.0925	-0.0925	-0.0925
8	24	0.2271	0.2271	0.2271



SVT Track-Based Geometry Alignment

Equation for derivative of DOCA with respect to a shift in x (Δx).



SVT Track-Based Geometry Alignment

Java code for derivative of DOCA with respect to a shift in x (Δx).

```
double dDOCAddx=(2*(1 - ((xd - xu)*((xd - xu)/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) - ((yd - yu)*(1/Math.tan(phis)))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Mat
h.pow(yd - yu,2) + Math.pow(zd - zu,2)))*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(
1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))) - ((xd - xu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),
2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))/Math.sqrt(Math.pow(xd
- xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((xd - xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*M
ath.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu
+ (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Mat
h.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)))) + 2*(-(1/Math.tan(phis)) - ((yd - yu)*((xd - xu)/Ma
th.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) - ((yd - yu)*(1/Math.tan(phis)))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)))*(yu - dx*
(1/Math.tan(phis)) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.
sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))) - ((yd - yu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz +
myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((xd
- xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.p
ow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl
*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))
)))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))) - (2*(zd - zu)*((xd - xu)/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) - ((yd - y
u)*(1/Math.tan(phis)))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)))*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2
))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))) - ((zd - zu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dbyx - Rl*M
ath.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.p
ow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((xd - xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2
) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow
(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.p
Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.s
qrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)))/(2.0*Math.sqrt(Math.pow(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz
 + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))) - ((xd - xu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dby
x - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2)
+ Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((xd - xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx
 + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) +
Math.pow(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + m
yx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)),
2) + Math.pow(yu - dx*(1/Math.tan(phis)) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(
dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)))) - ((yd - yu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx
,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.p
ow(zd - zu,2)) + ((xd - xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(
dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - R
l*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Ma
th.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)))) - ((zd - zu)*(((yd - yu)*(yu - dx*(1/Math.tan(phis))) + (Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2))/(Math.sqrt(1 + Math.pow(dmyx + myx
,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((xd - xu)*(-byx - dbyx + dx + xu + ((dmyx + myx)*(Math.cos(phis)*(byx + dbyx
 - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2)
+ Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)) + ((zd - zu)*(-byz - dbyz + zu + (myz*(Math.cos(phis)*(byx + dbyx - Rl*Math.cos(phis)) - Rl*Math.pow(Math.sin(phis),2)))/(Math.sqrt(1 + Math.pow(dmyx + myx,2) + Ma
th.pow(dmyz + myz,2))*(((dmyx + myx)*Math.cos(phis))/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2)) + Math.sin(phis)/Math.sqrt(1 + Math.pow(dmyx + myx,2) + Math.pow(dmyz + myz,2))))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd -
zu,2))))/Math.sqrt(Math.pow(xd - xu,2) + Math.pow(yd - yu,2) + Math.pow(zd - zu,2)),2)));
```