

# Status of constraints in alignment

---

Problem: *we can increase quality of alignment if we take into account various constant geometrical structure of the detector stations and the spectrometer, such as: the distance between two planes, the constant right angle between some U and V axis etc.*

There were some efforts to introduce constraints.

One of them is reparam option for Z coordinate (incomplete).

My idea is to try doing it using theory of alignment.

From the theory side, constraint is an inhomogeneous linear form which must equal zero:

$$f(\vec{a}) = \alpha_1 a^1 + \alpha_2 a^2 + \dots + \alpha_n a^n + \beta = 0 \quad (1)$$

$\vec{a}$  – vector of aligning parameters.

Example: the constant distance  $d$  between X and Y planes in some det station:

$$1 \cdot Z_{X\_plane} - 1 \cdot Z_{Y\_plane} - d = 0 \quad (2)$$

# Status of constraints in alignment

Alignment in COMPASS is based on millepede programm by Volker Blobel.  
millepede.F provides CONSTF function:

```
387 ..... SUBROUTINE CONSTF(DERCS,RHS)
388 *..... optional: constraints
389 *..... -----
390 *..... Basic dimension parameters
391 ..... PARAMETER (MGLOBAL=3000,MLOCAL=10,NSTORE=100000,MCS=10) ! dimensions
392 ..... PARAMETER (MGL=MGLOBAL+MCS) ..... ! derived parameter
393 *..... derived parameters
394 ..... PARAMETER (MSYMGB = (MGLOBAL*MGLOBAL+MGLOBAL)/2,
395 ..... + ..... MSYM = (MGL*MGL+MGL)/2,
396 ..... + ..... MSYMLC= (MLOCAL*MLOCAL+MLOCAL)/2,
397 ..... + ..... MRECTA= MGLOBAL*MLOCAL,
398 ..... + ..... MGLOCS= MGLOBAL*MCS,
399 ..... + ..... MSYMCS= (MCS*MCS+MCS)/2 )
```

- RHS - free parameter  $\beta$  in linear form (1), i.e. distance or the value of angle between two planes.
- DERCS - an array with length in NGLB elements (number of global parameters, it should be the dimension of the vector  $\vec{a}$  in (1)).

# Status of constraints in alignment

## Attempts

---

- I modified *Align.cc*, *Align.h*, *Main.align.cc* in the alignment source folder in order to test and debug.

Align procedure sees that «constraint» was added, console outputs:

```
Constraint 1      Sum - RHS =  0.0000      - 0.50000E-01 = -0.50000E-01
```

- There is no different between the alignment \*.out files with constraints and without ones.
- I was trying various numbers in the filling of the CONSTR array for:

$$Z_{GM09Y1} - Z_{GM09X1} = 0.05mm \quad (3)$$

But the result is the same.

# Status of constraints in alignment

## Conclusion

---

- It is clear how to modify alignment code in order to introduce new option for constraints.
- It is clear how to connect Align.cc and other with millepede.F sending the necessary constraint array.
- But it is not clear how to correct fill this array...

**Now the main problem for me is to understand the form (1) from the code side.**

There is official example in millepede how to use constraint.

The array filling is the following:

```
· FROT(1)=0.0  
· DO I=2,10  
· FROT(I)=1.0/(X(I)-X(1))  
· END DO  
· CALL CONSTF(FROT,0.0) ····· ! constraint: total rotation zero
```

## By the way

Error when Z coordinate is unfixed in align.opt

---

```
*** Break *** bus error
=====
There was a crash.
This is the entire stack trace of all threads:
=====
#0 0x00007efe8e9047be in waitpid () from /lib64/libc.so.6
#1 0x00007efe8e8965c9 in do_system () from /lib64/libc.so.6
#2 0x00007efe93e8993c in TUnixSystem::StackTrace() () from /afs/cern.ch
/sw/lcg/app/releases/R00T/5.34.07a/x86_64-slc6-gcc47-opt/root/lib/libCore.so
#3 0x00007efe93e8c173 in TUnixSystem::DispatchSignals(ESignals) () from
/afs/cern.ch/sw/lcg/app/releases/R00T/5.34.07a/x86_64-slc6-gcc47-opt/root/
lib/libCore.so
#4 <signal handler called>
#5 0x00000000004266e9 in Align::_Dump(std::basic_ostream<char, std::char_traits
<char> >&, char const*) ()
#6 0x00000000004278a0 in Align::DumpToFile(char const*, char const*) ()
#7 0x000000000041dae4 in main ()
=====
```