Concerted rotations

This notebook implement the equation needed to compute a concerted rotation of 7 consecutive dihedral angles on a protein backbone. The algorithm adopted is the one proposed by Zamuner et al, PLoSOne 10(3):e0118342.

The basic idea is to assign to each bond of the chain an orthonormal base, following the Denavit-Hartenberg convention. The constraints of moving a series of bonds while keeping the first and last of them fixed can then be
interpreted as a constraint on the base transformation matrix bringing the base attached to the first bond into the one attached to the last bond.

Such constraints correspond to 6 independent equation -3 for the orientation and 3 for the translation-for the last bond. Given a set of n>6 variables, the constraint equations identify a manifold of dimension n-6 in R^n. Given
a solution to the equations, a new one can then be found through a random displacement on the tangent space of the manifold followed by a minimization to project the new point on the tangent space back to the manifold.
Every such operation correspond to a concerted rotation of the selected portion of the chain. The tangent space can be identifying applying Dini's theorem once a free variable is chosen.

Here we consider only a variation of the torsion angles.

Notebook general parameters

In[380]:=
```
SetDirectory[NotebookDirectory[]]
```

Out[380]=
```
/home/tubiana/Science/Projects/ABM/concerted_rotations/Code/generic_move
```

In[381]:=
```
Directory[]
```

Out[381]=
```
/home/tubiana/Science/Projects/ABM/concerted_rotations/Code/generic_move
```

Equations implementing the move

Transformation matrix

In this case we consider a general transformation matrix.
In the following, α is the joint angle, θ the torsion angle, r=$\vec{S}\cdot\hat{z}$ is the z component of $\vec{S}$, and d=||$\vec{S}$-(d$\cdot\hat{z}$) $\hat{z}$|| , with $\vec{S}$ the vector joinint the base i with the base i+1 .

In[382]:=
```
T[θ_, α_, d_, r_] := {{Cos[θ], -Cos[α] Sin[θ], Sin[θ] Sin[α], d * Sin[θ]},
   {Sin[θ], Cos[θ] Cos[α], -Cos[θ] Sin[α], -d * Cos[θ]},
   {0., Sin[α], Cos[α], r},
   {0., 0., 0., 1.}}
```

In[383]:=
```
T[θ, α, d, r] // MatrixForm
```

Out[383]//MatrixForm=

$$\begin{pmatrix} \text{Cos}[\theta] & -\text{Cos}[\alpha]\,\text{Sin}[\theta] & \text{Sin}[\alpha]\,\text{Sin}[\theta] & \text{d}\,\text{Sin}[\theta] \\ \text{Sin}[\theta] & \text{Cos}[\alpha]\,\text{Cos}[\theta] & -\text{Cos}[\theta]\,\text{Sin}[\alpha] & -\text{d}\,\text{Cos}[\theta] \\ 0. & \text{Sin}[\alpha] & \text{Cos}[\alpha] & \text{r} \\ 0. & 0. & 0. & 1. \end{pmatrix}$$

Tranformation matrix for seven dihedral angles.PHI MOVE

The Final transformation matrix is obtained by composing the 3 elementary transformation matrices given above.
Going from one N to a C, 4 Cα apart, I move 7 dihedrals. The transformation Matrix is:

In[384]:=
```
T7[t1_, t2_, t3_, t4_, t5_, t6_, t7_] :=
 T[t1, α1, d1, r1].T[t2, α2, d2, r2].T[t3, α3, d3, r3].
  T[t4, α4, d4, r4].T[t5, α5, d5, r5].T[t6, α6, d6, r6].T[t7, α7, d7, r7]
```

In[385]:=
```
DeltaT7[t1_, t2_, t3_, t4_, t5_, t6_, t7_, ot1_, ot2_, ot3_, ot4_, ot5_, ot6_, ot7_] :=
 T7[t1, t2, t3, t4, t5, t6, t7] - T7[ot1, ot2, ot3, ot4, ot5, ot6, ot7]
```

The elements chosen for the constrain equations are the following: [[1, 3]], [[2, 3]], [[1,2]], [[1, 4]], [[2, 4]], [[3, 4]].
The first three should fix the orientation of the vector z in the last ref. system according to the first ref. system. The last Three its spatial position.

Derivative of T7 for different dihedral angles.

These are needed to apply Dini’s theorem.

In[386]:=
```
T7eqs[t1_, t2_, t3_, t4_, t5_, t6_, t7_] := {(T7[t1, t2, t3, t4, t5, t6, t7])[[1, 2]],
   (T7[t1, t2, t3, t4, t5, t6, t7])[[1, 3]],
   (T7[t1, t2, t3, t4, t5, t6, t7])[[2, 3]],
   (T7[t1, t2, t3, t4, t5, t6, t7])[[1, 4]],
   (T7[t1, t2, t3, t4, t5, t6, t7])[[2, 4]],
   (T7[t1, t2, t3, t4, t5, t6, t7])[[3, 4]]}
```

In[387]:=
```
DT7t1 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t1}];
DT7t2 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t2}];
DT7t3 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t3}];
DT7t4 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t4}];
DT7t5 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t5}];
DT7t6 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t6}];
DT7t7 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {t7}];
```

Jacobians

To apply Dini’s theorem, I have to choose one free variable, compute the jacobian of the equations with respect to the other 6 variables and invert it. Inversion will be done numerically. Nonetheless, it will be possible to print an expression for the determinant of the jacobian, allowing one to check if the inversion is possible.

In[394]:=
```
jacT7t1 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t2, t3, t4, t5, t6, t7}}];
jacT7t2 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t3, t4, t5, t6, t7}}];
jacT7t3 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t4, t5, t6, t7}}];
jacT7t4 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t3, t5, t6, t7}}];
jacT7t5 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t3, t4, t6, t7}}];
jacT7t6 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t3, t4, t5, t7}}];
jacT7t7 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t3, t4, t5, t6}}];
```

In[401]:=
```
JACT7 = D[T7eqs[t1, t2, t3, t4, t5, t6, t7], {{t1, t2, t3, t4, t5, t6, t7}}];
```

Printing equations in C format

All the equations and functions derived before must be plugged in a C code. In order to do so, we can print in C form. For the C code to be fast and reliable, we need to do two things:

1. substitute all sines and cosines with some variables c and s, which will be calculated beforehand and passed to the function.
2. Put the equations in the simplest possible polynomial form. This is achieved by using the HornerForm routine.

Point 1. above is achieved by using the following substitution rule.

In[402]:=
```
substitRule = {Cos[t1] → c1, Cos[t2] → c2, Cos[t3] → c3,
   Cos[t4] → c4, Cos[t5] → c5, Cos[t6] → c6, Cos[t7] → c7,
   Sin[t1] → s1, Sin[t2] → s2, Sin[t3] → s3, Sin[t4] → s4,
   Sin[t5] → s5, Sin[t6] → s6, Sin[t7] → s7,
   Cos[α1] → ca1, Cos[α2] → ca2, Cos[α3] → ca3, Cos[α4] → ca4,
   Cos[α5] → ca5, Cos[α6] → ca6, Cos[α7] → ca7,
   Sin[α1] → sa1, Sin[α2] → sa2, Sin[α3] → sa3, Sin[α4] → sa4,
   Sin[α5] → sa5, Sin[α6] → sa6, Sin[α7] → sa7
  };
```

All equations are printed one line per file. A python wrapper then puts together the C headers and source files.

Export T7

In[403]:=
```
Export["T7_12.c",
  CForm[HornerForm[T7[t1, t2, t3, t4, t5, t6, t7][[1, 2]] /. substitRule]], "Text"];
Export["T7_13.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[1, 3]] /. substitRule]], "Text"];
Export["T7_23.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[2, 3]] /. substitRule]], "Text"];
Export["T7_14.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[1, 4]] /. substitRule]], "Text"];
Export["T7_24.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[2, 4]] /. substitRule]], "Text"];
Export["T7_34.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[3, 4]] /. substitRule]], "Text"];
Export["T7_33.c", CForm[HornerForm[
    T7[t1, t2, t3, t4, t5, t6, t7][[3, 3]] /. substitRule]], "Text"];
```

Export DT7

In[410]:=
```
Do[Module[{bname = "DT7_t1"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t1[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[411]:=
```
Do[Module[{bname = "DT7_t2"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t2[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[412]:=
```
Do[Module[{bname = "DT7_t3"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t3[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[413]:=
```
Do[Module[{bname = "DT7_t4"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t4[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[414]:=
```
Do[Module[{bname = "DT7_t5"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t5[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[415]:=
```
Do[Module[{bname = "DT7_t6"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t6[[j]]] /. substitRule], "Text"]], {j, 6}]
```

In[416]:=
```
Do[Module[{bname = "DT7_t7"},
  fname = bname <> "_" <> ToString[j] <> ".c";
Export[fname, CForm[HornerForm[DT7t7[[j]]] /. substitRule], "Text"]], {j, 6}]
```

Export the Jacobians

In[417]:=
```
Do[
 Do[
  Module[{bname = "jac_t1"},
   el = CForm[HornerForm[jacT7t1[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[418]:=
```
Do[
 Do[
  Module[{bname = "jac_t2"},
   el = CForm[HornerForm[jacT7t2[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[419]:=
```
Do[
 Do[
  Module[{bname = "jac_t3"},
   el = CForm[HornerForm[jacT7t3[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[420]:=
```
Do[
 Do[
  Module[{bname = "jac_t4"},
   el = CForm[HornerForm[jacT7t4[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[421]:=
```
j
```

Out[421]=
```
j
```

In[422]:=
```
Do[
 Do[
  Module[{bname = "jac_t5"},
   el = CForm[HornerForm[jacT7t5[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[423]:=
```
Do[
 Do[
  Module[{bname = "jac_t6"},
   el = CForm[HornerForm[jacT7t6[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

In[424]:=
```
Do[
 Do[
  Module[{bname = "jac_t7"},
   el = CForm[HornerForm[jacT7t7[[i, j]]] /. substitRule];
   fname = bname <> "_" <> ToString[i] <> ToString[j] <> ".c";
   Export[fname, el, "Text"]
  ], {j, 6}],
 {i, 6}]
```

Call wrapper

In[425]:=
```
Run["python ./Precompiler.py"]
Run["rm DT7*.c jac_t*c T7_??.c"]
(*Run["./create_C_files_LUdecomp_inner_check.sh"]*)
```

Out[425]= 0

Out[426]= 0