```
> restart;
> with(VectorCalculus) : with(Student[VectorCalculus]) : with(PolynomialTools) :
      with(LinearAlgebra) :
```

Cas du cyclohexane :

Cette molécule présente des symétries particulières, étant donné que les atomes principaux ne sont que des atomes de Carbone. Ainsi les angles formés à chaque sommet entre deux liaison Carbone-Carbone sont égales, et les liaisons Carbone-Carbone ont toutes la même longueur, qu'on va prendre comme unité de mesure... i.e. on considère que les liaisons C-C sont de mesure 1 unité de longueur.

```
>
```

Détermination et résolution du système de solutions :

```
> Cyclo := proc( )
  local B, FF, α1, u, v, w, AH, HB, AB, AFF, AF, R, prod, POL, P, PP, Puv, Pvw, Pwu, Puu,
      Pw1, Pw2, S1, S2, S3, α, Φ, f, g, h, l;
  α := π/3 : Φ := 109·π/180 : α1 := (π − Φ)/2 : f := cos(Φ) :
  h := cos(α) : l := sin(α) :
  R := Matrix([[h, -l, 0], [l, h, 0], [0, 0, 1]]);
  AH := cos(α1) : HB := sin(α1) :
  B[1] := AH: B[2] := HB·(1 − u^2)/(1 + u^2) : B[3] := HB·2·u/(1 + u^2) :
  AB := Matrix([B[1], B[2], B[3]]);
  FF[1] := AH: FF[2] := HB·(1 − v^2)/(1 + v^2) : FF[3] := HB·2·v/(1 + v^2) :
  AFF := Transpose(Matrix([FF[1], FF[2], FF[3]]));
  AF := MatrixMatrixMultiply(R, AFF);
  AB := convert(AB, Vector) :
  AF := convert(AF, Vector) :
  prod := AB.AF;
  P := simplify((prod − f)·(1 + u^2)·(1 + v^2));
  PP := subs(v = 1/v, P);
  Puv := simplify(PP·v^2);
  Pvw := subs(v = w, u = v, Puv);
  Pwu := subs(u = w, v = u, Puv);
  POL := [Puv, Pvw, Pwu];
  Puu := subs(v = u, Puv);
  S1 := fsolve(Puu);
  Pw1 := subs(u = S1[1], Puv);
  S2 := fsolve(Pw1);
  Pw2 := subs(u = S1[2], Puv);
  S3 := fsolve(Pw2);
```

```
    [[S1], [S2], [S3]];
    end:

> SYM := Cyclo( );
SYM := [[-3.081236390, 3.081236390], [-3.081236390, 1.283677708],
    [-1.283677708, 3.081236390]]                                   (1.1.1)
```

On obtient ainsi un sytème d'équations par symétries :
POL[1]=Puv=0, POL[2]=Puw=0 et POL[3]=Pwu=0, où P est la liste des polynômes renvoyée par la procédure Cyclo. Il reste à trouver les solutions de ce sytème. S1 donne les solutions de Puu=0, c'est à dire les solutions pour u=v=w. Ensuite, on cherche les solutions pour u=v et w différent de u en substituant dans Puv u par une des solutions donnée par S1, et en résolvant Pu1w=0, où u1 est une solution de S1.
On obtient 4 solutions :
2 pour u=v=w : u=v=w=SYM[1][1], ou u=v=w=SYM[1][2];
et 4 autres pour u=v et w différent de u :
u=SYM[2][1] et w=SYM[2][2], ou u=SYM[2][2] et w=SYM[2][1]; ou bien u=SYM[3][1] et w= SYM[3][2], ou u=SYM[3][2] et w=SYM[3][1].
On a obtenu les 6 solutions symétriques du sytème.
On voit que les 2 solutions pour u=v=w donnent géométriquement la même forme de molécule, en forme de bateau, et que les 4 solutions pour u=v et w différent de u donnent géométriquement la même forme de molécule, en forme de chaise.
Il y a des solutions moins symétriques, qu'on peut obtenir grâce à l'étude des résultants.
En chimie, on sait que la molécule admet des symétries telles que les solutions non symétriques, possible mathématiquement, sont impossible chimiquement, du point de vue de la stéréométrie.

```
>
```

Solutions non symétriques :

Pour résoudre le système avec u, v, et w distincts deux à deux, on utilise la notion de résultant. En effet, le résultant de deux polynômes est nul en a si et seulement si les deux polynômes s'annulent en a. On réécrit une procédure, qui commence comme celle nommée Cyclo, mais cette procédure va renvoyer les solutions non symétriques.

```
> Cyclo2 := proc( )
  local B, FF, α1, u, v, w, AH, HB, AB, AFF, AF, R, prod, POL, P, PP, Puv, Pvw, Pwu, Puu,
      Pw1, Pw2, R1, R2, S4, S5, S6, S7, S8, T1, T2, T3, T4, α, Φ, f, g, h, l,
  α := π/3 : Φ := 109·π/180 : α1 := (π − Φ)/2 : f := cos(Φ) :
  h := cos(α) : l := sin(α) :
  R := Matrix([[h, -l, 0], [l, h, 0], [0, 0, 1]]);
  AH := cos(α1) : HB := sin(α1) :
  B[1] := AH: B[2] := HB·(1 − u^2)/(1 + u^2) : B[3] := HB·2·u/(1 + u^2) :
  AB := Matrix([B[1], B[2], B[3]]) :
  FF[1] := AH: FF[2] := HB·(1 − v^2)/(1 + v^2) : FF[3] := HB·2·v/(1 + v^2) :
```

```
    AFF := Transpose( Matrix( [ FF[1], FF[2], FF[3] ] ) ) :
    AF := MatrixMatrixMultiply( R, AFF ) :
    AB := convert( AB, Vector ) :
    AF := convert( AF, Vector ) :
    prod := AB.AF:
    P := simplify( ( prod - f ) · ( 1 + u^2 ) · ( 1 + v^2 ) ) :
    PP := subs( v = 1/v, P ) :
    Puv := simplify( PP · v^2 ) :
    Pvw := subs( v = w, u = v, Puv ) :
    Pwu := subs( u = w, v = u, Puv ) :
    R1 := evala( Resultant( Pvw, Pwu, w ) ) :
    R2 := evala( Resultant( R1, Puv, v ) );
    S4 := [ fsolve( R2 ) ] :
    T1 := subs( u = S4[1], Puv ) :
    S5 := [ fsolve( evalf ( T1 ) ) ] :
    T2 := subs( u = S5[1], Puv ) :
    S6 := [ fsolve( evalf ( T2 ) ) ];
    T3 := subs( u = S4[4], Puv );
    S7 := [ fsolve( evalf ( T3 ) ) ];
    T4 := subs( v = S7[1], Pvw );
    S8 := [ fsolve( evalf ( T4 ) ) ];
    [ S4, S5, S6, S7, S8 ];
    end:
> Cyclo2( );
      #Normalement, on doit obtenir des solutions réelles possibles du système, mais elles
      ne conviennent pas à la stéréochimie.
[ [ -2.619018437, -.3219077457, 0.3219077457, 2.619018437 ], [ -3.742921916,
    1.291963189 ], [ -2.619018434, 1.291963188 ], [ -1.291963189, 3.742921916 ],
    [ 2.619018432, 3.742921933 ] ]                                              (1.2.1)
> Cyclo2( );
[ [ -1.789686414, -.2894295962, 0.2894295962, 1.789686414 ], [ -11.79874051,
    1.437447980 ], [ -1.789686413, 1.437447979 ], [ -1.437447980, 11.79874051 ],
    [ 1.789686409, 11.79874072 ] ]                                              (1.2.2)
```

On trouve alors les solutions réelles non symétriques du système de départ.
Ces solutions ne sont pas intéressantes en Chimie.

Cas du glucose :