3a. Real Lie algebra

Implementation of definitions in §2.2.2, §2.3, §3.1.

A basis for the real Lie algebra is given in the list **XWlist**. The names in this list are at the basis of the computations; the order in the list determines the standard basis of the universal enveloping algebra. The routine **Liesub** transforms the Lie algebra elements into matrices. The subsitution **Liesub0** is for formal manipulations.

In[ ]:=
```
Clear[HHr, HHi, XX0, XX1, XX2, WW0, WW1, WW2, CKi, Liesub]
Liesub[xx_] := Block[{x}, x = xx //. Liesub0; x //. Liesub1 // Expand]
XWlist = {XX0, XX1, XX2, HHr, HHi, WW0, WW1, WW2};

Liesub0a = {lb[xx_, yy_] → Simplify[xx.yy - yy.xx], CKi → 3 WW0 - 2 HHi};
Liesub1 = {nul → {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}},
    HHr → {{0, 0, 1}, {0, 0, 0}, {1, 0, 0}},
    HHi → {{I, 0, 0}, {0, -2 I, 0}, {0, 0, I}}, XX0 → {{ⅈ/2, 0, -ⅈ/2}, {0, 0, 0}, {ⅈ/2, 0, -ⅈ/2}},
    XX1 → {{0, 1, 0}, {-1, 0, 1}, {0, 1, 0}}, XX2 → {{0, ⅈ, 0}, {ⅈ, 0, -ⅈ}, {0, ⅈ, 0}},
    WW0 → {{I, 0, 0}, {0, -I, 0}, {0, 0, 0}},
    WW1 → {{0, 1, 0}, {-1, 0, 0}, {0, 0, 0}},
    WW2 → {{0, I, 0}, {I, 0, 0}, {0, 0, 0}}};
Liesub0b = {};(* will be redefined later on *)
Liesub0 := Union[Liesub0a, Liesub0b];
```

Lie bracket

The routine **inLie** checks whether an element is in the real Lie algebra. The Lie bracket is given by **lb**.

In[ ]:=
```
Clear[inLie, lb, setlb, expc, t, nul]
inLie[X_] := Block[{XX}, XX = Liesub[X];
  FullSimplify[Conjugate[Transpose[XX]].I21 + I21.XX == nul] // Liesub]
lb[xx_, xx_] := nul
lb[-xx_, yy_] := -lb[xx, yy]
lb[xx_ + yy_, zz_] := lb[xx, zz] + lb[yy, zz]
lb[ff_ xx_, yy_] := ff lb[xx, yy] /; NumberQ[ff]
lb[xx_, -yy_] := -lb[xx, yy]
lb[xx_, yy_ + zz_] := lb[xx, yy] + lb[xx, zz]
lb[ xx_, ff_ yy_] := ff lb[xx, yy] /; NumberQ[ff]
lb[xx_, nul] := nul
lb[nul, yy_] := nul
setlb[X_, Y_, Z_] := Block[{tr}, tr = FullSimplify[X.Y - Y.X == Z] // Liesub;
  If[tr, lb[X, Y] = Z;
   lb[Y, X] = -Z;]; tr]
expc[X_, gt_] := FullSimplify[Simplify[MatrixExp[t X // Liesub] == gt] //.
    {Re[t] → t, Im[t] → 0, Abs[t] → t, Conjugate[t] → t}] /. Conjugate[t] → t
```

Illustration

In[ ]:=
```
inLie[HHr]
inLie[I HHr]
```

Out[ ]= True

Out[ ]= False

Some checks

In[ ]:=
```
inLie[HHr] // Liesub
inLie[HHi] // Liesub
inLie[XX0] // Liesub
inLie[XX1] // Liesub
inLie[XX2] // Liesub
inLie[WW0] // Liesub
inLie[WW1] // Liesub
inLie[WW2] // Liesub
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

The routine **setlb** defines the values for the formal function **lb**, and checks it by comparison with a matrix computation.

In[ ]:=
```
setlb[HHr, HHi, nul]
```

Out[ ]= True

In[ ]:=
```
setlb[HHr, XX0, 2 XX0]
setlb[HHr, XX1, XX1]
setlb[HHr, XX2, XX2]
setlb[HHi, XX0, nul]
setlb[HHi, XX1, 3 XX2]
setlb[HHi, XX2, -3 XX1]
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

In[ ]:=
```
setlb[XX1, XX2, 4 XX0]
setlb[XX1, XX0, nul]
setlb[XX2, XX0, nul]
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

In[ ]:=
```
setlb[WW0, WW1, 2 WW2]
setlb[WW0, WW2, -2 WW1]
setlb[WW1, WW2, 2 WW0]
setlb[HHi, WW0, nul]
setlb[HHi, WW1, 3 WW2]
setlb[HHi, WW2, -3 WW1]
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

In[ ]:=
```
setlb[XX0, WW0, -(1/2) HHr]
setlb[XX0, WW1, (1/2) XX2]
setlb[XX0, WW2, -(1/2) XX1]
setlb[XX1, WW0, -XX2 - WW2]
setlb[XX1, WW1, -HHr]
setlb[XX1, WW2, 2 XX0 + HHi]
setlb[XX2, WW0, XX1 + WW1]
setlb[XX2, WW1, -HHi - 2 XX0]
setlb[XX2, WW2, -HHr]
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

Out[ ]= True

In[ ]:=
```
setlb[HHr, WW0, -2 WW0 + HHi + 2 XX0]
setlb[HHr, WW1, XX1 - WW1]
setlb[HHr, WW2, XX2 - WW2]
```

Out[ ]= True

Out[ ]= True

Out[ ]= True

In[ ]:=
```
lb[CKi, XX_] := 3 lb[WW0, XX] - 2 lb[HHi, XX]
lb[XX_, CKi] := 3 lb[XX, WW0] - 2 lb[XX, HHi]
```

In[ ]:=
```
CKi /. Liesub0
3 WW0 - 2 HHi /. Liesub0
```

Out[ ]= -2 HHi + 3 WW0

Out[ ]= -2 HHi + 3 WW0

In[ ]:=