Appendix F

Direct Sum of Vector Spaces

Let U and V be vector spaces with $\dim(U) = n$ and $\dim(V) = m$. Additionally, let the sets of vectors $\{\boldsymbol{u}_i\}$ and $\{\boldsymbol{v}_j\}$ be basis sets for U and V respectively. The direct sum, $U \oplus V$, is defined to be

$$U \oplus V \equiv \{(\boldsymbol{u}, \boldsymbol{v}) \,|\, \boldsymbol{u} \in U, \boldsymbol{v} \in V\}, \tag{F.1}$$

where vector addition and scalar multiplication are defined by for all $(\boldsymbol{a}_1, \boldsymbol{b}_1), (\boldsymbol{a}_2, \boldsymbol{b}_2) \in U \oplus V$ and $\alpha \in \Re$

$$(\boldsymbol{a}_1, \boldsymbol{b}_1) + (\boldsymbol{a}_2, \boldsymbol{b}_2) \equiv (\boldsymbol{a}_1 + \boldsymbol{a}_2, \boldsymbol{b}_1 + \boldsymbol{b}_2) \tag{F.2}$$

$$\alpha (\boldsymbol{a}_1, \boldsymbol{b}_1) \equiv (\alpha \boldsymbol{a}_1, \alpha \boldsymbol{b}_1). \tag{F.3}$$

Now define the maps $i_U : U \rightarrow U \oplus V$ and $i_V : V \rightarrow U \oplus V$ by

$$i_U(\boldsymbol{u}) \equiv (\boldsymbol{u}, \boldsymbol{0}) \tag{F.4}$$

$$i_V(\boldsymbol{v}) \equiv (\boldsymbol{0}, \boldsymbol{v}). \tag{F.5}$$

Thus $(\boldsymbol{u}, \boldsymbol{v}) = i_U(\boldsymbol{u}) + i_V(\boldsymbol{v})$ and the set $\{i_U(\boldsymbol{u}_i), i_V(\boldsymbol{v}_j) \,|\, 0 < i \leq n, 0 < j \leq m\}$ form a basis for $U \oplus V$ so that if $\boldsymbol{x} \in U \oplus V$ then $(\alpha^i, \beta^j \in \Re)$ and using the Einstein summation convention

$$\boldsymbol{x} = \alpha^i i_U(\boldsymbol{u}_i) + \beta^j i_V(\boldsymbol{v}_j). \tag{F.6}$$

If from context we known that $\boldsymbol{x} \in U \oplus V$ and we are expanding $\boldsymbol{x}$ in terms of a basis of $U \oplus V$ we will write as a notational convenience $\boldsymbol{u}_i$ for $i_U(\boldsymbol{u}_i)$ and $\boldsymbol{v}_i$ for $i_V(\boldsymbol{v}_i)$ so that we may write

$$\boldsymbol{x} = \alpha^i \boldsymbol{u}_i + \beta^j \boldsymbol{v}_j. \tag{F.7}$$

Also for notational convenience denote for any $\boldsymbol{x} \in U \oplus V$, $x_U = i_U(\boldsymbol{x})$ and $x_V = i_V(\boldsymbol{x})$.

Likewise we define the mappings $p_U : U \oplus V \to U$ and $p_V : U \oplus V \to V$ by

$$p_U((\boldsymbol{u}, \boldsymbol{v})) \equiv \boldsymbol{u} \tag{F.8}$$

$$p_V((\boldsymbol{u}, \boldsymbol{v})) \equiv \boldsymbol{v}. \tag{F.9}$$

For notational convenience if $p_V((\boldsymbol{u}, \boldsymbol{v})) = \boldsymbol{0}$ we write $\boldsymbol{u} = p_U((\boldsymbol{u}, \boldsymbol{v}))$ and if $p_U((\boldsymbol{u}, \boldsymbol{v})) = \boldsymbol{0}$ we write $\boldsymbol{v} = p_V((\boldsymbol{u}, \boldsymbol{v}))$. We always make the identifications

$$\begin{aligned} \boldsymbol{u} &\leftrightarrow (\boldsymbol{u}, \boldsymbol{0}) \\ \boldsymbol{v} &\leftrightarrow (\boldsymbol{0}, \boldsymbol{v}). \end{aligned}$$

Which one to use will be clear from context or will be explicitly identified.

One final notational convenience is that for $(\boldsymbol{u}, \boldsymbol{v}) \in U \oplus V$ we make the equivalence

$$(\boldsymbol{u}, \boldsymbol{v}) \leftrightarrow \boldsymbol{u} + \boldsymbol{v}.$$

Appendix G

sympy/galgebra evaluation of $GL(n, \Re)$ structure constants

The structure constants of $GL(n, \Re)$ can be calculated using the *sympy* python computer algebra system wiht the latest *galgebra* modules (`https://github.com/brombo/sympy`). The python program used for the calculation is shown below:

```
#Lie Algebras
from sympy import symbols
from sympy.galgebra.ga import Ga
from sympy.galgebra.mv import Com
from sympy.galgebra.printer import Format, xpdf

#General Linear Group E generators
def E(i,j):
    global e,eb
    B = e[i]*e[j] - eb[i]*eb[j]
    Bstr = 'E_{' + i + j + '} '
    print '%' + Bstr + ' =',B
    return Bstr, B

#General Linear Group F generators
def F(i,j):
    global e,eb
    B = e[i]*eb[j] - eb[i]*e[j]
```

```
    Bstr = 'F_{' + i + j + '} '
    print '%' + Bstr + ' =',B
    return Bstr, B

#General Linear Group K generators
def K(i):
    global e,eb
    B = e[i]*eb[i]
    Bstr = 'K_{' + i + '} '
    print '%' + Bstr + ' =',B
    return Bstr, B

#Print Commutator
def ComP(A,B):
    AxB = Com(A[1],B[1])
    AxBstr = '%' + A[0] + r' \times ' + B[0] + ' ='
    print AxBstr, AxB
    return

Format()

(glg,ei,ej,em,en,ebi,ebj,ebm,ebn) = Ga.build(r'e_i e_j e_k e_l \bar{e}_i \bar

e = {'i':ei,'j':ej,'k':em,'l':en}
eb = {'i':ebi,'j':ebj,'k':ebm,'l':ebn}

print r'#\centerline{General Linear Group of Order $n$\newline}'
print r'#Lie Algebra Generators: $1\le i < j \le n$ and $1 \le i < l \le n$'

#Calculate Lie Algebra Generators
Eij = E('i','j')
Fij = F('i','j')
Ki = K('i')
Eil = E('i','l')
Fil = F('i','l')

print r'#Non Zero Commutators'
```

```
#Calculate and Print Non-Zero Generator Commutators
ComP(Eij,Fij)
ComP(Eij,Ki)
ComP(Fij,Ki)
ComP(Eij,Eil)
ComP(Fij,Fil)
ComP(Fij,Eil)

xpdf(paper='letter',pt='12pt',debug=True,prog=True)
```

Only those commutators that share one or two indices are calculated (all others are zero). The LaTeX output of the program follows:

General Linear Group of Order n

Lie Algebra Generators: $1 \le i < j \le n$ and $1 \le i < l \le n$

$$E_{ij} = e_i \wedge e_j - \bar{e}_i \wedge \bar{e}_j$$

$$F_{ij} = e_i \wedge \bar{e}_j + e_j \wedge \bar{e}_i$$

$$K_i = e_i \wedge \bar{e}_i$$

$$E_{il} = e_i \wedge e_l - \bar{e}_i \wedge \bar{e}_l$$

$$F_{il} = e_i \wedge \bar{e}_l + e_l \wedge \bar{e}_i$$

Non Zero Commutators

$$E_{ij} \times F_{ij} = 2e_i \wedge \bar{e}_i - 2e_j \wedge \bar{e}_j$$

$$E_{ij} \times K_i = -e_i \wedge \bar{e}_j - e_j \wedge \bar{e}_i$$

$$F_{ij} \times K_i = -e_i \wedge e_j + \bar{e}_i \wedge \bar{e}_j$$

$$E_{ij} \times E_{il} = -e_j \wedge e_l + \bar{e}_j \wedge \bar{e}_l$$

$$F_{ij} \times F_{il} = e_j \wedge e_l - \bar{e}_j \wedge \bar{e}_l$$

$$F_{ij} \times E_{il} = e_j \wedge \bar{e}_l + e_l \wedge \bar{e}_j$$

The program does not completely determine the structure constants since the *galgebra* module cannot currently solve a bivector equation. One must inspect the calculated commutator to see what the linear expansion of the commuatator is in terms of the Lie algebra generators. For this case the answer is:

$$E_{ij} \times F_{ij} = 2\left(K_i - K_j\right) \tag{G.1}$$
$$E_{ij} \times K_i = -F_{ij} \tag{G.2}$$
$$F_{ij} \times K_i = -E_{ij} \tag{G.3}$$
$$E_{ij} \times E_{il} = -E_{jl} \tag{G.4}$$
$$F_{ij} \times F_{il} = E_{jl} \tag{G.5}$$
$$F_{ij} \times E_{il} = F_{jl}. \tag{G.6}$$