```
def basic_multivector_operations_3D():
    Print_Function()

    g3d = Ga('e*x|y|z')
    (ex,ey,ez) = g3d.mv()

    A = g3d.mv('A','mv')

    print('A =',A)
    print('A =',A.Fmt(2))
    print('A =',A.Fmt(3))

    print('A_{+} =',A.even())
    print('A_{ } =',A.odd())

    X = g3d.mv('X','vector')
    Y = g3d.mv('Y','vector')

    print('g_{ij} = ',g3d.g)

    print('X =',X)
    print('Y =',Y)

    print('XY =',(X*Y).Fmt(2))
    print(r'X\W Y =',(X^Y).Fmt(2))
    print(r'X\cdot Y =',(X|Y).Fmt(2))
    print(r'X\times Y =',cross(X,Y).Fmt(3))
    return
```

Code Output:

$$A = A + A^{x}\boldsymbol{e}_{x} + A^{y}\boldsymbol{e}_{y} + A^{z}\boldsymbol{e}_{z} + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + A^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + A^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} + A^{xyz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z}$$

$$A = \begin{aligned}[t] & A \\ & + A^{x}\boldsymbol{e}_{x} + A^{y}\boldsymbol{e}_{y} + A^{z}\boldsymbol{e}_{z} \\ & + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + A^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + A^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \\ & + A^{xyz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$A = \begin{aligned}[t] & A \\ & + A^{x}\boldsymbol{e}_{x} \\ & + A^{y}\boldsymbol{e}_{y} \\ & + A^{z}\boldsymbol{e}_{z} \\ & + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} \\ & + A^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} \\ & + A^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \\ & + A^{xyz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$A_{+} = \begin{aligned}[t] & A \\ & + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} \\ & + A^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} \\ & + A^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$A_{-} = \begin{aligned}[t] & A^{x}\boldsymbol{e}_{x} \\ & + A^{y}\boldsymbol{e}_{y} \\ & + A^{z}\boldsymbol{e}_{z} \\ & + A^{xyz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$g_{ij} = \left[\begin{array}{ccc} (e_x\cdot e_x) & (e_x\cdot e_y) & (e_x\cdot e_z) \\ (e_x\cdot e_y) & (e_y\cdot e_y) & (e_y\cdot e_z) \\ (e_x\cdot e_z) & (e_y\cdot e_z) & (e_z\cdot e_z) \end{array}\right]$$

$$\begin{aligned} X = & \; X^{x}\boldsymbol{e}_{x} \\ & + X^{y}\boldsymbol{e}_{y} \\ & + X^{z}\boldsymbol{e}_{z} \end{aligned}$$

$$\begin{aligned} Y = & \; Y^{x}\boldsymbol{e}_{x} \\ & + Y^{y}\boldsymbol{e}_{y} \\ & + Y^{z}\boldsymbol{e}_{z} \end{aligned}$$

$$\begin{aligned} XY = & \left((e_x\cdot e_x) X^{x}Y^{x} + (e_x\cdot e_y) X^{x}Y^{y} + (e_x\cdot e_y) X^{y}Y^{x} + (e_x\cdot e_z) X^{x}Y^{z} + (e_x\cdot e_z) X^{z}Y^{x} + (e_y\cdot e_y) X^{y}Y^{y} + (e_y\cdot e_z) X^{y}Y^{z} + (e_y\cdot e_z) X^{z}Y^{y} + (e_z\cdot e_z) X^{z}Y^{z} \right) \\ & + \left(X^{x}Y^{y} - X^{y}Y^{x} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + \left(X^{x}Y^{z} - X^{z}Y^{x} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + \left(X^{y}Y^{z} - X^{z}Y^{y} \right) \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$X\wedge Y = \left(X^{x}Y^{y} - X^{y}Y^{x} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + \left(X^{x}Y^{z} - X^{z}Y^{x} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + \left(X^{y}Y^{z} - X^{z}Y^{y} \right) \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z}$$

$$X\cdot Y = (e_x\cdot e_x) X^{x}Y^{x} + (e_x\cdot e_y) X^{x}Y^{y} + (e_x\cdot e_y) X^{y}Y^{x} + (e_x\cdot e_z) X^{x}Y^{z} + (e_x\cdot e_z) X^{z}Y^{x} + (e_y\cdot e_y) X^{y}Y^{y} + (e_y\cdot e_z) X^{y}Y^{z} + (e_y\cdot e_z) X^{z}Y^{y} + (e_z\cdot e_z) X^{z}Y^{z}$$

$$\begin{aligned} X\times Y = & \frac{(e_x\cdot e_y)(e_y\cdot e_z) X^{x}Y^{y} - (e_x\cdot e_y)(e_y\cdot e_z) X^{y}Y^{x} + (e_x\cdot e_y)(e_z\cdot e_z) X^{x}Y^{z} - (e_x\cdot e_y)(e_z\cdot e_z) X^{z}Y^{x} - (e_x\cdot e_z)(e_y\cdot e_y) X^{x}Y^{y} + (e_x\cdot e_z)(e_y\cdot e_y) X^{y}Y^{x} - (e_x\cdot e_z)(e_y\cdot e_z) X^{x}Y^{z} + (e_x\cdot e_z)(e_y\cdot e_z) X^{z}Y^{x} + (e_y\cdot e_y)(e_z\cdot e_z) X^{y}Y^{z} - (e_y\cdot e_y)(e_z\cdot e_z) X^{z}Y^{y} - (e_y\cdot e_z)^{2} X^{y}Y^{z} + (e_y\cdot e_z)^{2} X^{z}Y^{y}}{\sqrt{(e_x\cdot e_x)(e_y\cdot e_y)(e_z\cdot e_z) - (e_x\cdot e_x)(e_y\cdot e_z)^{2} - (e_x\cdot e_y)^{2}(e_z\cdot e_z) + 2(e_x\cdot e_y)(e_x\cdot e_z)(e_y\cdot e_z) - (e_x\cdot e_z)^{2}(e_y\cdot e_y)}}\boldsymbol{e}_{x} \\ & + \frac{-(e_x\cdot e_x)(e_y\cdot e_z) X^{x}Y^{y} + (e_x\cdot e_x)(e_y\cdot e_z) X^{y}Y^{x} - (e_x\cdot e_x)(e_z\cdot e_z) X^{x}Y^{z} + (e_x\cdot e_x)(e_z\cdot e_z) X^{z}Y^{x} + (e_x\cdot e_y)(e_x\cdot e_z) X^{x}Y^{y} - (e_x\cdot e_y)(e_x\cdot e_z) X^{y}Y^{x} - (e_x\cdot e_y)(e_z\cdot e_z) X^{y}Y^{z} + (e_x\cdot e_y)(e_z\cdot e_z) X^{z}Y^{y} + (e_x\cdot e_z)^{2} X^{x}Y^{z} - (e_x\cdot e_z)^{2} X^{z}Y^{x} + (e_x\cdot e_z)(e_y\cdot e_z) X^{y}Y^{z} - (e_x\cdot e_z)(e_y\cdot e_z) X^{z}Y^{y}}{\sqrt{(e_x\cdot e_x)(e_y\cdot e_y)(e_z\cdot e_z) - (e_x\cdot e_x)(e_y\cdot e_z)^{2} - (e_x\cdot e_y)^{2}(e_z\cdot e_z) + 2(e_x\cdot e_y)(e_x\cdot e_z)(e_y\cdot e_z) - (e_x\cdot e_z)^{2}(e_y\cdot e_y)}}\boldsymbol{e}_{y} \\ & + \frac{(e_x\cdot e_x)(e_y\cdot e_y) X^{x}Y^{y} - (e_x\cdot e_x)(e_y\cdot e_y) X^{y}Y^{x} + (e_x\cdot e_x)(e_y\cdot e_z) X^{x}Y^{z} - (e_x\cdot e_x)(e_y\cdot e_z) X^{z}Y^{x} - (e_x\cdot e_y)^{2} X^{x}Y^{y} + (e_x\cdot e_y)^{2} X^{y}Y^{x} - (e_x\cdot e_y)(e_x\cdot e_z) X^{x}Y^{z} + (e_x\cdot e_y)(e_x\cdot e_z) X^{z}Y^{x} + (e_x\cdot e_y)(e_y\cdot e_z) X^{y}Y^{z} - (e_x\cdot e_y)(e_y\cdot e_z) X^{z}Y^{y} - (e_x\cdot e_z)(e_y\cdot e_y) X^{y}Y^{z} + (e_x\cdot e_z)(e_y\cdot e_y) X^{z}Y^{y}}{\sqrt{(e_x\cdot e_x)(e_y\cdot e_y)(e_z\cdot e_z) - (e_x\cdot e_x)(e_y\cdot e_z)^{2} - (e_x\cdot e_y)^{2}(e_z\cdot e_z) + 2(e_x\cdot e_y)(e_x\cdot e_z)(e_y\cdot e_z) - (e_x\cdot e_z)^{2}(e_y\cdot e_y)}}\boldsymbol{e}_{z} \end{aligned}$$

```python
def basic_multivector_operations_2D():
    Print_Function()
    g2d = Ga('e*x|y')
    (ex,ey) = g2d.mv()

    print('g_{ij} =',g2d.g)

    X = g2d.mv('X','vector')
    A = g2d.mv('A','spinor')

    print('X =',X)
    print('A =',A)

    print(r'X\cdot A =',(X|A).Fmt(2))
    print(r'X\lfloor A =',(X<A).Fmt(2))
    print(r'X\rfloor A =',(X>A).Fmt(2))
    return
```

Code Output:

$$g_{ij} = \left[\begin{array}{cc} (e_x\cdot e_x) & (e_x\cdot e_y) \\ (e_x\cdot e_y) & (e_y\cdot e_y) \end{array}\right]$$

$$\begin{aligned} X = & \; X^{x}\boldsymbol{e}_{x} \\ & + X^{y}\boldsymbol{e}_{y} \end{aligned}$$

$$\begin{aligned} A = & \; A \\ & + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} \end{aligned}$$

$$X\cdot A = -A^{xy}\left((e_x\cdot e_y) X^{x} + (e_y\cdot e_y) X^{y} \right)\boldsymbol{e}_{x} + A^{xy}\left((e_x\cdot e_x) X^{x} + (e_x\cdot e_y) X^{y} \right)\boldsymbol{e}_{y}$$

$$X\lfloor A = -A^{xy}\left((e_x\cdot e_y) X^{x} + (e_y\cdot e_y) X^{y} \right)\boldsymbol{e}_{x} + A^{xy}\left((e_x\cdot e_x) X^{x} + (e_x\cdot e_y) X^{y} \right)\boldsymbol{e}_{y}$$

$$X\rfloor A = AX^{x}\boldsymbol{e}_{x} + AX^{y}\boldsymbol{e}_{y}$$

```
def basic_multivector_operations_2D_orthogonal():
    Print_Function()
    o2d = Ga('e*x|y',g=[1,1])
    (ex,ey) = o2d.mv()
    print('g_{ii} =',o2d.g)

    X = o2d.mv('X','vector')
    A = o2d.mv('A','spinor')

    print('X =',X)
    print('A =',A)

    print('XA =',(X*A).Fmt(2))
    print(r'X\cdot A =',(X|A).Fmt(2))
    print(r'X\lfloor A =',(X<A).Fmt(2))
    print(r'X\lfloor A =',(X>A).Fmt(2))

    print('AX =',(A*X).Fmt(2))
    print(r'A\cdot X =',(A|X).Fmt(2))
    print(r'A\lfloor X =',(A<X).Fmt(2))
    print(r'A\lfloor X =',(A>X).Fmt(2))
    return
```

Code Output:

$$g_{ii} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$X = X^{x}\boldsymbol{e}_{x} + X^{y}\boldsymbol{e}_{y}$$

$$A = \begin{array}{l} A \\ + A^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} \end{array}$$

$$XA = \left (AX^{x} - A^{xy}X^{y}\right) \boldsymbol{e}_{x} + \left (AX^{y} + A^{xy}X^{x}\right) \boldsymbol{e}_{y}$$

$$X\cdot A = - A^{xy}X^{y}\boldsymbol{e}_{x} + A^{xy}X^{x}\boldsymbol{e}_{y}$$

$$X\lfloor A = - A^{xy}X^{y}\boldsymbol{e}_{x} + A^{xy}X^{x}\boldsymbol{e}_{y}$$

$$X\lfloor A = AX^{x}\boldsymbol{e}_{x} + AX^{y}\boldsymbol{e}_{y}$$

$$AX = \left (AX^{x} + A^{xy}X^{y}\right) \boldsymbol{e}_{x} + \left (AX^{y} - A^{xy}X^{x}\right) \boldsymbol{e}_{y}$$

$$A\cdot X = A^{xy}X^{y}\boldsymbol{e}_{x} - A^{xy}X^{x}\boldsymbol{e}_{y}$$

$$A\lfloor X = AX^{x}\boldsymbol{e}_{x} + AX^{y}\boldsymbol{e}_{y}$$

$$A\lfloor X = A^{xy}X^{y}\boldsymbol{e}_{x} - A^{xy}X^{x}\boldsymbol{e}_{y}$$

```
def rounding_numerical_components():
    Print_Function()
    o3d = Ga('e_x e_y e_z',g=[1,1,1])
    (ex,ey,ez) = o3d.mv()

    X = 1.2*ex+2.34*ey+0.555*ez
    Y = 0.333*ex+4*ey+5.3*ez

    print('X =',X)
    print('Nga(X,2) =',Nga(X,2))
    print('XY =',X*Y)
    print('Nga(XY,2) =',Nga(X*Y,2))
    return
```

Code Output:

$$X = 1.2\boldsymbol{e}_{x} + 2.34\boldsymbol{e}_{y} + 0.555\boldsymbol{e}_{z}$$

$$Nga(X,2) = 1.2\boldsymbol{e}_{x} + 2.3\boldsymbol{e}_{y} + 0.55\boldsymbol{e}_{z}$$

$$XY = \begin{aligned} & 12.7011 \\ & + 4.02078\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + 6.175185\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + 10.182\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$Nga(XY,2) = \begin{aligned} & 13.0 \\ & + 4.0\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + 6.2\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + 10.0\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

```
def derivatives_in_rectangular_coordinates():
    Print_Function()
    X = (x,y,z) = symbols('x y z')
    o3d = Ga('e_x e_y e_z',g=[1,1,1],coords=X)
    (ex,ey,ez) = o3d.mv()
    grad = o3d.grad

    f = o3d.mv('f','scalar',f=True)
    A = o3d.mv('A','vector',f=True)
    B = o3d.mv('B','bivector',f=True)
    C = o3d.mv('C','mv')
    print('f =',f)
    print('A =',A)
    print('B =',B)
    print('C =',C)

    print(r'\nabla f =',grad*f)
    print(r'\nabla\cdot A =',grad|A)
    print(r'\nabla A =',grad*A)

    print(r' I(\nabla\W A) =', o3d.I()*(grad^A))
    print(r'\nabla B =',grad*B)
    print(r'\nabla\W B =',grad^B)
    print(r'\nabla\cdot B =',grad|B)
    return
```

Code Output:

$$f = f$$

$$A = A^{x}\boldsymbol{e}_{x} + A^{y}\boldsymbol{e}_{y} + A^{z}\boldsymbol{e}_{z}$$

$$B = B^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + B^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + B^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z}$$

$$C = \begin{aligned} & C \\ & + C^{x}\boldsymbol{e}_{x} + C^{y}\boldsymbol{e}_{y} + C^{z}\boldsymbol{e}_{z} \\ & + C^{xy}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + C^{xz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + C^{yz}\boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \\ & + C^{xyz}\boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$\nabla f = \partial_{x} f\boldsymbol{e}_{x} + \partial_{y} f\boldsymbol{e}_{y} + \partial_{z} f\boldsymbol{e}_{z}$$

$$\nabla \cdot A = \partial_{x} A^{x} + \partial_{y} A^{y} + \partial_{z} A^{z}$$

$$\nabla A = \begin{aligned} & \left (\partial_{x} A^{x} + \partial_{y} A^{y} + \partial_{z} A^{z} \right) \\ & + \left (- \partial_{y} A^{x} + \partial_{x} A^{y} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y} + \left (- \partial_{z} A^{x} + \partial_{x} A^{z} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{z} + \left (- \partial_{z} A^{y} + \partial_{y} A^{z} \right) \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$-I(\nabla \wedge A) = \left (- \partial_{z} A^{y} + \partial_{y} A^{z} \right) \boldsymbol{e}_{x} + \left (\partial_{z} A^{x} - \partial_{x} A^{z} \right) \boldsymbol{e}_{y} + \left (- \partial_{y} A^{x} + \partial_{x} A^{y} \right) \boldsymbol{e}_{z}$$

$$\nabla B = \begin{aligned} & \left (- \partial_{y} B^{xy} - \partial_{z} B^{xz} \right) \boldsymbol{e}_{x} + \left (\partial_{x} B^{xy} - \partial_{z} B^{yz} \right) \boldsymbol{e}_{y} + \left (\partial_{x} B^{xz} + \partial_{y} B^{yz} \right) \boldsymbol{e}_{z} \\ & + \left (\partial_{z} B^{xy} - \partial_{y} B^{xz} + \partial_{x} B^{yz} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z} \end{aligned}$$

$$\nabla \wedge B = \left (\partial_{z} B^{xy} - \partial_{y} B^{xz} + \partial_{x} B^{yz} \right) \boldsymbol{e}_{x}\wedge \boldsymbol{e}_{y}\wedge \boldsymbol{e}_{z}$$

$$\nabla \cdot B = \left (- \partial_{y} B^{xy} - \partial_{z} B^{xz} \right) \boldsymbol{e}_{x} + \left (\partial_{x} B^{xy} - \partial_{z} B^{yz} \right) \boldsymbol{e}_{y} + \left (\partial_{x} B^{xz} + \partial_{y} B^{yz} \right) \boldsymbol{e}_{z}$$

```
def derivatives_in_spherical_coordinates():
    Print_Function()
    X = (r,th,phi) = symbols('r theta phi')
    s3d = Ga('e_r e_theta e_phi',g=[1,r**2,r**2*sin(th)**2],coords=X,norm=True)
    (er,eth,ephi) = s3d.mv()
    grad = s3d.grad

    f = s3d.mv('f','scalar',f=True)
    A = s3d.mv('A','vector',f=True)
    B = s3d.mv('B','bivector',f=True)

    print('f =',f)
    print('A =',A)
    print('B =',B)

    print(r'\nabla f =',grad*f)
    print(r'\nabla\cdot A =',grad|A)
    print(r' I*(\nabla\W A) =',(s3d.E()*(grad^A)).simplify())
    print(r'\nabla\W B =',grad^B)
```

Code Output:

$$f = f$$

$$A = A^{r}\boldsymbol{e}_{r} + A^{\theta}\boldsymbol{e}_{\theta} + A^{\phi}\boldsymbol{e}_{\phi}$$

$$B = B^{r\theta}\boldsymbol{e}_{r}\wedge \boldsymbol{e}_{\theta} + B^{r\phi}\boldsymbol{e}_{r}\wedge \boldsymbol{e}_{\phi} + B^{\theta\phi}\boldsymbol{e}_{\theta}\wedge \boldsymbol{e}_{\phi}$$

$$\nabla f = \partial_{r} f\boldsymbol{e}_{r} + \frac{\partial_{\theta} f}{r}\boldsymbol{e}_{\theta} + \frac{\partial_{\phi} f}{r\sin{\left(\theta \right)}}\boldsymbol{e}_{\phi}$$

$$\nabla \cdot A = \frac{r\partial_{r} A^{r} + 2A^{r} + \frac{A^{\theta}}{\tan{\left(\theta \right)}} + \partial_{\theta} A^{\theta} + \frac{\partial_{\phi} A^{\phi}}{\sin{\left(\theta \right)}}}{r}$$

$$-I*(\nabla \wedge A) = \frac{\frac{A^{\phi}}{\tan{\left(\theta \right)}} + \partial_{\theta} A^{\phi} - \frac{\partial_{\phi} A^{\theta}}{\sin{\left(\theta \right)}}}{r}\boldsymbol{e}_{r} + \frac{-r\partial_{r} A^{\phi} - A^{\phi} + \frac{\partial_{\phi} A^{r}}{\sin{\left(\theta \right)}}}{r}\boldsymbol{e}_{\theta} + \frac{r\partial_{r} A^{\theta} + A^{\theta} - \partial_{\theta} A^{r}}{r}\boldsymbol{e}_{\phi}$$

$$\nabla \wedge B = \frac{r\partial_{r} B^{\theta\phi} - \frac{B^{r\phi}}{\tan{\left(\theta \right)}} + 2B^{\theta\phi} - \partial_{\theta} B^{r\phi} + \frac{\partial_{\phi} B^{r\theta}}{\sin{\left(\theta \right)}}}{r}\boldsymbol{e}_{r}\wedge \boldsymbol{e}_{\theta}\wedge \boldsymbol{e}_{\phi}$$

```
def noneuclidian_distance_calculation():
    Print_Function()
    from sympy import solve,sqrt
    Fmt(1)

    g = '0 # #,# 0 #,# # 1'
    nel = Ga('X Y e',g=g)
    (X,Y,e) = nel.mv()

    print('g_{ij} =',nel.g)

    print(r'(X\W Y)^{2} =',(X^Y)*(X^Y))

    L = X^Y^e
    B = L*e # D&L 10.152
    Bsq = (B*B).scalar()
    print(r'L = X\W Y\W e \T{ is a non euclidian line}')
    print('B = Le =',B)

    BeBr =B*e*B.rev()
    print(r'BeB^{\dagger} =',BeBr)
    print('B^{2} =',B*B)
```

```
print('L^{2} =',L*L) # D&L 10.153
(s,c,Binv,M,S,C,alpha) = symbols('s c (1/B) M S C alpha')

XdotY = nel.g[0,1]
Xdote = nel.g[0,2]
Ydote = nel.g[1,2]

Bhat = Binv*B # D&L 10.154
R = c+s*Bhat # Rotor R = exp(alpha*Bhat/2)
print(r's = \f{\sinh}{\alpha/2} \T{ and } c = \f{\cosh}{\alpha/2}')
print(r'e^{\alpha B/{2\abs{B}}} =',R)

Z = R*X*R.rev() # D&L 10.155
Z.obj = expand(Z.obj)
Z.obj = Z.obj.collect([Binv,s,c,XdotY])
print(r'RXR^{\dagger}',Z.Fmt(3))
W = Z|Y # Extract scalar part of multivector
# From this point forward all calculations are with sympy scalars
#print '#Objective is to determine value of C = cosh(alpha) such that W = 0'
W = W.scalar()
print(r'W = Z\cdot Y =',W)
W = expand(W)
W = simplify(W)
W = W.collect([s*Binv])

M = 1/Bsq
W = W.subs(Binv**2,M)
W = simplify(W)
Bmag = sqrt(XdotY**2 2*XdotY*Xdote*Ydote)
W = W.collect([Binv*c*s,XdotY])

#Double angle substitutions

W = W.subs(2*XdotY**2 4*XdotY*Xdote*Ydote,2/(Binv**2))
W = W.subs(2*c*s,S)
W = W.subs(c**2,(C+1)/2)
W = W.subs(s**2,(C 1)/2)
W = simplify(W)
W = W.subs(1/Binv,Bmag)
W = expand(W)

print(r'S = \f{\sinh}{\alpha} \T{ and } C = \f{\cosh}{\alpha}')

print('W =',W)

Wd = collect(W,[C,S],exact=True,evaluate=False)

Wd_1 = Wd[one]
Wd_C = Wd[C]
Wd_S = Wd[S]

print(r'\T{Scalar Coefficient} =',Wd_1)
print(r'\T{Cosh Coefficient} =',Wd_C)
print(r'\T{Sinh Coefficient} =',Wd_S)

print(r'\abs{B} =',Bmag)
Wd_1 = Wd_1.subs(Bmag,1/Binv)
Wd_C = Wd_C.subs(Bmag,1/Binv)
Wd_S = Wd_S.subs(Bmag,1/Binv)
```

```
    lhs = Wd_1+Wd_C*C
    rhs =  Wd_S*S
    lhs = lhs**2
    rhs = rhs**2
    W = expand(lhs  rhs)
    W = expand(W.subs(1/Binv**2,Bmag**2))
    W = expand(W.subs(S**2,C**2 1))
    W = W.collect([C,C**2],evaluate=False)

    a = simplify(W[C**2])
    b = simplify(W[C])
    c = simplify(W[one])

    print(r'\T{Require } aC^{2}+bC+c = 0')

    print('a =',a)
    print('b =',b)
    print('c =',c)

    x = Symbol('x')
    C =  solve(a*x**2+b*x+c,x)[0]
    print('b^{2} 4ac =',simplify(b**2 4*a*c))
    print(r'\f{\cosh}{\alpha} = C =  b/(2a) =',expand(simplify(expand(C))))
    return
```

Code Output:

$$g_{ij} = \left[\begin{array}{ccc} 0 & (X\cdot Y) & (X\cdot e) \\ (X\cdot Y) & 0 & (Y\cdot e) \\ (X\cdot e) & (Y\cdot e) & 1 \end{array}\right]$$

$(X\wedge Y)^2 = (X\cdot Y)^2$

$L = X\wedge Y\wedge e$ is a non-euclidian line

$B = Le = \boldsymbol{X}\wedge\boldsymbol{Y} - (Y\cdot e)\,\boldsymbol{X}\wedge\boldsymbol{e} + (X\cdot e)\,\boldsymbol{Y}\wedge\boldsymbol{e}$

$BeB^{\dagger} = (X\cdot Y)\left(-(X\cdot Y) + 2\,(X\cdot e)\,(Y\cdot e)\right)\boldsymbol{e}$

$B^2 = (X\cdot Y)\left((X\cdot Y) - 2\,(X\cdot e)\,(Y\cdot e)\right)$

$L^2 = (X\cdot Y)\left((X\cdot Y) - 2\,(X\cdot e)\,(Y\cdot e)\right)$

$s = \sinh(\alpha/2)$ and $c = \cosh(\alpha/2)$

$e^{\alpha B/2|B|} = c + (1/B)s\boldsymbol{X}\wedge\boldsymbol{Y} - (1/B)\,(Y\cdot e)\,s\boldsymbol{X}\wedge\boldsymbol{e} + (1/B)\,(X\cdot e)\,s\boldsymbol{Y}\wedge\boldsymbol{e}$

$$RXR^{\dagger} \begin{array}{l} \left((1/B)^2\,(X\cdot Y)^2\,s^2 - 2(1/B)^2\,(X\cdot Y)\,(X\cdot e)\,(Y\cdot e)\,s^2 + 2(1/B)\,(X\cdot Y)\,cs - 2(1/B)\,(X\cdot e)\,(Y\cdot e)\,cs + c^2\right)\boldsymbol{X} \\ + 2(1/B)\,(X\cdot e)^2\,cs\boldsymbol{Y} \\ + 2(1/B)\,(X\cdot Y)\,(X\cdot e)\,s\left(-(1/B)\,(X\cdot Y)\,s + 2(1/B)\,(X\cdot e)\,(Y\cdot e)\,s - c\right)\boldsymbol{e} \end{array}$$

$W = Z\cdot Y = (1/B)^2\,(X\cdot Y)^3\,s^2 - 4(1/B)^2\,(X\cdot Y)^2\,(X\cdot e)\,(Y\cdot e)\,s^2 + 4(1/B)^2\,(X\cdot Y)\,(X\cdot e)^2\,(Y\cdot e)^2\,s^2 + 2(1/B)\,(X\cdot Y)^2\,cs - 4(1/B)\,(X\cdot Y)\,(X\cdot e)\,(Y\cdot e)\,cs + (X\cdot Y)\,c^2$

$S = \sinh(\alpha)$ and $C = \cosh(\alpha)$

$W = (X\cdot Y)\,C - (X\cdot e)\,(Y\cdot e)\,C + (X\cdot e)\,(Y\cdot e) + S\sqrt{(X\cdot Y)^2 - 2\,(X\cdot Y)\,(X\cdot e)\,(Y\cdot e)}$

Scalar Coefficient $= (X\cdot e)\,(Y\cdot e)$

Cosh Coefficient $= (X\cdot Y) - (X\cdot e)\,(Y\cdot e)$

Sinh Coefficient $= \sqrt{(X\cdot Y)^2 - 2\,(X\cdot Y)\,(X\cdot e)\,(Y\cdot e)}$

$|B| = \sqrt{(X\cdot Y)^2 - 2\,(X\cdot Y)\,(X\cdot e)\,(Y\cdot e)}$

Require $aC^2 + bC + c = 0$

$$a = (X \cdot e)^2 (Y \cdot e)^2$$

$$b = 2 (X \cdot e) (Y \cdot e) ((X \cdot Y) - (X \cdot e) (Y \cdot e))$$

$$c = (X \cdot Y)^2 - 2 (X \cdot Y) (X \cdot e) (Y \cdot e) + (X \cdot e)^2 (Y \cdot e)^2$$

$$b^2 - 4ac = 0$$

$$\cosh(\alpha) = C = -b/(2a) = -\frac{(X \cdot Y)}{(X \cdot e) (Y \cdot e)} + 1$$

```
def conformal_representations_of_circles_lines_spheres_and_planes():
    Print_Function()
    global n,nbar
    Fmt(1)
    g = '1 0 0 0 0,0 1 0 0 0,0 0 1 0 0,0 0 0 0 2,0 0 0 2 0'

    c3d = Ga('e_1 e_2 e_3 n \\bar{n}',g=g)
    (e1,e2,e3,n,nbar) = c3d.mv()

    print('g_{ij} =',c3d.g)

    e = n+nbar
    #conformal representation of points

    A = make_vector(e1, ga=c3d)     # point a = (1,0,0)  A = F(a)
    B = make_vector(e2, ga=c3d)     # point b = (0,1,0)  B = F(b)
    C = make_vector( e1, ga=c3d)    # point c = ( 1,0,0) C = F(c)
    D = make_vector(e3, ga=c3d)     # point d = (0,0,1)  D = F(d)
    X = make_vector('x',3, ga=c3d)

    print('F(a) =',A)
    print('F(b) =',B)
    print('F(c) =',C)
    print('F(d) =',D)
    print('F(x) =',X)

    print(r'a = e1, b = e2, c =  e1, \T{ and } d = e3')
    print(r'A = F(a) = 1/2(a^2 n+2a nbar)\T{, etc.}')
    print(r'\T{Circle through $a$, $b$, and $c$}')
    print(r'\T{Circle: } A\W B\W C\W X = 0 =',(A^B^C^X))
    print(r'\T{Line through $a$ and $b$}')
    print(r'\T{Line  : } A\W B\W n\W X = 0 =',(A^B^n^X))
    print(r'\T{Sphere through $a$, $b$, $c$, and $d$}')
    print(r'\T{Sphere: } A\W B\W C\W D\W X = 0 =',(((A^B)^C)^D)^X)
    print(r'\T{Plane through $a$, $b$, and $d$}')
    print(r'\T{Plane : } A\W B\W n\W D\W X = 0 =',(A^B^n^D^X))

    L = (A^B^e)^X

    print(r'\T{Hyperbolic\;\; Circle: } (A\W B\W e)\W X = 0',L.Fmt(3))
    return
```

Code Output:

$$g_{ij} = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{array}\right]$$

$$F(a) = \boldsymbol{e}_1 + \frac{1}{2}\boldsymbol{n} - \frac{1}{2}\bar{\boldsymbol{n}}$$

$F(b) = \boldsymbol{e}_2 + \frac{1}{2}\boldsymbol{n} - \frac{1}{2}\bar{\boldsymbol{n}}$

$F(c) = -\boldsymbol{e}_1 + \frac{1}{2}\boldsymbol{n} - \frac{1}{2}\bar{\boldsymbol{n}}$

$F(d) = \boldsymbol{e}_3 + \frac{1}{2}\boldsymbol{n} - \frac{1}{2}\bar{\boldsymbol{n}}$

$F(x) = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + x_3\boldsymbol{e}_3 + \left(\frac{(x_1)^2}{2} + \frac{(x_2)^2}{2} + \frac{(x_3)^2}{2}\right)\boldsymbol{n} - \frac{1}{2}\bar{\boldsymbol{n}}$

$a = e1, b = e2, c = -e1,$ and $d = e3$

$A = F(a) = 1/2(a^2 n + 2a - nbar)$, etc.

Circle through a, b, and c

Circle: $A \wedge B \wedge C \wedge X = 0 = -x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} + x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \bar{\boldsymbol{n}} + \left(\frac{(x_1)^2}{2} + \frac{(x_2)^2}{2} + \frac{(x_3)^2}{2} - \frac{1}{2}\right)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}}$

Line through a and b

Line : $A \wedge B \wedge n \wedge X = 0 = -x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} + \left(\frac{x_1}{2} + \frac{x_2}{2} - \frac{1}{2}\right)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}} + \frac{x_3}{2}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}} - \frac{x_3}{2}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}}$

Sphere through a, b, c, and d

Sphere: $A \wedge B \wedge C \wedge D \wedge X = 0 = \left(-\frac{(x_1)^2}{2} - \frac{(x_2)^2}{2} - \frac{(x_3)^2}{2} + \frac{1}{2}\right)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}}$

Plane through a, b, and d

$$\begin{aligned}\text{Plane : } A \wedge B \wedge n \wedge D \wedge X = 0 = &\left(-\frac{x_1}{2} - \frac{x_2}{2} - \frac{x_3}{2} + \frac{1}{2}\right)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}} \\ &- x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \\ &- x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \bar{\boldsymbol{n}}\end{aligned}$$

$$\begin{aligned}\text{Hyperbolic Circle: } (A \wedge B \wedge e) \wedge X = 0 + &\left(-\frac{(x_1)^2}{2} + x_1 - \frac{(x_2)^2}{2} + x_2 - \frac{(x_3)^2}{2} - \frac{1}{2}\right)\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}} \\ &+ x_3\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}} \\ &- x_3\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{n} \wedge \bar{\boldsymbol{n}}\end{aligned}$$

```python
def properties_of_geometric_objects():
    Print_Function()
    global n, nbar
    Fmt(1)
    g = '# # # 0 0,'+ \
        '# # # 0 0,'+ \
        '# # # 0 0,'+ \
        '0 0 0 0 2,'+ \
        '0 0 0 2 0'

    c3d = Ga('p1 p2 p3 n \\bar{n}',g=g)
    (p1,p2,p3,n,nbar) = c3d.mv()

    print('g_{ij} =',c3d.g)

    P1 = F(p1)
    P2 = F(p2)
    P3 = F(p3)

    tprint('Extracting direction of line from $L = P1\W P2\W n$')

    L = P1^P2^n
    delta = (L|n)|nbar
```

```
    print(r'(L\cdot n)\cdot \bar{n} =',delta)

    tprint('Extracting plane of circle from $C = P1\W P2\W P3$')

    C = P1^P2^P3
    delta = ((C^n)|n)|nbar
    print(r'((C\W n)\cdot n)\cdot \bar{n}=',delta)
    print(r'(p2 p1)\W (p3 p1)=',(p2 p1)^(p3 p1))
    return
```

Code Output:

$$g_{ij} = \begin{bmatrix} (p_1 \cdot p_1) & (p_1 \cdot p_2) & (p_1 \cdot p_3) & 0 & 0 \\ (p_1 \cdot p_2) & (p_2 \cdot p_2) & (p_2 \cdot p_3) & 0 & 0 \\ (p_1 \cdot p_3) & (p_2 \cdot p_3) & (p_3 \cdot p_3) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

Extracting direction of line from $L = P1 \wedge P2 \wedge n$

$(L \cdot n) \cdot \bar{n} = 2\boldsymbol{p}_1 - 2\boldsymbol{p}_2$

Extracting plane of circle from $C = P1 \wedge P2 \wedge P3$

$((C \wedge n) \cdot n) \cdot \bar{n} = 2\boldsymbol{p}_1 \wedge \boldsymbol{p}_2 - 2\boldsymbol{p}_1 \wedge \boldsymbol{p}_3 + 2\boldsymbol{p}_2 \wedge \boldsymbol{p}_3$

$(p2 - p1) \wedge (p3 - p1) = \boldsymbol{p}_1 \wedge \boldsymbol{p}_2 - \boldsymbol{p}_1 \wedge \boldsymbol{p}_3 + \boldsymbol{p}_2 \wedge \boldsymbol{p}_3$

```
def extracting_vectors_from_conformal_2_blade():
    Print_Function()
    Fmt(1)
    print(r'B = P1\W P2')

    g = '0  1  #,'+ \
        ' 1  0  #,'+ \
        '# # #'

    c2b = Ga('P1 P2 a',g=g)
    (P1,P2,a) = c2b.mv()

    print('g_{ij} =',c2b.g)

    B = P1^P2
    Bsq = B*B
    print('B^{2} =',Bsq)
    ap = a ( a^B)*B
    print(r"a' = a ( a\W )B =",ap)

    Ap = ap+ap*B
    Am = ap  ap*B

    print("A+ = a'+a'B =",Ap)
    print("A  = a ' a'B =",Am)

    print('(A+)^{2} =',Ap*Ap)
    print('(A )^{2} =',Am*Am)

    aB = a|B
    print(r'a\cdot B =',aB)
    return
```

Code Output:

$B = P1 \wedge P2$

$$g_{ij} = \left[\begin{array}{ccc} 0 & -1 & (P_1 \cdot a) \\ -1 & 0 & (P_2 \cdot a) \\ (P_1 \cdot a) & (P_2 \cdot a) & (a \cdot a) \end{array} \right]$$

$$B^2 = 1$$

$$a' = a - (a \wedge)B = -(P_2 \cdot a)\,\boldsymbol{P}_1 - (P_1 \cdot a)\,\boldsymbol{P}_2$$

$$A+ = a' + a'B = -2\,(P_2 \cdot a)\,\boldsymbol{P}_1$$

$$A- = a' - a'B = -2\,(P_1 \cdot a)\,\boldsymbol{P}_2$$

$$(A+)^2 = 0$$

$$(A-)^2 = 0$$

$$a \cdot B = -(P_2 \cdot a)\,\boldsymbol{P}_1 + (P_1 \cdot a)\,\boldsymbol{P}_2$$

```
def reciprocal_frame_test():
    Print_Function()
    Fmt(1)
    g = '1 # #,'+ \
        '# 1 #,'+ \
        '# # 1'

    ng3d = Ga('e1 e2 e3',g=g)
    (e1,e2,e3) = ng3d.mv()

    print('g_{ij} =',ng3d.g)

    E = e1^e2^e3
    Esq = (E*E).scalar()
    print('E =',E)
    print('E^{2} =',Esq)
    Esq_inv = 1/Esq

    E1 = (e2^e3)*E
    E2 = ( 1)*(e1^e3)*E
    E3 = (e1^e2)*E

    print(r'E1 = (e2\W e3)E =',E1)
    print(r'E2 =(e1\W e3)E =',E2)
    print(r'E3 = (e1\W e2)E =',E3)

    w = (E1|e2)
    w = w.expand()
    print(r'E1\cdot e2 =',w)

    w = (E1|e3)
    w = w.expand()
    print(r'E1\cdot e3 =',w)

    w = (E2|e1)
    w = w.expand()
    print(r'E2\cdot e1 =',w)

    w = (E2|e3)
    w = w.expand()
    print(r'E2\cdot e3 =',w)

    w = (E3|e1)
    w = w.expand()
    print(r'E3\cdot e1 =',w)
```

```
    w = (E3|e2)
    w = w.expand()
    print(r'E3\cdot e2 =',w)

    w = (E1|e1)
    w = (w.expand()).scalar()
    Esq = expand(Esq)
    print(r'(E1\cdot e1)/E^{2} =',simplify(w/Esq))

    w = (E2|e2)
    w = (w.expand()).scalar()
    print(r'(E2\cdot e2)/E^{2} =',simplify(w/Esq))

    w = (E3|e3)
    w = (w.expand()).scalar()
    print(r'(E3\cdot e3)/E^{2} =',simplify(w/Esq))
    return
```

Code Output:

$$g_{ij} = \left[\begin{array}{ccc} 1 & (e_1 \cdot e_2) & (e_1 \cdot e_3) \\ (e_1 \cdot e_2) & 1 & (e_2 \cdot e_3) \\ (e_1 \cdot e_3) & (e_2 \cdot e_3) & 1 \end{array}\right]$$

$$E = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$$

$$E^2 = (e_1 \cdot e_2)^2 - 2\,(e_1 \cdot e_2)\,(e_1 \cdot e_3)\,(e_2 \cdot e_3) + (e_1 \cdot e_3)^2 + (e_2 \cdot e_3)^2 - 1$$

$$E1 = (e2 \wedge e3)E = \left((e_2 \cdot e_3)^2 - 1\right)\boldsymbol{e}_1 + ((e_1 \cdot e_2) - (e_1 \cdot e_3)\,(e_2 \cdot e_3))\,\boldsymbol{e}_2 + (-\,(e_1 \cdot e_2)\,(e_2 \cdot e_3) + (e_1 \cdot e_3))\,\boldsymbol{e}_3$$

$$E2 = -(e1 \wedge e3)E = ((e_1 \cdot e_2) - (e_1 \cdot e_3)\,(e_2 \cdot e_3))\,\boldsymbol{e}_1 + \left((e_1 \cdot e_3)^2 - 1\right)\boldsymbol{e}_2 + (-\,(e_1 \cdot e_2)\,(e_1 \cdot e_3) + (e_2 \cdot e_3))\,\boldsymbol{e}_3$$

$$E3 = (e1 \wedge e2)E = (-\,(e_1 \cdot e_2)\,(e_2 \cdot e_3) + (e_1 \cdot e_3))\,\boldsymbol{e}_1 + (-\,(e_1 \cdot e_2)\,(e_1 \cdot e_3) + (e_2 \cdot e_3))\,\boldsymbol{e}_2 + \left((e_1 \cdot e_2)^2 - 1\right)\boldsymbol{e}_3$$

$$E1 \cdot e2 = 0$$

$$E1 \cdot e3 = 0$$

$$E2 \cdot e1 = 0$$

$$E2 \cdot e3 = 0$$

$$E3 \cdot e1 = 0$$

$$E3 \cdot e2 = 0$$

$$(E1 \cdot e1)/E^2 = 1$$

$$(E2 \cdot e2)/E^2 = 1$$

$$(E3 \cdot e3)/E^2 = 1$$

```
def signature_test():
    Print_Function()

    e3d = Ga('e1 e2 e3',g=[1,1,1])
    print('g =', e3d.g)
    print(r'\T{Signature = (3,0)\:} I =', e3d.I(),'\: I^{2} =', e3d.I()*e3d.I())

    e3d = Ga('e1 e2 e3',g=[2,2,2])
    print('g =', e3d.g)
    print('r\T{Signature = (3,0)\:} I =', e3d.I(),'\; I^{2} =', e3d.I()*e3d.I())

    sp4d = Ga('e1 e2 e3 e4',g=[1, 1, 1, 1])
    print('g =', sp4d.g)
    print(r'\T{Signature = (1,3)\:} I =', sp4d.I(),'\: I^{2} =', sp4d.I()*sp4d.I())
```

```
    sp4d = Ga('e1 e2 e3 e4',g=[2, 2, 2, 2])
    print('g =', sp4d.g)
    print(r'\T{Signature = (1,3)\:} I =', sp4d.I(),'\: I^{2} =', sp4d.I()*sp4d.I())

    e4d = Ga('e1 e2 e3 e4',g=[1,1,1,1])
    print('g =', e4d.g)
    print(r'\T{Signature = (4,0)\:} I =', e4d.I(),'\: I^{2} =', e4d.I()*e4d.I())

    cf3d = Ga('e1 e2 e3 e4 e5',g=[1,1,1,1, 1])
    print('g =', cf3d.g)
    print(r'\T{Signature = (4,1)\:} I =', cf3d.I(),'\: I^{2} =', cf3d.I()*cf3d.I())

    cf3d = Ga('e1 e2 e3 e4 e5',g=[2,2,2,2, 2])
    print('g =', cf3d.g)
    print(r'\T{Signature = (4,1)\:} I =', cf3d.I(),'\: I^{2} =', cf3d.I()*cf3d.I())

    return
```

Code Output:

$$g = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Signature = (3,0) $I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$ $I^2 = -1$

$$g = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

rSignature = (3,0) $I = \frac{\sqrt{2}}{4}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$ $I^2 = -1$

$$g = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Signature = (1,3) $I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4$ $I^2 = -1$

$$g = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}$$

Signature = (1,3) $I = \frac{1}{4}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4$ $I^2 = -1$

$$g = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Signature = (4,0) $I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4$ $I^2 = 1$

$$g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Signature = (4,1) $I = \boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4 \wedge \boldsymbol{e}_5$ $I^2 = -1$

$$g = \left[\begin{array}{ccccc} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & -2 \end{array}\right]$$

Signature = (4,1) $I = \frac{\sqrt{2}}{8}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \wedge \boldsymbol{e}_4 \wedge \boldsymbol{e}_5$ $I^2 = -1$

```
def Fmt_test():
    Print_Function()

    e3d = Ga('e1 e2 e3',g=[1,1,1])

    v = e3d.mv('v','vector')
    B = e3d.mv('B','bivector')
    M = e3d.mv('M','mv')

    Fmt(2)

    tprint('Global $Fmt = 2$')

    print('v =',v)
    print('B =',B)
    print('M =',M)

    tprint('Using $.Fmt()$ Function')

    print('v.Fmt(3) =',v.Fmt(3))
    print('B.Fmt(3) =',B.Fmt(3))
    print('M.Fmt(2) =',M.Fmt(2))
    print('M.Fmt(1) =',M.Fmt(1))

    print('Global $Fmt = 1$')

    Fmt(1)

    print('v =',v)
    print('B =',B)
    print('M =',M)

    return
```

Code Output:

Global $Fmt = 2$

$v = v^{1}\boldsymbol{e}_1 + v^{2}\boldsymbol{e}_2 + v^{3}\boldsymbol{e}_3$

$B = B^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + B^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + B^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3$

$$M = \begin{array}{l} M \\ + M^{1}\boldsymbol{e}_1 + M^{2}\boldsymbol{e}_2 + M^{3}\boldsymbol{e}_3 \\ + M^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + M^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + M^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \\ + M^{123}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \end{array}$$

Using $.Fmt()$ Function

$$v.Fmt(3) = \begin{array}{l} v^{1}\boldsymbol{e}_1 \\ + v^{2}\boldsymbol{e}_2 \\ + v^{3}\boldsymbol{e}_3 \end{array}$$

$$B.Fmt(3) = \begin{aligned} & B^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \\ & + B^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 \\ & + B^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \end{aligned}$$

$$M.Fmt(2) = \begin{aligned} & M \\ & + M^{1}\boldsymbol{e}_1 + M^{2}\boldsymbol{e}_2 + M^{3}\boldsymbol{e}_3 \\ & + M^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + M^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + M^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \\ & + M^{123}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3 \end{aligned}$$

$$M.Fmt(1) = M + M^{1}\boldsymbol{e}_1 + M^{2}\boldsymbol{e}_2 + M^{3}\boldsymbol{e}_3 + M^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + M^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + M^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 + M^{123}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$$

$$Global\mathrm{Fmt} = 1$$

$$v = v^{1}\boldsymbol{e}_1 + v^{2}\boldsymbol{e}_2 + v^{3}\boldsymbol{e}_3$$

$$B = B^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + B^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + B^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3$$

$$M = M + M^{1}\boldsymbol{e}_1 + M^{2}\boldsymbol{e}_2 + M^{3}\boldsymbol{e}_3 + M^{12}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 + M^{13}\boldsymbol{e}_1 \wedge \boldsymbol{e}_3 + M^{23}\boldsymbol{e}_2 \wedge \boldsymbol{e}_3 + M^{123}\boldsymbol{e}_1 \wedge \boldsymbol{e}_2 \wedge \boldsymbol{e}_3$$