```
def Maxwells_Equations_in_Geometric_Calculus():
    Print_Function()
    X = symbols('t x y z')
    (g0,g1,g2,g3,grad) = MV.setup('gamma*t|x|y|z',metric='[1,-1,-1,-1]',coords=X)
    I = MV.I
    B = MV('B','vector',fct=True)
    E = MV('E','vector',fct=True)
    B.set_coef(1,0,0)
    E.set_coef(1,0,0)
    B *= g0
    E *= g0
    J = MV('J','vector',fct=True)
    F = E+I*B
    print r'\text{Pseudo Scalar\;\;}I =',I
    print '\\text{Magnetic Field Bi-Vector\\;\\;} B = \\bm{B\\gamma_{t}} =',B
    print '\\text{Electric Field Bi-Vector\\;\\;} E = \\bm{E\\gamma_{t}} =',E
    print '\\text{Electromagnetic Field Bi-Vector\\;\\;} F = E+IB =',F
    print '%\\text{Four Current Density\\;\\;} J =',J
    gradF = grad*F
    print '#Geometric Derivative of Electomagnetic Field Bi-Vector'
    gradF.Fmt(3,'grad*F')
    print '#Maxwell Equations'
    print 'grad*F = J'
    print '#Div $E$ and Curl $H$ Equations'
    (gradF.grade(1)-J).Fmt(3,'%\\grade{\\nabla F}_{1} -J = 0')
    print '#Curl $E$ and Div $B$ equations'
    (gradF.grade(3)).Fmt(3,'%\\grade{\\nabla F}_{3} = 0')
    return
```

Code Output:

$$\text{Pseudo Scalar}\;\; I = \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_y} \wedge \boldsymbol{\gamma_z}$$

$$\text{Magnetic Field Bi-Vector}\;\; B = \boldsymbol{B\gamma_t} = -B^x \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} - B^y \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_y} - B^z \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_z}$$

$$\text{Electric Field Bi-Vector}\;\; E = \boldsymbol{E\gamma_t} = -E^x \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} - E^y \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_y} - E^z \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_z}$$

$$\text{Electromagnetic Field Bi-Vector}\;\; F = E + IB = -E^x \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} - E^y \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_y} - E^z \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_z} - B^z \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_y} + B^y \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_z} - B^x \boldsymbol{\gamma_y} \wedge \boldsymbol{\gamma_z}$$

$$\text{Four Current Density}\;\; J = J^t \boldsymbol{\gamma_t} + J^x \boldsymbol{\gamma_x} + J^y \boldsymbol{\gamma_y} + J^z \boldsymbol{\gamma_z}$$

Geometric Derivative of Electomagnetic Field Bi-Vector

$$\begin{aligned}
\boldsymbol{\nabla} F = & \left(\partial_x E^x + \partial_y E^y + \partial_z E^z\right) \boldsymbol{\gamma_t} \\
& + \left(-\partial_z B^y + \partial_y B^z - \partial_t E^x\right) \boldsymbol{\gamma_x} \\
& + \left(\partial_z B^x - \partial_x B^z - \partial_t E^y\right) \boldsymbol{\gamma_y} \\
& + \left(-\partial_y B^x + \partial_x B^y - \partial_t E^z\right) \boldsymbol{\gamma_z} \\
& + \left(-\partial_t B^z + \partial_y E^x - \partial_x E^y\right) \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_y} \\
& + \left(\partial_t B^y + \partial_z E^x - \partial_x E^z\right) \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_z} \\
& + \left(-\partial_t B^x + \partial_z E^y - \partial_y E^z\right) \boldsymbol{\gamma_t} \wedge \boldsymbol{\gamma_y} \wedge \boldsymbol{\gamma_z} \\
& + \left(\partial_x B^x + \partial_y B^y + \partial_z B^z\right) \boldsymbol{\gamma_x} \wedge \boldsymbol{\gamma_y} \wedge \boldsymbol{\gamma_z}
\end{aligned}$$

Maxwell Equations

$$\boldsymbol{\nabla} F = J$$

Div E and Curl H Equations

$$\begin{aligned}\langle\nabla F\rangle_1 - J = 0 = &\left(-J^t + \partial_x E^x + \partial_y E^y + \partial_z E^z\right)\boldsymbol{\gamma_t}\\ &+ \left(-J^x - \partial_z B^y + \partial_y B^z - \partial_t E^x\right)\boldsymbol{\gamma_x}\\ &+ \left(-J^y + \partial_z B^x - \partial_x B^z - \partial_t E^y\right)\boldsymbol{\gamma_y}\\ &+ \left(-J^z - \partial_y B^x + \partial_x B^y - \partial_t E^z\right)\boldsymbol{\gamma_z}\end{aligned}$$

Curl E and Div B equations

$$\begin{aligned}\langle\nabla F\rangle_3 = 0 = &\left(-\partial_t B^z + \partial_y E^x - \partial_x E^y\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y}\\ &+ \left(\partial_t B^y + \partial_z E^x - \partial_x E^z\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_z}\\ &+ \left(-\partial_t B^x + \partial_z E^y - \partial_y E^z\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z}\\ &+ \left(\partial_x B^x + \partial_y B^y + \partial_z B^z\right)\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z}\end{aligned}$$

```
def Dirac_Equation_in_Geometric_Calculus():
    Print_Function()
    vars = symbols('t x y z')
    (g0,g1,g2,g3,grad) = MV.setup('gamma*t|x|y|z',metric='[1,-1,-1,-1]',coords=vars)
    I = MV.I
    (m,e) = symbols('m e')
    psi = MV('psi','spinor',fct=True)
    A = MV('A','vector',fct=True)
    sig_z = g3*g0
    print '\\text{4-Vector Potential\\;\\;}\\bm{A} =',A
    print '\\text{8-component real spinor\\;\\;}\\bm{\\psi} =',psi
    dirac_eq = (grad*psi)*I*sig_z-e*A*psi-m*psi*g0
    dirac_eq.simplify()
    dirac_eq.Fmt(3,r'%\text{Dirac Equation\;\;}\nabla \bm{\psi} I \sigma_{z}-e\bm{A}\bm{\psi}-m\bm{\psi}\gamma_{t} = 0')

    return
```

Code Output:

4-Vector Potential $\boldsymbol{A} = A^t\boldsymbol{\gamma_t} + A^x\boldsymbol{\gamma_x} + A^y\boldsymbol{\gamma_y} + A^z\boldsymbol{\gamma_z}$

8-component real spinor $\boldsymbol{\psi} = \psi + \psi^{tx}\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x} + \psi^{ty}\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_y} + \psi^{tz}\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_z} + \psi^{xy}\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y} + \psi^{xz}\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_z} + \psi^{yz}\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z} + \psi^{txyz}\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z}$

Dirac Equation
$$\begin{aligned}\nabla\boldsymbol{\psi}I\sigma_z - e\boldsymbol{A}\boldsymbol{\psi} - m\boldsymbol{\psi}\gamma_t = 0 = &\left(-eA^t\psi - eA^x\psi^{tx} - eA^y\psi^{ty} - eA^z\psi^{tz} - m\psi - \partial_y\psi^{tx} - \partial_z\psi^{txyz} + \partial_x\psi^{ty} + \partial_t\psi^{xy}\right)\boldsymbol{\gamma_t}\\ &+ \left(-eA^t\psi^{tx} - eA^x\psi - eA^y\psi^{xy} - eA^z\psi^{xz} + m\psi^{tx} + \partial_y\psi - \partial_t\psi^{ty} - \partial_x\psi^{xy} + \partial_z\psi^{yz}\right)\boldsymbol{\gamma_x}\\ &+ \left(-eA^t\psi^{ty} + eA^x\psi^{xy} - eA^y\psi - eA^z\psi^{yz} + m\psi^{ty} - \partial_x\psi + \partial_t\psi^{tx} - \partial_y\psi^{xy} - \partial_z\psi^{xz}\right)\boldsymbol{\gamma_y}\\ &+ \left(-eA^t\psi^{tz} + eA^x\psi^{xz} + eA^y\psi^{yz} - eA^z\psi + m\psi^{tz} + \partial_t\psi^{txyz} - \partial_z\psi^{xy} + \partial_y\psi^{xz} - \partial_x\psi^{yz}\right)\boldsymbol{\gamma_z}\\ &+ \left(-eA^t\psi^{xy} + eA^x\psi^{ty} - eA^y\psi^{tx} - eA^z\psi^{txyz} - m\psi^{xy} - \partial_t\psi + \partial_x\psi^{tx} + \partial_y\psi^{ty} + \partial_z\psi^{tz}\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y}\\ &+ \left(-eA^t\psi^{xz} + eA^x\psi^{tz} + eA^y\psi^{txyz} - eA^z\psi^{tx} - m\psi^{xz} + \partial_x\psi^{txyz} + \partial_z\psi^{ty} - \partial_y\psi^{tz} - \partial_t\psi^{yz}\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_z}\\ &+ \left(-eA^t\psi^{yz} - eA^x\psi^{txyz} + eA^y\psi^{tz} - eA^z\psi^{ty} - m\psi^{yz} - \partial_z\psi^{tx} + \partial_y\psi^{txyz} + \partial_x\psi^{tz} + \partial_t\psi^{xz}\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z}\\ &+ \left(-eA^t\psi^{txyz} - eA^x\psi^{yz} + eA^y\psi^{xz} - eA^z\psi^{xy} + m\psi^{txyz} + \partial_z\psi - \partial_t\psi^{tz} - \partial_x\psi^{xz} - \partial_y\psi^{yz}\right)\boldsymbol{\gamma_x}\wedge\boldsymbol{\gamma_y}\wedge\boldsymbol{\gamma_z}\end{aligned}$$

```
def Lorentz_Tranformation_in_Geometric_Algebra():
    Print_Function()
    (alpha,beta,gamma) = symbols('alpha beta gamma')
    (x,t,xp,tp) = symbols("x t x' t'")
    (g0,g1) = MV.setup('gamma*t|x',metric='[1,-1]')
    from sympy import sinh,cosh
    R = cosh(alpha/2)+sinh(alpha/2)*(g0^g1)
```

```
    X = t*g0+x*g1
    Xp = tp*g0+xp*g1
    print 'R =',R
    print r"#%t\bm{\gamma_{t}}+x\bm{\gamma_{x}} = t'\bm{\gamma'_{t}}+x'\bm{\gamma'_{x}} = R\lp t'\bm{\gamma_{t}}+x'\bm{\gamma_{x}}\rp R^{\dagger}"
    Xpp = R*Xp*R.rev()
    Xpp = Xpp.collect([xp,tp])
    Xpp = Xpp.subs({2*sinh(alpha/2)*cosh(alpha/2):sinh(alpha),sinh(alpha/2)**2+cosh(alpha/2)**2:cosh(alpha)})
    print r"%t\bm{\gamma_{t}}+x\bm{\gamma_{x}} =",Xpp
    Xpp = Xpp.subs({sinh(alpha):gamma*beta,cosh(alpha):gamma})
    print r'%\f{\sinh}{\alpha} = \gamma\beta'
    print r'%\f{\cosh}{\alpha} = \gamma'
    print r"%t\bm{\gamma_{t}}+x\bm{\gamma_{x}} =",Xpp.collect(gamma)
    return
```

Code Output:

$$R = \cosh\left(\frac{1}{2}\alpha\right) + \sinh\left(\frac{1}{2}\alpha\right)\boldsymbol{\gamma_t}\wedge\boldsymbol{\gamma_x}$$

$$t\boldsymbol{\gamma_t} + x\boldsymbol{\gamma_x} = t'\boldsymbol{\gamma'_t} + x'\boldsymbol{\gamma'_x} = R\left(t'\boldsymbol{\gamma_t} + x'\boldsymbol{\gamma_x}\right)R^{\dagger}$$

$$t\boldsymbol{\gamma_t} + x\boldsymbol{\gamma_x} = \left(t'\cosh\left(\alpha\right) - x'\sinh\left(\alpha\right)\right)\boldsymbol{\gamma_t} + \left(-t'\sinh\left(\alpha\right) + x'\cosh\left(\alpha\right)\right)\boldsymbol{\gamma_x}$$

$$\sinh\left(\alpha\right) = \gamma\beta$$

$$\cosh\left(\alpha\right) = \gamma$$

$$t\boldsymbol{\gamma_t} + x\boldsymbol{\gamma_x} = \left(\gamma\left(-\beta x' + t'\right)\right)\boldsymbol{\gamma_t} + \left(\gamma\left(-\beta t' + x'\right)\right)\boldsymbol{\gamma_x}$$