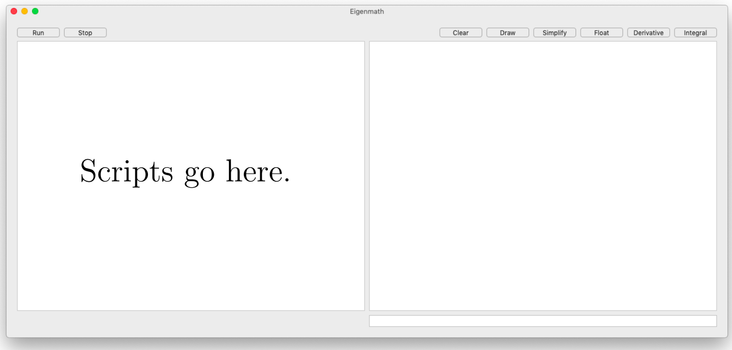
EXAMPLES

Here are some simple but interesting problems with example scripts. The scripts, shown in blue type, can be pasted into the Eigenmath script window and then run by clicking the Run button.



1. What is the probability that 23 people have different birthdays? The first person has a birthday on any day. For the second person, the probability is 364/365 that the second person's birthday is different from the first. For the third person the probability is 363/365 that the third person's birthday is different from the first and second. The pattern repeats for all the people. Multiply together all the probabilities to obtain

$$p = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{343}{365} = \frac{365!/(365-23)!}{365^{23}}$$

for the probability that 23 people have different birthdays.

```
-- www.eigenmath.org/birthday.txt
"Product method"
p = product(k,1,23,(365-k+1)/365)
float(p)
"Factorial method"
p = 365! / (365 - 23)! / 365^23
float(p)
"Probability of at least one shared birthday"
1.0 - p
```

2. Let

$$\mathbf{E} = \begin{pmatrix} A\sin(kz - \omega t + \phi) \\ 0 \\ 0 \end{pmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{pmatrix} 0 \\ A\sin(kz - \omega t + \phi) \\ 0 \end{pmatrix}$$

where $k = \omega/c$. Verify that $\mathbf{E}$ and $\mathbf{B}$ are solutions to the free-field Maxwell equations

$$\nabla \cdot \mathbf{E} = 0$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} + c^{-1}\frac{\partial}{\partial t}\mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\nabla \times \mathbf{B} - c^{-1}\frac{\partial}{\partial t}\mathbf{E} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

```
k = omega/c
E = (A*sin(k*z - omega*t + phi), 0, 0)
B = (0, A*sin(k*z - omega*t + phi), 0)
div(E)
div(B)
curl(E) + d(B,t)/c
curl(B) - d(E,t)/c
```

3. Let

$$\psi = \exp(ik_x x + ik_y y + ik_z z - i\omega t)$$

where

$$\omega = \sqrt{k_x^2 + k_y^2 + k_z^2 + m^2}$$

Verify that ψ is a solution to the Klein-Gordon equation

$$\frac{\partial^2}{\partial t^2}\psi - \frac{\partial^2}{\partial x^2}\psi - \frac{\partial^2}{\partial y^2}\psi - \frac{\partial^2}{\partial z^2}\psi + m^2\psi = 0$$

```
omega = sqrt(kx^2 + ky^2 + kz^2 + m^2)
psi = exp(i*kx*x + i*ky*y + i*kz*z - i*omega*t)
d(psi,t,t) - d(psi,x,x) - d(psi,y,y) - d(psi,z,z) + m^2*psi
```

4. Verify the following equation.

$$\vec{k}\cdot\vec{\sigma} = \begin{pmatrix} k_z & k_x - ik_y \\ k_x + ik_y & -k_z \end{pmatrix}$$

```
sigmax = ((0,1),(1,0))
sigmay = ((0,-i),(i,0))
sigmaz = ((1,0),(0,-1))
sigma = (sigmax,sigmay,sigmaz)
k = (kx,ky,kz)
dot(k,sigma)
```

The vector dot product is commutative but attempting to calculate `dot(sigma,k)` causes an error. This is because $\vec{\sigma}$ is a tensor, not a vector. To compute $\vec{\sigma}\cdot\vec{k}$ it is necessary to transpose indices.

```
sigmax = ((0,1),(1,0))
sigmay = ((0,-i),(i,0))
sigmaz = ((1,0),(0,-1))
sigma = (sigmax,sigmay,sigmaz)
sigma = transpose(sigma,1,2)
sigma = transpose(sigma,2,3)
k = (kx,ky,kz)
dot(sigma,k)
```

Another method is to contract an outer product.

```
sigmax = ((0,1),(1,0))
sigmay = ((0,-i),(i,0))
sigmaz = ((1,0),(0,-1))
sigma = (sigmax,sigmay,sigmaz)
k = (kx,ky,kz)
T = outer(k,sigma)
contract(T,1,2)
T = outer(sigma,k)
contract(T,1,4)
```

5. Show that

$$u_1\bar{u}_1 + u_2\bar{u}_2 = (E+m)(\not{p}+m)$$

The energy is

$$E = \sqrt{p_x^2 + p_y^2 + p_z^2 + m^2}$$

The spinors are

$$u_1 = \begin{pmatrix} E+m \\ 0 \\ p_z \\ p_x + ip_y \end{pmatrix} \qquad u_2 = \begin{pmatrix} 0 \\ E+m \\ p_x - ip_y \\ -p_z \end{pmatrix}$$

The adjoint of u is $\bar{u} = u^\dagger\gamma^0$. Note that $\bar{u}$ is a row vector hence $u\bar{u}$ is an outer product. This problem demonstrates a trick regarding normalization constants. The normalization constant $E+m$ is placed on the right-hand side of the equation instead of normalizing the spinors with $1/\sqrt{E+m}$. This is because the software is better at simplifying expressions with multipliers instead of divisors.

```
E = sqrt(px^2 + py^2 + pz^2 + m^2)
u1 = (E+m, 0, pz, px+i*py)
u2 = (0, E+m, px-i*py, -pz)
p = (E, px, py, pz)
I = ((1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1))
gmunu = ((1,0,0,0),(0,-1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma0 = ((1,0,0,0),(0,1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma1 = ((0,0,0,1),(0,0,1,0),(0,-1,0,0),(-1,0,0,0))
gamma2 = ((0,0,0,-i),(0,0,i,0),(0,i,0,0),(-i,0,0,0))
gamma3 = ((0,0,1,0),(0,0,0,-1),(-1,0,0,0),(0,1,0,0))
gamma = (gamma0,gamma1,gamma2,gamma3)
pslash = dot(gmunu,p,gamma)
bar(u) = dot(conj(u),gamma0)
A = outer(u1,bar(u1)) + outer(u2,bar(u2))
B = (E + m) * (pslash + m*I)
A-B
```

6. Prove the following Gordon decomposition by direct calculation (p_1 and p_2 have the same rest mass m).

$$\bar{u}(p_2)\gamma^\mu u(p_1) = \bar{u}(p_2)\left[\frac{(p_2+p_1)^\mu}{2m} + i\sigma^{\mu\nu}\frac{(p_2-p_1)_\nu}{2m}\right]u(p_1)$$

The following vectors and spinors are used. Spinors u_{11} and u_{21} are spin-up, u_{12} and u_{22} are spin-down.

$$p_1 = \begin{pmatrix} E_1 \\ p_{1x} \\ p_{1y} \\ p_{1z} \end{pmatrix} \quad u_{11} = \begin{pmatrix} E_1+m \\ 0 \\ p_{1z} \\ p_{1x}+ip_{1y} \end{pmatrix} \quad u_{12} = \begin{pmatrix} 0 \\ E_1+m \\ p_{1x}-ip_{1y} \\ -p_{1z} \end{pmatrix} \quad E_1 = \sqrt{p_{1x}^2+p_{1y}^2+p_{1z}^2+m^2}$$

$$p_2 = \begin{pmatrix} E_2 \\ p_{2x} \\ p_{2y} \\ p_{2z} \end{pmatrix} \quad u_{21} = \begin{pmatrix} E_2+m \\ 0 \\ p_{2z} \\ p_{2x}+ip_{2y} \end{pmatrix} \quad u_{22} = \begin{pmatrix} 0 \\ E_2+m \\ p_{2x}-ip_{2y} \\ -p_{2z} \end{pmatrix} \quad E_2 = \sqrt{p_{2x}^2+p_{2y}^2+p_{2z}^2+m^2}$$

Tensor $\sigma^{\mu\nu}$ is defined as

$$\sigma^{\mu\nu} = \frac{i}{2}\left[\gamma^\mu, \gamma^\nu\right]$$

The inner product $\sigma^{\mu\nu}(p_2-p_1)_\nu$ is computed this way.

$$\sigma^{\mu\nu}(p_2-p_1)_\nu = \texttt{dot(gmunu,p2-p1,sigmamunu[mu])}$$

Multiplication by the metric tensor lowers the index of p_2-p_1. Tensor $\sigma^{\mu\nu}$ is placed on the right-hand side of the dot product so that index ν is correctly oriented. The complete index for $\sigma^{\mu\nu}$ is $\sigma^{\mu\nu\alpha\beta}$. The μ is removed by indexing leaving $\sigma^{\nu\alpha\beta}$. The dot product sums over ν in $(p_2-p_1)_\nu\sigma^{\nu\alpha\beta}$.

```
-- www.eigenmath.org/gordon.txt
E1 = sqrt(p1x^2 + p1y^2 + p1z^2 + m^2)
E2 = sqrt(p2x^2 + p2y^2 + p2z^2 + m^2)
p1 = (E1,p1x,p1y,p1z)
p2 = (E2,p2x,p2y,p2z)
u11 = (E1+m,0,p1z,p1x+i*p1y)
u12 = (0,E1+m,p1x-i*p1y,-p1z)
u21 = (E2+m,0,p2z,p2x+i*p2y)
u22 = (0,E2+m,p2x-i*p2y,-p2z)
u1 = (u11,u12)
u2 = (u21,u22)
I = ((1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1))
gmunu = ((1,0,0,0),(0,-1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma0 = ((1,0,0,0),(0,1,0,0),(0,0,-1,0),(0,0,0,-1))
gamma1 = ((0,0,0,1),(0,0,1,0),(0,-1,0,0),(-1,0,0,0))
gamma2 = ((0,0,0,-i),(0,0,i,0),(0,i,0,0),(-i,0,0,0))
gamma3 = ((0,0,1,0),(0,0,0,-1),(-1,0,0,0),(0,1,0,0))
gamma = (gamma0,gamma1,gamma2,gamma3)
sigmamunu = i/2 * sum(mu,1,4,sum(nu,1,4,
  outer(I[mu],I[nu],dot(gamma[mu],gamma[nu])-dot(gamma[nu],gamma[mu]))
))
```

```
for(s1,1,2,for(s2,1,2,for(mu,1,4,
  T = 1/(2*m) * ((p2+p1)[mu]*I + i*dot(gmunu,p2-p1,sigmamunu[mu])),
  A = dot(conj(u2[s2]),gamma0,T,u1[s1]),
  B = dot(conj(u2[s2]),gamma0,gamma[mu],u1[s1]),
  print(A-B)
)))
```