Modeling the Free Electron

© 2004 Blacklight Power Inc.

```
<< Graphics`ParametricPlot3D`
<< Miscellaneous`PhysicalConstants`
<< Graphics`Graphics3D`
<< Calculus`VectorAnalysis`
<< Graphics`Colors`
```

This is a computational appendix to the Chapter 3: Electron in Free Space section of R. Mills, *The Grand Unified Theory of Classical Quantum Mechanics*, January 2004 Edition, posted at: http://www.blacklightpower.com/bookdownload.shtml. Here the free electron is modeled to scale using the physical constants and a variable relative velocity.

The free electron is a flat disk of charge whose radius (ρ_0) is proportional to Planck's constant bar, and inversely proportional to the relative velocity and electron mass, as,

$$\rho_0 = \frac{\hbar}{m_e v_z}$$

The charge distribution (σ_e) on the surface of the disk in cylindrical coordinates is a function of the radius (ρ), with the highest charge in the center and a slow taper until it drops off to zero when $\rho = \rho_0$.When the velocity goes to the speed of light ($v_z \to c$) the electron behaves as a point particle. When it goes to zero (as $v_z \to 0$), it behaves as a uniformly charged plane wave. The equation used to model the three-dimensional charge distribution is given below, and stored in "ChargeDens."

$$\sigma_e(\rho, \phi, z) = \frac{e}{\frac{2}{3}\pi \rho_0^3} \sqrt{\rho_0^2 - \rho^2}\, \delta(z)$$

The physical constants are provided byMathematica,

```
PlanckConstantReduced
```

1.05457×10^{-34} Joule Second

```
ElectronMass
```

9.10938×10^{-31} Kilogram

```
ElectronCharge
```

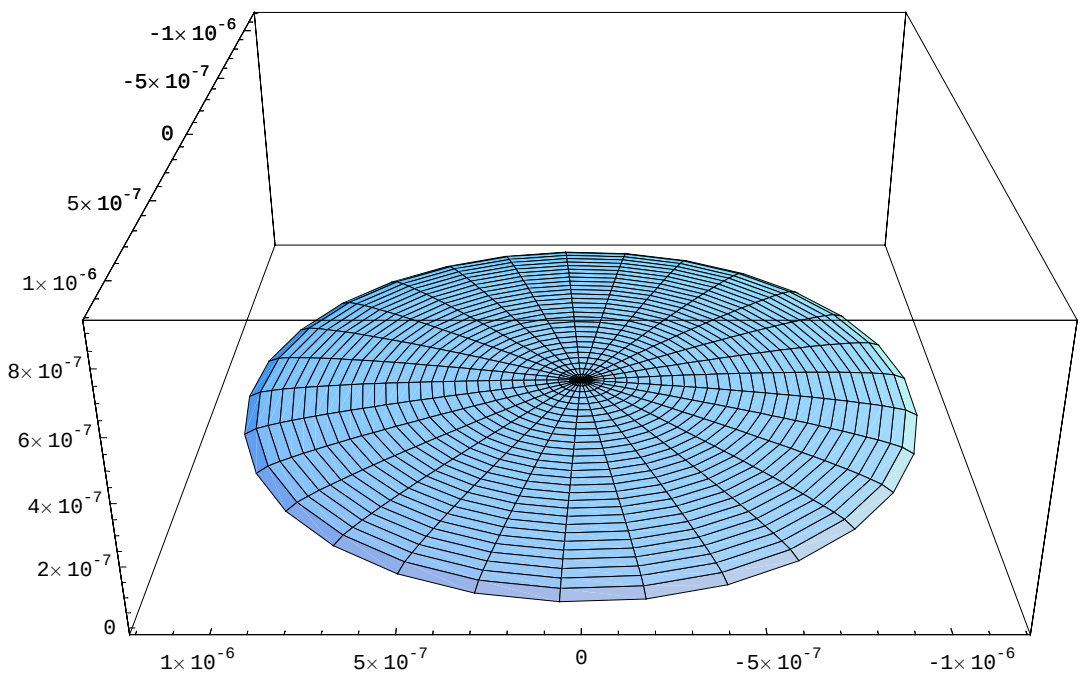
1.60218×10^{-19} Coulomb

We animate the change in the charge distribution for the range of velocities 100 to 400 m/s:

```
Array[ChargeDist, {60}];
```

```
Do[

 Vel := 100 + 5 * i;
 ρnot = (1.0545715964207855`*^-34 / (9.10938188`*^-31 * Vel));
 ChargeDens = FullSimplify[1.602176462`*^-19 / ((2/3) π (ρnot)^3) Sqrt[(ρnot)^2 - (ρ)^2]];
 ChargeDist[i] = CylindricalPlot3D[ChargeDens, {ρ, 0, ρnot}, {θ, 0, 2 π},
   ViewPoint → {0, 2, 1}, PlotRange → {{-1.208604160266866`*^-6, 1.215390508631058`*^-6},
     {-1.2137774436356162`*^-6, 1.2137774436356162`*^-6},
     {-2.2831655476212466`*^-8, 9.36097874524711`*^-7}}, ImageSize → 72 * 7],
{i,
 1,
 60}]
```



Click **GIF** or **AVI** to see animation.

Next we will animate the free electron charge distribution using ShadowPlot, a different plotting function, the benefit being that it "shadows" the distribution into a plane. It requires cartesian coordinates,

```
CoordinatesFromCartesian[{x , y, z}, Cylindrical]
```

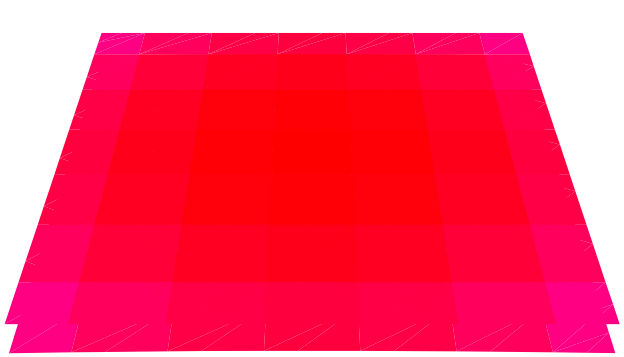
$$\left\{\sqrt{x^2 + y^2}\text{, ArcTan[x, y], z}\right\}$$

Note that x and y must be less than ρ_o, hence an infinitesimal constant is added to it in the CartChargeDens function. Note also that I am normalizing the scale of the graph to the 250 m/s electron radius.

```
ρnotnorm = (1.0545715964207855`*^-34 / (9.10938188`*^-31 * (250)))
```

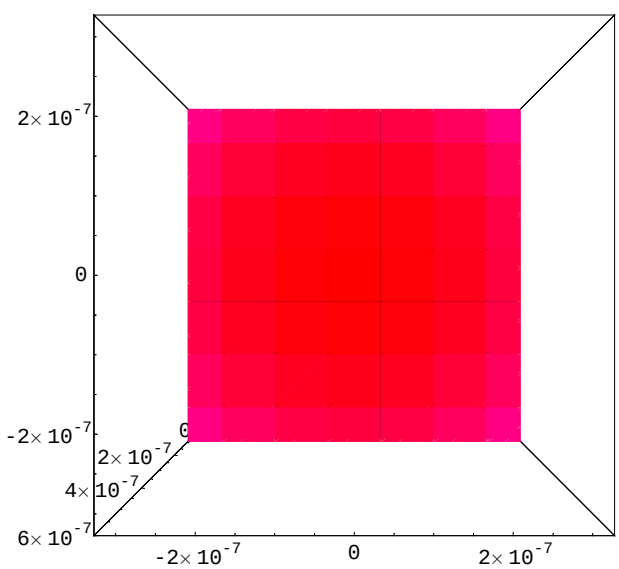
4.63071×10^{-7}

```
Do[
  Vel := 100 + 5 * i;
  ρnot = (1.0545715964207855`*^-34 / (9.10938188`*^-31 * Vel));
  CartChargeDens = FullSimplify[1.602176462`*^-19 / ((2/3) π (ρnot)^3) Sqrt[(ρnot + 1 * 10^-18)^2 - (x^2 + y^2)]];
  CartChargeDist = ShadowPlot3D[CartChargeDens, {x, -ρnot/Sqrt[2], ρnot/Sqrt[2]},
    {y, -ρnot/Sqrt[2], ρnot/Sqrt[2]}, ShadowPosition → 1, ViewPoint → {0, 2, 1}, ShadowMesh → False,
    SurfaceMesh → False, Axes → False, Boxed → False, ImageSize → 72 * 5,
    PlotRange → {{-ρnotnorm/Sqrt[2], ρnotnorm/Sqrt[2]}, {-ρnotnorm/Sqrt[2], ρnotnorm/Sqrt[2]}, {0, 6 * 10^-7}}],
  {i,
   1,
   32}];
```



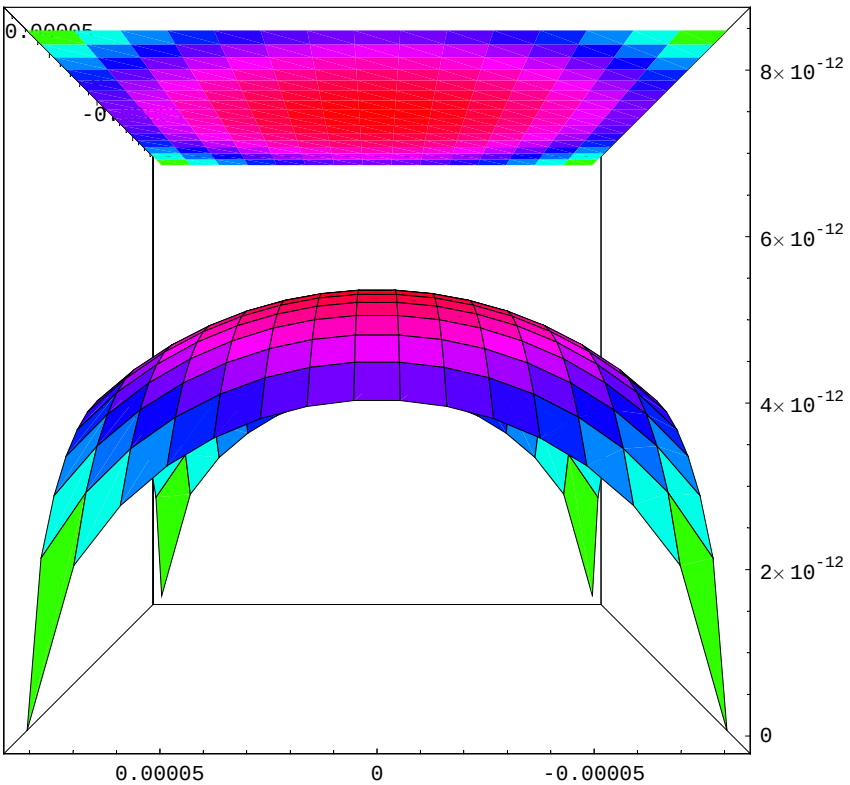
(Animation available in *Mathematica* only.) Viewed from the direction of propagation of the electron, see below

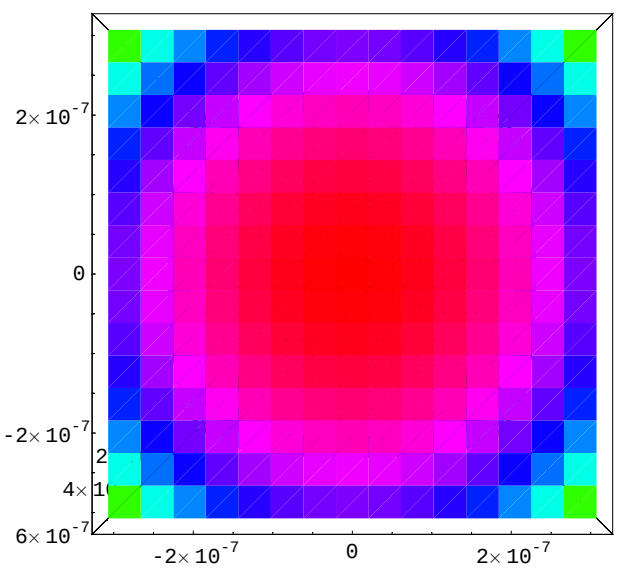
```
Do[
  Vel := 100 + 5 * i;
  ρnot = (1.0545715964207855`*^-34 / (9.10938188`*^-31 * Vel));
  CartChargeDens = FullSimplify[1.602176462`*^-19 / ((2/3) π (ρnot)^3) Sqrt[(ρnot + 1 * 10^-18)^2 - (x^2 + y^2)]];
  CartChargeDist = ShadowPlot3D[CartChargeDens, {x, -ρnot/Sqrt[2], ρnot/Sqrt[2]}, {y, -ρnot/Sqrt[2], ρnot/Sqrt[2]},
    ShadowPosition → 1, ViewPoint → {0, 0, 2}, ShadowMesh → False, Axes → True,
    PlotRange → {{-ρnotnorm/Sqrt[2], ρnotnorm/Sqrt[2]}, {-ρnotnorm/Sqrt[2], ρnotnorm/Sqrt[2]}, {0, 6 * 10^-7}}],
  {i,
   1,
   32}];
```



Two static images from the above animations are:

```
Show[CartChargeDist, ViewPoint → {0, 2, 0}];
```


8× 10^-12
6× 10^-12
4× 10^-12
2× 10^-12
0
0.00005
0
-0.00005


2× 10^-7
0
-2× 10^-7
-2× 10^-7
0
2× 10^-7