Integration

November 11, 2015

In [1]:
```
from sympy import init_session
from sympy.physics.vector import *
import sympy.physics.units as u
from sympy.mpmath import asin
from sympy.physics.vector import init_vprinting
from sympy.assumptions.assume import global_assumptions
init_session()
```

```
IPython console for SymPy 0.7.6.1 (Python 2.7.10-64-bit) (ground types: python)

These commands were executed:
>>> from __future__ import division
>>> from sympy import *
>>> x, y, z, t = symbols('x y z t')
>>> k, m, n = symbols('k m n', integer=True)
>>> f, g, h = symbols('f g h', cls=Function)
>>> init_printing()

Documentation can be found at http://www.sympy.org
```

In [2]:
```
c = 2.99 * u.m / u.s
eo = 8.85418782e-12 * u.m**-3 * u.kg**-1 * u.s**4 * u.A**2
me = 9.10e-31 * u.kg
```

In [3]:
```
q, t = symbols('q t',real=True)
d, beta = symbols('d \\beta',real=True,positive=True)

global_assumptions.add(Q.positive(1-beta**2))

N = ReferenceFrame('N')
z = beta * t
sin_theta_sqr = d**2 / (d**2 + z**2)
rp = d * N.y + z * N.z
np = rp.normalize()
```

Potential given by https://en.wikipedia.org/wiki/Li%C3%A9nard%E2%80%93Wiechert_potential

In [4]:
```
E = (q*(1-beta**2)/(rp.dot(rp) * (1 - beta**2 * sin_theta_sqr )**(3/2))) * np
E
```

Out[4]:

$$\frac{dq\left(-\beta^2+1\right)}{\left(\beta^2t^2+d^2\right)^{\frac{3}{2}}\left(-\frac{\beta^2d^2}{\beta^2t^2+d^2}+1\right)^{1.5}}\hat{\mathbf{n}}_{\mathbf{y}} + \frac{\beta qt\left(-\beta^2+1\right)}{\left(\beta^2t^2+d^2\right)^{\frac{3}{2}}\left(-\frac{\beta^2d^2}{\beta^2t^2+d^2}+1\right)^{1.5}}\hat{\mathbf{n}}_{\mathbf{z}}$$

We only care about y kick, let's consider the particle stationary

In [5]:
```
Ey = E.dot(N.y)
Ey
```

Out[5]:

$$\frac{dq\left(-\beta^2+1\right)}{\left(\beta^2t^2+d^2\right)^{\frac{3}{2}}\left(-\frac{\beta^2d^2}{\beta^2t^2+d^2}+1\right)^{1.5}}$$

Which I assert is equal to

In [6]:
```
simplification = q * (1 - beta**2) * d * 1/(d**2 + (beta**2) * (t**2 - d**2))**(3/2)
integrand = q * simplification
integrand
```

Out[6]:

$$\frac{dq^2\left(-\beta^2+1\right)}{\left(\beta^2(-d^2+t^2)+d^2\right)^{1.5}}$$

0.1 Unitless

In [7]:
```
result = integrate(integrand,(t,-oo,oo))
simplify(result)
```

Out[7]:

$$\begin{cases} \frac{1.12837916709551\sqrt{\pi}q^2(\beta-1)(\beta+1)\left(\beta^2-1\right)}{\beta d^{1.0}\,\mathrm{polar\_lift}^{2.0}\,(-\beta^2+1)} & \text{for } \left|\mathrm{periodic}_{\mathrm{argument}}\left(\frac{1}{\mathrm{polar\_lift}\,(-\beta^2+1)},\infty\right)\right| < \pi \\ \int_{-\infty}^{\infty} -\frac{dq^2\left(\beta^2-1\right)}{\left(-\beta^2(d^2-t^2)+d^2\right)^{1.5}}\,dt & \text{otherwise} \end{cases}$$

Substituting $\beta = 1$ should be correct results, as β approaches zero then my assumptions break down

In [11]:
```
result.subs(beta,0.9).evalf()
```

Out[11]:

$$\frac{2.22222222222222q^2}{d^{1.0}}$$

0.2 Adding units

Now add appropriate units, involving c and $4\pi\epsilon_0$

In [12]:
```
with_units = integrand.subs(q,q*u.coulomb).subs(d,d*u.m).subs(t,t*c*u.s)/(4*pi*eo)
simplify(with_units)
```

Out[12]:

$$-\frac{28235226661.3653dq^2kgm^{1.0}\left(\beta^2-1\right)}{\pi s^2(-\beta^2(d^2-8.9401t^2)+d^2)^{1.5}}$$

In [13]:
```
k = simplify(integrate(with_units,(t,-oo,oo))) * u.s # Integration doesn't seem to work with u
k
```

Out[13]:

$$s\begin{cases} \frac{10655532288.5968q^2kgm^{1.0}(\beta-1)(\beta+1)\left(\beta^2-1\right)}{\sqrt{\pi}\beta d^{1.0}s^2\,\mathrm{polar\_lift}^{2.0}\,(-\beta^2+1)} & \text{for } \left|\mathrm{periodic}_{\mathrm{argument}}\left(\frac{1}{\mathrm{polar\_lift}\,(-\beta^2+1)},\infty\right)\right| < \pi \\ \int_{-\infty}^{\infty} -\frac{28235226661.3653dq^2kgm^{1.0}\left(\beta^2-1\right)}{\pi s^2(-\beta^2(d^2-8.9401t^2)+d^2)^{1.5}}\,dt & \text{otherwise} \end{cases}$$

In [14]:
```
def enter_vals(b=0.999,charge=1.6e-19):
    return simplify(k.subs(beta,b)).subs(q,charge)
enter_vals()
```

Out[14]:

$$\frac{2.73054681269351 \cdot 10^{-28} kgm^{1.0} e^{0}}{\sqrt{\pi} d^{1.0} s}$$

In [15]:
```
kick = enter_vals().subs(d,1)
kick.evalf()
```

Out[15]:

$$\frac{kgm^{1.0}}{s} 1.54054606911121 \cdot 10^{-28}$$

In [18]:
```
dv = (kick / me)
dv.evalf()
```

Out[18]:

$$\frac{169.290776825407 m^{1.0}}{s}$$

So apparently $169.3 m.s^{-1}$ at a distance of $1m$

In [19]:
```
n = 1e12
(enter_vals().subs(d,0.001) *n).evalf()
```

Out[19]:

$$\frac{kgm^{1.0}}{s} 1.54054606911121 \cdot 10^{-13}$$

In [ ]:
```
ev = 1.6e-19
(enter_vals().subs(d,0.001) *n).evalf()
```