Setup

In [35]:
```
from __future__ import division
```

In [36]:
```
import numpy as np
```

In [37]:
```
import matplotlib.pyplot as plt
```

In [38]:
```
import quantities as pq
```

Initial Conditions

In [39]:
```
d = 0.0641 * pq.inch # wire diameter
```

In [40]:
```
r1 = 1.5*pq.inch / 2 # coil inner radius
```

In [41]:
```
n = 600 # number of turns per coil
```

In [42]:
```
B = 0.3 * pq.T # desired magnetic field
```

Geometric Calculations

Circle packing density is roughly 0.9, so the area occupied by N turns with diameter d each is:

In [43]:
```
A = n * 1/0.9 * np.pi * (d/2)**2
```

In [44]:
```
print A
```

```
2.15136788517 in**2
```

This area corresponds to a circular bundle with radius

In [45]:
```
r2 = np.sqrt(A / np.pi)
```

In [46]:
```
print r2
```

```
0.82752744164 in
```

Therefore, the effective radius of the Helmholtz coil is

In [47]:
```
r3 = r1 + r2
```

In [48]:
```
print r3
```

```
1.57752744164 in
```

Magnetics Calculations

To achieve our desired magnetic field B, we need a current I1 running through our coil:

In [49]:
```
mu_0 = pq.Quantity(4 * np.pi * 10**-7,'V*s/(A*m)')
```

In [50]:
```
I1 = B * r3 / (n * mu_0)*(4/5)**(-3/2)
```

In [51]:
```
print I1.simplified
```

```
22.2810574859 A
```

Power calculations

The resistivity of copper:

In [52]:
```
rho_c = pq.Quantity(1.7*10**-8, 'ohm*m')
```

And the total length of wire in each coil is:

In [53]:
```
l = n * 2 * np.pi * r3
```

In [54]:
```
print l.simplified
```

```
151.057313983 m
```

So the resistance of the wire is:

In [55]:
```
R = rho_c * l / (np.pi * (d/2)**2)
```

In [56]:
```
R.units = 'ohm'; print R
```

```
1.2334379479 ohm
```

And the voltage and power for each coil are:

In [57]:
```
V = I1 * R
```

In [58]:
```
print V.rescale('volt')
```

```
27.4823018225 V
```

In [59]:
```
P = I1 * V
```

In [60]:
```
print P.rescale('watt')
```

```
612.334746752 W
```

In [61]:
```
def helmholtz_power(d, n):
    r1 = 1.5*pq.inch / 2
    B = 0.3 * pq.T
    A = n * 1/0.9 * np.pi * (d/2)**2
    r2 = np.sqrt(A/np.pi)
    r3 = r1 + r2
    mu_0 = pq.Quantity(4 * np.pi * 10**-7, 'V*s/(A*m)')
    I1 = B * r3 / (n * mu_0) * (4/5)**(-3/2)
    rho_c = pq.Quantity(1.7*10**-8, 'ohm * m')
    l = n * 2 * np.pi * r3
    R = rho_c * l / (np.pi * (d/2)**2)
    V = I1 * R
    P = I1 * V
    P.units = 'watt'
    return P
```

In [62]:
```
print helmholtz_power(0.0641*pq.inch, 600)
```

```
612.334746752 W
```

In [63]:
```
xs = np.linspace(0.01*pq.inch, 0.1*pq.inch)
```

In [64]:
```
ys = np.linspace(100, 2000)
```

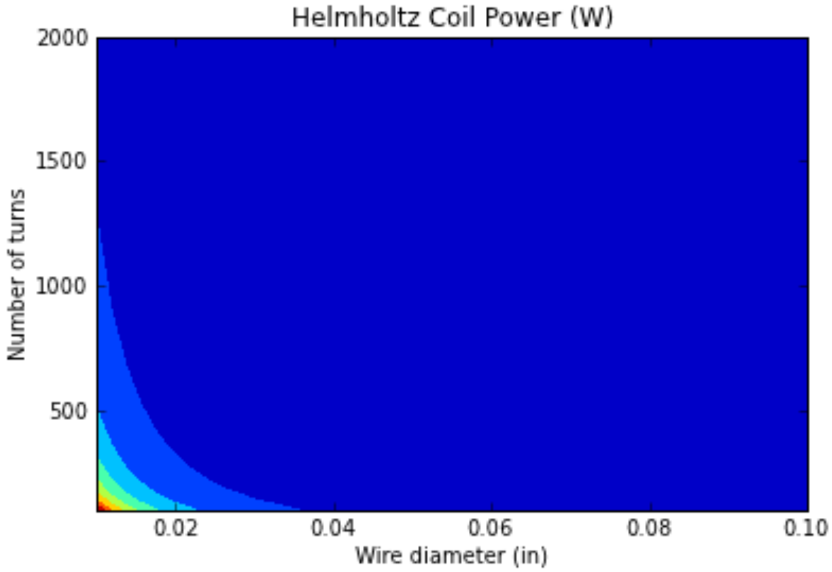
In [65]:
```
[X, Y] = np.meshgrid(xs, ys)
```

In [66]:
```
Z = np.zeros(np.shape(X))
```

In [67]:
```
for i, x in enumerate(xs):
    for j, y in enumerate(ys):
        Z[j, i] = helmholtz_power(x, y)
```

In [74]:
```
contourf(X, Y, Z)
xlabel('Wire diameter (in)')
ylabel('Number of turns')
title('Helmholtz Coil Power (W)')
```

Out[74]: <matplotlib.text.Text at 0x10828f050>



In [ ]: