Setup

In [2]:
```
from __future__ import division
```

In [3]:
```
import numpy as np
```

In [4]:
```
import matplotlib.pyplot as plt
```

In [5]:
```
import quantities as pq
```

Initial Conditions

In [6]:
```
d = 0.0641 * pq.inch # wire diameter
```

In [7]:
```
r1 = 1.5*pq.inch / 2 # coil inner radius
```

In [8]:
```
n = 400 # number of turns per coil
```

In [9]:
```
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 [10]:
```
A = n * 1/0.9 * np.pi * (d/2)**2
```

In [11]:
```
print A
```

```
1.43424525678 in**2
```

This area corresponds to a circular bundle with radius

In [12]:
```
r2 = np.sqrt(A / np.pi)
```

In [13]:
```
print r2
```

```
0.675673326723 in
```

Therefore, the effective radius of the Helmholtz coil is

In [14]:
```
r3 = r1 + r2
```

In [15]:
```
print r3
```

```
1.42567332672 in
```

Magnetics Calculations

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

In [16]:
```
mu_0 = pq.Quantity(4 * np.pi * 10**-7,'V*s/(A*m)')
```

In [17]:
```
I1 = B * r3 / (n * mu_0)*(4/5)**(-3/2)
```

In [18]:
```
print I1.simplified
```

```
30.2043963012 A
```

Power calculations

The resistivity of copper:

In [19]:
```
rho_c = pq.Quantity(1.7*10**-8, 'ohm*m')
```

And the total length of wire in each coil is:

In [20]:
```
l = n * 2 * np.pi * r3
```

In [21]:
```
print l.simplified
```

```
91.0109401449 m
```

So the resistance of the wire is:

In [22]:
```
R = rho_c * l / (np.pi * (d/2)**2)
```

In [23]:
```
R.units = 'ohm'; print R
```

```
0.743137450872 ohm
```

And the voltage and power for each coil are:

In [24]:
```
V = I1 * R
```

In [25]:
```
print V.rescale('volt')
```

```
22.4460180724 V
```

In [26]:
```
P = I1 * V
```

In [27]:
```
print P.rescale('watt')
```

```
677.968425241 W
```

Sanity checks

In [28]:
```
assert 2 * r2 < r3, 'The coils will intersect each other'
```

In [29]:
```
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 [30]:
```
print helmholtz_power(0.0641*pq.inch, 600)
```

```
612.334746752 W
```

In [31]:
```
xs = np.linspace(0.01*pq.inch, 0.1*pq.inch)
```

In [32]:
```
ys = np.linspace(100, 2000)
```

In [33]:
```
[X, Y] = np.meshgrid(xs, ys)
```

In [34]:
```
Z = np.zeros(np.shape(X))
```

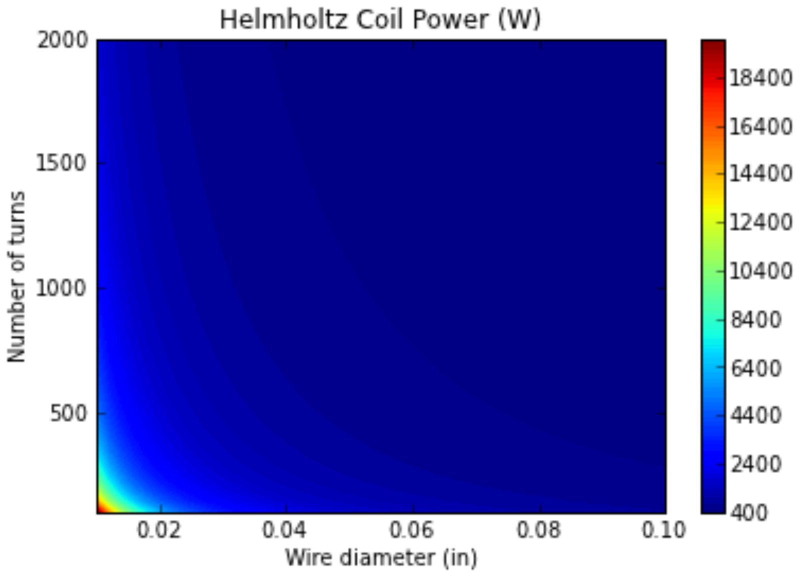
In [35]:
```
for i, x in enumerate(xs):
    for j, y in enumerate(ys):
```

```
        Z[j, i] = helmholtz_power(x, y)
```

In [37]:

```
contourf(X, Y, Z, 100)
xlabel('Wire diameter (in)')
ylabel('Number of turns')
title('Helmholtz Coil Power (W)')
colorbar()
```

Out[37]: <matplotlib.colorbar.Colorbar instance at 0x1081cb638>



In [36]: