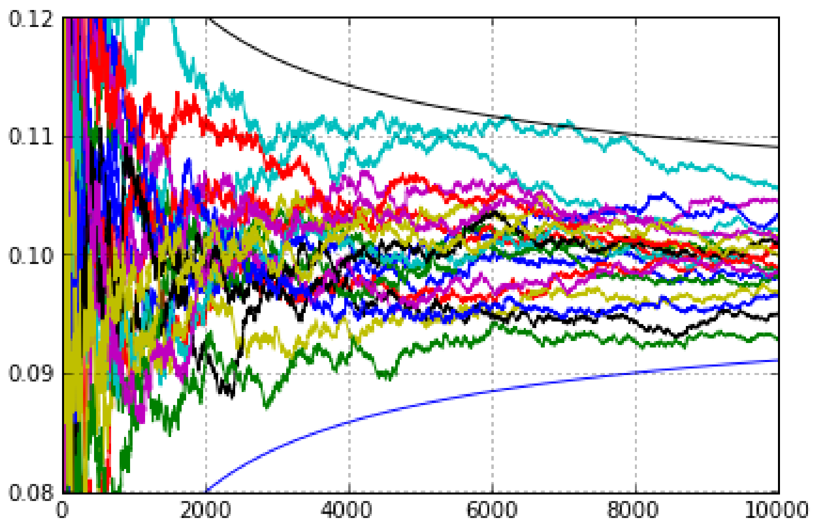
In [83]:

```
from random import random
from pylab import *

n=1000 # length of sequence
m=1000   # number of trials
p=0.02 # the true probability
q=0.025   # the hypothesized (or model) probability

count=0
for j in range(m):
    outcomes=[(1 if random()<p else 0) for i in range(n)]
    if sum(outcomes)>=int(n*q): count += 1
print (count+0.0)/m
```

```
0.156
```

In [84]:

```
n=10000 # length of sequence
m=20    # number of trials
p=0.1   # the true probability

count=0
for j in range(m):
    outcomes=[(1 if random()<p else 0) for i in range(n)]
    cs=cumsum(outcomes)
    run_aver=[cs[i]/(i+1.0) for i in range(len(cs))]
    plot(run_aver)

var=p*(1-p)
upper=[p+3*sqrt(var/(i+1.0)) for i in range(n)]
lower=[p-3*sqrt(var/(i+1.0)) for i in range(n)]
plot(upper)
plot(lower)
r=3*sqrt(var/(n/5))
ylim([p-r,p+r])
grid()
```

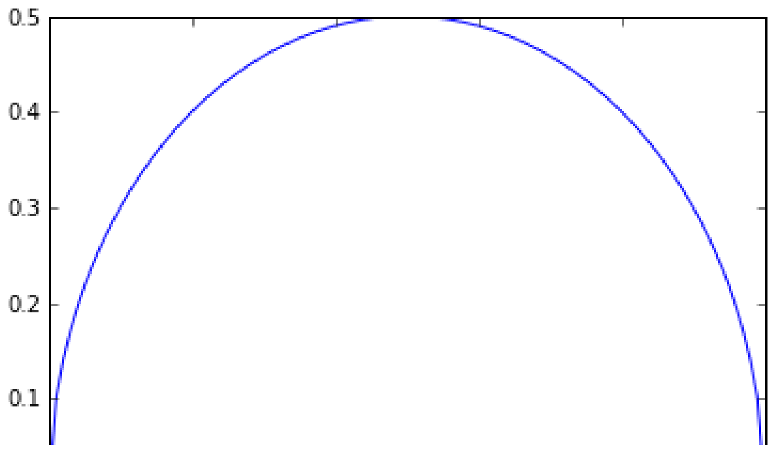


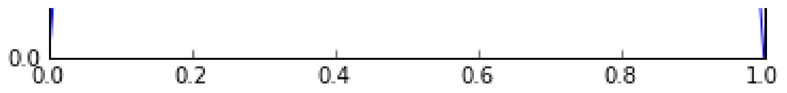
In [85]:

```
P=[(i+0.0)/100.0 for i in range(101)]
std=[sqrt(P[i]*(1-P[i])) for i in range(len(P))]
plot(P,std)
```

Out[85]:

```
[<matplotlib.lines.Line2D at 0x111f75510>]
```



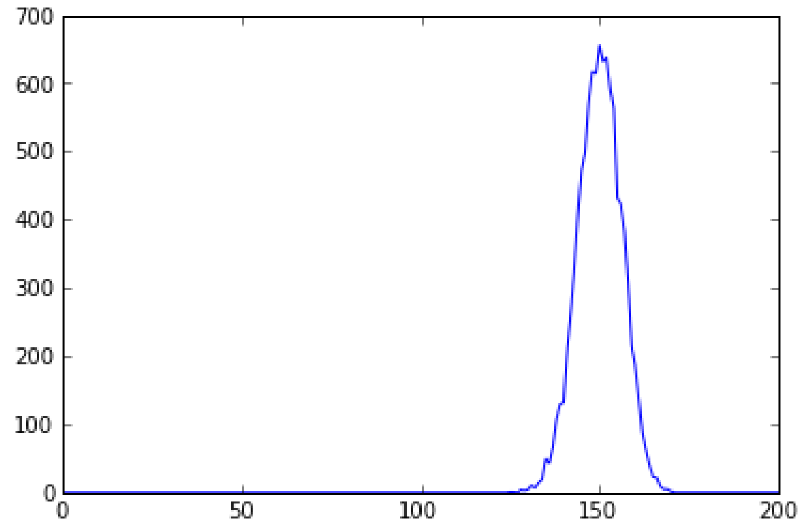


In [86]:

```
n=200
m=10000
p=0.75

counts=[0]*(n+1)
for j in range(m):
    outcomes=[(1 if random()<p else 0) for i in range(n)]
    S=sum(outcomes)
    counts[S] += 1

plot(counts)
```

Out[86]:

```
[<matplotlib.lines.Line2D at 0x10ffbfa90>]
```

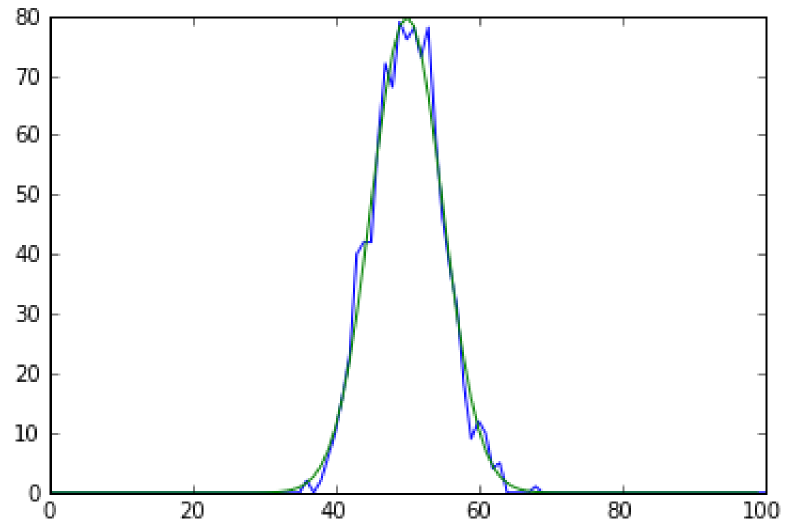


In [99]:

```
n=100
m=1000
p=0.5

sigma=sqrt(n*p*(1-p))
Z=sigma*sqrt(2*pi)

counts=[0]*(n+1)
for j in range(m):
    outcomes=[(1 if random()<p else 0) for i in range(n)]
    S=sum(outcomes)
    counts[S] += 1

def nDist(i):
    diff=i-p*n
    return (m/Z)*exp(-(diff/sigma)**2/2)

ND = [nDist(i) for i in range(n) ]
plot(counts)
plot(ND)
```

Out[99]:

```
[<matplotlib.lines.Line2D at 0x110b60710>]
```



In [87]:

In [77]:

In [63]:

In [63]:

In [ ]: