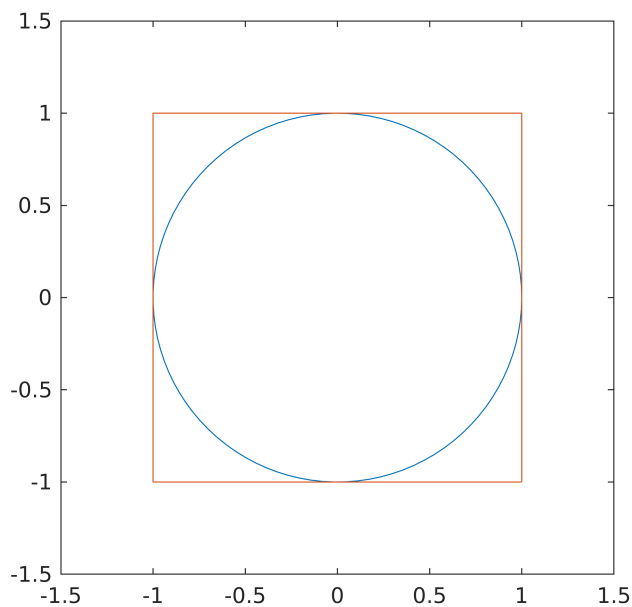


Exercise 6, approx of π , area of ellipse

Monte Carlo simulation
HA, 4.12.2018

(a) Draw square and circle (dartboard) inside.

```
close all;format short;format compact % squeeze off extra blank lines
t = linspace(0,2*pi);
x = cos(t);y = sin(t);
plot(x,y,[1 1 -1 -1 1 ], [-1 1 1 -1 -1]);
axis([-1.5 1.5 -1.5 1.5])
axis square;shg
```

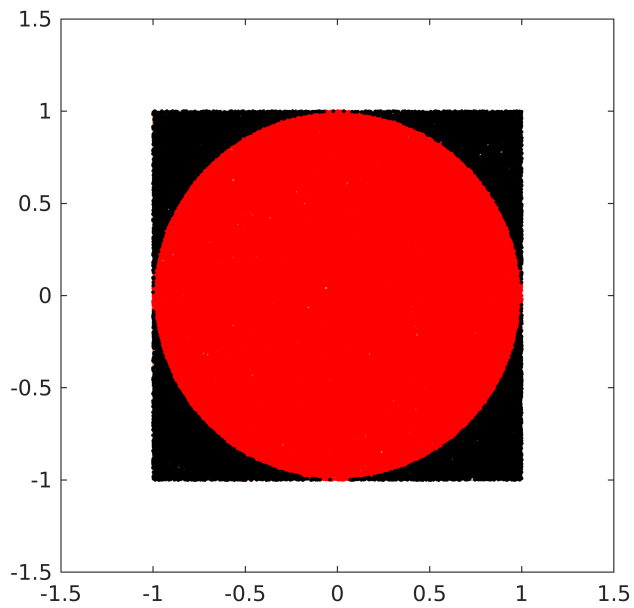


(b) Draw points (throw darts)

n = 10000 % Nr. of "darts", start with "small"

```
n = 100000; % Increase n
X = 2*rand(n,1)-1; % Random x-coords on (-1,1).
Y = 2*rand(n,1)-1; % Random y-coords on (-1,1).
```

```
in = (X.^2+Y.^2)<=1; % Logical indices for points inside circle
out=~in;
hold on
plot(X(in),Y(in),'r.',X(out),Y(out),'k.');
```



```
Nin=sum(in) % in is a logical vector sum(in):nr.of ones.
```

```
Nin = 78529
```

```
%
```

Display in 2 ways:

1)

```
disp('Display using disp: ')
```

```
Display using disp:
```

```
disp(' ')
```

```
disp([' 4*inside/n ' ' pi'])
```

```
4*inside/n pi
```

```
disp([4*Nin/n, pi])
```

```
3.1412 3.1416
```

```
disp(' ')
```

```
% 2)
```

```
disp('Display using fprintf:')
```

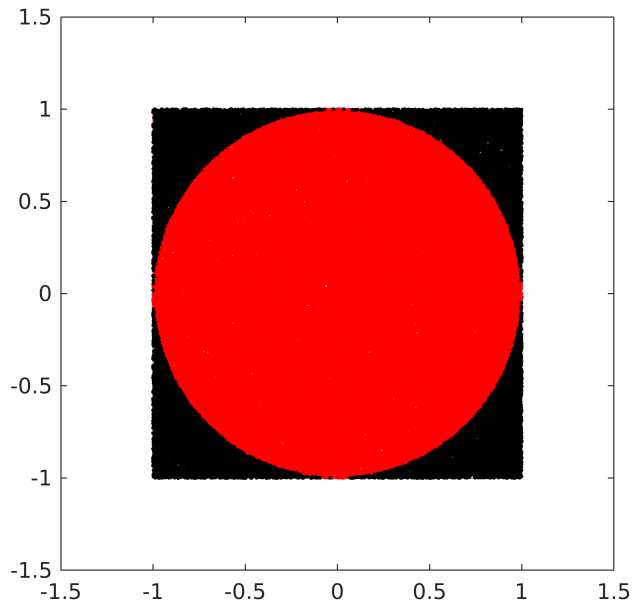
Display using fprintf:

```
disp(' ')
```

```
fprintf('4*hits/total = %1.6f pi = %1.6f\n',4*Nin/n,pi);
```

4*hits/total = 3.141160 pi = 3.141593

```
hold off
```



(c) Ellipse

```
help myAellipse
```

Use Monte Carlo simulation for approximation of area of ellipse.
Call: [Aappr,Atrue] = **myAellipse**(a,b,N,Nmax) % Default Nmax=10⁶
if N > Nmax, skip plotting of "darts".

Example:

```
[Aappr,Atrue] = myAellipse(4,2,106) % Plot darts  
[Aappr,Atrue] = myAellipse(4,2,106+1) % Don't plot darts.
```

```
type myAellipse
```

```
function [Aappr,Atrue] = myAellipse(a,b,N,Nmax)  
% Use Monte Carlo simulation for approximation of area of ellipse.  
% Call: [Aappr,Atrue] = myAellipse(a,b,N,Nmax) % Default Nmax=106
```

```

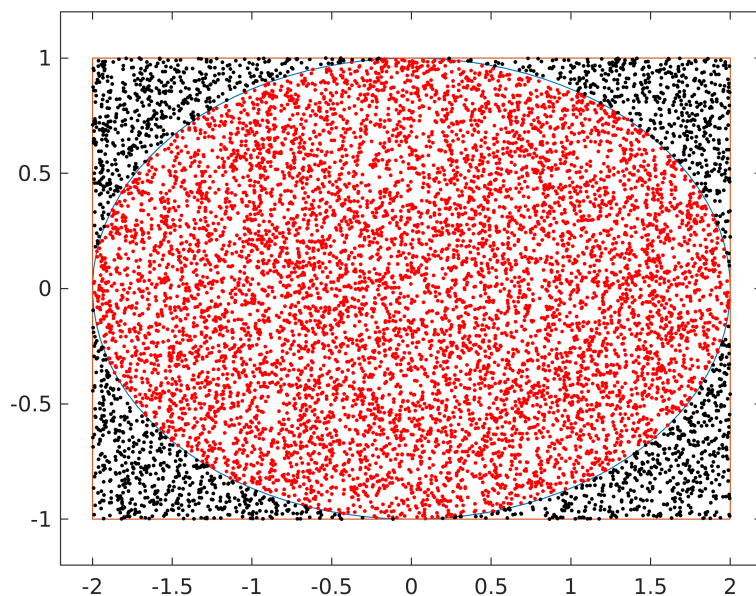
% if N > Nmax, skip plotting of "darts".
% Example:
% [Aappr,Atrue] = myAellipse(4,2,10^6) % Plot darts
% [Aappr,Atrue] = myAellipse(4,2,10^6+1) % Don't plot darts.
%
if nargin==3
    Nmax=10^6;
end
t = linspace(0,2*pi);
x = a*cos(t);y = b*sin(t);
plot(x,y,[a a -a -a a], [-b b b -b -b]);
axis([-a-.2 a+.2 -b-.2 b+.2])
%axis equal;shg
X = 2*a*rand(N,1)-a; % Random x-coords on (-a,a)
Y = 2*b*rand(N,1)-b; % Random y-coords on (-b,b)
in = ((X.^2./a^2+Y.^2./b^2)<=1);
out=~in;
% Plot "darts" if N <= Nmax
if N<=Nmax
    hold on
    plot(X(in),Y(in),'r',X(out),Y(out),'.k');
    hold off;shg;
else
    display(' ')
    display(['N = ' num2str(N) ', too large for plotting darts'])
end

Nin=sum(in); % in is a logical vector sum=nr. of ones.
% n = nnz(in); % Matlab has a built-in function
% The ratio Nin/N = Rect_area/ell_area, hence:
Aappr=(Nin/N)*4*a*b;
Atrue=pi*a*b;
end

```

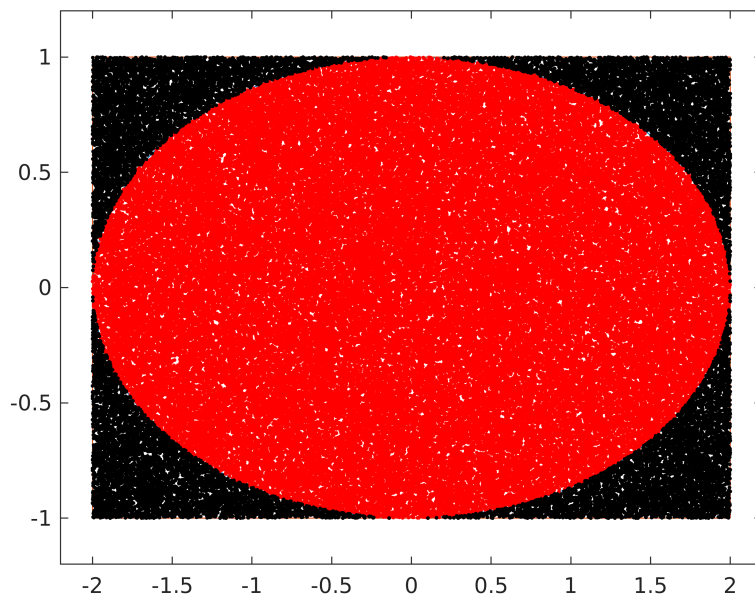
```
%
```

```
[Aappr,Atrue] = myAellipse(2,1,10000)
```



```
Aappr = 6.3408
Atrue = 6.2832
```

```
[Aappr,Atrue] = myAellipse(2,1,100000)
```



```
Aappr = 6.2937
Atrue = 6.2832
```

```
%
```

Notes:

- Some student solutions just used an approximation for π obtained in the circle case to compute πab . This was a little too much of a short cut. Anyway, we were quite generous in evaluation.
- Another problem was far too small value of nr. of points leading to unnecessarily poor approximations. One can get easily about 3 correct digits, but hardly many more.

Some advanced Matlab-functions

There was at least one student who referred to the functions below:

- **nnz** Matlab has a built-in function "number of nonzeros". In the basic course we prefer teaching the "basic core" Matlab thinking
- **polyarea** This is an advanced function to compute numerically the area inside a polygon. One student (cleverly) realized that Matlab's circle, ellipse etc. are in fact polygons. Anyway sticking to basics is again preferred. How many of you would fluently read and understand the code of polyarea as compared to the method of logical indexing.