# Question 3. function in function

### 3.1 Explain why t = fzero(@(y) (sin(y) - x) , [0 pi/2]) is equivalent to solve t = asin(x)?

The two expressions are actually the same when some work’s done to the first expression. In order to find the zero point, we get:

sin(y) - x = 0 => sin(y)=x => y=asin(x)

where y is replaced by t.

Apparently, the two expressions are the same. Moreover, x should be between 0.1 and 0.9 as described in the given function f(x), since x weights the function. fzero cannot find any zero point, if x is less than 0, because the curve goes toward negative far away from zero point (see figure 1). Whereas the zero point could not be found if x is greater than 1, as the function is heading to positive far from zero(see figure 2).



Figure 1 asin(x)



Figure 2 sin(x)

### 3.2 write a function Fun that could calculate f(x) for a given x, which asin(x) can be calculated by the technique from question 3.1.

The underneath is the function Fun, which is just a function inside another function. The technique from question 3.1 is applied, (sin(y) - x) and asin(x) are equivalent.

function [ fx ] = Fun( x )

% Functions navn er sat sum FUN med variabel: x and

% output parameter som fx. x er et tal mellem 0.1 and 0.9

roden = fzero( @ (y) (sin(y) - x), [0 pi/2]);

fx = roden\*x^2+sqrt(x)-1;

end

### 3.3 apply the function fzero to calculate the zero point for f(x) with fzero(@Fun,0.5), the root should be around 0.6.

nulpunkt = fzero(@Fun,0.5)

nulpunkt =0.5966

clearly seen from figure 3 that when y= 0, x is close to 0.6.



Figure 3 fzero(@Fun,0.5)

# Question 4

### 4.1

The purpose of using Minimization golden section search is to find the minimum for function f and to provide the total amount of interactions to find the minimum. Using the given initial value [0 2,], the expected error es=0.1 that refers to 10^-3.

x =0.9792

iter = 14

The matlab output indicates that the function f is minimum at x=0.9792, and it takes 14iterations to find this corresponding x value.

### 4.2

The question asks to find a proper interaction’s time such that the absolution error after k time’s interaction is less than the tolerance tol=10^12. Based on the given equation, we did some reduction to finally get a function of k. k is rounded to 58, which means it takes at least 58 interactions to manage the absolute error below the tolerance.

tol=10^(-12);

%phi er definiret i side 187

phi=(1+sqrt(5))/2;

xl=0;

xu=2;

k=(log(phi-1)+log(tol/(phi\*xl-phi\*xu-2\*xl+2\*xu)))/log(phi-1);

k = 57.8601 (58)

### 4.3

option43= optimset('TolX',1e-12,'display','iter');

[xmin, fval, exitflag, output] = fminbnd(f,xl,xu,option43);

output =

iterations: 13

funcCount: 14

algorithm: 'golden section search, parabolic interpolation'

message: 'Optimization terminated:

the current x satisfi...'

X= 0.9788

It is seen that fminbnd takes much less interactions to find the minimum than the golden section search method. The reason of this difference is that fminbnd uses both the robust but slow algorithm golden section search and the fast parabolic interpolation.