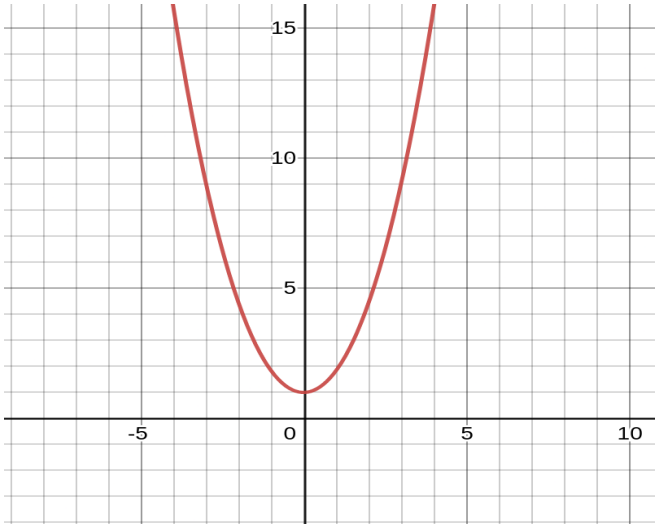
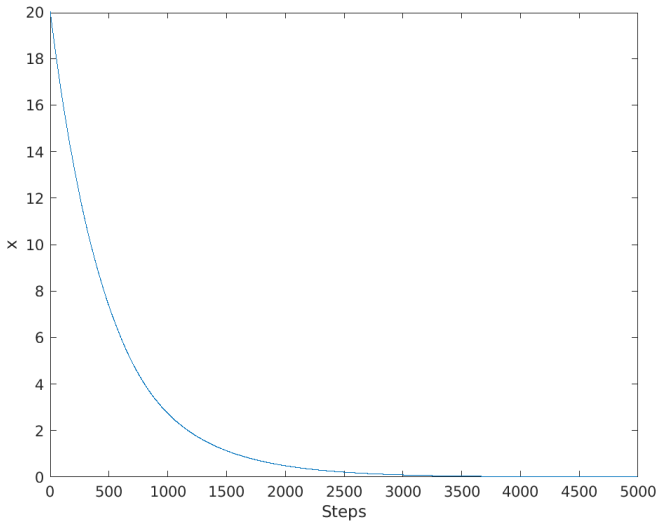
Function:

i) $f(x) = \sin(x)/x + x^2$; (not defined at x = 0)

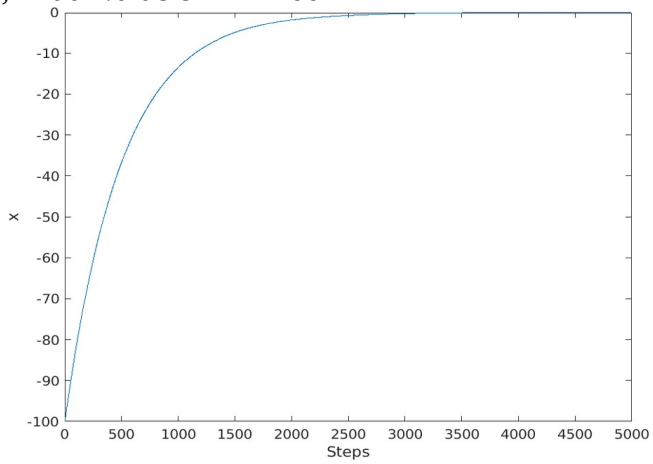


Parameters set:

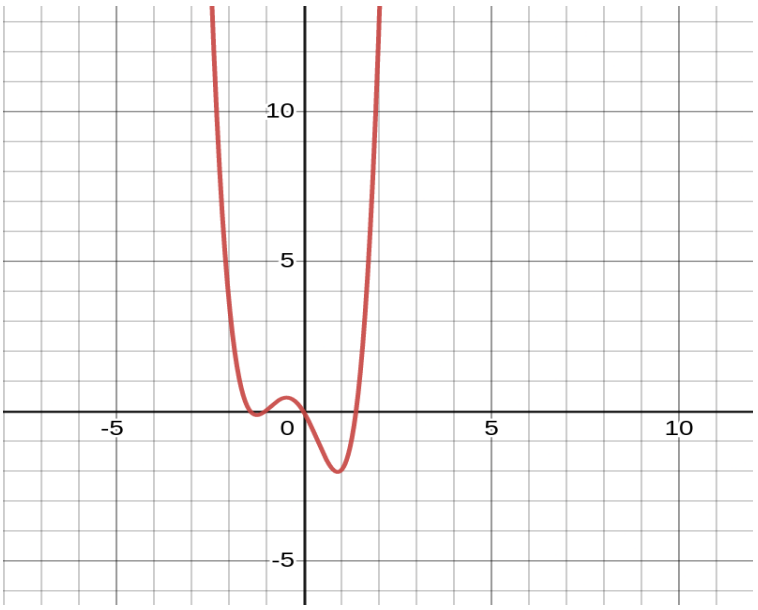
M = 20; a = 0.001; b = 0.1; initial value of x = 20



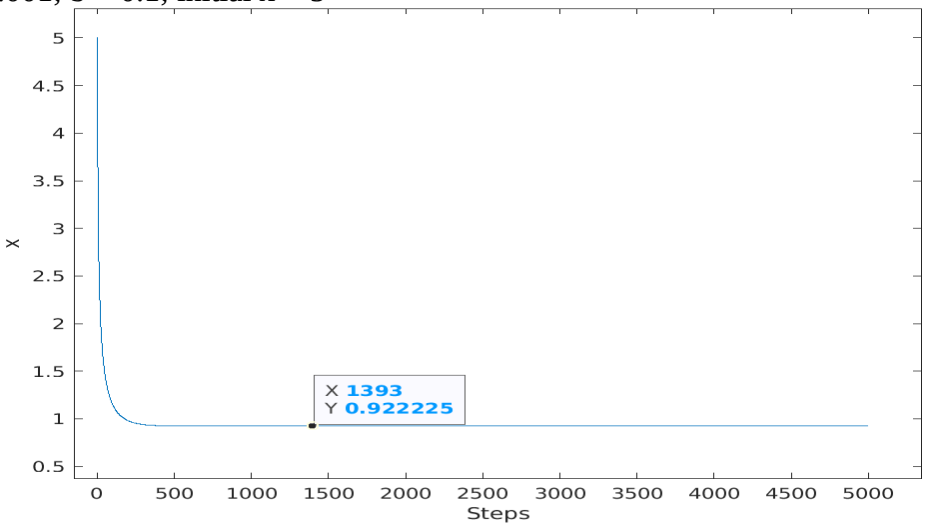
M = 5; a = 0.001;b = 0.1; initial value of x = -100



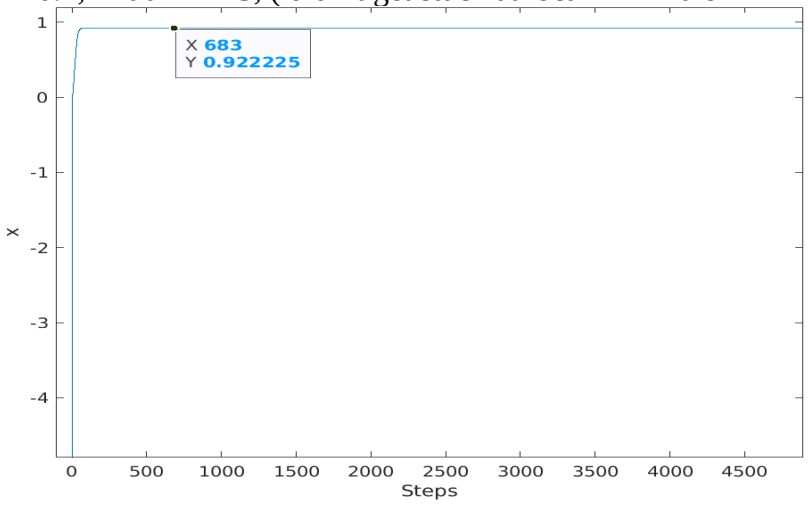
ii) $x^4 + x^3 - 2x^2 - 2x$



M = 20; a = 0.001; b = 0.1; initial x = 5

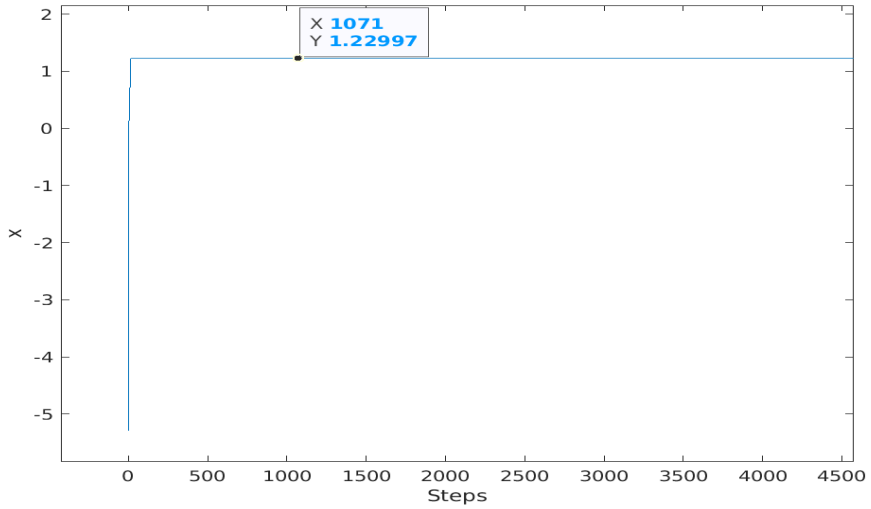


M = 20; a = 0.01;b = 0.1; initial x = -5; (didn't get stuck at local minima of x = -1.22)

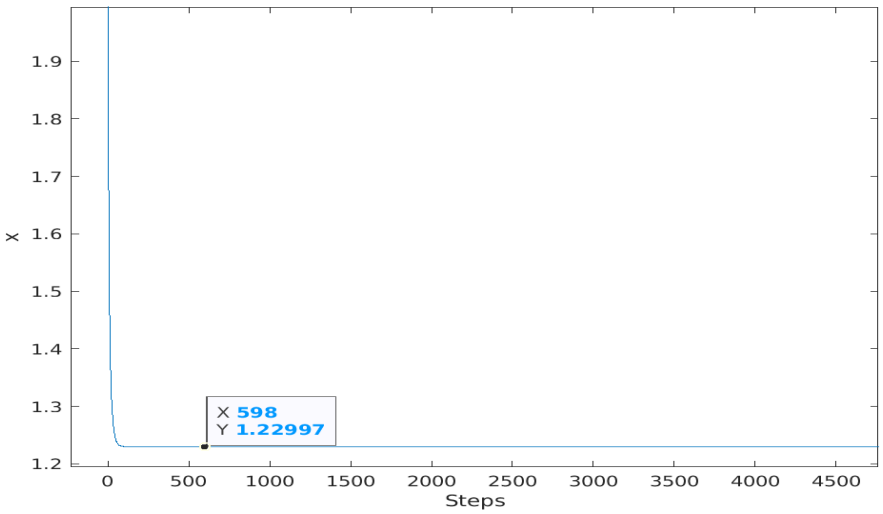


iii) $x^5 + 4x^4 - 3x^3 - 10x^2 - 3x$

M = 10; a = 0.01; b = 0.3; initial x = -5



M = 10; a = 0.001; b = 0.3; initial x = 3



Code:

```
clear all;
M = 10;
a = 0.001;
b = 0.3;
ee = 1 + 2.*rand(1,M); // i.i.d distributed-in-range (-1,1)
xn = zeros(1,5000);      // xn is x referred to as in algorithm
xn(1) = 2;                  // holds the initial value of x
count = 0;
for n = 1 : 5000
        sum = 0;
        for i = 1 : M
                sum = sum + noncon(xn(n) + b*ee(i));
        end
        sum = double(sum) / M;
        if noncon(xn(n)) <= sum
                xn(n+1) = xn(n) - a*grad(xn(n));
        else
                min = 1e+09;
                mini = 0;       // will hold minimum i, referred to as i* in algorithm
                for i = 1 : M
                        y = noncon(xn(n) + b*ee(i));
                        if y < min
                                min = y;
                                mini = i;
                        end
                end
                count = count+1;
                xn(n) = xn(n) + b*ee(mini);    // will update x(n)
                n = n – 1;                          // because we are updating x(n) without changing any
                                                    //future values
        end
end
x = 0:1:5000;
plot(x,xn);
xlabel('Steps')
ylabel('x')
```

%noncon is the function containing the expression for the function to be evaluated
%grad is the function containing the gradient of the noncon function
%Also code is available at https://github.com/bkpcoding/sgd