Homework 3

Problem 1

Python (Number of cores = 8)

In [28]:
```
import math
import time
import numpy as np
from joblib import Parallel, delayed
from scipy.optimize import fsolve

def root(x):
    return x**2+5*x+6
n=80000
result1=[]

start1 = time.time();
for i in range(n):
    result1.append(fsolve(root,i))
end1 = time.time();
t1=end1-start1;
print("Execution Time for Normal Computation: ",t1);

start2 = time.time();
result2 = Parallel(n_jobs=8)(delayed(fsolve)(root,i) for i in range(n))
end2 = time.time();
t2=end2-start2;
print("Execution Time for Parallel Computation: ",t2)

sp=t1/t2;

print("Speedup: ", sp)
print("Average Efficiency: ", sp/8, "=", str(int((sp/8)*100)) + "%")
```

```
Execution Time for Normal Computation:  11.763902187347412
Execution Time for Parallel Computation:  2.644016981124878
Speedup:  4.4492536437275625
Average Efficiency:  0.5561567054659453 = 55%
```

R (Number of cores = 8)

In [4]:
```
library(tictoc)
library(doParallel)
library(parallel)
library(pracma)

n <- 80000

root <- function(x) {
   return(x**2+5*x+6)
}

start1 <- Sys.time()
result1 <- foreach(i=1:n) %do% fzero(root,i)
end1 <- Sys.time()
t1 <- (end1-start1);

sprintf("Execution Time for Normal Computation: %f",t1);

start2 <- Sys.time()
result2 <- parallel::mclapply(1:n, function(x) fzero(root, x))
end2 <- Sys.time()
t2 <- end2-start2;

sprintf("Execution Time for Parallel Computation: %f",t2)
sp <- as.numeric(t1)/as.numeric(t2);

sprintf("Speedup: %f", sp)
sprintf("Average Efficiency: %f = %d%%", sp/8, as.integer((sp/8)*100))
```

'Execution Time for Normal Computation: 8.467319'

'Execution Time for Parallel Computation: 1.578543'

'Speedup: 5.364009'

'Average Efficiency: 0.670501 = 67%'

Problem 2

Code (Number of Cores = 8)

```
clear;

if isempty(gcp())
 parpool();
end

files = dir('TextFiles/*.txt'); % Text files are stored in this Directory
totalFiles=numel(files);

arr1=cell(1, totalFiles);
arr2=cell(1, totalFiles);

tic
for i=1:totalFiles
    data = fileread(files(i).name);
    [p,l,s] = palindromes(data,'Length', 5);
    if l==5
        output1=1;
        arr1(i)=s; % Identified Palindromes are stored in this cell array
    else
        output1=0;
    end
end
t1 = toc;
x = sprintf("Execution Time for Normal Computation: %f",t1);
disp(x);

tic
parfor i=1:totalFiles
    data = fileread(files(i).name);
    [p,l,s] = palindromes(data,'Length', 5);
    if l==5
        output2=1;
        arr2(i)=s; % Identified Palindromes are stored in this cell array
    else
        output2=0;
    end
end
t2 = toc;

y = sprintf("Execution Time for Embarrassingly Parallel Computation: %f",t2);
disp(y);

sp = t1/t2;
z = sprintf("Speedup: %f", sp);
disp(z);

v = sprintf("Average Efficiency: %f = %d%%", sp/8, int8((sp/8)*100));
disp(v);
```

Sample Text File Content

SIHGVLHDNSERQNFNALUTSYXDGYYKDEQTXTJOWPBPUKPINNCQADFHLIQCNGO
NVFFRHXZYBPGIDFOCKDCALRFHWOVVZBSLDIGYGMHQOTKUITIQKDMOKDUTD
LLCWTJNZVTORXHIKGJRWZEVYPAPTYNRKBXPMEUGFLDEBYGPBZLRHXJLUXFX
GJJHWIEMSBEZSKTRILKLFEUKDERVFCQNYDIZIELCRLCGGVBOEDANSMWGOCQZ
ZAIBDZDYLJNLJZBMAIWOCMKMQYDUWSOEKBLMHCEQRQIGLPFKBSCNFKXEIHZ
QNKYRVJREVYGAVVFLWDCXFLFDHXCQQYDDTGZPDXXINTBKDNUADIPAIUIXSK
OBUUNPQAAHNJZILBYKIKMXCTLQIYFHWRLNZOHTMIFGNFOIGENCEBFPKFZKEP
HUQQMYXCISATEBNSYYLCYRPBLWBCMUCTOIZRHTFTFNEHJNWFXSJUTWEVOCD
MMNTEIJIIWYCSBDVWIBUMXUZRRCMQJXEMFDTHGIWNYZLIZYLNJXILLDDPLLE
OKBGARYORQWBMXCLGCCNEHGUASDOWKPAHGGLBEXEXGVMMIHIHQRBPMHU
KFOZTMDAYWEANWYLXAELMXVVRSUWCVHMBJKDERDMATAMNLBUTXKBXFG
DJATXSPZRDWTBPYUGLHRAMKRCWYBVQISECKYHRLBPZKTUITKNESGHOONXHS
UIVTTGRAECVJLTMBIMTBSVSTHCGLLTGBWUBYAXNBCJDHEXAWCUWYTTQRJG
CPVNZZPCIVADLIDAPYAWTYBBDYYBEBNOQGEIMFCWJYITKUCGVHJPTYFOVHW
CALFAVEJBZWRHUFTHUJXMMBNXFOEVDHDLAMXNPOATNMEOJRRJAIXHCMAIB
MKWPAFYVYSQCDDKHETCIRNRWJZBVEXKUDKSWIMPKPDXUGDLVOYXSQFORK
FWOHMZLLCVKZLLADYBUDEESHOEQDPIRNRYBOSZXCOHBEJFURDVRESEGZQAY
TFTLNYEIBLQJKAZGNEAMO

Result for 100 Text Files

Execution Time for Normal Computation: 0.023313
Execution Time for Embarrassingly Parallel Computation: 0.172930
Speedup: 0.134813
Average Efficiency: 0.016852 = 2%

Result for 10000 Text Files

Execution Time for Normal Computation: 1.168037
Execution Time for Embarrassingly Parallel Computation: 0.560343
Speedup: 2.084505
Average Efficiency: 0.260563 = 26%

Result for 80000 Text Files

Execution Time for Normal Computation: 12.414793
Execution Time for Embarrassingly Parallel Computation: 4.400903
Speedup: 2.820965
Average Efficiency: 0.352621 = 35%

Problem 3

Code (Number of Cores = 8)

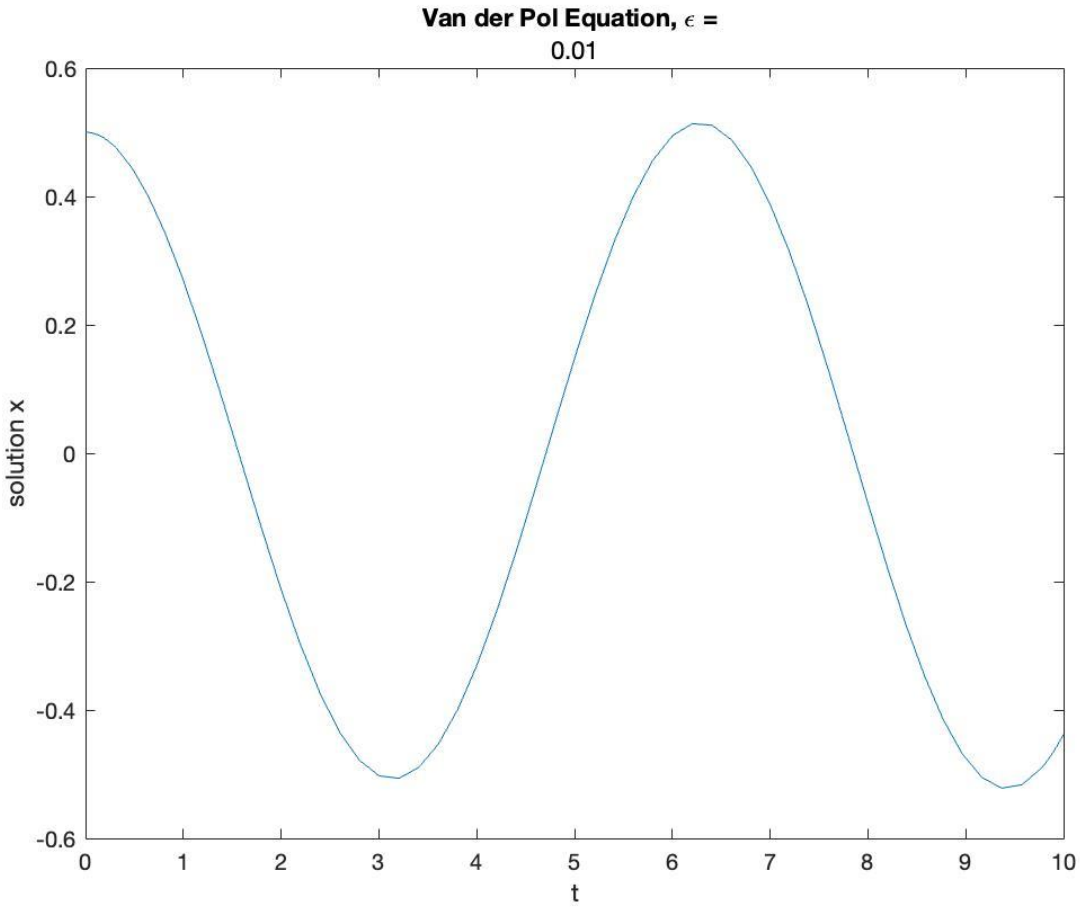
```
clear;

if isempty(gcp())
    parpool();
end

tspan = [0 10]; % t = 0 to 10
x0 = [0.5; 0]; % x(0) = 0.5

tic
for i = 1:5000 % 5000 different values of Œµ from 0.01 to 4

    if i<400
        Ep = i/100;
    elseif i<4000
        Ep = i/1000;
    else
        Ep = i/1250;
    end

    ode = @(t,x) vanderpoldemo(t,x,Ep);
    [t,x] = ode45(ode, tspan, x0);

    if ismember(i, [1,10,500,2000,5000]) % plot the solution for 5 Œµ values
        figure(i);
        plot(t,x(:,1));
        xlabel('t');
        ylabel('solution x');
        title('Van der Pol Equation, \epsilon = ', Ep);
    end
end
t1 = toc;

m = sprintf("Execution Time for Normal Computation: %f",t1);
disp(m);

tic
parfor i = 1:5000 % 5000 different values of Œµ from 0.01 to 4

    if i<400
        Ep = i/100;
    elseif i<4000
        Ep = i/1000;
    else
        Ep = i/1250;
    end

    ode = @(t,x) vanderpoldemo(t,x,Ep);
    [t,x] = ode45(ode, tspan, x0);

    if ismember(i, [1,10,500,2000,5000]) % plot the solution for 5 Œµ values
        figure(i);
        plot(t,x(:,1));
        xlabel('t');
        ylabel('solution x');
        title('Van der Pol Equation, \epsilon = ', Ep);
    end
```

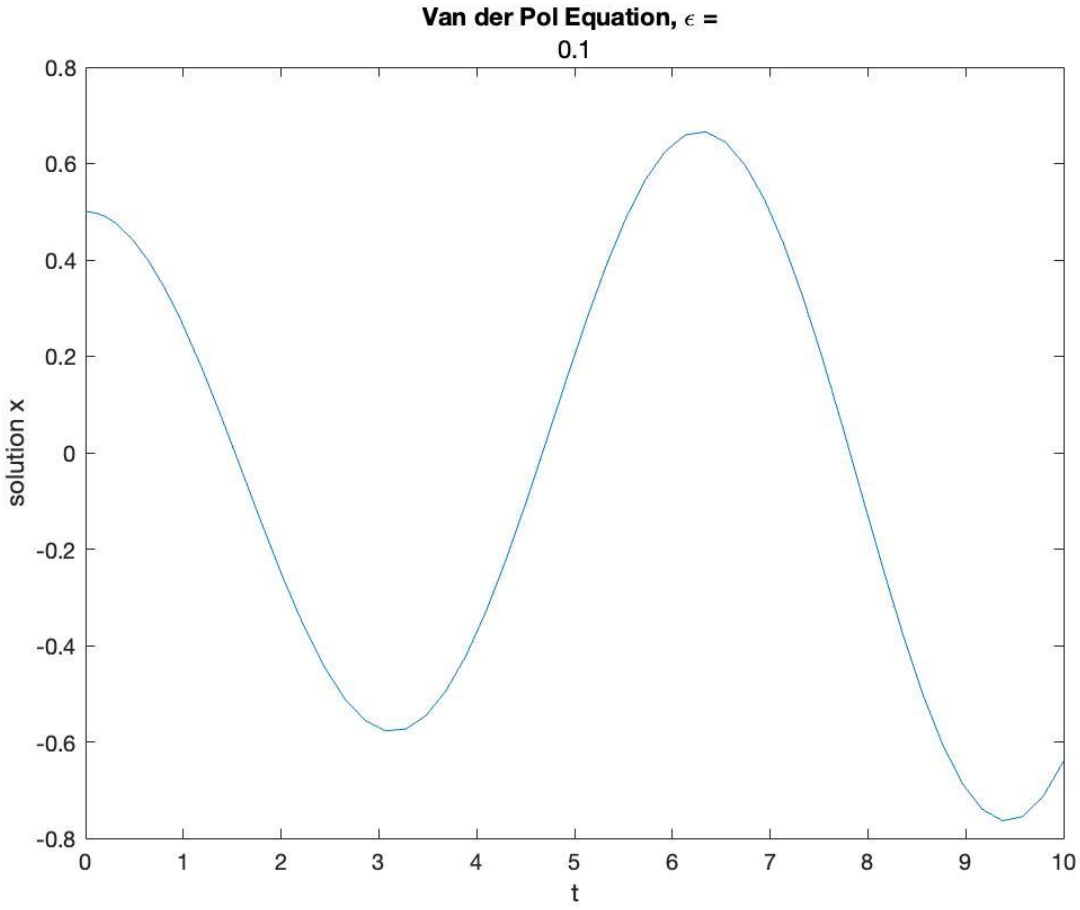
```
end
t2 = toc;

n = sprintf("Execution Time for Embarrassingly Parallel Computation: %f",t2);
disp(n);

sp = t1/t2;
z = sprintf("Speedup: %f", sp);
disp(z);

v = sprintf("Average Efficiency: %f = %d%%", sp/8, int8((sp/8)*100));
disp(v);
```

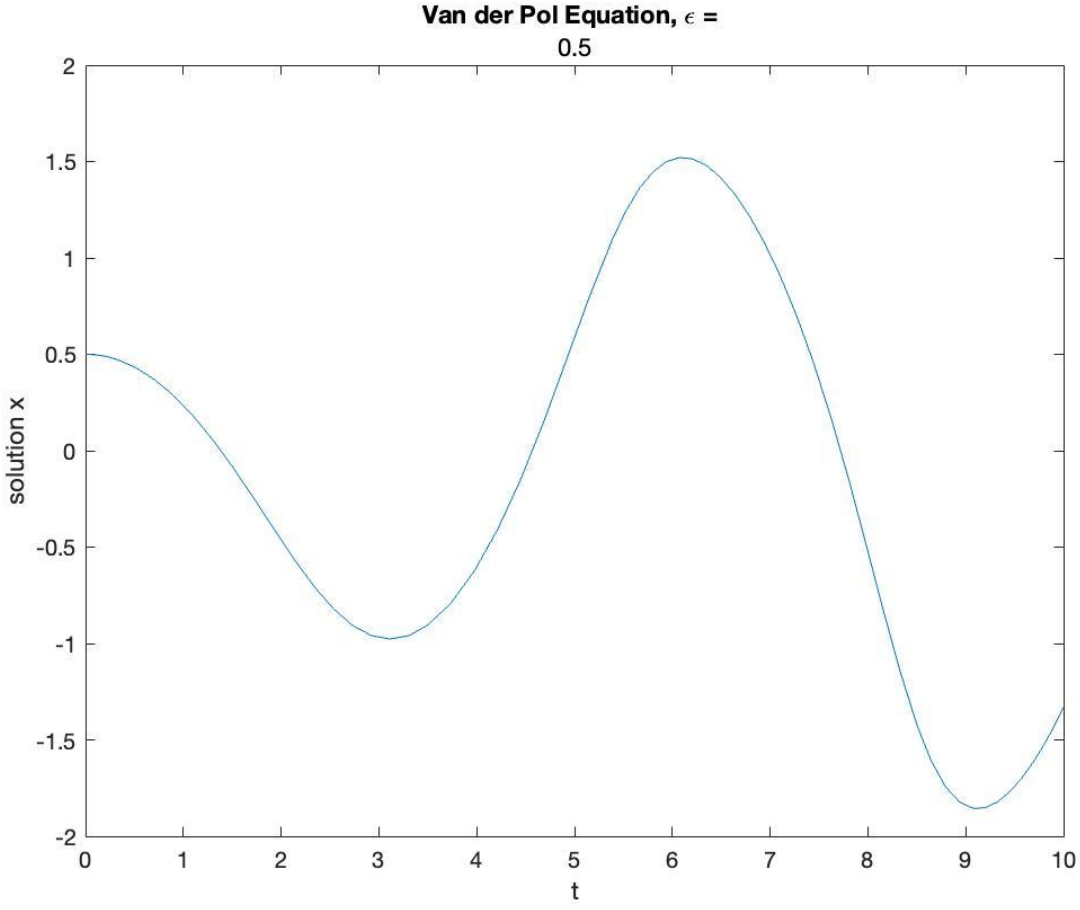
Result

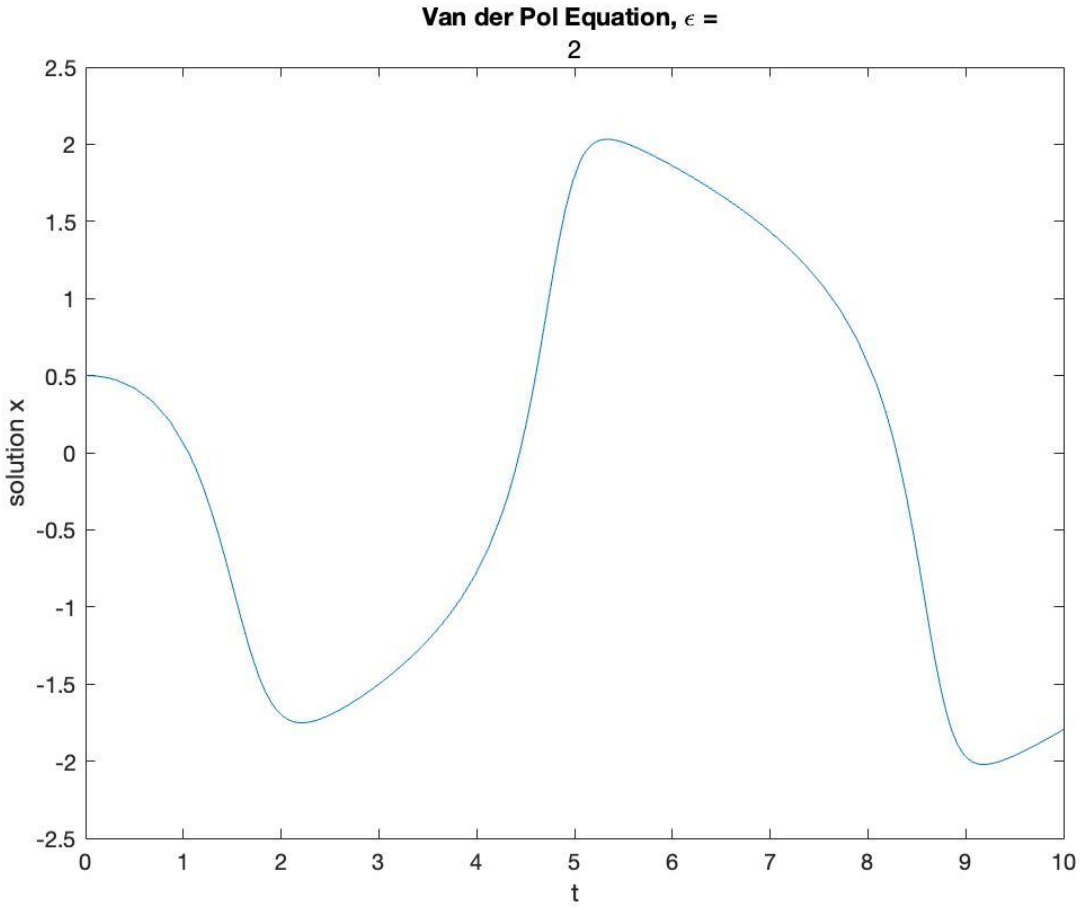
Execution Time for Normal Computation: 1.904806
Execution Time for Embarrassingly Parallel Computation: 0.472971
Speedup: 4.027316
Average Efficiency: 0.503415 = 50%

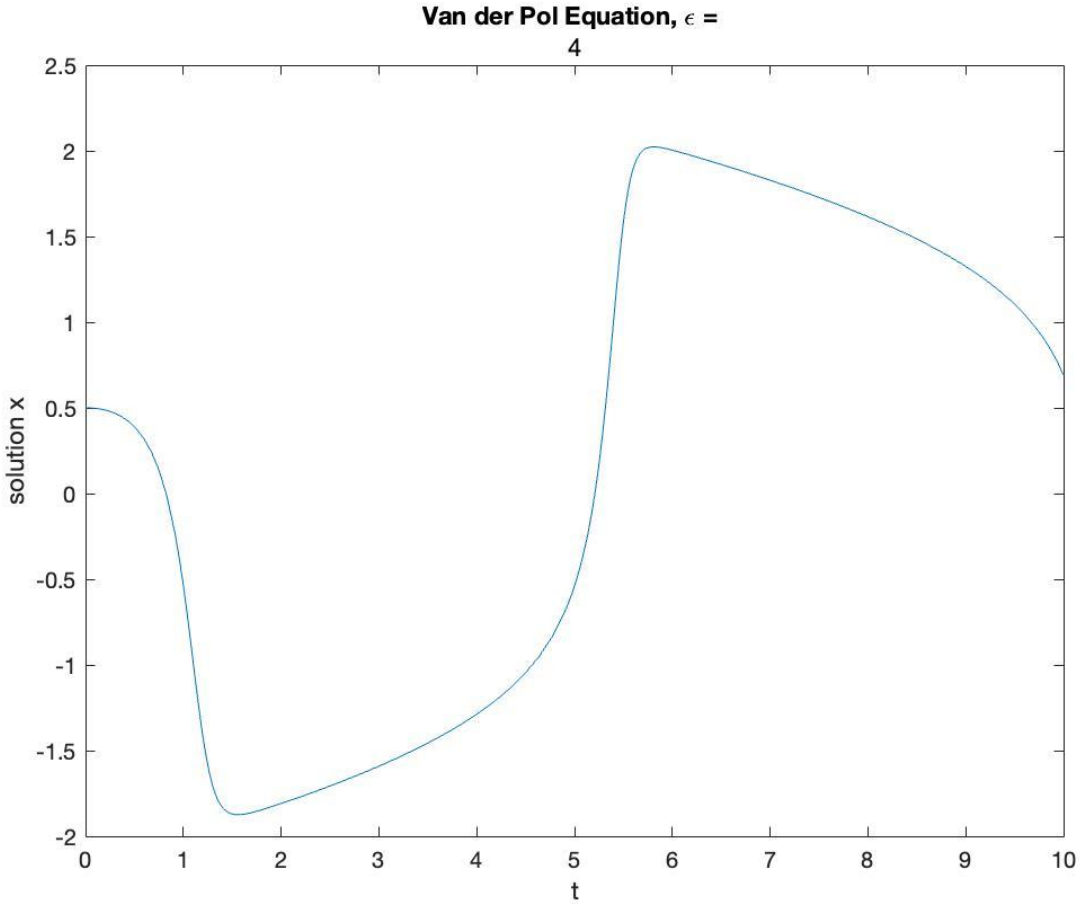
Solutions and Plots




Van der Pol Equation, ϵ =
0.1
solution x
t
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8
9
10


Van der Pol Equation, ϵ =
0.5
solution x
t
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8
9
10


Van der Pol Equation, ϵ =
2
solution x
t
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6
7
8
9
10


Van der Pol Equation, ϵ =
4
solution x
t
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8
9
10