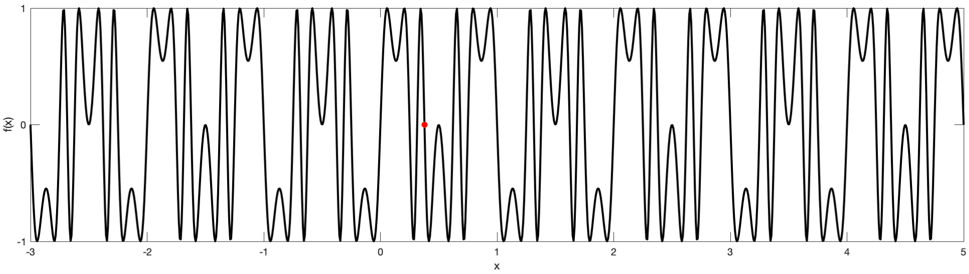
Matlab Output

Finding root for the equation

```
clear all; close all; % Start fresh

f = @(x) sin(3*pi*cos(2*pi*x).*sin(pi*x));
a = -3; b = 5;
x0 = 0.5;

tic
    q = fzero(f,x0);
toc


%Plot the function and roots if possible
xx = linspace(a,b,1001);
fig = figure('Position',[100 100 1200 300]);
plot(xx,f(xx),'-k','linewidth',2);
hold on
plot(q,f(q),'o','markerfacecolor','r')
xlim([a,b]); ylim([-1,1]);
yticks([-1 0 1])
xlabel('x');
ylabel('f(x)');
pbaspect([4 1 1])
```

Output:



Finding T1:

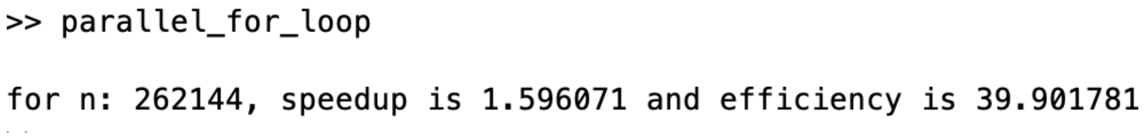
```
function t1 = getT1(f, n, x0)

q = zeros(size(x0)); % Preallocate a vector for storing roots.
```

```
tic
for i=1:n
    q(i) = fzero(f,x0(i));
end
t1 = toc;

%Processing Outputs%
q = unique(q); % keep roots with unique values only.


end
```

Parallel Processing to find root

```
clear all; close all;

p = feature('numcores');

% Create a parallel pool if none exists
if isempty(gcp())
    parpool();
end


% Inputs
f = @(x) sin(3*pi*cos(2*pi*x).*sin(pi*x));
a = -3;
b = 5;
n = 4^9;

x0 = linspace(a,b,n); % Vector containing initial starting points
q = zeros(size(x0)); % Preallocate a vector for storing roots.


% Parallel Processing
tic
parfor i=1:n
    q(i) = fzero(f,x0(i));
end
tp = toc;


% Processing Outputs
q = unique(q); % keep roots with unique values only.

% Calling function to find 'T1'
t1 = getT1(f, n, x0);

speedup = t1/tp;
efficiency = (speedup/p) * 100;

fprintf("\nfor n: %d, speedup is %f and efficiency is %f\n", n, speedup, efficiency)
```

Output:



```
>> parallel_for_loop

for n: 262144, speedup is 1.596071 and efficiency is 39.901781
```