Problem 3 Matlab

Van der Pol oscillator:

Calculating `ep,` in `getEP.m`

```
function ep = getEP(i)
if i < 400
    ep  = i/100;
elseif i < 4000
    ep = i/1000;
else
    ep = i/1250;
end
```

Code for plotting the diagram, `plot_diagram.m`

```
function plot_ = plot_diagram(i, t, x, ep)
    figure(i);
    plot(t, x(:,1));
    xlabel('t');
    ylabel('solution x');
    title('Van der pol oscillator, epsilon = ', ep);
end
```

Code for calculating the ODE, in `do_vanderpol`

```
function [t, x] = do_vanderpol(tspan, x0, ep)

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

end
```

Driving code, `van_der_pol.m`

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

tspan = [0 10];
x0 = [0.5; 0];

p = feature('numcores');


% Running code serially
tic
for i = 1:5000
    ep = getEP(i);

    [t,x] = do_vanderpol(tspan, x0, ep);

    if ismember(i, [1, 100, 1500, 2500, 5000])
        % Plotting
        plot_diagram(i, t, x, ep)
    end
end
t1 = toc;

% Embarrasingly Parallel Computation
tic
parfor i = 1:5000
    ep = getEP(i);

    [t,x] = do_vanderpol(tspan, x0, ep);

    if ismember(i, [1, 100, 1500, 2500, 5000])
        % Plotting
        plot_diagram(i, t, x, ep)
    end
end
tp = toc;

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

fprintf("t1: %f\n", t1);
fprintf("tp: %f\n", tp);

fprintf("SpeedUp: %f\n", speedup);
fprintf("Efficiency: %f\n", efficiency);
```

Output:

```
>> van_der_pol
t1: 2.255206
tp: 0.820363
SpeedUp: 2.749036
Efficiency: 68.725906
```

Output Graph:

I have uploaded in a different file called `problem_3_vanderpol_output_graph.pdf`