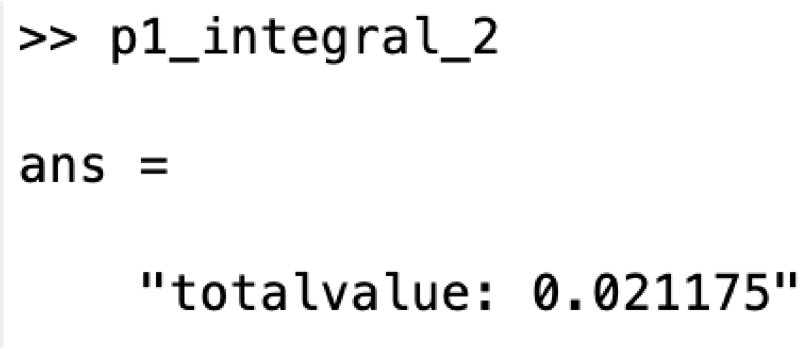
Assignment 4

Problem 1

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

nworkers = gcp().NumWorkers;


% Define the function
f = @(x,y) cos(x).^2.*y.^2;


% Discretize the interval on the client
x = linspace(1,2,nworkers+1);
y = linspace(-1,1,nworkers+1);


% On the workers
spmd
    ainit = x(spmdIndex());
    bfin =  x(spmdIndex()+1);
    cinit = y(spmdIndex());
    dfin = y(spmdIndex()+1);

    locint = integral2(f,ainit,bfin,cinit,dfin);
    totalint = spmdPlus(locint);
end

totalvalue = totalint{1};
sprintf("totalvalue: %f", totalvalue)
```

Output:

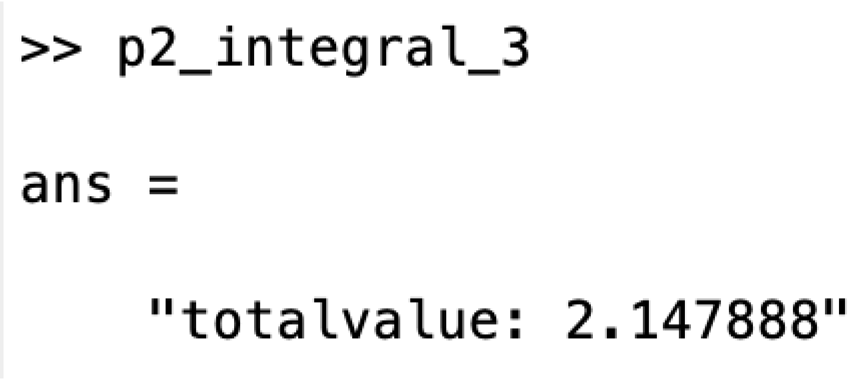


```
>> p1_integral_2

ans =

    "totalvalue: 0.021175"
```

Problem 2

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

n_workers = gcp().NumWorkers;

% Define the function
f = @(x,y,z) x.^3.*y.^3.*z.^3;


% Discretize the interval on the client
x = linspace(1,2,nworkers+1);
y = linspace(-1,1,nworkers+1);
z = linspace(2,3,nworkers+1);


% On the workers
spmd
    ainit = x(spmdIndex());
    bfin = x(spmdIndex()+1);
    cinit = y(spmdIndex());
    dfin = y(spmdIndex()+1);
    einit = z(spmdIndex());
    ffin = z(spmdIndex()+1);

    locint = integral3(f,ainit,bfin,cinit,dfin,einit,ffin);
    totalint = spmdPlus(locint);
end


totalvalue = totalint{1};
sprintf("totalvalue: %f", totalvalue)
```

Output:



```
>> p2_integral_3

ans =

    "totalvalue: 2.147888"
```

Problem 3

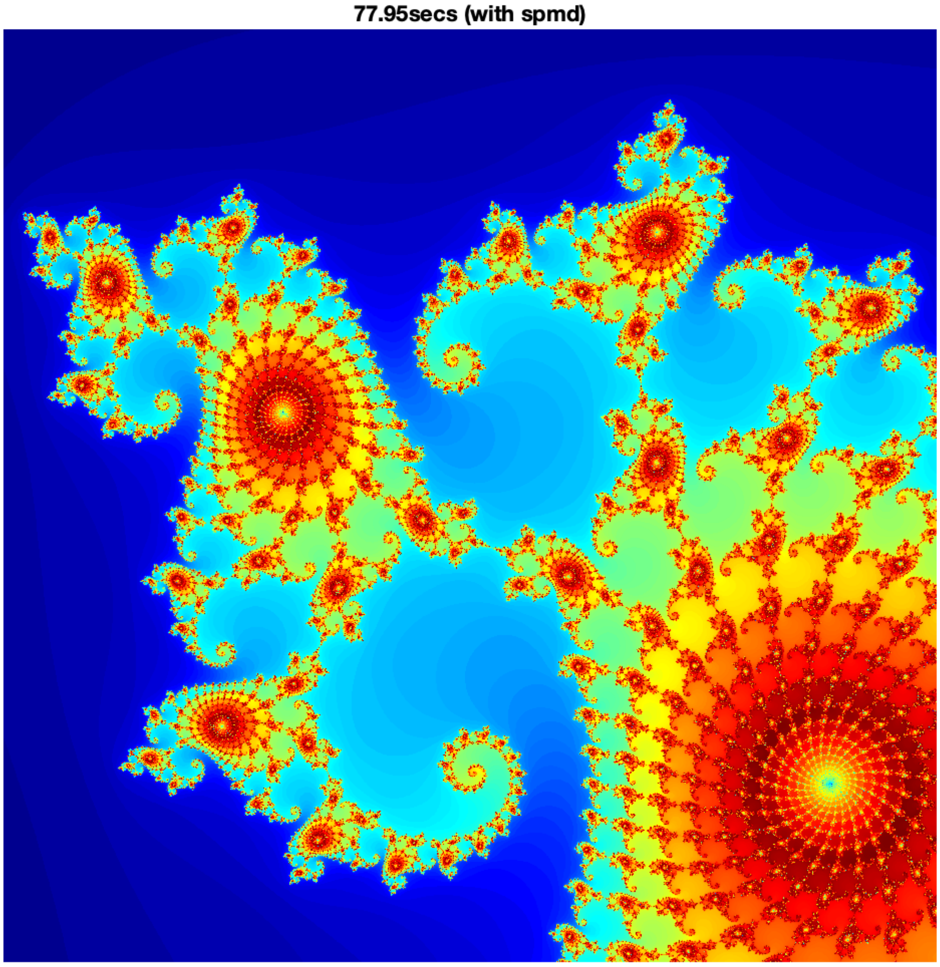
file: `main.m`

```
clear; close all;

if isempty(gcp())
    parpool(4);
end

n_workers = gcp().NumWorkers;

grid_size = 2000;
max_iterations = 2000;

x_lim = [-0.748766713922161, -0.748766707771757];
y_lim = [ 0.123640844894862, 0.123640851045266];
x_lim = linspace(x_lim(1),x_lim(end),(n_workers/2)+1);
y_lim = linspace(y_lim(1),y_lim(end),(n_workers/2)+1);

% Setup on the workers
tic();
spmd
    [m,n] = getMN(spmdIndex());

    x = linspace(x_lim(m),x_lim(m+1),grid_size/spmdSize()*2);
    y = linspace(y_lim(n),y_lim(n+1),grid_size/spmdSize()*2);

    [x_grid,y_grid] = meshgrid(x,y);
    a1 = x_grid + 1i*y_grid; count = ones(size(a1));

    % Calculate the iterations
    a = a1;
    for n = 0:max_iterations
        a = a.*a + a1;
        In = abs(a)<=2; count = count + In;
    end

    count = log(count);
end


% Display on the client
cpu_time = toc();
set( gcf,'Position',[200 200 600 600] );
imagesc(cat(1,cat(2,count{1},count{3}),cat(2,count{2},count{4})));
axis image;axis off; colormap([jet();flipud(jet());0 0 0]); drawnow;
title( sprintf('%1.2fsecs (with spmd)',cpu_time));
```

function: `getMN.m`:

```
function [m,n] = getMN(li)
if li==1
    m=1;
    n=1;

elseif li==2
    m=1;
    n=2;

elseif li==3
    m=2;
    n=1;

else
    m=2;
    n=2;
end
end
```

Output:



77.95secs (with spmd)