```
% Example_5_1.mlx
%
% This is a live script version of Example 5.1 from the paper
% 'Geometric modelling of polycrystalline materials: Laguerre tessellations and periodic
% semi-discrete optimal transport' by D.P. Bourne, M. Pearce & S.M. Roper.
```

Example 5.1 from the paper *"Geometric modelling of polycrystalline materials: Laguerre tessellations and periodic semi-discrete optimal transport" by D.P. Bourne, M. Pearce and S. M. Roper.*

Grain sizes for three different microstructures are specified. The microstructures considered are single phase, dual phase and log-normal. Semi-discrete optimal transport is used to find a **periodic Laguerre diagram** with prescribed random seeds whose cell volumes match the grain volumes of the target microstructure. The calculation involves solving a concave maximisation problem, for which we used the damped Newton method with backtracking. In this example we report average run-times for each microstructure.

Microstructures

A single-phase microstructure is one in which all the grains have the same volume. A dual-phase microstructure has two populations of grains, within each population the grains have the same volume, but grains from different populations have distinct volumes, both the sizes of the populations and the relative volumes between populations can be varied. A log-normal microstructure has grain volumes drawn from an approximate log-normal distribution.

The box is the unit cube $[0,1]\times[0,1]\times[0,1]$

```
%%% Specify box dimensions
bx=[1,1,1];
```

We compute periodic Laguerre diagrams

```
%%% Specify periodicity
periodic=true;
```

Specify the maximum percentage error in the cell volumes

```
%%% Specify maximal percentage error
percent_tol=1;
```

The vector `n` is a row vector of numbers of cells. In the example below we choose `n=1000`

```
%%% Specify number of grains (seeds)
% Set n=[100,250,500,1000,2500,5000,10000,25000,50000,100000]
% to reproduce the results from the paper
n=[1000];
nn=length(n);
```

The number of random draws over which the average run-time will be calculated

```
%%% Specify number of experiments for each number of seeds
nexp=100;
```

We control which microstructures are considered using the boolean variables below

```
%%% Specify whether the microstructure is simulated or not
singlephase=true;
dualphase=false; % Set to true to reproduce the results from the paper
lognormal=false; % Set to true to reproduce the results from the paper
```

Single phase

All grains have the same volume $v = 1/N$ where N is the number of cells.

```
%%% Single phase run times

if(singlephase)
    disp('Single phase')
    for idx_n=1:nn
        % For each number of cells

        disp(sprintf('Calculating run times for n = %d',n(idx_n)));

        % Initial guess is always w=0
        w_0=zeros(n(idx_n),1);

        for idx_exp=1:nexp
            % For each experiment

            fprintf('\t Experiment number %d',idx_exp);

            % Seeds locations and target volumes
            [X,target_vols]=nPhase(n(idx_n),1,bx);

            % Store the X and target_vols
            X_sp{idx_n,idx_exp}=X;
            tv_sp{idx_n,idx_exp}=target_vols;

            tic
            [w,max_percent_err,actual_vols,EXITFLAG] = ...
                SDOT_damped_Newton(w_0,X,target_vols,bx,periodic,percent_tol);
            t=toc;

            fprintf(' : completed in %f\n',t);

            % Store the solution for w
            w_sp{idx_n,idx_exp}=w;

            runtime_sp(idx_n,idx_exp)=t; % run times
        end

        disp(sprintf('Average run time %f',mean(runtime_sp(idx_n,:)')));
    end
end
```

```
Single phase
```

```
Calculating run times for n = 1000
     Experiment number 1
 : completed in 0.591395
     Experiment number 2
 : completed in 0.446159
     Experiment number 3
 : completed in 0.459348
     Experiment number 4
 : completed in 0.453556
     Experiment number 5
 : completed in 0.419105
     Experiment number 6
 : completed in 0.443767
     Experiment number 7
 : completed in 0.426289
     Experiment number 8
 : completed in 0.411200
     Experiment number 9
 : completed in 0.471978
     Experiment number 10
 : completed in 0.439541
     Experiment number 11
 : completed in 0.446884
     Experiment number 12
 : completed in 0.432571
     Experiment number 13
 : completed in 0.626587
     Experiment number 14
 : completed in 0.437245
     Experiment number 15
 : completed in 0.454437
     Experiment number 16
 : completed in 0.441623
     Experiment number 17
 : completed in 0.476867
     Experiment number 18
 : completed in 0.445830
     Experiment number 19
 : completed in 0.469297
     Experiment number 20
 : completed in 0.595207
     Experiment number 21
 : completed in 0.448934
     Experiment number 22
 : completed in 0.506838
     Experiment number 23
 : completed in 0.451690
     Experiment number 24
 : completed in 0.461690
     Experiment number 25
 : completed in 0.467137
     Experiment number 26
 : completed in 0.511982
     Experiment number 27
 : completed in 0.427202
     Experiment number 28
 : completed in 0.432271
     Experiment number 29
 : completed in 0.433404
     Experiment number 30
 : completed in 0.444821
     Experiment number 31
 : completed in 0.459688
     Experiment number 32
```

```
: completed in 0.440436
    Experiment number 33
: completed in 0.587965
    Experiment number 34
: completed in 0.466514
    Experiment number 35
: completed in 0.457397
    Experiment number 36
: completed in 0.503858
    Experiment number 37
: completed in 0.470570
    Experiment number 38
: completed in 0.468437
    Experiment number 39
: completed in 0.826026
    Experiment number 40
: completed in 0.806283
    Experiment number 41
: completed in 0.854106
    Experiment number 42
: completed in 0.963485
    Experiment number 43
: completed in 0.783150
    Experiment number 44
: completed in 0.907366
    Experiment number 45
: completed in 0.780219
    Experiment number 46
: completed in 1.162291
    Experiment number 47
: completed in 0.765395
    Experiment number 48
: completed in 0.800486
    Experiment number 49
: completed in 0.648777
    Experiment number 50
: completed in 0.710920
    Experiment number 51
: completed in 0.704078
    Experiment number 52
: completed in 1.026277
    Experiment number 53
: completed in 0.779648
    Experiment number 54
: completed in 0.800961
    Experiment number 55
: completed in 0.830485
    Experiment number 56
: completed in 0.805784
    Experiment number 57
: completed in 0.818660
    Experiment number 58
: completed in 0.770208
    Experiment number 59
: completed in 0.809277
    Experiment number 60
: completed in 0.806292
    Experiment number 61
: completed in 1.069739
    Experiment number 62
: completed in 0.885339
    Experiment number 63
: completed in 0.861221
    Experiment number 64
```

```
: completed in 0.879138
    Experiment number 65
: completed in 0.802033
    Experiment number 66
: completed in 0.782411
    Experiment number 67
: completed in 0.800011
    Experiment number 68
: completed in 0.787911
    Experiment number 69
: completed in 0.814544
    Experiment number 70
: completed in 0.846892
    Experiment number 71
: completed in 0.846683
    Experiment number 72
: completed in 0.795891
    Experiment number 73
: completed in 0.793165
    Experiment number 74
: completed in 0.826484
    Experiment number 75
: completed in 0.765824
    Experiment number 76
: completed in 0.791195
    Experiment number 77
: completed in 0.798384
    Experiment number 78
: completed in 0.746678
    Experiment number 79
: completed in 0.521759
    Experiment number 80
: completed in 0.424891
    Experiment number 81
: completed in 0.451504
    Experiment number 82
: completed in 0.409857
    Experiment number 83
: completed in 0.442239
    Experiment number 84
: completed in 0.486373
    Experiment number 85
: completed in 0.455838
    Experiment number 86
: completed in 0.456864
    Experiment number 87
: completed in 0.488034
    Experiment number 88
: completed in 0.448331
    Experiment number 89
: completed in 0.449329
    Experiment number 90
: completed in 0.523702
    Experiment number 91
: completed in 0.488059
    Experiment number 92
: completed in 0.437709
    Experiment number 93
: completed in 0.495402
    Experiment number 94
: completed in 0.631792
    Experiment number 95
: completed in 0.488700
    Experiment number 96
```

```
 : completed in 0.486260
    Experiment number 97
 : completed in 0.597449
    Experiment number 98
 : completed in 0.571758
    Experiment number 99
 : completed in 0.833079
    Experiment number 100
 : completed in 0.564597
Average run time 0.620370
```

Dual phase

There are N_1 grains of volume v_1 and N_2 grains of volume v_2 where $N = N_1 + N_2$ and $N_1 v_1 + N_2 v_2 = 1$ (since the total volume must be the same as the volume of $[0,1] \times [0,1] \times [0,1]$, which is 1).

```
%%% Dual phase run times

if(dualphase)
    % Set a size ratio
    r=5;

    % Set a fraction of seeds that are the smaller phase
    f=0.5;

    disp('Dual phase');

    for idx_n=1:nn
        % For each number of cells

        disp(sprintf('Calculating run times for n = %d',n(idx_n)));

        % Initial guess is always w=0
        w_0=zeros(n(idx_n),1);

        for idx_exp=1:nexp
            % For each experiment

            fprintf('\t Experiment number %d',idx_exp);

            pn=round(n(idx_n)*[f 1-f]); % number of cells of each volume fraction
            vr=[1,r]; % volume fractions

            % Seeds locations and target volumes
            [X,target_vols]=nPhase(pn,vr,bx);

            % Store the X and target_vols
            X_dp{idx_n,idx_exp}=X;
            tv_dp{idx_n,idx_exp}=target_vols;

            tic
            [w,max_percent_err,actual_vols,EXITFLAG] = ...
                SDOT_damped_Newton(w_0,X,target_vols,bx,periodic,percent_tol);
            t=toc;
```

```
            fprintf(' : completed in %f\n',t);

            % Store the solution for w
            w_dp{idx_n,idx_exp}=w;

            runtime_dp(idx_n,idx_exp)=t; % run times
        end

        disp(sprintf('Average run time %f',mean(runtime_dp(idx_n,:)')));

    end
end
```

Log-normal

In the log-normal microstructure the grain volumes are approximately log-normally distributed.

```
%%% Log-normal distribution run times

if(lognormal)

    % Parameters in log-normal distribution
    ln_mean=1; % mean
    std_dev=0.35; % standard deviation
    sigma=sqrt((log(1+(std_dev/ln_mean)^2))); % log-normal parameter sigma
    mu=-0.5*sigma^2+log(ln_mean); % log-normal parameter mu

    disp('Log-normal')
    for idx_n=1:nn
        % For each number of cells

        disp(sprintf('Calculating run times for n = %d',n(idx_n)));

        % Initial guess is always w=0
        w_0=zeros(n(idx_n),1);

        for idx_exp=1:nexp
            % For each experiment

            fprintf('\t Experiment number %d',idx_exp);

            % Seed locations
            X=rand(n(idx_n),3);

            % Compute target volumes:
            % Draw radii from log-normal distribution
            rad=lognrnd(mu,sigma,n(idx_n),1);
            % Calculate the corresponding grain volumes
            % (we don't need the factor 4pi/3 as we'll be renormalising)
            target_vols=rad.^3;
            % Normalise the volumes so that they add to the volume of the box
            target_vols=target_vols*prod(bx)/sum(target_vols); % target volumes of the grains
```

```
            % Store the X and target_vols
            X_ln{idx_n,idx_exp}=X;
            tv_ln{idx_n,idx_exp}=target_vols;

            tic
            [w,max_percent_err,actual_vols,EXITFLAG] = ...
                SDOT_damped_Newton(w_0,X,target_vols,bx,periodic,percent_tol);
            t=toc;

            fprintf(' : completed in %f\n',t);
            % Store the solution for w
            w_ln{idx_n,idx_exp}=w;

            runtime_ln(idx_n,idx_exp)=t; % run times
        end

        disp(sprintf('Average run time %f',mean(runtime_ln(idx_n,:)')));

    end
end
```