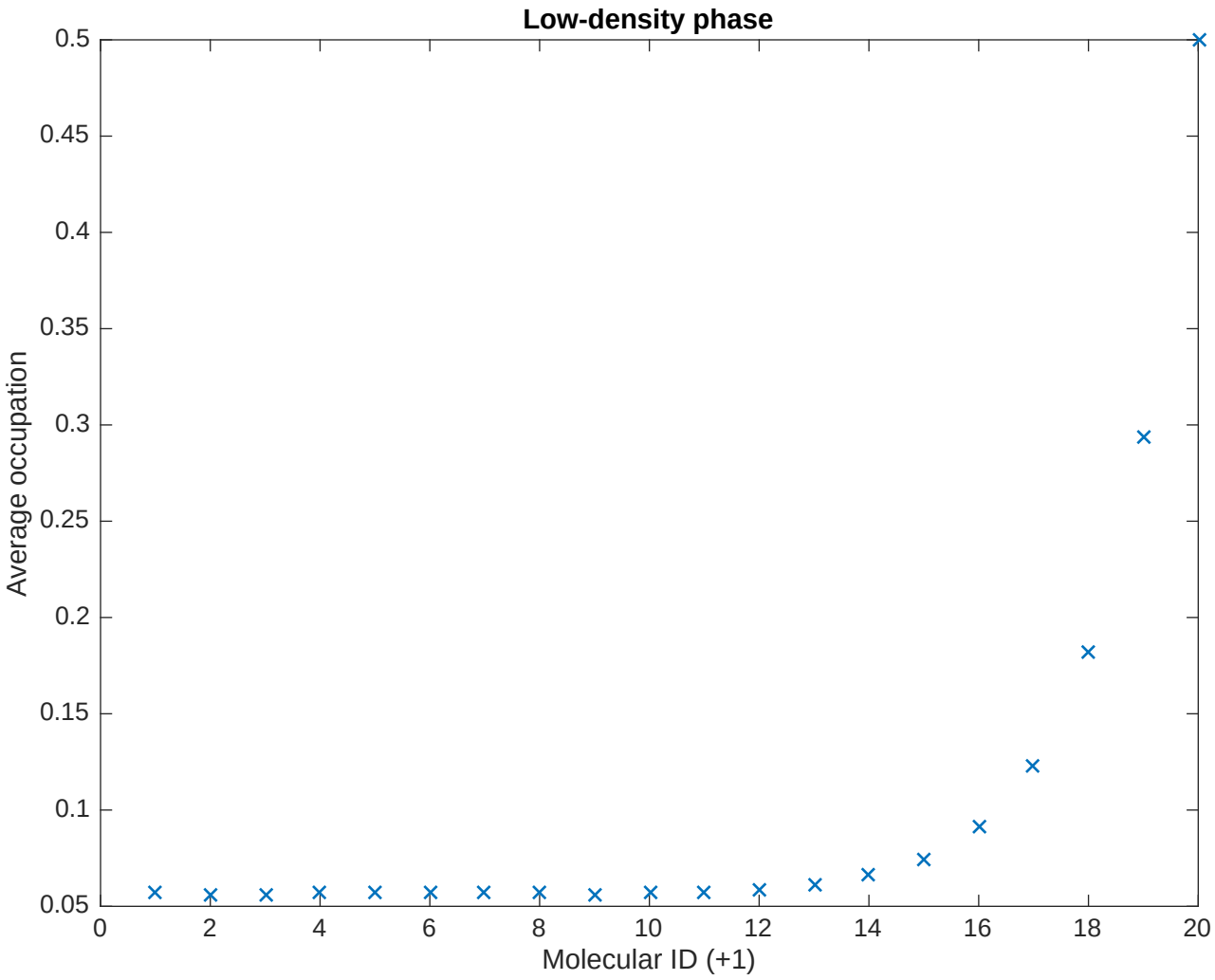
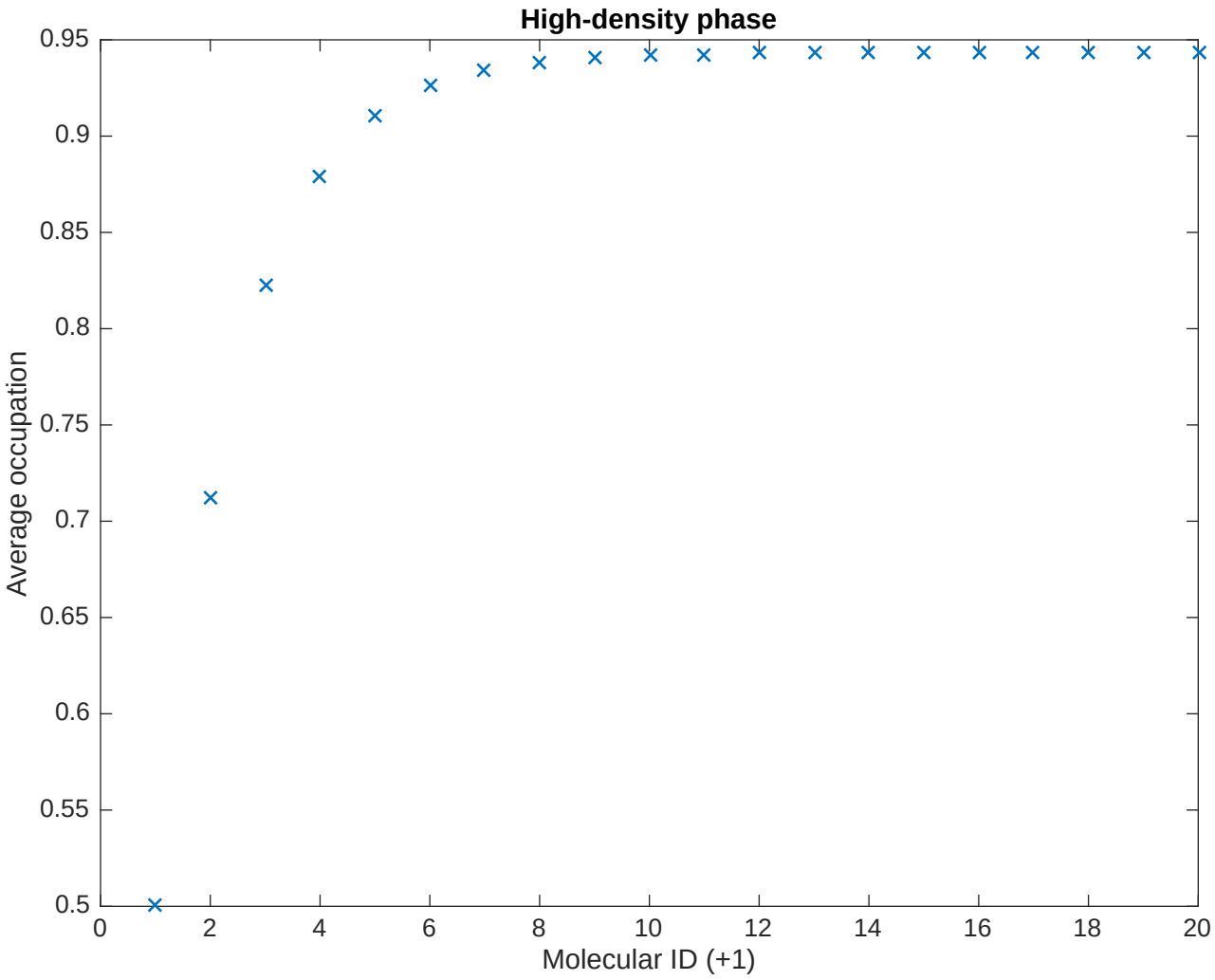
```c
//
//  KMCet.c
//
//  Kinetic Monte Carlo Simulation of Electron Transfer
//
//  Created by Cathie So on 18/4/2016.
//

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define N 20
#define Max_step 1e6
#define k_hop 1e9
#define Max_ngb 1

//changing inputs
#define k_red 1e7
#define k_ox 2e7

int main() {
    double occ[N]={0},sum_occ[N]={0};
    double rate[N][Max_ngb+2]={0};
    double rate_occ[N][Max_ngb+2];
    int lsngb[N][Max_ngb+1];
    int i,k,step;

    //populate lsngb
    for (i=0; i<N; i++) {
        lsngb[i][0]=1;
        lsngb[i][1]=i+1; // only outgoing neightbor is the next molecule
    }

    //populate rate (constants)
    rate[0][0]=k_red;
    rate[N-1][1]=k_ox;
    for (i=0; i<N-1; i++) {
        rate[i][2]=k_hop;
    }

    int n_red=0, n_ox=0;
    double t=0;
    double r, r_th, r_acc;

    for (step=1; step<=Max_step; step++) {
        r = 0;
        //update rate_occ
        for (i=0; i<N; i++) {
            rate_occ[i][0]=rate[i][0]*(1-occ[i]);
            rate_occ[i][1]=rate[i][1]*occ[i];
            rate_occ[i][2]=rate[i][2]*(1-occ[lsngb[i][1]])*occ[i];
        }

        for (i=0; i<N; i++) {
```

```
            r += rate_occ[i][0];
            r += rate_occ[i][1];
            for (k=1; k<=lsngb[i][0]; k++) {
                r += rate_occ[i][k+1];
            }
        }

        //t -= log(rand()/RAND_MAX)/r;
        r_th = r*rand()/RAND_MAX;
        r_acc = 0;

        for (i=0; i<N; i++) {
            if (r_th<(r_acc+=rate_occ[i][0]))   {   //reduction
                occ[i] = 1;
                //printf("%le %d %d\n",t,++n_red,n_ox);
                break;
            }
            else if (r_th<(r_acc+=rate_occ[i][1]))  {   //oxidation
                occ[i] = 0;
                //printf("%le %d %d\n",t,n_red,++n_ox);
                break;
            }
            else
                for (k=1; k<=lsngb[i][0]; k++) {    //hopping
                    if (r_th<(r_acc+=rate_occ[i][k+1])) {
                        occ[lsngb[i][k]]=1;
                        occ[i]=0;
                        break;
                    }
                }
            if (r_th<r_acc) break;
        }

        for (i=0; i<N; i++) {
            sum_occ[i] += occ[i];
        }
    }

    for (i=0; i<N; i++) {
        occ[i] = sum_occ[i]/Max_step;
        printf("%le\n",occ[i]);
    }
}
```


Low-density phase
Average occupation
Molecular ID (+1)
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2
4
6
8
10
12
14
16
18
20


High-density phase
Average occupation
Molecular ID (+1)
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
2
4
6
8
10
12
14
16
18
20