

```

%{
U ovoj vežbi ćemo analizirati Metropolis Monte Karlo metodu za kanonske
(NVT) ansamble primenjene na Lenard-Džonsov potencijal 6-12.
Koristićemo standardne redukovane jedinice ( $u^*=u/\epsilon$  za energiju i
 $r^*=r/\sigma$  za dužinu). Cilj je da procenimo svojstva fluida na temperaturama
i gustinama blizu  $T^*=1.0$  ( $T^*=kT/\epsilon$ ) i  $\rho^*=0.85$  ( $\rho^*=N\sigma^3/V$ ).
%}

%Definišimo parametre za simulaciju:
%broj čestica (ukupno  $l^3$ ), gustinu i temperaturu:
l=2;
rho=0.85;
tstar=1.0;
%Ukupan broj čestica u sistemu:
n = l^3;

%drmax je maksimalni početni pomeraj.
%niter je broj Monte Karlo ciklusa.
niter=100;
drmax=0.15;

%Veličina čestica na slici:
size=1000;

%Određujemo dimenzije jedinične ćelije:
boxl = l/(rho^(1/3));
boxl2 = boxl/2;
boxl3 = boxl/l;
%{
Definišimo položaje čestica u simulacionoj kutiji.
Počecemo sa prostom kubnom rešetkom, kako bi bili sigurni da se u početnom
trenutku nijedan par čestica ne nalazi suviše blizu.
%}
%položaji čestica:
coord = zeros(n,3);
for i = 1:n
    j = ceil(i/l^2);
    j1 = j-1;
    k1 = i-j1*l^2;
    k = ceil(k1/l);
    m = rem(i,l) + 1;

    coord(i,1) = j;
    coord(i,2) = k;
    coord(i,3) = m;
end
coord = coord - (l+1)/2;
coord = coord*boxl3;

figure('Name','Čestice u početnoj ćeliji');
scatter3(coord(:,1),coord(:,2),coord(:,3),size,[0 0 1],'filled');
axis([-1 1 -1 1 -1 1]*boxl2);

%Izračunajmo potencijalnu energiju i faktor kompresibilnosti početne konfiguracije.

pot = 0;
vir = 0;

%računanje rastojanja između i-te i j-te čestice
for i=1:(n-1)

```

```

rx_i = coord(i,1);
ry_i = coord(i,2);
rz_i = coord(i,3);

for j = i+1:n
    rx_ij = rx_i - coord(j,1);
    ry_ij = ry_i - coord(j,2);
    rz_ij = rz_i - coord(j,3);

    rx_ij = rx_ij - round(rx_ij/box1)*box1;
    ry_ij = ry_ij - round(ry_ij/box1)*box1;
    rz_ij = rz_ij - round(rz_ij/box1)*box1;

    %rastojanja između svih čestica na kvadrat r^2
    rijsq = rx_ij^2 + ry_ij^2 + rz_ij^2;

    sr2 = 1/rijsq;
    sr6 = sr2^3;
    srl2 = sr6^2;
    vij = srl2-sr6;

    %sabiranje potencijalnih energija između svih čestica
    pot = pot + vij;
    wij = vij + srl2;
    vir = vir + wij;
end
end

%redukovana potencijalna energija: 4(1/r^12-1/(r^6))
u = pot*4.;
%faktor kompresibilnosti: 24/3(2/r^12-1/(r^6))
w = vir*24./3.;

%početna potencijalna energija u*
ustar = pot*4./n;

%Z*=1+Z/nkT*
zstar = 1 + vir*24/(3*n*tstar);

%output
ustar
zstar

%Monte Karlo simulacija
%Metropolis Monte Carlo Rutinae

%Sada ćemo definisati veličine koje će sprečiti promenu vrednosti potencijalne
energije i faktora kompresibilnosti:
uvect = zeros(niter,1);
zvect = zeros(niter,1);
step = zeros(niter,1);
naccept = 0;

for k = 1:niter    %petlja po ciklusima

    for i = 1:n    %petlja po cesticama

        %stare koordinate
        rx_iold = coord(i,1);
        ry_iold = coord(i,2);

```

```

rziold = coord(i,3);

%računanje energije interakcije čestica u staroj konfiguraciji
pot = 0;
vir = 0;

for j = 1:n

    if (j ~=i)
        rxij = rxiold - coord(j,1);
        ryij = ryiold - coord(j,2);
        rzij = rziold - coord(j,3);
        rxij = rxij - round(rxij/box1)*box1;
        ryij = ryij - round(ryij/box1)*box1;
        rzij = rzij - round(rzij/box1)*box1;

        rijsq = rxij^2 + ryij^2 + rzij^2;

        sr2 = 1/rijsq;
        sr6 = sr2^3;
        sr12 = sr6^2;
        vij = sr12-sr6;

        pot = pot + vij;
        wij = vij + sr12;
        vir = vir + wij;
    end
end

potold = pot*4.;
viroid = 24.*vir/3;

%probni pomeraj čestice i
rxinew = rxiold + random('unif', -drmax,drmax);
ryinew = ryiold + random('unif', -drmax,drmax);
rzinew = rziold + random('unif', -drmax,drmax);

%računanje energije interakcije čestice i u novoj konfiguraciji
pot = 0;
vir = 0;
for j = 1:n
    if (j ~=i)

        rxij = rxinew - coord(j,1);
        ryij = ryinew - coord(j,2);
        rzij = rzinew - coord(j,3);
        rxij = rxij - round(rxij/box1)*box1;
        ryij = ryij - round(ryij/box1)*box1;
        rzij = rzij - round(rzij/box1)*box1;

        rijsq = rxij^2 + ryij^2 + rzij^2;

        sr2 = 1/rijsq;
        sr6 = sr2^3;
        sr12 = sr6^2;
        vij = sr12-sr6;

        pot = pot + vij;
        wij = vij + sr12;
        vir = vir + wij;
    end
end

```

```

        end
    end
    potnew = 4.*pot;
    virnew = 24.*vir/3;

    %računanje promene energije pri probom pomeraju
    deltu = potnew - potold;
    deltw = virnew - virold;
    delta = deltu/tstar;

    if(delta < 75.) %prihvatanje ili odbijanje probnog pomeraja
        ranno = random('unif',0,1);
        %Ako je  $\Delta U \leq 0$  ili  $e^{-\Delta U} >$  slučajnog broja između 0 i 1
        if((deltu<=0) || ( exp(-delta) > ranno))
            %pomeraj prihvacen
            u = u+deltu;
            w = w+deltw;
            coord(i,1) = rxinew;
            coord(i,2) = ryinew;
            coord(i,3) = rzinew;
            naccept = naccept + 1;
        end
    end
end

%zapamtiti potencijalnu energiju po čestici "u" i faktor
%kompresibilnosti "z" za datu konfiguraciju
step(k) = k;
uvect(k) = u/n;
zvect(k) = 1+w/(n*tstar);
end

fractiona = num2str(naccept/(n*niter));
disp(['Deo pomeraja koji je prihvaćen = ',fractiona]);
disp('Dakle, ima nešto manju vrednost od idealnih 0.5');

%Pogledajmo sada kako su raspoređene čestice nakon proračuna
figure('Name','Čestice nakon simulacije');
scatter3(coord(:,1),coord(:,2),coord(:,3),size,[0 0 1],'filled');
axis([-1 1 -1 1 -1 1]*box1/2);

%Grafički prikaz zavisnosti energije od broja ciklusa
figure('Name','Zavisnost energije od broja ciklusa');
scatter(step(:),uvect(:),'filled');
ylabel('Uvect');

%Grafički prikaz zavisnosti faktora kompresibilnosti od broja ciklusa
figure('Name','Zavisnost faktora kompresibilnosti od broja ciklusa');
scatter(step(:),zvect(:),'filled');
ylabel('Zvect');

%Statistika
disp([' '])
disp(['Statistika'])
meanU = num2str(mean(uvect));
varU = num2str(var(uvect));
sdU = num2str(std(uvect));
rangeU = num2str(range(uvect));
medU = num2str(mad(uvect));

```

```
normmedU = num2str(1.253*mad(uvect));
mediandevU = num2str(mad(uvect,1));
normmediandevU = num2str(1.4785*mad(uvect,1));
iqrdevU = num2str(0.7413*iqr(uvect));

disp([' Mean = ', meanU]);
disp([' Variance = ', varU]);
disp([' Standard Deviation = ', sdU]);
disp([' Sample Range = ', rangeU]);
disp([' Mean Deviation = ', medU]);
disp([' Norm Mean Deviation = ', normmedU]);
disp([' Median Deviation = ', mediandevU]);
disp([' Norm Median Deviation = ', normmediandevU]);
disp([' Quartile Deviation = ', iqrdevU]);
disp(' ');
%//Statistika nije idealna zbog malog broja ciklusa i malog broja čestica.
%Povećajte broj ciklusa na 1 000 (2 000, 5 000, 10 000) i analizirajte rezultate.
%Promenite temperaturu na T*=1.5 (0.85) i ponovite račun sa većim brojem ciklusa.
%Uporedite rezultate.
```