

```

clc;
clear;
close all;

colormap('jet');
hold on;

saveflag = 1;

mu = zeros(2, 3);
mu(:, 1) = [3; 5];
mu(:, 2) = [2; 0.4];
mu(:, 3) = [4; 3];

Sigma = zeros(2, 6);
Sigma(:, 1:2) = [1, 0; 0, 1];
Sigma(:, 3:4) = [1, 0; 0, 1];
Sigma(:, 5:6) = [1, 0; 0, 1];

N = 100;
MaxIt = 50;

R1 = zeros(6, N);
R2 = zeros(6, N);
R3 = zeros(6, N);

Sigmahit = zeros(1, 3);

s = zeros(2, 3);
t = zeros(2, 3);
milplus1 = zeros(2, 3);

ss = zeros(2, 6);
tt = zeros(2, 6);
silplus1 = zeros(2, 6);

% Sample Data #1
R1(1:2, :) = mvnrnd([-2, -2], [1, 0; 0, 1], N)';
R2(1:2, :) = mvnrnd([2, 2], [1, 0; 0, 1], N)';
R3(1:2, :) = mvnrnd([6, 6], [1, 0; 0, 1], N)';

% Sample Data #2
R1(1:2, :) = mvnrnd([-2, -2], [1, 0; 0, 16], N)';
R2(1:2, :) = mvnrnd([2, 2], [1, 0; 0, 16], N)';
R3(1:2, :) = mvnrnd([6, 6], [1, 0; 0, 16], N)';

R = [R1, R2, R3];

MinXY = min(min(min(R1(1, :), R1(2, :)), min(min(R2(1, :), R2(2, :)))));
MaxXY = max(max(max(R1(1, :), R1(2, :)), max(max(R2(1, :), R2(2, :)))));
MinXXYY = min(MinXY, min(min(R3(1, :), R3(2, :))));
MaxXXYY = max(MaxXY, max(max(R3(1, :), R3(2, :))));
X = MinXXYY:0.1:MaxXXYY;
Y = MinXXYY:0.1:MaxXXYY;

```

```

plot(R1(1, :)', R1(2, :)', 'or', 'MarkerSize', 2);
plot(R2(1, :)', R2(2, :)', 'og', 'MarkerSize', 2);
plot(R3(1, :)', R3(2, :)', 'ob', 'MarkerSize', 2);
xlabel('x1');
ylabel('x2');
axis([MinXXYY, MaxXXYY, MinXXYY, MaxXXYY]);

PG = zeros(1, 3);
PG(1, 1) = numel(R1(1, :)) / numel(R(1, :));
PG(1, 2) = numel(R2(1, :)) / numel(R(1, :));
PG(1, 3) = numel(R3(1, :)) / numel(R(1, :));

for it = 1:MaxIt
    hold on;
    for i = 1:3 * N
        for j = 1:3
            R(j+2, i) = PG(1, j) * det(Sigma(:, 2*j-1:2*j))^-0.5 * exp(-
0.5*(R(1:2, i) - mu(:, j))'*inv(Sigma(:, 2*j-1:2*j))*(R(1:2, i) - mu(:, j)));
        end
        R(6, i) = sum(R(3:5, i));
        for j = 1:3
            R(j+2, i) = R(j+2, i) / R(6, i);
        end
    end
    for i = 1:3
        Sigmahit(i) = sum(R(i+2, :));
    end
    for i = 1:3 * N
        for j = 1:3
            s(:, j) = (R(j+2, i) .* R(1:2, i));
        end
        for j = 1:3
            t(:, j) = s(:, j) + t(:, j);
        end
    end
    for i = 1:3
        milplus1(:, i) = t(:, i) ./ Sigmahit(i);
    end
    mu = milplus1;

    for i = 1:3 * N
        for j = 1:3
            ss(:, 2*j-1:2*j) = (R(j+2, i) .* ((R(1:2, i) - milplus1(:, j)) *
(R(1:2, i) - milplus1(:, j))'));
        end
        for j = 1:3
            tt(:, 2*j-1:2*j) = ss(:, 2*j-1:2*j) + tt(:, 2*j-1:2*j);
        end
    end
    for i = 1:3
        silplus1(:, 2*i-1:2*i) = tt(:, 2*i-1:2*i) ./ Sigmahit(i);
    end
    Sigma = silplus1;

    PG(1, 1) = Sigmahit(1) / numel(R(1, :));

```

```

PG(1, 2) = Sigmahit(2) / numel(R(1, :));
PG(1, 3) = Sigmahit(3) / numel(R(1, :));

f1 = zeros(size(X, 1), size(X, 1));
f2 = zeros(size(X, 1), size(X, 1));
f3 = zeros(size(X, 1), size(X, 1));

i = 1;
for y = MinXXYY:0.1:MaxXXYY
    j = 1;
    for x = MinXXYY:0.1:MaxXXYY
        f1(i, j) = mvar(x, y, mu(1, 1), mu(2, 1), Sigma(:, 1:2));
        f2(i, j) = mvar(x, y, mu(1, 2), mu(2, 2), Sigma(:, 3:4));
        f3(i, j) = mvar(x, y, mu(1, 3), mu(2, 3), Sigma(:, 5:6));
        j = j + 1;
    end;
    i = i + 1;
end;

contour(MinXXYY:0.1:MaxXXYY, MinXXYY:0.1:MaxXXYY, f1, 8, 'LineWidth', 3);
contour(MinXXYY:0.1:MaxXXYY, MinXXYY:0.1:MaxXXYY, f2, 8, 'LineWidth', 3);
contour(MinXXYY:0.1:MaxXXYY, MinXXYY:0.1:MaxXXYY, f3, 8, 'LineWidth', 3);

plot(R1(1, :)', R1(2, :)', 'or', 'MarkerSize', 2);
plot(R2(1, :)', R2(2, :)', 'og', 'MarkerSize', 2);
plot(R3(1, :)', R3(2, :)', 'ob', 'MarkerSize', 2);

t(:) = 0;
tt(:) = 0;

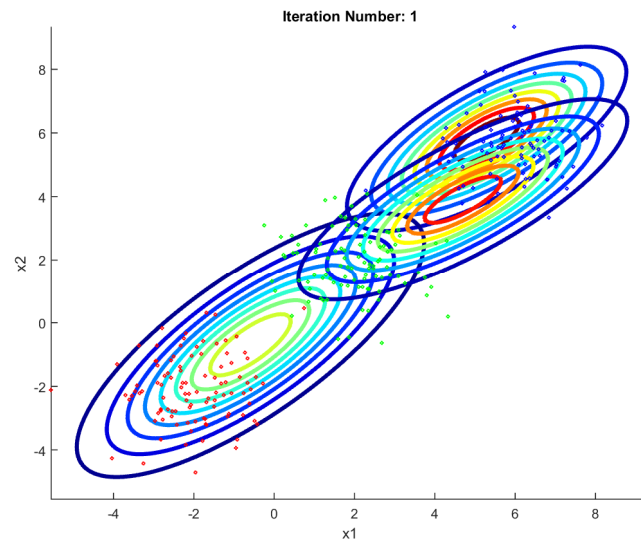
title(['Iteration Number: ', num2str(it)]);
if saveflag
    print(['Frame ', num2str(it, '%04d')], '-dpng', '-r150');
end
pause(0.001);
if it < MaxIt
    clf;
end
end

if saveflag
    GifName = 'EM.gif';
    delay = 0.5;
    for i = 1:MaxIt
        [AA, ~] = imread(['Frame ', num2str(i, '%04d'), '.png']);
        [X, map] = rgb2ind(AA, 256);
        if i == 1
            imwrite(X, map, GifName, 'gif', 'LoopCount', inf, 'DelayTime',
delay)
        else
            imwrite(X, map, GifName, 'gif', 'WriteMode', 'append',
'DelayTime', delay)
        end
    end
end
end

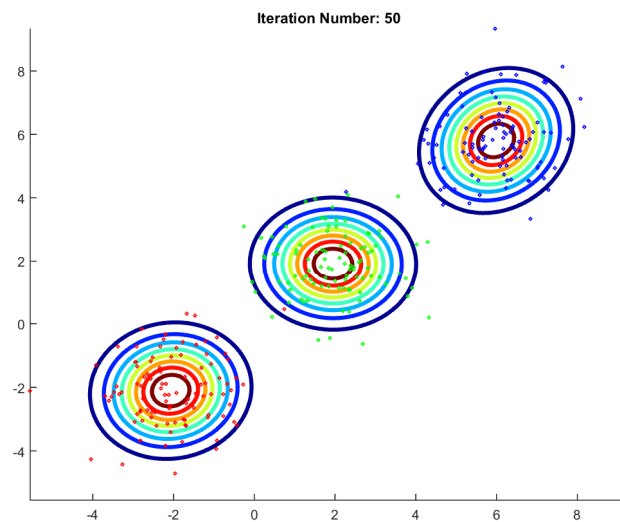
```

اجرای برنامه با داده‌های مشخص شده در نمونه اول:

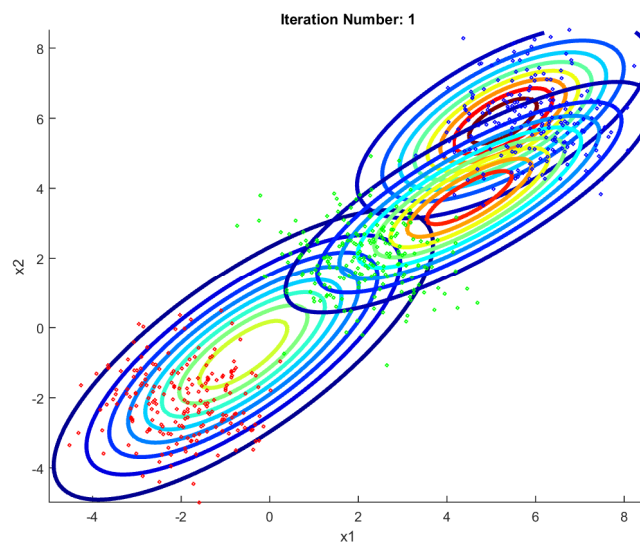
```
R1(1:2, :) = mvnrnd([-2, -2], [1, 0; 0, 1], N)';  
R2(1:2, :) = mvnrnd([2, 2], [1, 0; 0, 1], N)';  
R3(1:2, :) = mvnrnd([6, 6], [1, 0; 0, 1], N)';
```



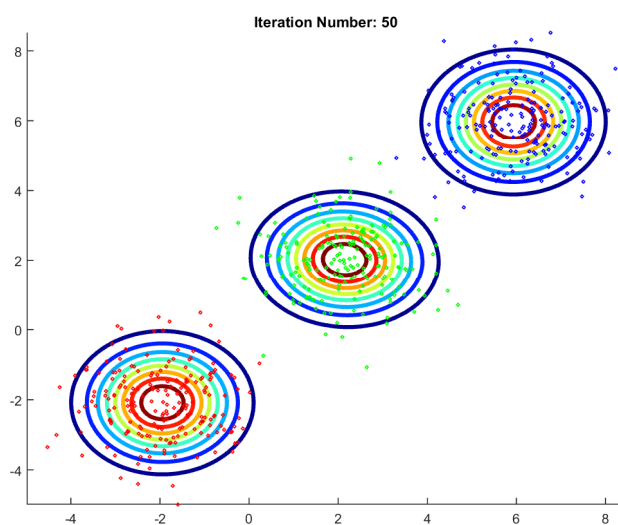
تکرار ۱-ام - تعداد داده‌ها ۱۰۰



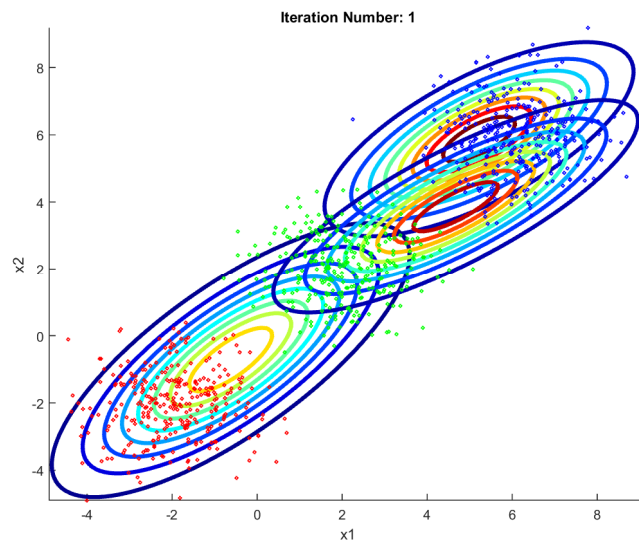
تکرار ۵۰-ام - تعداد داده‌ها ۱۰۰



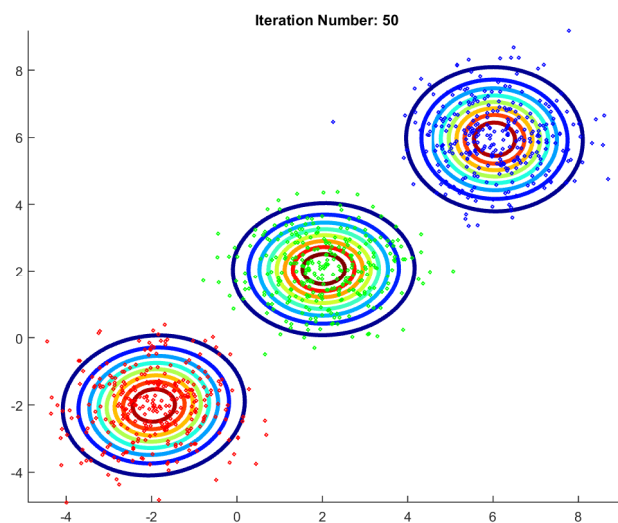
تکرار ۱-ام - تعداد داده‌ها ۲۰۰



تکرار ۵۰-ام - تعداد داده‌ها ۲۰۰



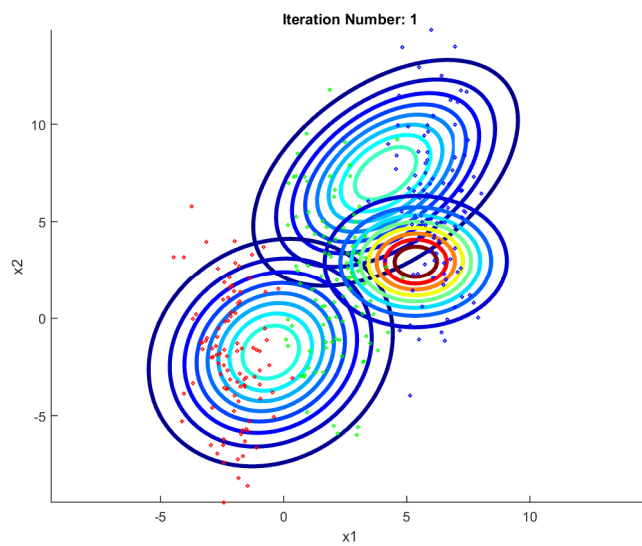
تکرار ۱-ام - تعداد داده‌ها ۳۰۰



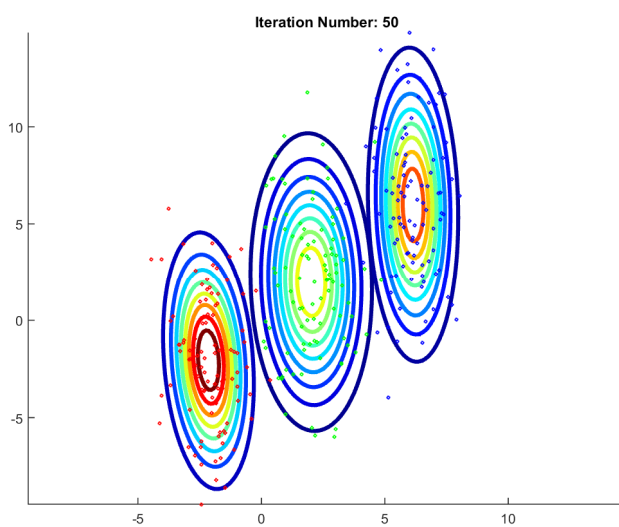
تکرار ۵۰-ام - تعداد داده‌ها ۳۰۰

اجرای برنامه با داده‌های مشخص شده در نمونه دوم:

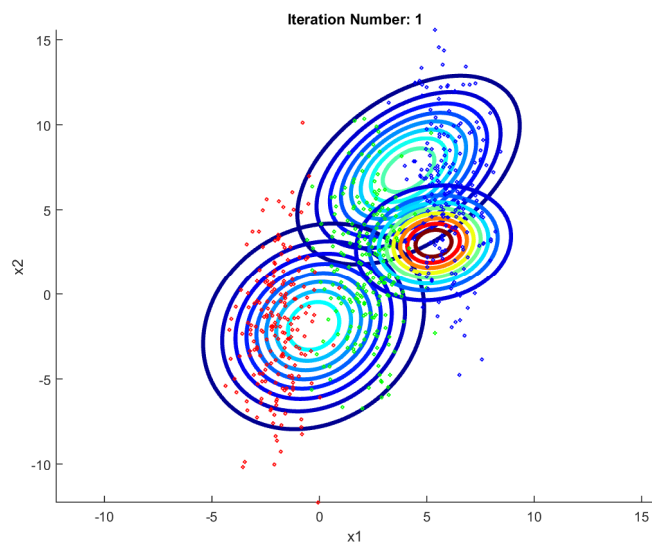
```
R1(1:2, :) = mvnrnd([-2, -2], [1, 0; 0, 16], N)';
R2(1:2, :) = mvnrnd([2, 2], [1, 0; 0, 16], N)';
R3(1:2, :) = mvnrnd([6, 6], [1, 0; 0, 16], N)';
```



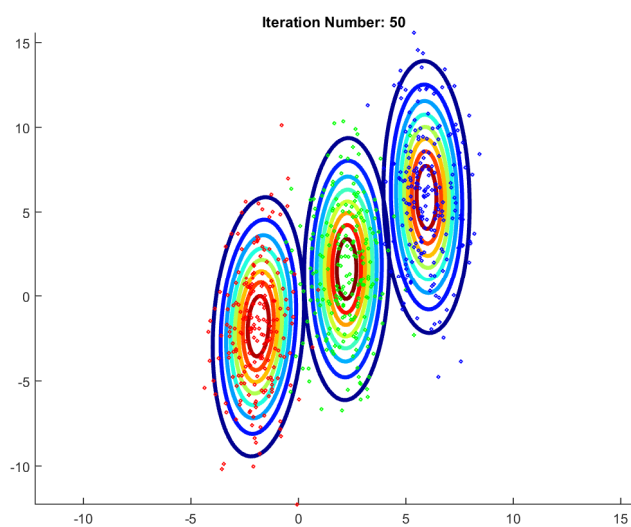
تکرار ۱-ام - تعداد داده‌ها ۱۰۰



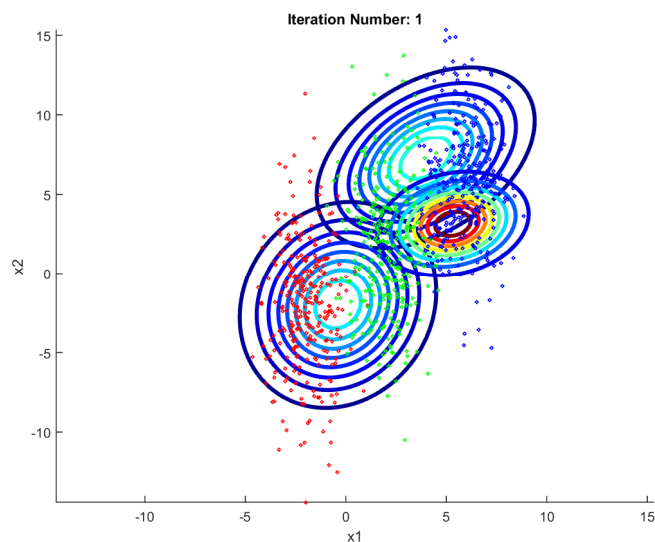
تکرار ۵۰-ام - تعداد داده‌ها ۱۰۰



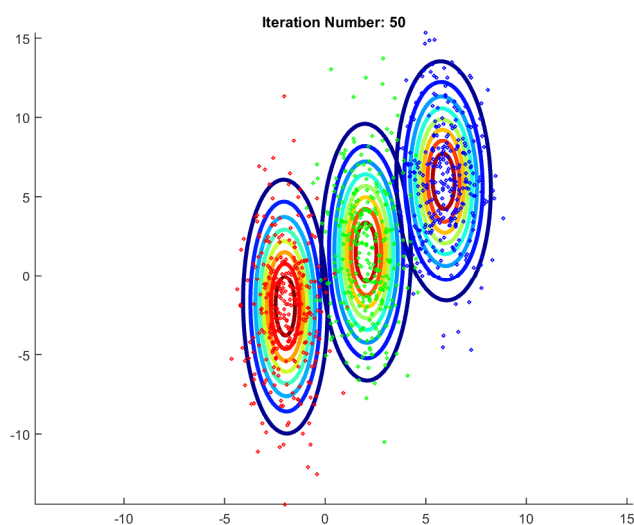
تکرار ۱-ام - تعداد داده‌ها ۲۰۰



تکرار ۵۰-ام - تعداد داده‌ها ۲۰۰



تکرار ۱-ام - تعداد داده‌ها ۳۰۰



تکرار ۵۰-ام - تعداد داده‌ها ۳۰۰

توضیحات برنامه:

ابتدا مقادیر اولیه میانگین و سیگمای ۰ الی ۳ به برنامه داده می‌شوند.

$$\mu_1^{(0)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \Sigma_1^{(0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mu_2^{(0)} = \begin{bmatrix} 2 \\ 0.4 \end{bmatrix}, \Sigma_2^{(0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mu_3^{(0)} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \Sigma_3^{(0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

سه دسته داده به تعداد ۱۰۰، ۲۰۰ و یا ۳۰۰ عدد در هر دسته با توجه به موارد مطرح‌شده در تکلیف، توسط دستور

mvnrnd در فضای دو بُعدی با میانگین و سیگمای دلخواه تولید می‌شوند.

سپس با استفاده از مقادیر اولیه میانگین و سیگمای ۰ الی ۳ و بهره‌گیری از داده‌های رندم تولید شده و بهره‌گیری

از رابطه زیر h_i^t محاسبه شده و پس از آن m_i^{l+1} و S_i^{l+1} محاسبه خواهند شد.

گفتنی است که در تکرار اول مقدار π_i از تقسیم تعداد هر خوشه بر تعداد کل شناسایی می‌گردد که در این حالت احتمال تمام کلاس‌ها برابر ۰/۳۳ است. پس از به‌روزرسانی m_i^{l+1} و S_i^{l+1} ، π_i نیز به‌روزرسانی می‌گردد تا در محاسبات بعدی مورد استفاده قرار گیرد.

$$h_i^t = \frac{\pi_i |S_i|^{-1/2} \exp\left[-(1/2)(x^t - m_i)^T S_i^{-1} (x^t - m_i)\right]}{\sum_j \pi_j |S_j|^{-1/2} \exp\left[-(1/2)(x^t - m_j)^T S_j^{-1} (x^t - m_j)\right]}$$

$$m_i^{l+1} = \frac{\sum_t h_i^t x^t}{\sum_t h_i^t}$$

$$S_i^{l+1} = \frac{\sum_t h_i^t (x^t - m_i^{l+1})(x^t - m_i^{l+1})^T}{\sum_t h_i^t}$$

$$\pi_i = \frac{\sum h_i^t}{N}$$

کد اصلی برنامه که محاسبات مربوط به این روابط را انجام می‌دهد مجدداً در ذیل آورده شده است:

```
for i = 1:3 * N
    for j = 1:3
        R(j+2, i) = PG(1, j) * det(Sigma(:, 2*j-1:2*j))^-0.5 * exp(-
0.5*(R(1:2, i) - mu(:, j))'*inv(Sigma(:, 2*j-1:2*j))*(R(1:2, i) - mu(:, j)));
    end
    R(6, i) = sum(R(3:5, i));
    for j = 1:3
        R(j+2, i) = R(j+2, i) / R(6, i);
    end
end
for i = 1:3
    Sigmahit(i) = sum(R(i+2, :));
end
for i = 1:3 * N
    for j = 1:3
        s(:, j) = (R(j+2, i) .* R(1:2, i));
    end
    for j = 1:3
        t(:, j) = s(:, j) + t(:, j);
    end
end
for i = 1:3
    milplus1(:, i) = t(:, i) ./ Sigmahit(i);
end
mu = milplus1;

for i = 1:3 * N
    for j = 1:3
        ss(:, 2*j-1:2*j) = (R(j+2, i) .* ((R(1:2, i) - milplus1(:, j)) *
(R(1:2, i) - milplus1(:, j))'));
    end
    for j = 1:3
        tt(:, 2*j-1:2*j) = ss(:, 2*j-1:2*j) + tt(:, 2*j-1:2*j);
    end
end
```

```

end
end
for i = 1:3
    silplus1(:, 2*i-1:2*i) = tt(:, 2*i-1:2*i) ./ Sigmahit(i);
end
Sigma = silplus1;

PG(1, 1) = Sigmahit(1) / numel(R(1, :));
PG(1, 2) = Sigmahit(2) / numel(R(1, :));
PG(1, 3) = Sigmahit(3) / numel(R(1, :));

```

پس از محاسبه تمامی موارد ذکرشده، تعدادی داده جدید توسط تابع mvar با توجه به میانگین‌ها و سیگماهای جدید محاسبه‌شده، تولید گردیده و فقط نمودارهای هم‌تراز این داده‌ها ترسیم می‌گردند؛ بنابراین، در هر تکرار از برنامه نمودارهای تراز ترسیم‌شده نیز با توجه به مقادیر میانگین‌ها و سیگماهای به‌روزرسانی می‌شوند. در انتهای فرایند (در صورتی که saveflag مقدار ۱ داشته باشد) تمامی ۵۰ فریم تصاویر ذخیره‌شده بر روی دیسک، در قالب یک فایل gif پویانمایی‌ای ذخیره خواهند شد.