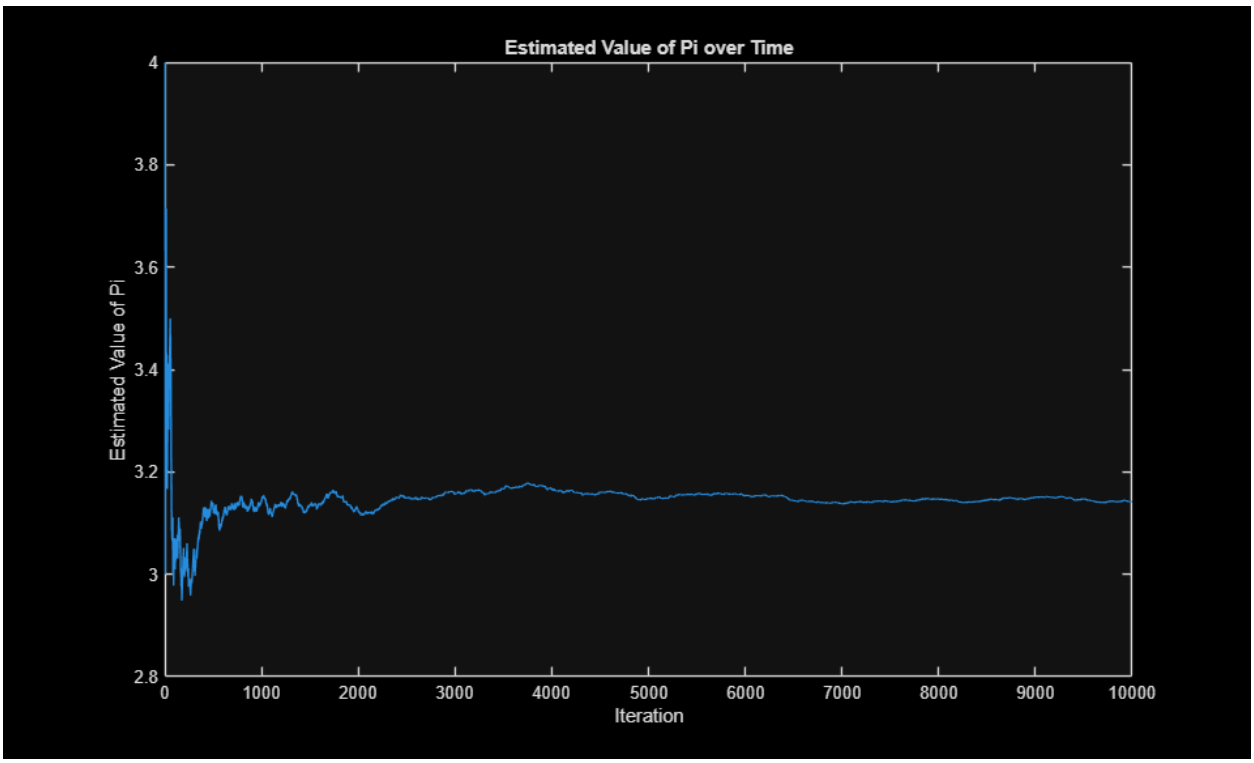
```
%{
Estimate pi based on a fixed number of random points and plot
the estimated value of pi as iterations increase and the difference
between the estimated and actual value of pi as iterations increase
%}

inside = 0;
num_points = 10000;
pis = [];

for i = 1:num_points
    x = rand;
    y = rand;
    r = sqrt(x^2 + y^2);
    % Determine if the randomly generated point is inside the circle
    if r <= 1
        inside = inside + 1;
    end
    pis(end+1) = 4*(inside / i);
end

diffs = pis - pi;

est_pi = 4*(inside / num_points);

figure;
plot(pis);
xlabel('Iteration');
ylabel('Estimated Value of Pi');
title('Estimated Value of Pi over Time');

figure;
plot(diffs);
xlabel('Iteration');
ylabel('Precision');
title('Difference Between Estimated and Actual Value of Pi over Time');

%{
Use different amounts of points and measure the computation time
and precision for each and plot these against each other
%}

points_array = [100, 1000, 10000, 100000, 1000000, 10000000];
new_pis = [];
times = [];

for points = points_array
    inside = 0;
    tic;
    for i = 1:points
        x = rand;
        y = rand;
```

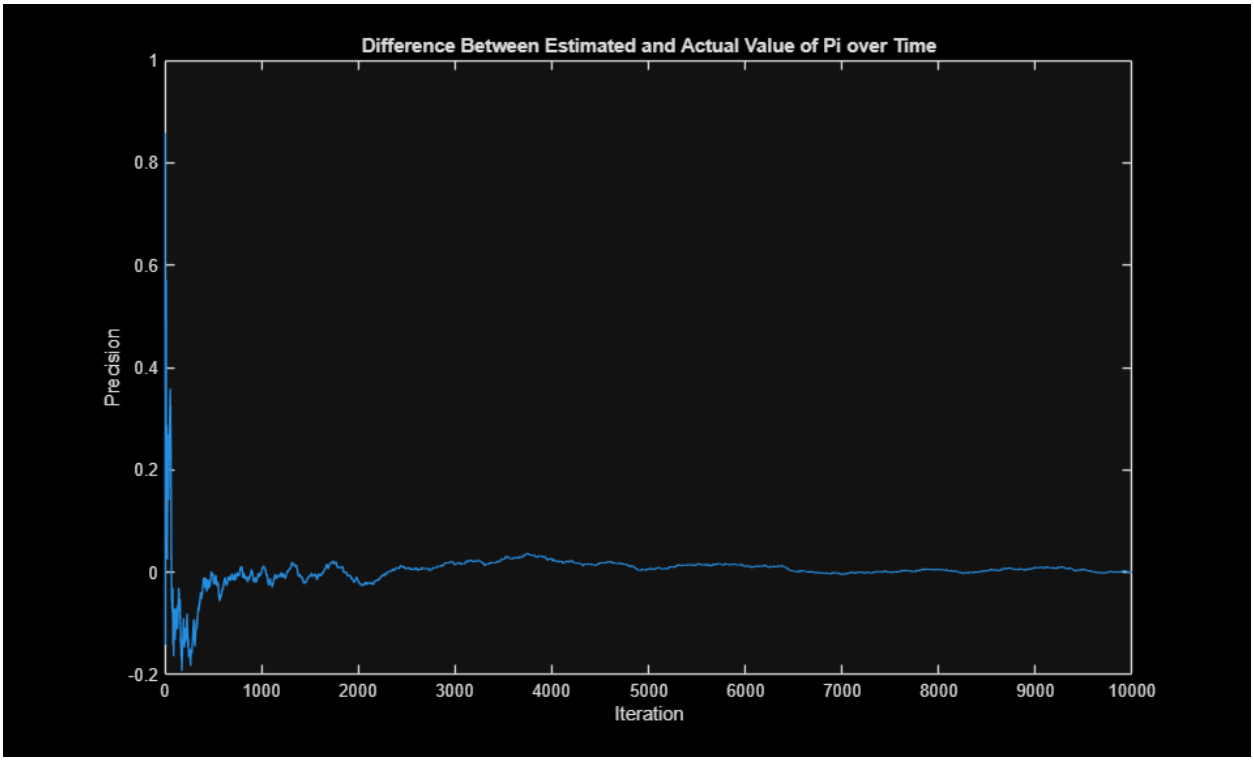
```
        r = sqrt(x^2 + y^2);
        % Determine if the randomly generated point is inside the circle
        if r <= 1
            inside = inside + 1;
        end
    end
    new_pis(end+1) = 4 * (inside / points);
    times(end+1) = toc;
end

new_diffs = new_pis - pi;

figure;
scatter(times, new_diffs, 'filled');
set(gca, 'XScale', 'log');
xlabel('Execution Time (log scale)');
ylabel('Precision');
title('Precision vs. Execution Time');

%{
Estimate pi to a specified number of significant figures
without using the true value of pi
%}

sigfigs = 3;
inside = 0;
total = 0;
prev = 0;
est_pi = 0;
stable_count = 0;
stable_required = 100;
batch_size = 100;
tol = .05 * 10^(-sigfigs);

while true
    x = rand(batch_size, 1);
    y = rand(batch_size, 1);
    % Determine which points in the batch are inside the circle
    inside = inside + sum(x.^2 + y.^2 <= 1);
    total = total + batch_size;
    est_pi = 4 * (inside / total);
    % Check the stability of the estimate
    if abs(est_pi - prev) < tol
        stable_count = stable_count + 1;
    else
        stable_count = 0;
    end
    % Stop if desired stability reached
    if stable_count >= stable_required
        est_pi = round(est_pi, sigfigs-1);
        break;
    end
    prev = est_pi;
```

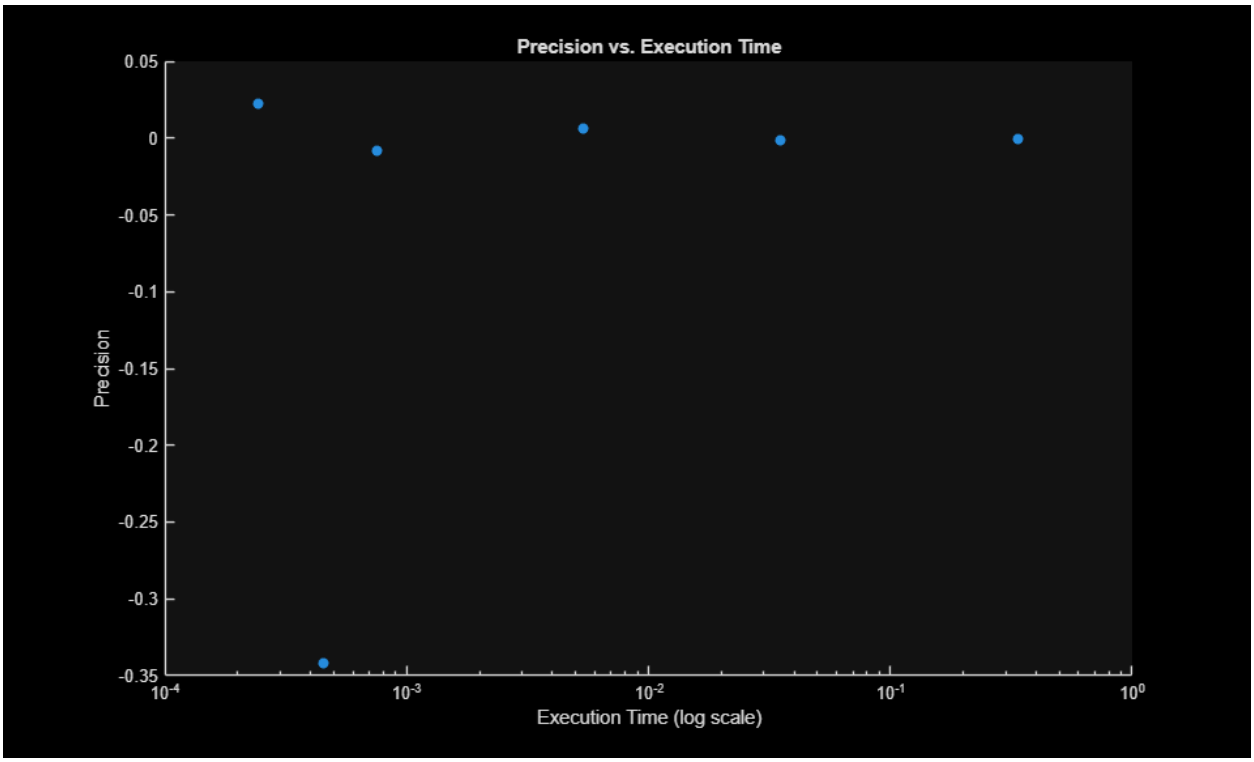
```
end

disp(est_pi);
disp(total);

%{
Estimate pi to a specified number of significant figures
without using the true value of pi using a function that
takes precision as input, plots points as they are generated,
displays the final value of pi, and returns the estimated
value of pi
%}

function estimate = estimate_pi(sigfigs)

    sigfigs = 2;
    inside = 0;
    total = 0;
    prev = 0;
    est_pi = 0;
    stable_count = 0;
    stable_required = 3;
    batch_size = 10;
    tol = .05 * 10^(-sigfigs);

    figure;
    hold on;
    axis equal;
    axis([0 1 0 1]);
    title('Estimating pi to User Defined Precision');
    xlabel('x');
    ylabel('y');
    theta = linspace(0, pi/2, 200);
    plot(cos(theta), sin(theta));

    while true
        x = rand(batch_size, 1);
        y = rand(batch_size, 1);
        r2 = x.^2 + y.^2;
        % Determine which points in the batch are inside the circle
        inside_index = r2 <= 1;
        % Plot points in the current batch
        plot(x(inside_index), y(inside_index), 'b.');
        plot(x(~inside_index), y(~inside_index), 'r.');
        inside = inside + sum(inside_index);
        total = total + batch_size;
        est_pi = 4 * (inside / total);
        % Check the stability of the estimate
        if abs(est_pi - prev) < tol
            stable_count = stable_count + 1;
        else
            stable_count = 0;
        end
```

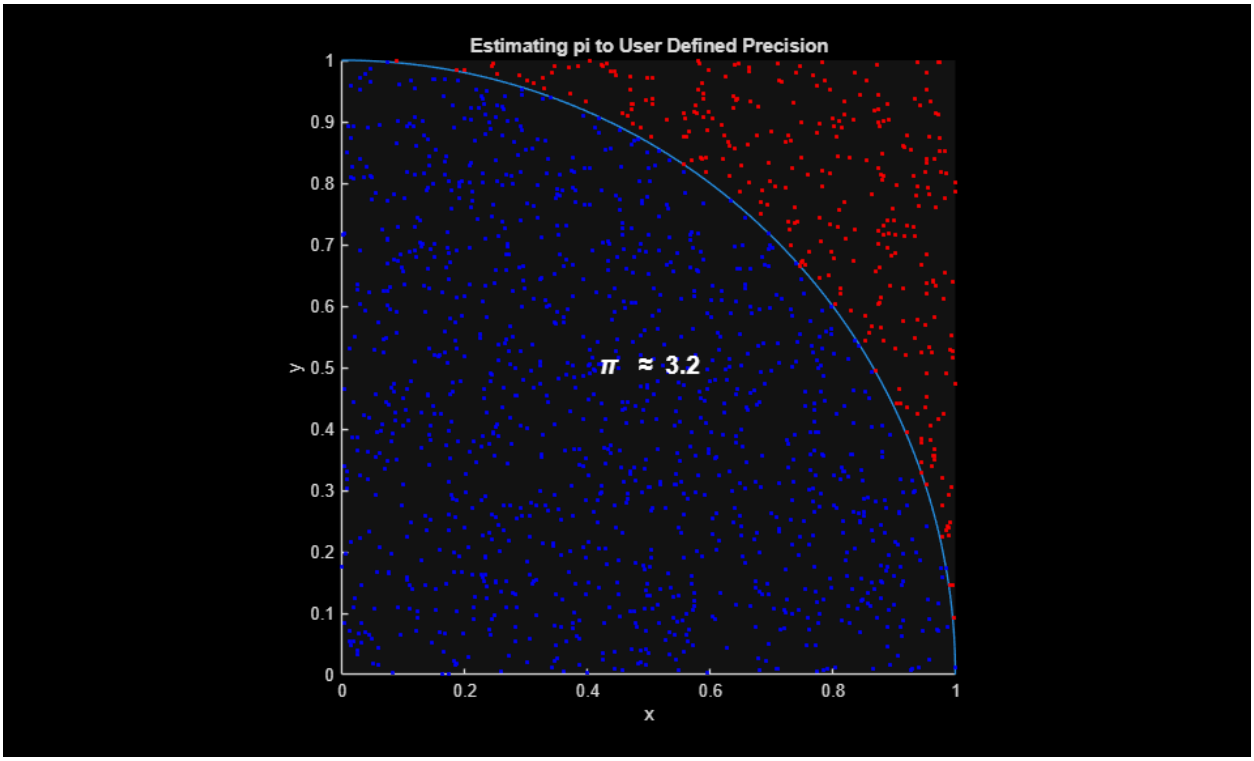
```
        % Stop if desired stability reached
        if stable_count >= stable_required
            est_pi = round(est_pi, sigfigs-1);
            break;
        end
        prev = est_pi;
        drawnow;
    end

    text(0.5, 0.5, sprintf('\\pi \\approx %.*g', sigfigs, est_pi), ...
        'FontSize', 14, 'FontWeight', 'bold', 'Color', 'w', ...
        'HorizontalAlignment', 'center');
    estimate = est_pi;

end

disp(estimate_pi(3));
```

```
    3.1400

      702900

    3.2000
```




Difference Between Estimated and Actual Value of Pi over Time
Precision
Iteration
1
0.8
0.6
0.4
0.2
0
-0.2
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Precision vs. Execution Time
Precision
Execution Time (log scale)
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
10^-4
10^-3
10^-2
10^-1
10^0



Published with MATLAB® R2025a