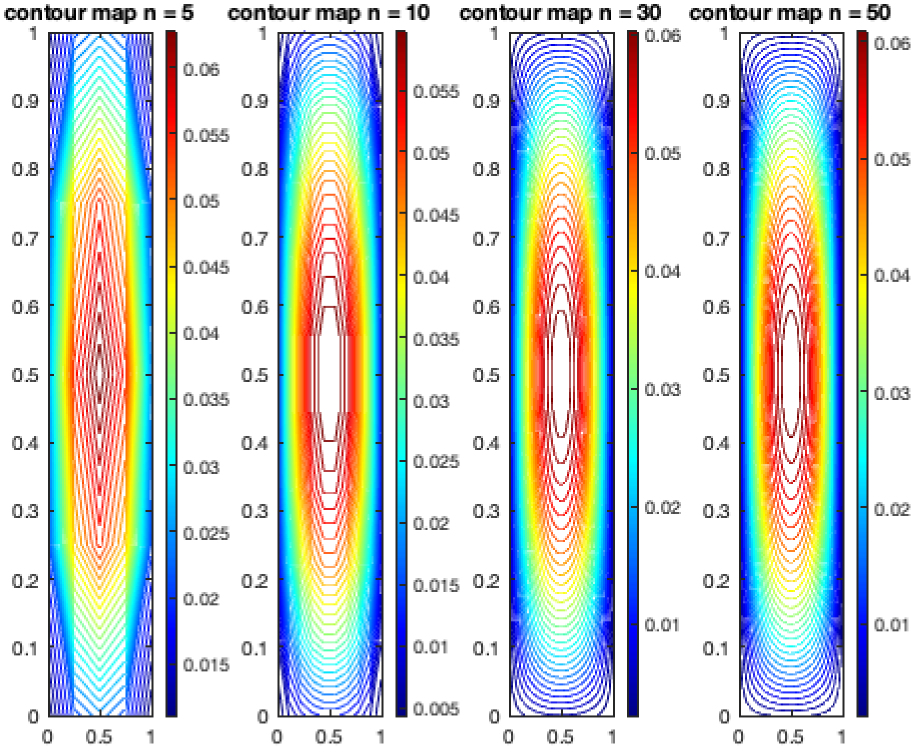
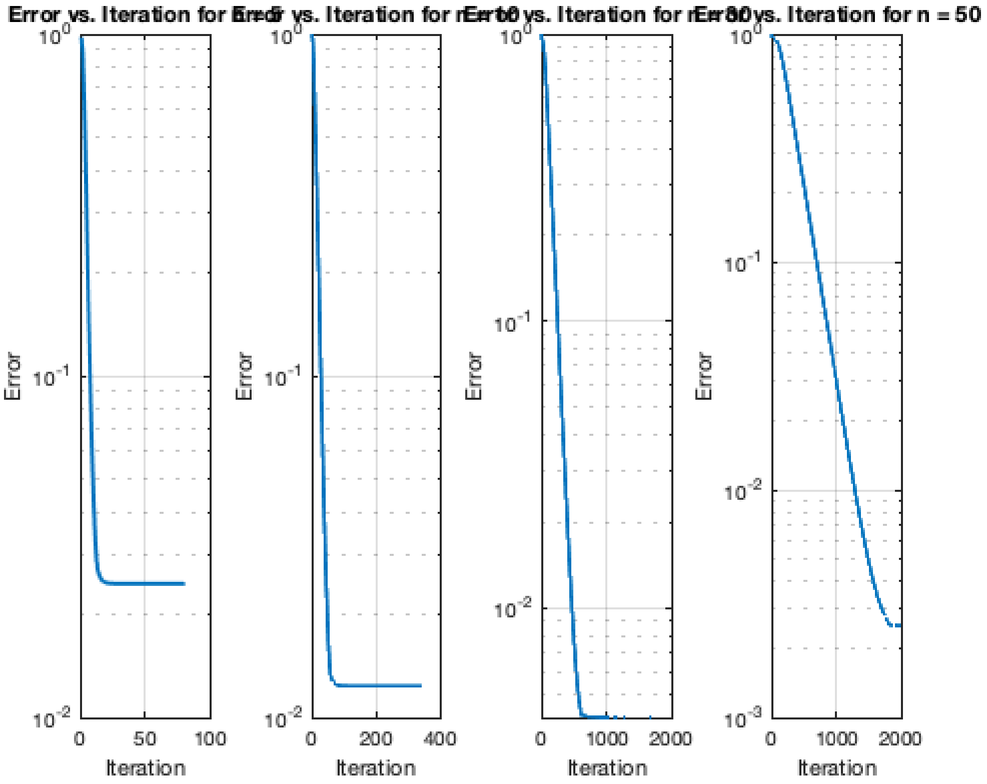
```
% Credit: Received help from Charlie, Christine

clc;
close all;
clear all;

% param
A = 2;
r = 5;
maxiter = 2000;
tol = 1e-15;
n_list = [5, 10, 30, 50];

figure;
for i = 1:length(n_list)
    n = n_list(i);
    subplot(1, length(n_list), i);
    [X, Y] = meshgrid(linspace(0, 1, n), linspace(0, 1, n));
    U = gs_solve(n, A, r, maxiter, tol);
    contour(X, Y, U(2:end-1, 2:end-1), 30);
    title(sprintf('contour map n = %d', n));
    colormap('jet');
    colorbar;
end
[U, err] = gs_solve(n, A, r, maxiter, tol);

% err
figure;
for i = 1:length(n_list)
    n = n_list(i);
    subplot(1, length(n_list), i);
    [U, err] = gs_solve(n, A, r, maxiter, tol);
    iter = 1:length(err);
    semilogy(iter, err, 'LineWidth', 2);
    title(sprintf('Error vs. Iteration for n = %d', n));
    xlabel('Iteration');
    ylabel('Error');
    grid on;
end

function result = mu(x, A, r)
    result = 1 + A * exp(-(x - 0.5).^2 / (2 * r^2));
end

function result = u(x, y)
    result = (x .* y) .* (1 - x) .* (1 - y);
end

function result = f(x, y, A, r)
    fx = 1/2 * (y-1) .* y .* ((A.*exp(-(1-2.*x).^2./(8*r^2)).*(4*r^2 - (1-2.*x).^2) ./r^2)+ 4);
    fy = -2 * (1-x) .* x .*(A.*exp(-(1-2.*x).^2./(8*r^2)) +1);
    result = -fx-fy;
end

function [U, err] = gs_solve(n, A, r, maxiter, tol)
    h = 1/n;
    x = linspace(0, 1, n+1); x = x(1:end-1);
    [X, Y] = meshgrid(x, x);
    err = [];
    Utrue = u(X, Y);
    U = zeros(n+2, n+2);
    F = padarray(f(X+ h/2, Y+ h/2, A, r)*h^2, [1, 1], 'both');
    U_prev = ones(size(U));

    for m = 1:maxiter
        for j = 2:n+1
            for k = 2:n+1
                a_ii = (mu(j*h, A, r) + mu((j-1)*h, A, r) + 2*mu((j-1/2)*h, A, r));
                b_i = F(j, k);
                sum_new = -U_prev(j+1, k) * mu(j*h, A, r) - U_prev(j, k+1) * mu((j-0.5)*h, A, r);
                sum_old = -U(j, k-1) * mu((j-0.5)*h, A, r) - U(j-1, k) * mu((j-1)*h, A, r);
                U(j, k) = (b_i - sum_new - sum_old) / a_ii;
            end
        end

        U(1, :) = -U(2, :);
        U(:, 1) = -U(:, 2);
        U(end, :) = -U(end-1, :);
        U(:, end) = -U(:, end-1);

        U_conc = U(2:end-1, 2:end-1);
        cur_err = max(max(abs(U_conc - Utrue)));
        err = [err; cur_err];

        if norm(U_prev - U, 'fro') < tol
            fprintf('Converge after %d iterations\n', m);
            break;
        end
        U_prev = U;
    end
end
```

```
Converge after 81 iterations
Converge after 343 iterations
Converge after 81 iterations
Converge after 343 iterations
```





Published with MATLAB® R2021a