```
% relax - Program to solve the Laplace equation using
% Jacobi, Gauss-Seidel and SOR methods on a square grid
clear all; help relax;  % Clear memory and print header

%* Initialize parameters (system size, grid spacing, etc.)
method = 3;
N = 60;
L = 1;          % System size (length)
h = L/(N-1);    % Grid spacing
x = (0:N-1)*h;  % x coordinate
y = (0:N-1)*h;  % y coordinate

%* Select over-relaxation factor (SOR only)
if( method == 3 )
  omegaOpt = 2/(1+sin(pi/N));  % Theoretical optimum
  fprintf('Theoretical optimum omega = %g \n',omegaOpt);
  omega = input('Enter desired omega: ');
end

% cage
cage = ones(N, N); % matrix to modify
choice = menu('Cage', 'None', 'a', 'b', 'c');
switch choice
 case 2
  cage(20, 20) = 0;
  cage(30, 20) = 0;
  cage(40, 20) = 0;
  cage(20, 30) = 0;
  cage(20 ,40) = 0;
  cage(30, 40) = 0;
  cage(40, 30) = 0;
  cage(40, 40) = 0;
 case 3
  cage(20, 20) = 0;
  cage(20, 40) = 0;
  cage(40, 20) = 0;
  cage(40, 40) = 0;
 case 4
  cage(20, 30) = 0;
  cage(30, 20) = 0;
  cage(40, 30) = 0;
  cage(30, 40) = 0;
end

% set up potential
phi = zeros(N, N);
phi(:, 1) = zeros(N, 1);
phi(:, end) = 100 .* ones(N, 1);
phi(1, :) = linspace(0, 100, N);
phi(end, :) = linspace(0, 100, N);

%* Loop until desired fractional change per iteration is obtained
```

```
newphi = phi;           % Copy of the solution (used only by Jacobi)
iterMax = N^2;          % Set max to avoid excessively long runs
changeDesired = 1e-4;   % Stop when the change is given fraction
fprintf('Desired fractional change = %g\n',changeDesired);
for iter=1:iterMax
  changeSum = 0;

    for i=2:(N-1)        % Loop over interior points only
     for j=2:(N-1)
       newphi = 0.25*omega*(phi(i+1,j)+phi(i-1,j)+ ...
               phi(i,j-1)+phi(i,j+1))  +  (1-omega)*phi(i,j);
       changeSum = changeSum + abs(1-phi(i,j)/newphi);
       phi(i,j) = newphi;
     end
    end

    phi = cage .* phi;

  %* Check if fractional change is small enough to halt the iteration
  change(iter) = changeSum/(N-2)^2;
  if( rem(iter,10) < 1 )
    fprintf('After %g iterations, fractional change = %g\n',...
                             iter,change(iter));
  end
  if( change(iter) < changeDesired )
    fprintf('Desired accuracy achieved after %g iterations\n',iter);
 fprintf('Breaking out of main loop\n');
    break;
  end
end

%* Plot final estimate of potential as contour and surface plots
figure(1); clf;
cLevels = 0:(0.1):1;    % Contour levels
cs = contour(x,y,flipud(rot90(phi)),cLevels);
xlabel('x'); ylabel('y'); clabel(cs);
title(sprintf('Potential after %g iterations',iter));
figure(2); clf;
mesh(x,y,flipud(rot90(phi)));
xlabel('x'); ylabel('y'); zlabel('\Phi(x,y)');

%* Plot the fractional change versus iteration
figure(3); clf;
semilogy(change);
xlabel('Iteration');  ylabel('Fractional change');
title(sprintf('Number of flops = %g\n',flops));
```

```
  relax - Program to solve the Laplace equation using
  Jacobi, Gauss-Seidel and SOR methods on a square grid

Theoretical optimum omega = 1.90053

Error using input
Cannot call INPUT from EVALC.
```

```
Error in relax (line 17)
  omega = input('Enter desired omega: ');
```

Published with MATLAB® R2019a