The voltage at an arbitrary point in space is given by

\[
V(\mathbf{r}) = \frac{Q}{4\pi \varepsilon_0 |\mathbf{r} - \mathbf{r}_Q|} + \sum_{\text{plate}} \frac{g(\mathbf{r}') \, ds'}{4\pi \varepsilon_0 |\mathbf{r} - \mathbf{r}'|}
\]

where \( \mathbf{r}_Q \) is the location of the point charge given by

\[
\mathbf{r}_Q = \frac{1}{2} \hat{x} + \frac{1}{2} \hat{y} + \frac{1}{2} \hat{z}.
\]

For observation points on the plate, the potential/voltage is zero. Let's have the observation points correspond to the center of each cell.

Assume there are \( N_{\text{side}} \) cells per side of the plate. Therefore the length of a side of a cell would be given by

\[
\Delta = \frac{1.0}{N_{\text{side}}} \quad \text{(total length of a side is 1.0 m)}
\]

For this assignment \( N_{\text{side}} = 20 \) so \( \Delta \) would be 0.05 m.

The center of the \( m \)th cell is given by

\[
\mathbf{r}_m = (i-\frac{1}{2})\Delta \hat{x} + (j-\frac{1}{2})\Delta \hat{y},
\]

where

\[
i = \text{int} \left( \frac{(m-1)}{N_{\text{side}}} \right) + 1
\]

\[
j = \text{mod} \left( (m-1), N_{\text{side}} \right) + 1
\]

Note that this is no different from the plate problem. Furthermore note that the \( \frac{1}{2} \)'s that appear in \( \mathbf{r}_m \) have nothing to do with the charge location. (These \( \frac{1}{2} \)'s just ensure we are at the center of the cell.)

Now if we discretize the plate with pulse basis function and make the observation point the center of one of the cells we can write

\[
V(\mathbf{r}_m) = \frac{Q}{4\pi \varepsilon_0 |\mathbf{r}_m - \mathbf{r}_Q|} + \sum_{n=1}^{N_{\text{side}}^2} a_n \int_{n_{\text{th cell}}} \frac{ds'}{4\pi \varepsilon_0 |\mathbf{r}_m - \mathbf{r}'|} Z_{mn} \quad \text{potential at } \mathbf{r}_m \text{ due to basis function } n. \text{ Same as plate problem,}
\]
Rearranging we obtain, after multiplying through by \( \epsilon_0 \):

\[
-\frac{Q}{|\mathbf{r}_m - \mathbf{r}_q|} = \sum_{n=1}^{N} a_n \int_{n^{th} \text{ cell}} \frac{ds'}{|\mathbf{r}_m - \mathbf{r}'|}
\]

We can write this in matrix-vector form as

\[
[V_m] = [Z_{mn}] [a_n]
\]

where

\[
V_m = \frac{-Q}{|\mathbf{r}_m - \mathbf{r}_q|} = \frac{-Q}{\left( [i - \frac{1}{2}, \frac{1}{2}]^2 + [j - \frac{1}{2}, \frac{1}{2}]^2 + [0, \frac{1}{2}]^2 \right)^{\frac{1}{2}}}
\]

and

\[
Z_{mn} = \begin{cases} 
4\Delta \ln(\sqrt{2}+1) & \text{if } m = n \\
\frac{\Delta}{\left( [i - i', \frac{1}{2}]^2 + [j - j', \frac{1}{2}]^2 \right)^{\frac{1}{2}}} & \text{if } m \neq n
\end{cases}
\]

Note that \( Z_{mn} \) is no different in form from that of plate problems. (But could differ by a factor of \( \alpha \epsilon_0 \) depending on what we do with that.)

Solution for charge-density coefficients \( a_n \) is given by

\[
[a_n] = [Z_{mn}]^{-1} [V_m]
\]

Matlab implementation is shown on following pages. Plots of the charge are shown on last three pages.
diary charge-over-plate.session

% Number of cells per side of the plate.
numPerSide = 20;

%%% Create the forcing vector.
V = makeV(1.e-9, numPerSide);

%%% An alternative implementation for creation of the Vm vector is
%%% also available called makeVfor() that uses a for-loop instead of
%%% using a vector-based implementation. makeVfor() shown on next page.

%%% Create the impedance matrix with 'numPerSide' segments
%%% per side of the plate. Time this to see how long it takes.
t0 = cputime;
z = makeZm(numPerSide);
disp(sprintf('Time to create matrix: %f', cputime - t0))

%%% Find the coefficients of the basis functions.
t0 = cputime;
an = z \backslash v;
disp(sprintf('Time to solve for coefficients: %f', cputime - t0))

%%% Plot coefficient of all basis functions.
figure(1)
plot(an)
title(['Fig. 1: Charge density vs. basis function number, \Delta=1\,', ...
      int2str(numPerSide)])
xlabel('Basis Function Number')
ylabel('Charge [Coulombs]')

%%% Create a 2D array of coefficients of basis function on
%%% plate and plot it using a surface plot.
plate = reshape(an, numPerSide, numPerSide);
del = 1.0 / numPerSide;  % length along a side of segment
x1 = del / 2;  % del : 1.0 - del / 2;  % x location of center of segments
y1 = del / 2;  % del : 1.0 - del / 2;  % y location of center of segments
figure(2)
surf(x1, y1, plate)
title(['Fig. 2: Charge density vs. position on plate, \Delta=1\,', ...
      int2str(numPerSide)])
xlabel('Displacement from edge')
ylabel('Charge [Coulombs]')

%%% Plot charge along the center of the plate.
figure(3)
plot(x1, plate(:, floor(numPerSide / 2) + 1))
title(['Fig. 3: Charge density vs. position along center of plate, \Delta=1\,', ...
       '\Delta=1\,', int2str(numPerSide)])
ylabel('Charge [Coulombs]')
function v = makeVfor(q, numPerSide)
% makeV Make the V vector for a square plate with a charge
% of q located 0.5 m above the plate.
% V = makeV(voltage, numPerSide) Returns the forcing vector V for a
% plate with numPerSide^2 basis functions. A charge of 'q' is
% located 0.5 m above the center of the plate. Here the origin
% is assumed to at the corner of the plate so that the charge
% is located at (0.5, 0.5, 0.5).
% This version uses a for-loop to construct the V vector.

numPerPlate = numPerSide^2;

del = length of the side of a cell.
del = 1 / numPerSide;

v = zeros(numPerPlate,1);
for m = 1 : numPerPlate
    % i and j indices for each cell.
    [i, j] = m2ij(m, numPerSide);
    % Subtract a half from indices to get center of cell. (The second
    % subtraction of 0.5 is the x or y location of the point charge.)
    delx = (i - 0.5) * del - 0.5;
    dely = (j - 0.5) * del - 0.5;

    % Forcing function is the negative of the charge divided by the
    % distance between the charge and the observation point.
    v(m) = -q / sqrt(delx^2 + dely^2 + 0.5^2);
end

return:

Many other ways of coding
[Vm] exist! I've just shown two.

function zmn = makeZmn(numPerSide)
% makeZmn Make the impedance matrix for a square plate.
% ZMN = makeZmn(numPerSide) Creates the impedance matrix for a
% square plate where each basis function is a square pulse. The
% area of each basis function is (1/numPerSide)^2, i.e., the plate
% is assumed to have unit area and is divided into numPerSide^2
% square segments.

del = 1 / numPerSide;

self = 2.0 * del * log((sqrt(2) + 1) / (sqrt(2) - 1));

zmn = zeros(numPerPlate);

% Since symmetric, only compute values for upper half of matrix.
% First, fill in diagonal terms, i.e., the 'self' terms.
for m = 1 : numPerPlate
    zmn(m, m) = self;
end

% Calculate off-diagonal terms in upper half of matrix.
for m = 1 : numPerPlate
    for n = m + 1 : numPerPlate
        % observation point
        [ip, jp] = m2ij(n, numPerSide);
        % source point
        zmn(m, n) = del / sqrt((i - ip)^2 + (j - jp)^2);
    end
end

% Fill in lower half -- things are symmetric since all elements the
% same size.
for m = 1 : numPerPlate
    for n = m + 1 : numPerPlate
        zmn(n, m) = zmn(m, n);
    end
end

return:
function [i, j] = m2ij(m, numPerSide)
% M2IJK Convert the global index to the local indices.
% [i, j] = m2ij(m, numPerSide) where m is the global index and
% numPerSide is the number of elements along one side of the
% plate.

j = floor((m - 1) / numPerSide) + 1;

i = mod(m - 1, numPerSide) + 1;

return;

Unchanged from plate problem.
Fig. 1: Charge density vs. basis function number, $\Delta = 1/20$
Fig. 2: Charge density vs. position on plate, $\Delta=1/20$
Fig. 3: Charge density vs. position along center of plate, $\Delta = 1/20$