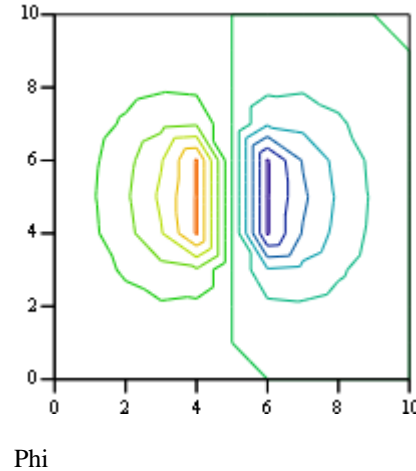


Laplace's equation: the potential between parallel plates

Laplace's equation describing the electric potential in two dimensions is: $\nabla^2\Phi(x, y) = 0$

At right is the potential around two parallel capacitor plates. In this exercise, we will learn how to find and plot the potential.

We will solve this using a two dimensional grid with values of $\Phi(x,y)$ at each grid point. It is a property of this equation that the value of Φ at each point is the average of the values of Φ at the points around it (the pair above and below and the pair to the left and right.) We can find the solution by making a guess and then averaging the points with the points around it until the answer converges. We will represent $\Phi(x,y)$ by a matrix of values $\Phi_{i,j} = \Phi_{\text{row,column}}$.



We will use a square array of points with 11 columns and rows:

$$\text{imax} := 10 \quad \text{jmax} := \text{imax}$$

The answer will be put in the array $\Phi_{i,j}$ which will be initialized to zero.

The subscripts will have the values: $i := 0.. \text{imax} \quad j := 0.. \text{jmax}$

We will assume that the spacing of points in the x direction is the same as the spacing in the y direction. If the spacing is different we would have to give different weightings to the values being averaged.

We will use a PROGRAM LOOP which will average each cell in the matrix Phi with the two cells to the left and right and the two cells above and below. The loop will do the averaging over and over again. This procedure will converge to the proper solution if we do it enough times. A PROGRAM LOOP returns a value for a variable. So we will make our matrix Phi the variable that is returned.

The LOOP is typed in using the Programming menu which can be turned on using the View|Toolbars menu. The black vertical lines indicate the extent of loops. We have put loops within loops so there are several nested vertical lines.

On the next page, the first pair of nested loops using variables i and j set the initial values of $\Phi_{i,j}$ equal to zero. Next there are three nested loops that solve the problem.

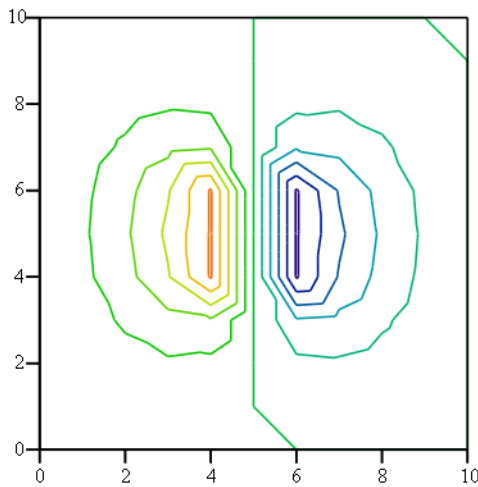
The outer loop does the averaging kmax times. The two inner loops put into temporary variable $\text{Avg}_{i,j}$ the average of the four cells around the i,j cell in the Phi matrix.

The boundary cells (rows 0 and imax, columns 0 and jmax) are not averaged because these boundary values are part of the input that defines the problem. The inner loops therefore start with 1 rather than 0 and end with imax -1 and jmax -1 rather than with imax and jmax.

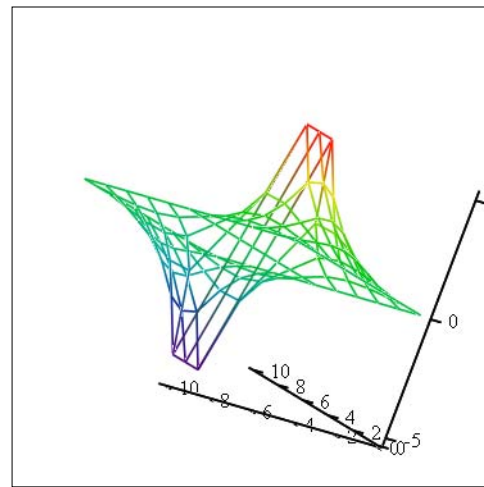
The next pair of nested loops puts the new values $\text{Avg}_{i,j}$ into the matrix $\Phi_{i,j}$. This unfortunately changes the values of the potentials on the capacitor plates which we want to hold fixed. These values should not have been averaged. So we add a last loop which restores the specified values to the capacitor plates.

On the left below is a contour plot. This type of plot can be selected from the graph menu. The potential values of 5.1 and -5.1 were used so that the "5" contour and the "-5" contour would pass close to the capacitor plates. Otherwise, we could not tell exactly where they are.

Plots of potential contours



Phi



Phi

You can click on the contour plot at right and rotate it. The axes are labelled with the grid point number. You can change this by clicking on the graph, then on axes, and filling in the limits.

Mathcad has a special functions "multgrid" and "relax" which will solve Laplace's equation. These solvers converge more quickly, but basically do the same thing we have done here.

Notes:

1. We initially loaded zeros into Phi as our initial guess. This is not a great guess. But the solver converges anyway. The values in the rows and columns at the boundaries are not averaged because this is a boundary value problem with fixed boundary conditions.
2. The averaging was done 40 times. Is that enough? One way to tell is to double the number of iterations. If they are within 1% of one another then the answer is probably good to 1%.

Being more clever with boundary conditions:

We could have a program that allowed us to put different shaped central objects without having to put new lines in the programming block. Let's put the values for our new object in a matrix called BC for boundary condition. We will set the unused members of this matrix equal to 1001. Then we will put into the program an "if" statement: if the value is greater than 1000, it is not averaged because it is a boundary value.

$BC_{i,j} := 1001$ This "works" for all i,j values. Test it with any i,j values: $BC_{3,4} = 1.001 \times 10^3$

Use the same capacitor plates, only turn them horizontally.

$BC_{4,4} := 5.1$ $BC_{5,4} := 5.1$ $BC_{6,4} := 5.1$ $BC_{4,6} := -5.1$ $BC_{5,6} := -5.1$ $BC_{6,6} := -5.1$

We average 60 times. $kmax := 60$

```
Phi := | i ← 0
        | for j ∈ 0 .. jmax
        |   for i ∈ 0 .. imax
        |     Phii,j ← if(BCi,j < 1000, BCi,j, 0)
        |     for k ∈ 0 .. kmax
        |       for i ∈ 1 .. imax - 1
        |         for j ∈ 1 .. jmax - 1
        |           Avgi,j ←  $\frac{\text{Phi}_{i-1,j} + \text{Phi}_{i+1,j} + \text{Phi}_{i,j-1} + \text{Phi}_{i,j+1}}{4}$ 
        |           for i ∈ 1 .. imax - 1
        |             for j ∈ 1 .. jmax - 1
        |               Phii,j ← if(BCi,j < 1000, BCi,j, Avgi,j)
        |     Phi
```

It is good practice to initialize a matrix.

The updated matrix value is set equal to the average of the surrounding values UNLESS it is a boundary value. If it is a boundary value, the original value is put into the matrix instead of the average.

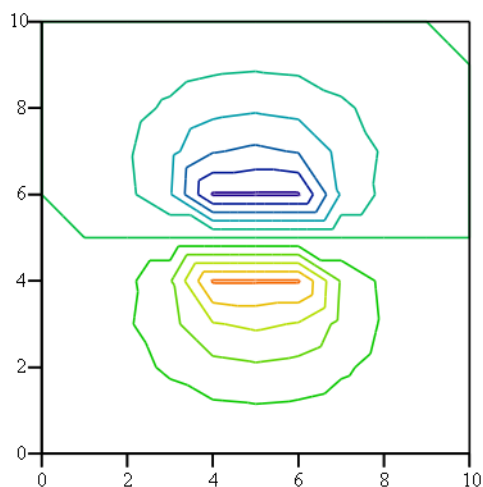
How "if" works: When the condition following the "if" is true, the first value (BC) is used, and if the condition is not true, the second value (Avg) is used.

Further reading:

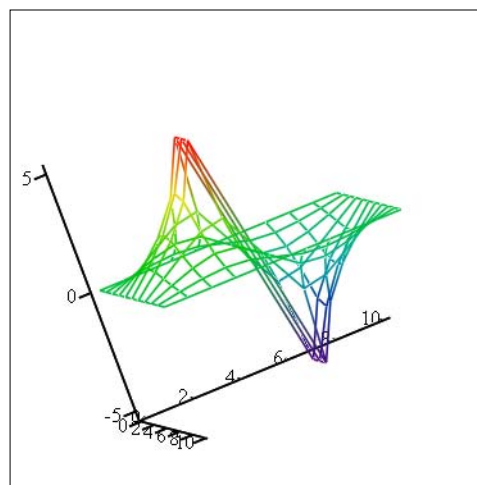
1. *Numerical Recipes, The art of scientific computing*, by Press, Flannery, Teukolsky and Vetterling (Cambridge Press, New York, 1985) Chapter 17.5.
2. "Solutions to Laplace's equation using a spreadsheet on a personal computer," T. T. Crow, *American Journal of Physics* 55, 817 (1987).

The plots are on the next page.

Plots of potential contours



Phi



Phi

We did the second problem with 60 iterations instead of 30. Are the answers the same as for 30? The two problems are the same except the capacitor plates are turned 90 degrees. Let's turn our second answer 90 degrees (a matrix transpose operation from the matrix menu) and compare to the first answer.A

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.163	0.351	0.571	0.774	0.85	0.774	0.571	0.351	0.163	0
2	0	0.3	0.672	1.158	1.676	1.85	1.676	1.158	0.672	0.3	0
3	0	0.365	0.878	1.713	2.922	3.198	2.922	1.713	0.878	0.365	0
4	0	0.282	0.764	1.894	5.1	5.1	5.1	1.894	0.764	0.282	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	-0.282	-0.764	-1.894	-5.1	-5.1	-5.1	-1.894	-0.764	-0.282	0
7	0	-0.365	-0.878	-1.713	-2.922	-3.198	-2.922	-1.713	-0.878	-0.365	0
8	0	-0.3	-0.672	-1.158	-1.676	-1.85	-1.676	-1.158	-0.672	-0.3	0
9	0	-0.163	-0.351	-0.571	-0.774	-0.85	-0.774	-0.571	-0.351	-0.163	0
10	0	0	0	0	0	0	0	0	0	0	0

There is a difference in the third decimal place for some of the values. So our answer is good to at least two decimal places.

Try it: Show that a smoother plot is obtained with a larger number of grid points.

Try it: Put a circular object in the center with a potential of +10. You could define a circular boundary of radius r using something like $BC_{ij} := 10$ if $(i-5)^2 + (j-5)^2 < r^2$ and 1001 otherwise.

Using the relax function

Mathcad has a built in function, relax, which performs the iterations. Relax has 8 arguments which you can see below. The arguments a,b,c and d are the weightings of the four neighboring values. These are each -1/4 if the weighting given to the central value is +1. To create an unchanging boundary, these weightings are set to zero at points that are on boundaries where the potential is specified. The initial guesses go in u and the boundary conditions go in f. The last argument rj is a value between 0 and 2 that determines the convergence of the process. Higher values cause faster convergence but are more likely to "crash." Use the largest value that doesn't cause the function to fail. We are using the boundary conditions for the initial guess when we set u = f.

$$a_{i,j} := \text{if}\left(\text{BC}_{i,j} > 1000, \frac{-1}{4}, 0\right) \quad b_{i,j} := a_{i,j} \quad c_{i,j} := a_{i,j} \quad d_{i,j} := a_{i,j}$$

$$e_{i,j} := 1 \quad f_{i,j} := \text{if}\left(\text{BC}_{i,j} < 1000, \text{BC}_{i,j}, 0\right) \quad u := f \quad rj := 0.99$$

$$S := \text{relax}(a, b, c, d, e, f, u, rj) \quad \text{Same result as using } \text{relax}(a, a, a, a, e, f, f, rj).$$

$$S^T =$$

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.163	0.351	0.571	0.774	0.85	0.774	0.571	0.351	0.163	0
2	0	0.3	0.672	1.158	1.676	1.85	1.676	1.158	0.672	0.3	0
3	0	0.365	0.879	1.713	2.922	3.198	2.922	1.713	0.879	0.365	0
4	0	0.282	0.764	1.894	5.1	5.1	5.1	1.894	0.764	0.282	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	-0.282	-0.764	-1.894	-5.1	-5.1	-5.1	-1.894	-0.764	-0.282	0
7	0	-0.365	-0.879	-1.713	-2.922	-3.198	-2.922	-1.713	-0.879	-0.365	0
8	0	-0.3	-0.672	-1.158	-1.676	-1.85	-1.676	-1.158	-0.672	-0.3	0
9	0	-0.163	-0.351	-0.571	-0.774	-0.85	-0.774	-0.571	-0.351	-0.163	0
10	0	0	0	0	0	0	0	0	0	0	0

Above the matrix transpose of S has been listed so it can be compared with the other listings. The transpose takes the horizontally oriented plates and makes them vertically oriented. Note that it agrees with the 60 iteration version and differs a little from the 30 iteration version. We did not have to tell the relax function how many iterations to use. The number of iterations is determined by the tolerance variable TOL which Mathcad sets to 0.001 if you do not specify a different value.

The equation below from the definition of the relax function will help you see why the values a,b,c,d,e and f have the values we have assigned to them:

$$e_{i,j} \text{Avg}_{i,j} := f_{i,j} - \left(a_{i,j} \text{Phi}_{i-1,j} + b_{i,j} \text{Phi}_{i+1,j} + c_{i,j} \text{Phi}_{i,j-1} + d_{i,j} \text{Phi}_{i,j+1} \right)$$

Our assignments make Avg equal to the average of the four adjacent values of Phi unless we are on an interior boundary point. In that case a,b,c and d are zero so the adjacent values are not averaged. f at an interior boundary is nonzero and is equal to boundary values. At these points the boundary value is placed in Avg.