

# Point Relaxation to Solve Laplace's Equation

Charles Kankelborg

March 2021

We have learned that if  $V(\mathbf{r})$  satisfies Laplace's equation,

$$\nabla^2 V = 0,$$

on some volume  $\mathcal{V}$ , then the average value of  $V$  over the surface of a sphere of any radius, so long as it is within  $\mathcal{V}$ , is equal to the value of  $V$  at the center of the sphere.

Conceptually, then, it may not surprise you to learn that it is possible to calculate a solution to Laplace's equation as follows:

1. Set up a grid of points in  $x, y, z$ .
2. Set the known values (boundary conditions) for  $V(x, y, z)$ , and hold them fixed.
3. Set every other grid point to the average of its 6 nearest neighbors.
4. Repeat the previous step until the solution converges.

The above is called point relaxation, and it is the subject of three spreadsheet demonstrations that I have saved in the lecture notes for § 3.1 of Griffiths.

Let's do a slightly more careful derivation to see why the answer reached by this procedure is correct. The partial derivative of  $V$  with respect to  $x$  can be approximated as follows:

$$\frac{\partial V}{\partial x} = \frac{V(x + \delta/2, y, z) - V(x - \delta/2, y, z)}{\delta}.$$

The displacement  $\delta$  is assumed to be very small compared to all important physical scales. Since we are taking finite differences, the derivative is not exact (it suffers from truncation error: contributions from higher order derivatives). The above formula is a little different from the one you learned in calculus; the two evaluations of  $V$  have been placed symmetrically about the point  $(x, y, z)$  to improve the quality of the approximation. This is called a centered finite difference. If I apply the above formula twice, the second partial is

$$\frac{\partial^2 V}{\partial x^2} = \frac{\partial}{\partial x} \left[ \frac{\partial V}{\partial x} \right] = \frac{V(x + \delta, y, z) - 2V(x, y, z) + V(x - \delta, y, z)}{\delta^2}.$$

The Laplacian is therefore

$$\begin{aligned}\nabla^2 V &= \frac{V(x + \delta, y, z) - 2V(x, y, z) + V(x - \delta, y, z)}{\delta^2} \\ &+ \frac{V(x, y + \delta, z) - 2V(x, y, z) + V(x, y - \delta, z)}{\delta^2} \\ &+ \frac{V(x, y, z + \delta) - 2V(x, y, z) + V(x, y, z - \delta)}{\delta^2}.\end{aligned}$$

If we set  $\nabla^2 V = 0$  and solve the above equation for  $V(x, y, z)$ , then

$$\begin{aligned}V(x, y, z) &= \frac{1}{6} [V(x + \delta, y, z) + V(x - \delta, y, z) + \\ &V(x, y + \delta, z) + V(x, y - \delta, z) + \\ &V(x, y, z + \delta) + V(x, y, z - \delta)].\end{aligned}$$

This helps us to understand step 3 in the point relaxation procedure. The only problem is that, upon setting each cell to the average of its neighbors, the neighbors change too. So we keep doing it iteratively. Once the values stop changing, then only at that point is the above equation satisfied at every point  $(x, y, z)$  in the grid. We then have a good approximate solution to  $\nabla^2 V = 0$ .

It is important to emphasize that the convergence is gradual. For a grid of  $N \times N$  (or  $N \times N \times N$ ) points, it takes about  $N^2$  points to decrease the error by half. So, if you want convergence to within about 0.1% for an array of  $100 \times 100 \times 100$  points, you will need about  $10^5$  iterations, each with  $10^6$  points, each requiring 5 additions and one division. So that's almost  $10^{12}$  floating point operations (FLOPs). This is quite doable for our personal computers, but you will have to wait a hot minute.