Investigation of the potentials and electric fields of an arbitrary discrete charge distribution using multipole expansion

Kenneth V. Domingo

National Institute of Physics, University of the Philippines, Diliman, Quezon City
kdomingo@nip.upd.edu.ph

1 Physics

In electrostatics, a multipole expansion is a useful tool in approximating the behavior of electromagnetic potentials and fields at a certain distance from a charge distribution. Multipole expansions are mathematical series that can be truncated in order to approximate a function within a reasonable margin of error [1]. The terms of the multipole expansion are expressed in powers of the reciprocal distance from the origin [2].

For an arbitrary, discrete charge distribution, the individual charges are assumed to have unit charge and are point-like. Thus, the potential at any point P defined by the position vector $\mathbf{r}$ is given by

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\frac{q_i}{|\mathbf{r}-\mathbf{r}_i|} \tag{1}$$

where ϵ_0 is the electric permittivity in a vacuum, q_i is the the ith charge, and $\mathbf{r}_i$ is the position vector from the origin to the ith charge. The electric field $\mathbf{E}$ is given by

$$\mathbf{E}(\mathbf{r}) = -\nabla\Phi(\mathbf{r}) \tag{2}$$

For $|\mathbf{r}-\mathbf{r}_i| \to \infty$, the potential's multipole expansion up to the quadrupole term is given by [3]

$$\Phi(\mathbf{r}) = \frac{q}{r} + \frac{\mathbf{p}\cdot\mathbf{r}}{r^3} + \frac{1}{2r^5}\mathbf{r}\mathbf{Q}\mathbf{r} + \mathcal{O}\left(\frac{1}{r^4}\right) \tag{3}$$

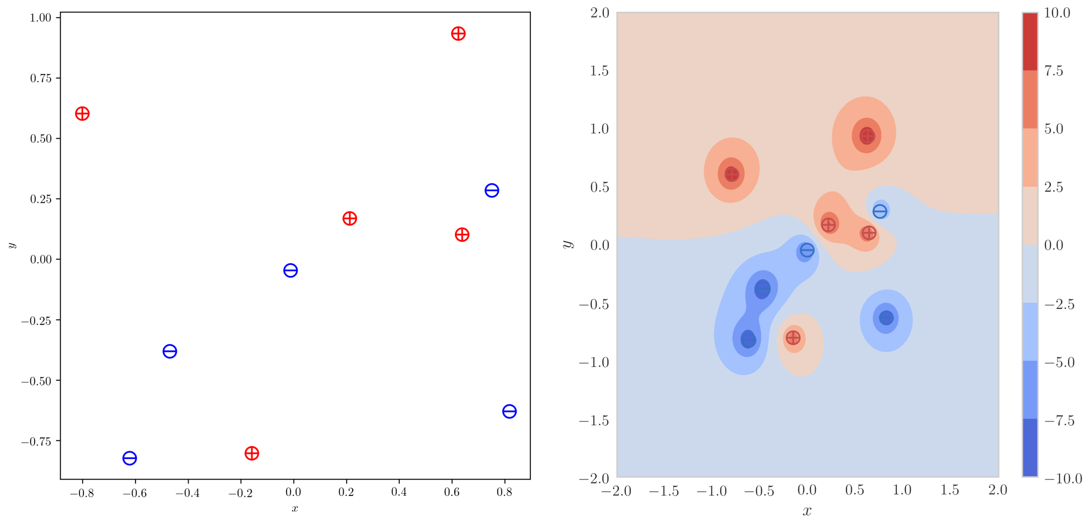
where q is the total charge, $\mathbf{p}$ is the dipole moment, and $\mathbf{Q}$ is the quadrupole tensor, given by

$$q = \sum_{i=1}^{N} q_i \tag{4}$$

$$\mathbf{p} = \sum_{i=1}^{N} q_i\mathbf{r}_i \tag{5}$$

$$\mathbf{Q}_{kl} = \sum_{i=1}^{N} q_i\left[3r_i^{(k)}r_i^{(l)} - \delta_{kl}\mathbf{r}_i^2\right] \tag{6}$$

where δ_{kl} is the Kronecker delta, and $\mathbf{r}_i^{(k)}$ is the kth component of $\mathbf{r}_i$.



(a)

(b)

Figure 1: (a) Random distribution of charges on the x-y plane. (b) Potential of 10 unit charges on the plane $z = 0.1$.

2 Algorithm

With the random seed fixed for reproducibility, all the pre-execution code was defined as a `MultipoleExpansion` class that takes an arbitrary even number N of charges as the sole argument. The behavior of the electrostatic potentials and fields was investigated by generating N unit point charges, scattered within a 2×2 meter box on the x-y plane, centered at the origin. Random numbers were then drawn from a uniform distribution and inserted into the row vectors of an array `r`. Thus, `r` is a $3 \times N$ matrix, with the rows representing the position vectors of the charges and all the z-components set to zero. From these vectors, the first half were assigned to be the positive charges, while the latter half were assigned as the negative charges.

The electric potential was calculated for a 4×4 meter box on the x-y plane by dividing it into a 500×500 `meshgrid` for $x, y \in [-2, 2]$, then evaluating the exact Φ for each point in the meshgrid using (1), and setting units of $4\pi\epsilon_0 = 1$. The potential was then plotted as a contour projection on the x-y plane using the `Axes3D` object [4]. To quantify the accuracy of the dipole and quadrupole approximations, the potential was also plotted as a function of y using the dipole and quadrupole approximations given by (3), superimposed with the exact potential given by (1).

Finally, the electric field was numerically evaluated using the central difference algorithm, as defined in [5][6], and was plotted using `streamplot` [7].

3 Results

A total of ten (10) point charges were generated for the purposes of this paper. Their distribution on the x-y plane is shown in Figure 1a. On the plane $z = 0.1$, they generate a potential, which is shown as contours of constant Φ in Figure 1b.

To compare the dipole and quadrupole approximations to the exact potential, a plane parallel to the y-axis was selected and “cut” at the location $x = z = 0.1$. This plane was of particular interest due to the fact that it passes close to two positive charges at $y \approx \pm 1$, and runs right in the middle of a dipole near $y \approx 0$. Figure 2 shows the result.

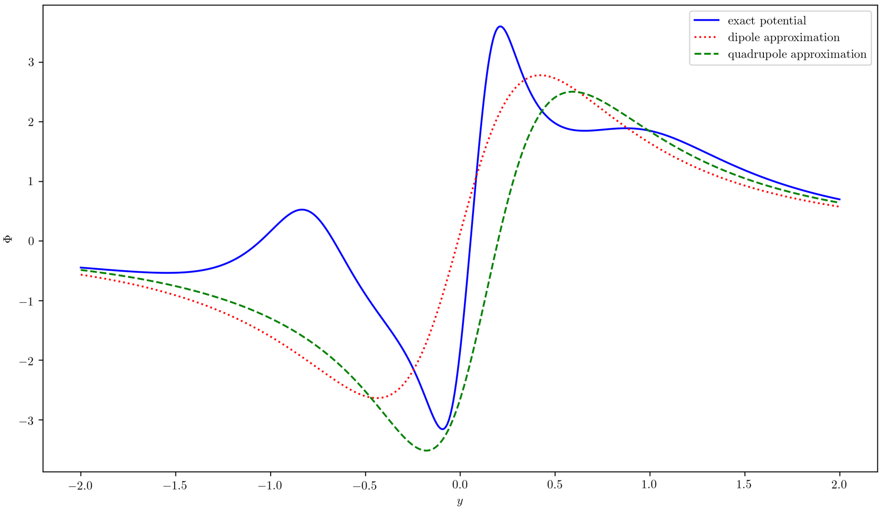
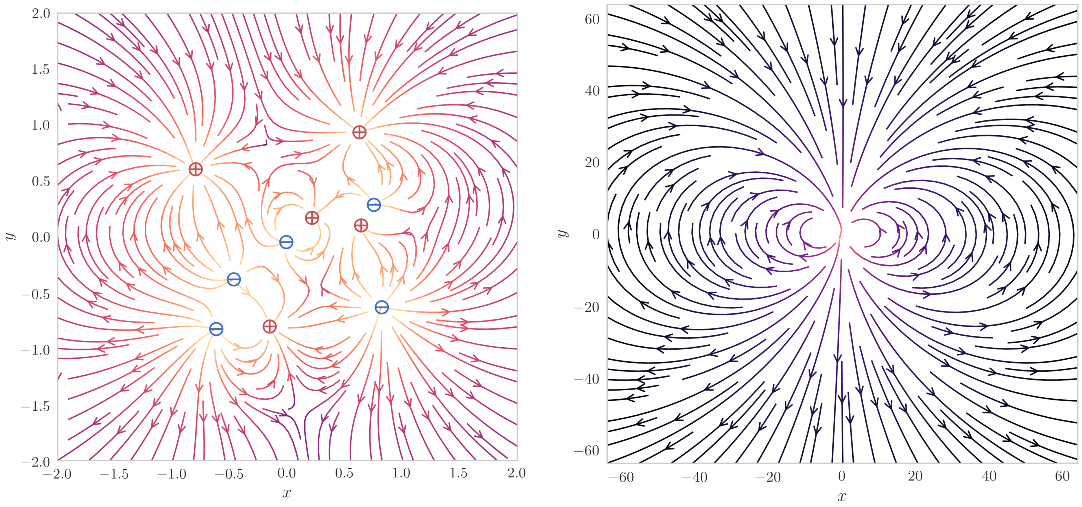


Figure 2: Potentials for different approximations at $x = z = 0.1$.



(a)

(b)

Figure 3: Electric field of 10 unit charges viewed at (a) 2 m, and (b) 64 m away from the charge distribution.

It can be observed that the dipole approximation cannot account for both of the positive charges on either side of the origin. This is due to the fact that the dipole approximation only allows for two turning points in its curve. Even with the quadrupole correction, the actual potential is emulated only in a very rough manner.

Calculation of the electric field yields the graphs in Figure 3. Figure 3a shows the complex structure of the electric field for multiple charges, and as one moves away from the charge distribution, the field appears to be that of a dipole, as can be seen in Figure 3b.

References

[1] Kinzel, W., and G. Reents (1998). Multipole expansion. In M. Clajus, and B. Freeland-Clajus (Trans.), *Physics by computer: programming physical problems using Mathematica and C*, pp. 30-36. New York: Springer (Original work published 1996).

[2] Griffiths, D.J. (1999). Multipole expansion. *Introduction to electrodynamics* (3rd ed.), pp. 146-159. New Jersey: Prentice-Hall.

[3] Gradshteyn, I.S., and I.M. Ryzhik. (1994). *Table of integrals, series, and products.* Boston, MA: Academic Press.

[4] Hunter, J.D. (2007). Matplotlib: a 2D graphics environment. *Computing in science and engineering* ***9***(3), 90-95.

[5] Wilmott, P., S. Howison, and J. Dewynne. (1995). *The mathematics of financial derivatives: a student introduction*, p. 137. Cambridge University Press.

[6] Olver, P. (2013). *Introduction to partial differential equations*, p. 182. Springer Science & Business Media.

[7] christian. (2015, 10 December). Visualizing a vector field with Matplotlib. Retrieved 12 October 2018, from *Learning scientific programming with Python*: https://scipython.com/blog/visualizing-a-vector-field-with-matplotlib/.