Using python to solve partial differential equations through the finite differences method

Anna Griffith
University of Bristol Physics Department

(Dated: March 11, 2019)

The finite differences method was used to solve two partial differential equations, the Poisson and the diffusion equation. The Poisson equation was used to model the potential and electric fields in and between the plates of a capacitor, whereas the diffusion equation was used to plot the temperature distribution along an iron rod placed in a furnace. Different boundary conditions were explored, as well as extending the diffusion model into two dimensions.

I. INTRODUCTION

Partial differential equations form many of the key fundamental equations in physics. There are a vast number of strategies to solve them, in this report the finite differences method, an approximation to the differential, will be examined.

The Poisson equation is an example of a boundary value problem, which is solved in two dimensions to find a function $u(x, y)$, independent of time, and satisfying some predetermined boundary conditions, which are kept constant throughout the calculation. The finite differences method allows a discretisation of the domain into a mesh of points where a numerical solution can be approximated, by simultaneously solving a system of algebraic equations [5]. The two dimensional Poisson equation,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y) \tag{1}$$

can be rewritten after discretisation with

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i-1,j} + u_{i+1,j} - 2u_{i,j}}{\Delta x^2}, \tag{2}$$

where $x_i = x_0 + i\Delta x$, $i \in (0, L_x)$. The same can be done for the y component, using $y_j = y_0 + j\Delta y$, $j \in (0, L_y)$ [4].

This program will be solving the Laplace equation, where $\rho = 0$ to find the potential $V(x, y)$ in and around a parallel place capacitor. This will be done using an iterative technique,

$$\begin{aligned} V(x_i, y_j) = \frac{1}{4}[&V(x_{i-1}, y_j) + V(x_{i+1}, y_j) \\ &+V(x_i, y_{j-1}) + V(x_i, y_{j+1})] \end{aligned} \tag{3}$$

where either the Jacobi or Gauss-Seidel relaxation method can be used. The Jacobi method averages over the old values of $V(x, y)$, and therefore require the storage of both an old and new grid of values. It is slowly converging, the number of iterations r required to reduce the error by a factor of 10^{-p} is

$$r \simeq \frac{1}{2} p J^2 \tag{4}$$

for a $J \times J$ grid [5]. The Gauss-Siedel method, however, makes use of new values as soon as they become available. This reduces the number of iterations required by a factor of two, however the speed of convergence is often still impractical [5]. For both methods the total computing time is $\mathcal{O}(J)^2$ [4].

The second partial differential equation to be solved is the diffusion equation

$$\alpha \nabla^2 u = \frac{\partial u}{\partial t} \tag{5}$$

where *u(x,y,t)* denotes the temperature as function of position and time, and α is the diffusion coefficient. This is an example of an initial value problem where $u(x, y, t_o)$ is known for an initial time and all position grid nodes, and is then allowed to propagate in time. This can either be modeled using an explicit, Forward Euler scheme, or an implicit, Backward Euler scheme. In the Forward Euler scheme, the value at each time step is calculated from the previous ones,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) \tag{6}$$

for $t_n = t_0 + n\Delta t$, $t \in (0, T)$. This is computationally less intensive than an implicit method, but requires $\Delta t \leq \Delta x^2/(2\alpha)$ to be stable [3].

The Backward Euler method instead replaces the right hand size of Equation 6 with elements of u at the *new time-step* , which are unknown. u^{n+1} can instead be found by solving $Au^{n+1} = u^n$, where A is a known sparse matrix composed in terms of the mesh Fourier number, $F = \alpha\Delta t/\Delta x^2$ [3]. Unlike the Forward Euler scheme this routine is stable with any time-step, and always gives a smooth, decaying solution [2].

When the diffusion equation is extended to two dimensions, however, the advantages of the Backward

Euler scheme become redundant. The time-step restriction for both schemes is the same for both, $\Delta t \leq (\Delta x^2 + \Delta y^2/(8\alpha)$, stricter than before [4].

II. METHOD

First, a routine for solving the two dimensional Laplace equation was written, to model both the potential and electric fields around a capacitor. The length of the plates, a, and their separation d are specified, and then a grid created, dimension $5d \times 2a$. The field and potential can then be approximated to be zero at the edges of the grid, as they are far from the capacitor (placed at the center), forming the boundary conditions. The iteration process to find the solution will continue until the process has converged; when the normalised difference,

$$\sum \frac{u_i^{n+1} - u_i^n}{u_i^n} \tag{7}$$

is less than some threshold value. These values will be chosen after plotting graphs of the normalised difference for each iteration. This process will be repeated for all further functions described. Both the Gauss-Seidel and Jacobi methods will be tested and compared, as well as the effect of the choice of grid spacing. The values for the electric field are simply found from $E = -\nabla V$, and both its absolute magnitude and field lines plotted.

The diffusion equation is solved in a variety of cases. First in one dimension, modelling an iron rod with one end in a furnace, allowing no heat loss. Then, also in one dimension, with the far end immersed in ice, modelled as an infinite ice bath, which, like the furnace, does not change temperature. These two cases allow for two separate boundary conditions; Dirichlet, where $u(x)$ if fixed, and Neumann where the first derivative is constant. The first of these cases is easily managed, by instead solving $Au^{n+1} = u^n + b$, where b is a vector containing the boundary conditions [3]. In comparison, Neumann boundary conditions require the matrix A to be altered. It can be shown that

$$u_x(x,t) = \frac{\partial u_i^k}{\partial x} \approx \frac{u_{i+2}^k - u_i^n}{2\Delta x} \tag{8}$$

which will be set to zero at the end of the rod where no heat loss occurs [4]. A only needs to inverted once since it has no time dependence, this was done using a sparse matrix scheme since it is tridiagonal. The distribution of heat along the rod for both boundary cases will then be compared as time progresses.

To solve the two dimensional diffusion process the Forward Euler scheme is instead used, as it is simple to implement and has the same stability criteria as the Backward scheme. My function models a square slab of iron with one end in a furnace, and as before, either one end in ice, or air with no heat loss. For a specified time, a 3D plot is created, showing the distribution of temperature along the rod. As an extension to this, a more realistic scenario is also considered, where instead of one end of the rod being submerged in an infinite ice bath, a finite block of ice is modelled, that is allowed to melt. This required a larger grid to be used, where on the edges, far from the rod, the temperature of the ice is assumed to be constant (Dirichlet). In this case the diffusivities of the iron, air and ice all need to be considered. The diffusivity of ice is weakly temperature dependent [1], but between small temperature intervals it can be approximated as constant.

Testing functions are also created for all of the above solutions. To verify the solution of the capacitor problem, the average value of electric field through the center of the capacitor is taken and compared to V/d. Quantitatively testing solutions of the diffusion equation is much harder. Instead, the temperature of the furnace and ice are modified to be the same as the initial temperature, and the heat distribution along the bar checked. In this case it should stay constant - at room temperature.

III. RESULTS

The contour plots of the potential and electric field in and surrounding the capacitor are shown in Figure 1, with both tending to zero far from the plates, and the value of the electric field between the plates matching $E = V/d$. The effects of the ratio a/d were examined, and a uniform field between the plates at large ratios observed, shown in Figure 2(a). From Figure 2(b) it is clear that a large ratio of a/d negatively effects the error reduction, with the normalised error not going below 5×10^{-2} even after 2000, iterations, whereas the other plots reach this value after less than 200.

The two different relaxation methods, Jacobi and Gauss-Siedel were also compared. It was found that on average the calculation time for the Gauss-Siedel method was longer, and contrary to the theory presented before, the error reduction was the same. The effect of the size of the grid spacing on the errors was also analysed. As expected, the finer the spacing, the fewer iterations it took to reduce the error to a specified tolerance level.

From Figure 3 it is clear that Neumann and Dirichlet boundary conditions have very different effects on the temperature distribution along the bar. When no heat loss is modelled (Neumann) an equilibrium point is reached where the bar is at the temperature of the furnace. By contrast, Dirichlet boundary conditions fix the temperatures at either end to be that of the furnace and ice bath, the equilibrium position a straight line between the two. Figure 4 shows how the bar with one end in ice

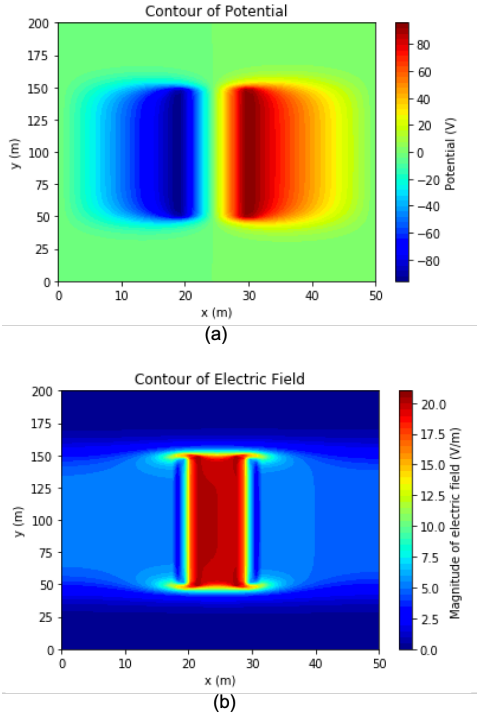

FIG. 1. (a) Plot of the potential field in and around a capacitor, plates at $\pm 100V$, length 100m and separation 10m. (b) Plot of the magnitude of the electric field around the same capacitor

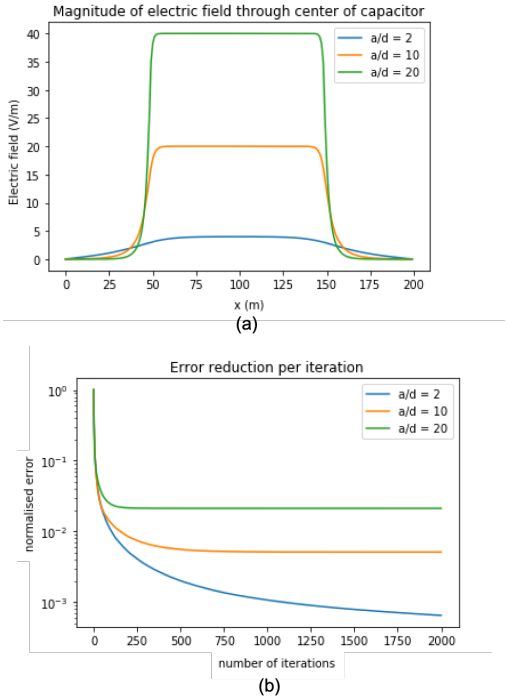

FIG. 2. (a) Magnitude of the electric field through the canter of the capacitor for differing ratios of a/d (b) Plot of the reduction of error by iteration for different ratios of a/d

reaches a (near) equilibrium state much faster than that in air, even after 300 minutes the bar is not completely at the furnace temperature. This effect is then repeated in the two dimensional model.

Figure 5 shows the two dimensional model, based on a $50 \times 50cm$ iron bar, with similar overall characteristics to Figure 3. There are differences around the edges, where it is submerged in the furnace/ice. Again, the case where the the bar is submerged in ice is much quicker to reach equilibrium, $5,258s$, than the case where the bar is in the air, 15,149s. The model created to allow for a finite sized block of ice, with heat loss allowed, is only valid over short time periods and is shown in Figure 6.

IV. DISCUSSION

The produced contours of the potential and electric fields qualitatively look as expected, with $E = V/d$ between the plates and both tending to zero outside the capacitor. The reduction of errors is heavily dependent on both the grid spacing, and the ratio of a/d, as shown in Figure 3. Instead of keeping the grid spacing constant at $1cm^2$, which works well for moderate ratios of a/d, it could be varied so that it becomes finer for larger ratios, therefore reducing the errors more effectively. Another option would be to explore successive over-relaxation or multigrid methods to speed up convergence, the latter obtains successive solutions on finer and finer grids [5].

The solution to the diffusion equation also behaves as you would expect, considering the boundary conditions on the two cases. The situation with the rod in air with no heat loss takes considerably longer to reach a relative equilibrium, this could perhaps be considered in the code so that there are different tolerance values for each case. The two dimensional version also created successful temperature distributions, with the same overall shape to the one dimensional models. This was extended to a more realistic case in Figure 6, qualitatively the heat distribution looks as expected for short timescales, with the far end of the rod exhibiting similar behaviour to the two dimensional case in air. Beyond $100s$ end of the bar reaches $600K$, which would render the assumption that the diffusivity of the ice remains constant with temperature invalid. Furthermore, at $200s$ the system reaches equilibrium, which compared to the timescales that this occurs in the other cases implies that the model is inaccurate beyond this point. To improve this model at larger timescales the temperature dependence of the ice could be taken into account, and the size of the grid could be increased, to ensure that the boundary conditions remain valid. However, this would further increase the already long computational time, $8.4s$ for 1000 iterations. This is likely due to the inability to vectorise the Forward Eu-

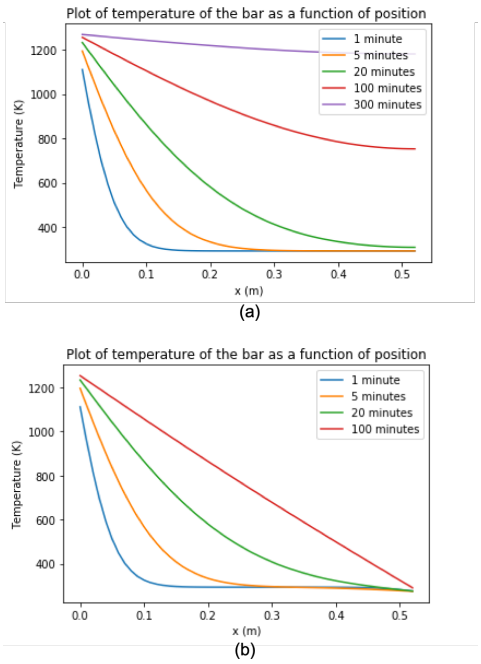

FIG. 3. (a) Temperature distribution along the bar at 5 different times, for one end of the rod in air and one in the furnace, assuming no heat loss. (b) Temperature distribution along the bar with the far end in an infinite ice bath, allowing for heat loss.

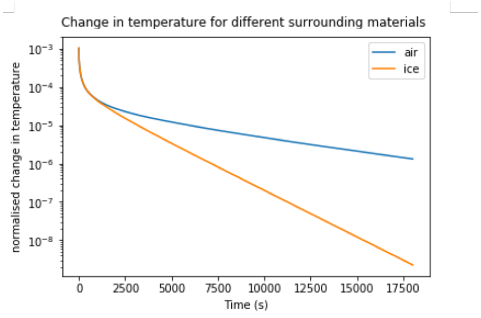

FIG. 4. Effect of the surrounding material and boundary conditions on the rate of change of temperature

ler routine, since the diffusivity of the material depends on the position in the grid. Furthermore, more research could be done into creating a quantitative test for the diffusion equation solutions, an 'order of magnitudes' estimation could be calculated, or two solutions by two different methods could be compared.

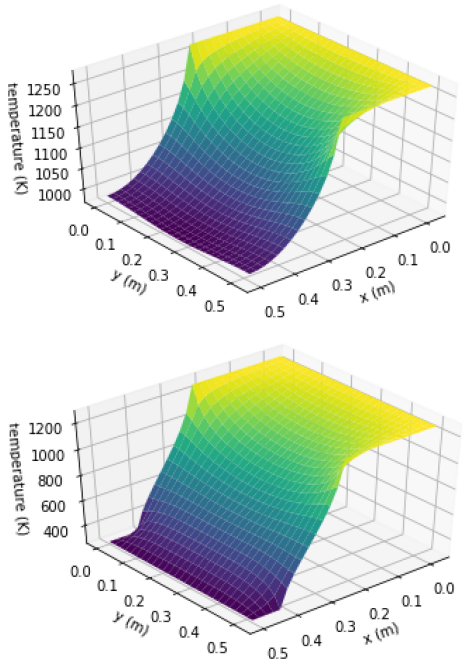

FIG. 5. (a) Temperature distribution across the bar at 5000s, for one end of the rod in air and one in the furnace, assuming no heat loss. (b) Temperature distribution across the bar at 5000s with the far end in an infinite ice bath, allowing for heat loss

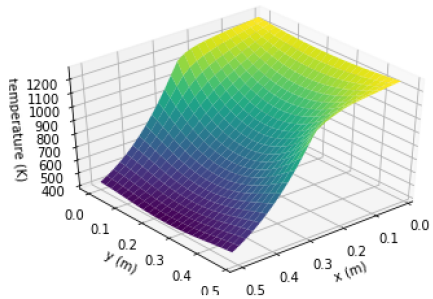

FIG. 6. Temperature distribution across the rod at 30s, with one end in a finite block of ice

[1] D. W. James. The thermal diffusivity of ice and water between 40 and + 60 C. *Journal of Materials Science*, 3(5):540–543, 9 1968.

[2] H. P. Langtangen and S. Linge. Finite Difference Computing with PDEs-A Modern Software Approach. Technical report, 2016.

[3] H. P. Langtangen and S. Linge. Finite difference methods for diffusion processes. Technical report, 2016.

[4] L. Olsen-Kettle. *Numerical solution of partial differential equations.*

[5] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, and H. Gould. *Numerical Recipes, The Art of Scientific Computing*, volume 55. 1987.