Partial Differential Equations

Ben Crabbe
Exercise 2, Computational Physics 301, University of Bristol.
(Dated: March 10, 2014)

Here we present a description and analysis of the solution to Poisson's equation by finite differencing and relaxation, and to the heat diffusion equation by the fully implicit backwards time method. The methods and programs presented are also easily adaptable to other initial value and boundary value problems.

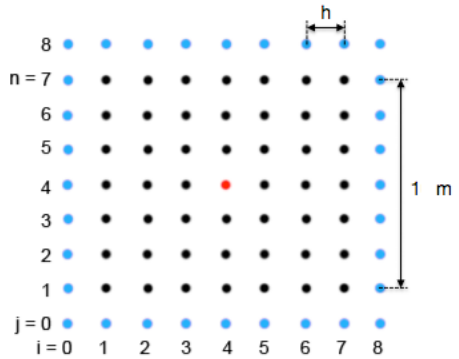


FIG. 1: An example of the discretisation with n=7. The grid spacing, $h = 10/(n-1)$ m. The blue nodes are outside the region are used to impose boundary conditions. The red node shows the location of the point source which is used in the following analysis.

SOLVING POISSON'S EQUATION

Poisson's equation In two dimensions:

$$\nabla^2 V(x,y) = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})V(x,y) = -\frac{\rho(x,y)}{\epsilon_0} \tag{1}$$

This defines the electrical potential function,$V(x,y)$, due to a charge distribution, $\rho(x,y)$. It is an example of a boundary value problem; we have a static i.e. time independent partial differential equation which may be solved uniquely in some closed region of space provided we know its behaviour on the boundary of our region. This solution may be approximated numerically by a finite differencing and relaxation as in the accompanying program poissons.c.

Here we model the potential on a region of unit length by calculating its values at n discrete points as shown in figure 1. We also include a set of points outside the region with which boundary conditions may be applied.

This discretisation enables use of a finite difference approximation of the second differential. We use a five-point stencil formula from [1], giving us a discrete form of (1) which may be rearranged to give the potential at any point in terms of its four nearest neighbours. Taking $\epsilon_0 = 1$, we have:

$$V(x_i,y_j) \simeq \frac{V(x_{i+1},y_j) + V(x_{i-1},y_j) + V(x_i,y_{j+1}) + V(x_i,y_{j-1})}{4} + \frac{h^2\rho(x_i,y_j)}{4} \tag{2}$$

We set the potential of the boundary points to 0 then move across all interior grid points calculating their new potential from (3). We use the Gauss-Seidel method of iteration where the values which have already been iterated are used where possible as this leads to a factor of 2 reduction in the convergence time over the Jacobi method [2]. The solutions are said to be fully converged when the potential at any point does not change between successive iterations by more than a given convergence factor. The resulting $V(x,y)$ for various charge configurations can be seen in figure 2.

The effect of the grid spacing on the resulting potential can be seen in figure 3. At large values of h the contours are clearly non circular indicating error. Large h also causes the potential to be more spread out around the point charge; these effects decrease as h is made smaller, however the time taken to compute quickly becomes impractical as can be seen in figure 4. This is due to both the required number of iterations and the time per iteration increasing with 1/h.

The effect of the convergence condition on the solutions can be seen in figure 5. In this case the correct value moves inwards from the boundary; this is due to initially allocating every grid position a random potential between 0 and 10 V , done to make the progression clear to see. The number of iterations required to reduce the error by factors of 10 is shown in figure 6. We observe that there is an inverse exponential relationship between the two, the slope of which is proportional on the number of grid points, n^2.

To examine the solution when the region contains a parallel plate capacitor the electric field has been calculated from the potential using the central difference formula. Chosen as it provides the greatest balance between accuracy and stability when computing differentials [3]. This gives:

$$E_y(x_i,y_i) = -(\frac{V(x_{i+1},y_j) - V(x_{i-1},y_j)}{2h}) + \mathcal{O}(h^2) \tag{3}$$

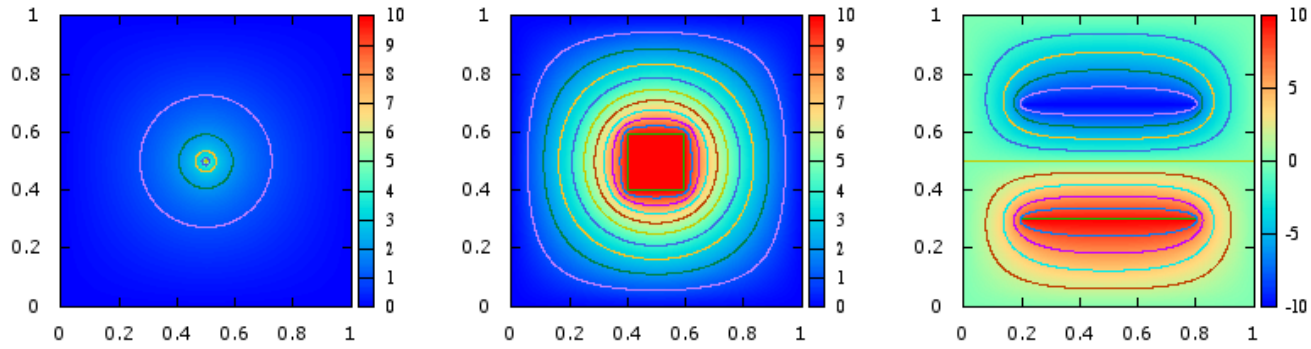


FIG. 2: The values of $V(x, y)$ around a point source, a 0.2 m^2 square conductor and a paralell plate capacitor all at a potential 10 V. The step size, h = 0.002 m, and the convergence condition is 1e-6. The boundary conditions are Dirichlet type, with $V(x, y) = 0$ on all sides. The contours represent 1V, 1V and 2V increments respectively.

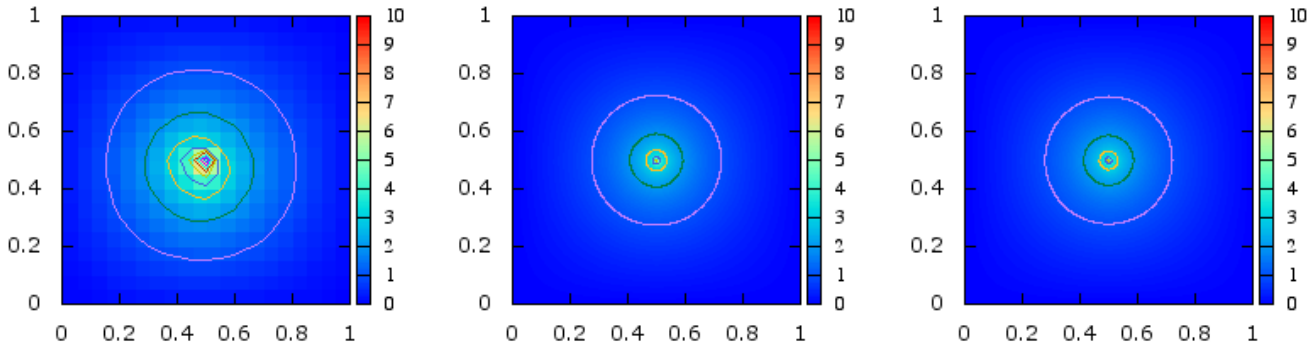


FIG. 3: The potential due to a point source for step sizes of 0.05 m, 0.002 m and 0.00125 m respectively. Each solution was converged to 1e-6. As the step size approaches zero the solution becomes more accurate. The contours represent 1V increments.

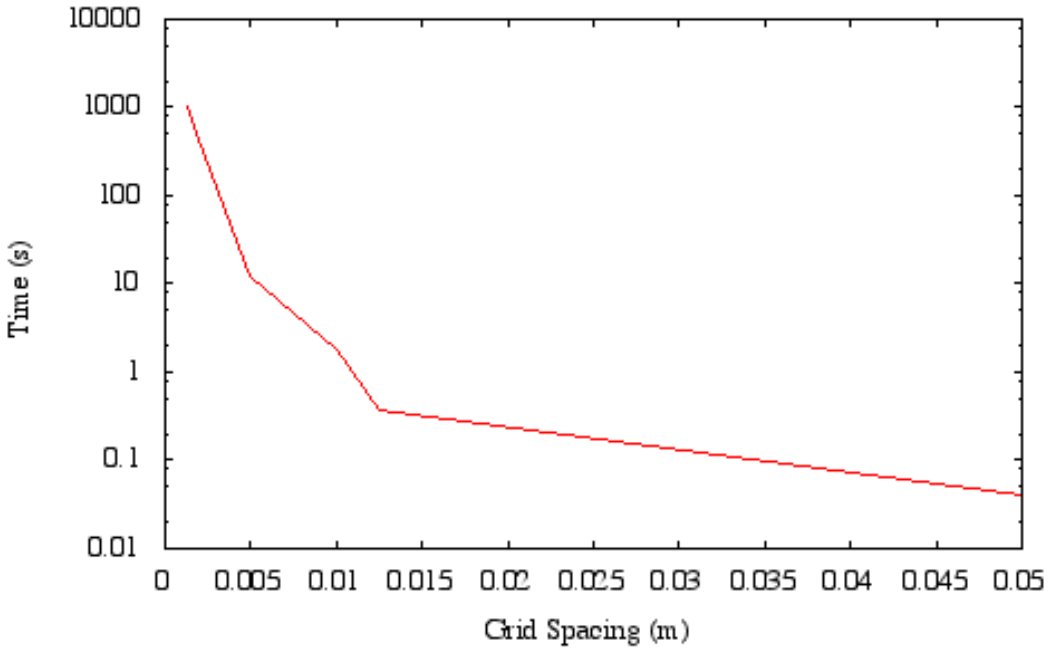


FIG. 4: The time taken for the program to converge on a solution to within 1×10^{-6} for different grid spacings.

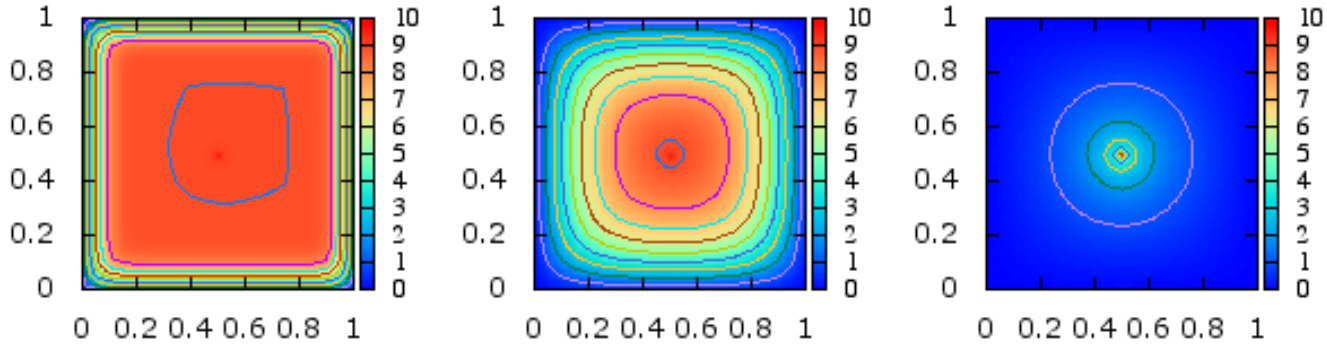


FIG. 5: Shows the potential around a point source when converged to 0.1, 0.01 and 0.0001. The gird was initialised to random values between 0 and 10, and the grid spacing is 0.01 m. The contours represent 1V increments.

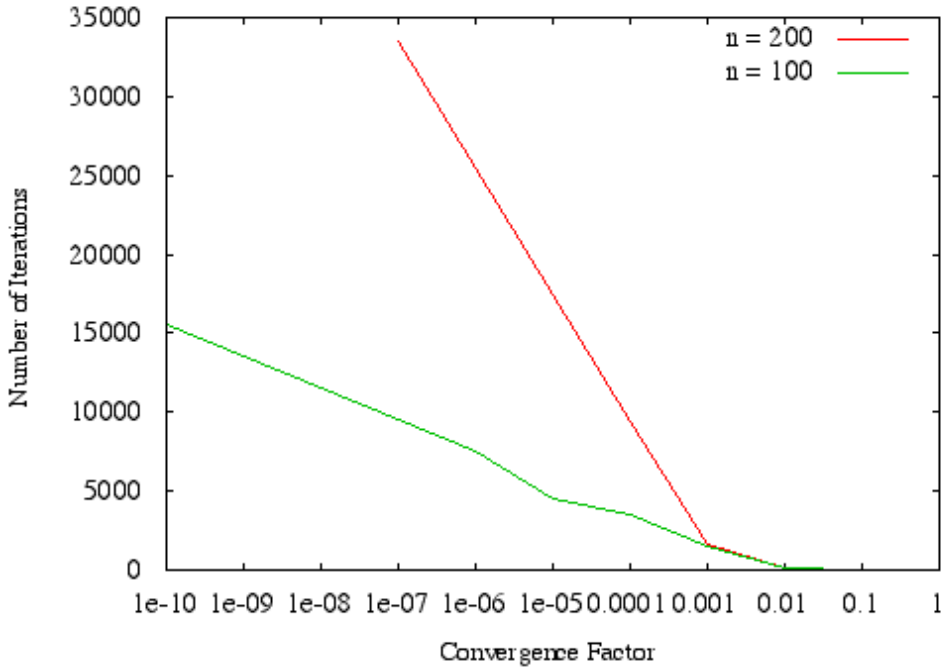


FIG. 6: The number of iterations required for different convergence conditions for n =100 and n=200.

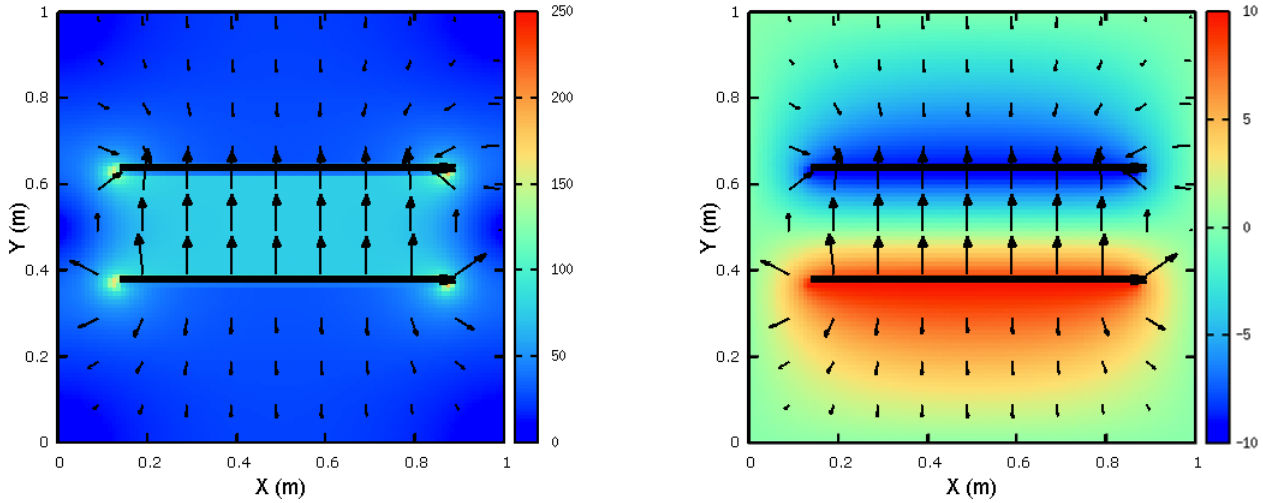
And similarly for E_x. Vector plots of the electric field are presented in figure 7. The behaviour of the electric field as the ratio of the capacitors length, A, to the distance between its plates, d, increases can be seen in figure8. The program is found to be in good agreement with the infinite plate solution at a A/d ratio of 7.

SOLVING THE HEAT EQUATION

We have a 50cm long, 1 dimensional, iron rod initially at 20°C, which at a time t=0 is subjected to a temperature of 1000°C at one end. The temperature distribution at later times is described by the heat equation:

$$\frac{\partial \phi(x,t)}{\partial t} = \alpha \frac{\partial^2 \phi(x,t)}{\partial x^2} \tag{4}$$

We start by discretising the length of the rod into n points, shown for the case of n = 4 in figure 9. We can then substitute finite difference approximations into 4, here we use a backwards difference approximation for $\partial/\partial t$ and a central difference



(a) Imposed on a heat map of the magnitude of the electric field.

(b) Imposed on a heat map of the potential.

FIG. 7: Plots of the electric field vectors around a paralell plate capacitor (show in black).

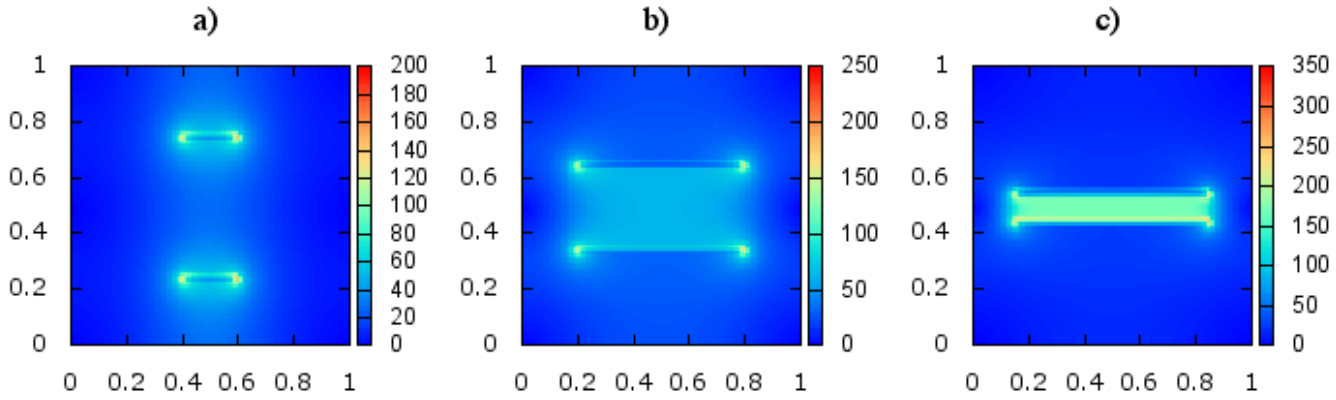


FIG. 8: Shows the magnitude of the electric field vectors on in the presence of a parallel plate capacitor with A/d ratios of a) 0.4, b) 2, and c) 7. The potential difference across the plates is 20 V giving a infinite plate solution for the electric field between the plates for c) of $E = V/d = 20/0.1 = 200$ which is seen to hold.

approximation to $\partial^2/\partial x^2$ producing the BTSC (Backwards Time Space Centred) or "implicit" form:

$$\frac{\phi'_i - \phi_i}{\Delta t} = \frac{\alpha}{h^2}[\phi'_{i-1} - 2\phi'_i + \phi'_{i+1}] \tag{5}$$

Where ϕ'_i represents the temperature at a grid point, i, at a time Δt later than the non dashed ϕ. The BTSC method has been chosen as it is unconditionally stable for all step sizes, as opposed to the more intuitive Forwards Time Space Centred (FTSC) method which is stable only when $\Delta t \leq 1/2(\Delta x)^2$ [6].

Rearranging (5) we find the temperature at a grid point i at time t in terms of itself and its nearest neighbours at at $t + \Delta t$:

$$-\frac{\alpha\Delta t}{h^2}\phi'_{i-1} + (1 + \frac{2\alpha\Delta t}{h^2})\phi'_i - \frac{\alpha\Delta t}{h^2}\phi'_{i+1} = \phi_i \tag{6}$$

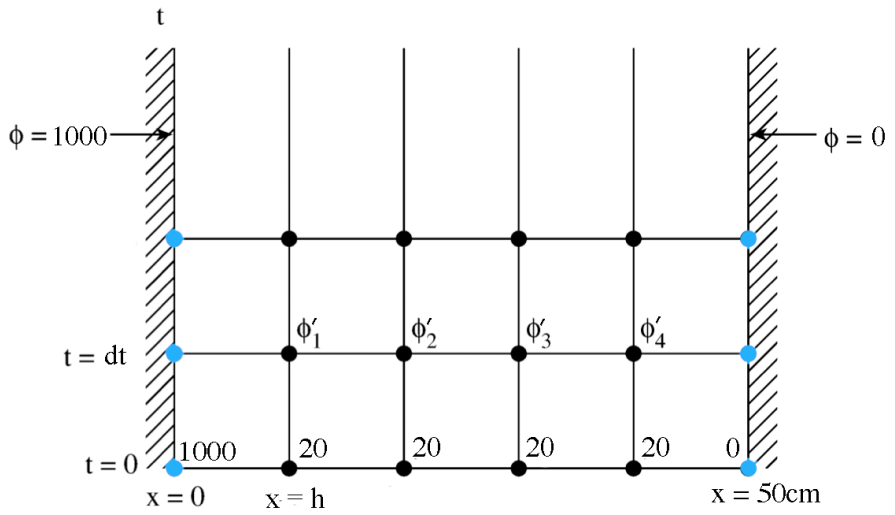


FIG. 9: The discretisation of our rod for $n = 4$, making $h = 50/(n-1)$ cm. The black points represent the rod itself, whilst the blue points, whose values are fixed at all t, communicate the boundary conditions. The temperatures at t=0 are marked in °C. ϕ'_n represents the temperatures of the nodes at time $t = dt$. the boundary condition on the right holds only for case 2 where we have it fixed to 0°C. Figure adapted from [5].

Which we can write down for all grid points in a matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ a & b & a & 0 & 0 & 0 \\ 0 & a & b & a & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a & b & a \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi'_0 \\ \phi'_1 \\ \phi'_2 \\ \vdots \\ \phi'_n \\ \phi'_{n+1} \end{bmatrix} = \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \\ \phi_{n+1} \end{bmatrix} \tag{7}$$

Where the points on the rod are those $i = 1 \to n$ with the two points on either side of these being unchanging boundary points, and $a = -\frac{\alpha\Delta t}{h^2}$, $b = (1 + \frac{2\alpha\Delta t}{h^2})$.

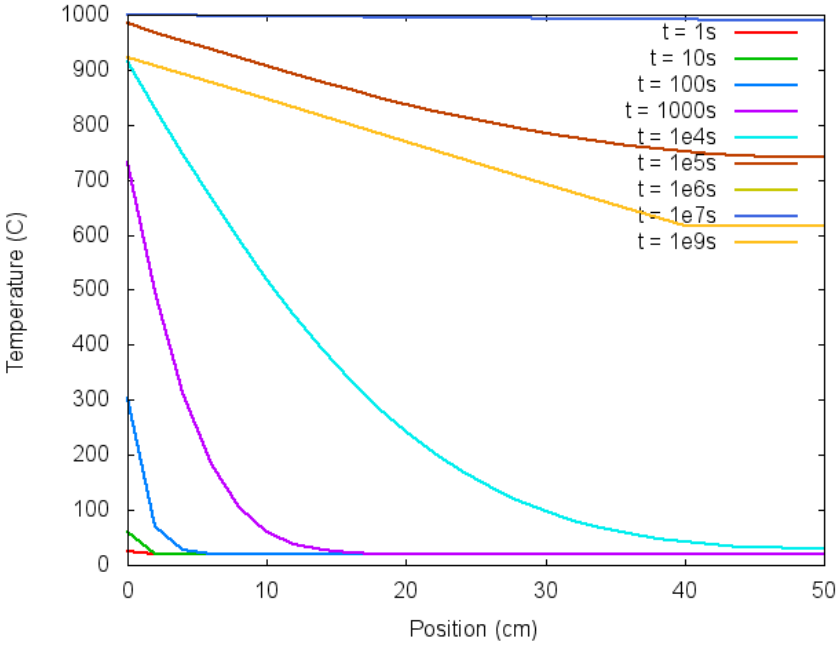
This matrix is then solved using the GNU Scientific Library LU decomposition algorithm [7], every time this is done we move the solution forwards in time by Δt.

This problem has been solved in two cases, representing two types of boundary conditions on far end of the rod. Case 1, there is no heat loss from the end. This imposes that $\partial\phi/\partial x = 0$ in the final node of the rod. This is a Von Neumann type boundary condition. Applying a backwards differential approximation tells us that for this to be true we must have $\phi_n = \phi_{n-1}$ at all t. This is applied simply by setting them equal after each iteration. Case 2, the other end of the rod is in ice at 0°C. This is another Dirichlet type boundary condition, and is handled in the same way as the 1000°C side as is shown in figure 9 and in (7). The computed temperature distributions for each case are shown in figure 10. The solutions behave as expected increasing from the initial state to a static final state within the bounds of the boundary conditions.

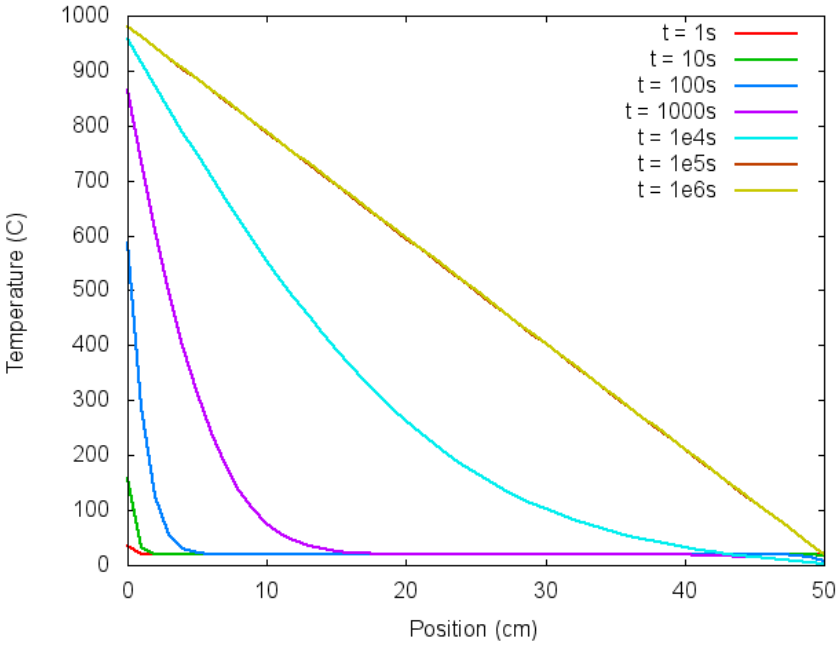
The errors in this computation come from two sources: 1) truncation errors due to our implementation of the finite differencing approximations made in (5), these should be $\mathcal{O}(h^2) + \mathcal{O}(\Delta t)$ and can be minimised by choosing small step sizes. And 2) round off errors due to the finite accuracy floating point calculations. These seem to be accumulating and making the solutions grow smaller at $t \geq 10^8$ as can be seen in figure 10 a). This should lead to an error of $N\epsilon_m$ where N is the number of computations, and ϵ_m is the machine accuracy of 3×10^-8 [8]. In the case of figure 10, $N = \frac{1\times10^9}{0.1}$, giving an potential error of 300 at completion, which is seen to be roughly correct.

[1] *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, M. Abramowitz, National Bureau of Standards Applied Mathematics **55**, ninth Printing, (1964), **25.3.30**.
[2] *Numerical Recipes*,W. H. Press et al, Cambridge University Press, second Edition (1973), **19.5**.
[3] *Numerical Recipes*,W. H. Press et al, Cambridge University Press, second Edition (1973), **5.7**.

[4] *Numerical Methods for Partial Differential Equations*,W. F. Ames, Academic Press Inc, second edition (1977) **1.5**.
[5] *Numerical Techniques in Electromagnetics*, M. N. O Sadiku, CRC Press, second edition (2001), **3.4**.
[6] *Numerical Solution of Partial Differential Equations* K. W. Morton, D. Mayer, Cambridge University Press, second edition (2005) **2.8**.
[7] *GNU Scientific Library Reference Manual*, Edition 1.0, for GSL Version 1.0 (2001)
[8] *Numerical Recipes*,W. H. Press et al, Cambridge University Press, second Edition (1973), **1.3**.



(a) Case 1



(b) Case 2

FIG. 10: Shows the temperature distribution $\phi(x)$ at increasing times. The node spacing, $h = 1$cm and $\Delta t = 0.1$s.