Explicit Finite Differencing in F Region Models

Robbie Morrison,[1]⋆ Student ID = 32965303,[2] Third Author[2,3] and Fourth Author[3]
[1] *Royal Astronomical Society, Burlington House, Piccadilly, London W1J 0BQ, UK*
[2] *Department, Institution, Street Address, City Postal Code, Country*
[3] *Another Department, Different Institution, Street Address, City Postal Code, Country*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
This paper analyses the usefulness of explicit finite difference schemes in modeling diffusion in the lower ionosphere (specifically the F region). A homogeneous 3D heat equation is solved for a density and brightness distribution using a forward time central spacing method, and compared to real data as well as theoretical results. It is concluded that for homogeneous equations explicit schemes work adequately well, but for heterogeneous heat equations non-explicit schemes must be used. For physically realistic models an alternative method entirely, such as the finite element method, is required.

Key words: Diffusion– Ionosphere – Explicit

1 INTRODUCTION

In the 1920s, much of the world had already been explored; even the stars had been charted and studied. It came to the attention of some geophysicists that the region between the mesosphere (85km) and the exosphere(600km) was practically an enigma. Little was known about it beyond its electrical conductivity properties and its effect on radio signals, as described in Rishbeth (1988).The ionosphere received its name due to its ability to ionise atoms and molecules, converting them into a plasma. Until the 1970s the best way of understanding this process was through theory and passive observation of the region, but as technology evolved it became feasible to send our own elements into the ionosphere and study how they reacted to the environment. This particular study will try to simulate the diffusion of a barium cloud 190km above the Earth's surface, and compare our results with the experimental results of Jindong et al. (2014). Photos of their barium cloud at different times are shown in Figure 1. The closer our results are to observation, the better understanding we have of ionospheric physics.

The F region is a section of the ionosphere between 140km and 220km (more detail can be found in Martyn (1959)). It has a particularly high free electron density so from now on the ionospheric physics we study will be referred to as F region physics, though much of our results may be applied to other layers of the ionosphere. The class of model we use is known as a finite difference method (FDM), where we take continuous partial differential equations and discretise them to form an array approximating their continuity. The particular type of FDM we use is known as an explicit FDM: this forms equations where each solution is solved sequentially from the previous solution. There is another type of FDM known as an implicit FDM where the equations form a system of linear equations and are solved simultaneously using linear algebra methods. They are typically harder to code but provide better temporal accuracy. Our paper aims to examine how reliable explicit FDMs are when modeling F region physics.

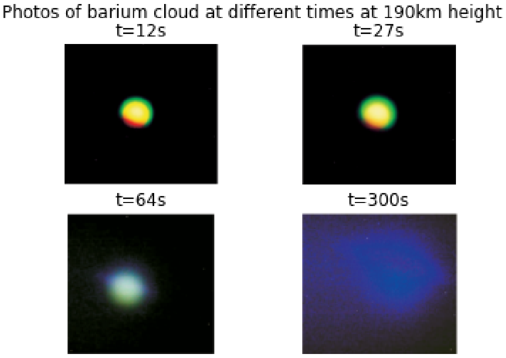


Figure 1. Barium cloud diffusion at different times. Unclear about the true scale of $t = 300s$ so no scales were added. Results from Jindong et al. (2014)

2 METHODS

2.1 The Heat Equation

The process of diffusion with convection can be represented by the heterogeneous PDE:

$$\frac{\partial C}{\partial t} = D\nabla^2 C + v\frac{\partial C}{\partial x} \qquad (1)$$

where C is the variable diffusing (in this case barium number density), t is time, D is the diffusion coefficient, ∇^2 is the Laplacian operator, and v is the wind speed. We will return to the heterogeneous term, but for now, we take $v = 0ms^{-1}$. For our problem, we will work in 3 dimensions so there are 3 second-order partial derivatives on the RHS of equation (1). To model the diffusion this way, there are various important assumptions we are making:

⋆ E-mail: publications@ras.ac.uk (KTS)

- A negligible number of the barium atoms ionise, and so there are no generation or loss terms in equation (1)
- There are no vertical forces (namely the gravitational force) and for now no planar forces such as wind
- The particles no longer have any velocity remaining from the initial explosion
- The initial density distribution is constant i.e $C_{t=0} = N_0$ where N_0 is the initial number density except on the surface of the sphere where it is equal to $\frac{N_0}{2}$

The only valid assumption is the third one; the others are bound to increase the error in our model. We are choosing to ignore gravitational forces because compared to wind its effect on diffusion will be relatively small, and for now we want to analyse the homogeneous equation. Additionally, the generation and loss terms would be present due to the intensity of UV radiation at those heights. The final assumption, the distribution, can be changed to an exponential decay fit (which would be more realistic) should we find a reason to when comparing it to experimental data.

2.2 Finite Differencing

The FDM takes the infinitesimally small width of a derivative (or anti-derivative) and increases its size to a real (though still small) number. As an example, consider $\frac{\partial C}{\partial t}$. To evaluate this numerically using the FDM, we represent the derivative as :

$$\frac{\partial C}{\partial t} \approx \frac{C_t^{x,y,z} - C_{t-1}^{x,y,z}}{\delta t} \tag{2}$$

where y_n represents the point n where the derivative is evaluated and δx is a small width. This is known as a forward time finite difference, as it evaluates the current step from the previous step sequentially. It is accurate to first order in δt.

In this section and throughout the paper, we describe a term or scheme as being nth-order accurate, where $n = 1, 2, 3....$ This stems from the fact that finite difference schemes for derivatives come from the derivative's Taylor series, where the nth term in the series has a coefficient h^n, where h is the spatial width for the finite differencing. If a term is first-order accurate, the error will come from the term with the coefficient h^2. The smaller this coefficient, the smaller the error. There are two ways of accomplishing this: the first is to increase the resolution (make h smaller), whilst the second is to have a better order of accuracy. Better orders of accuracy are preferable because they tend to change the solution's computational expense marginally and dramatically improve the error.

Using FDM and writing $\delta x^2 = \delta y^2 = \delta z^2 = 6D\delta t$ where $\delta(x, y, z)$ represent the steps in the spatial components and δt represents the time step, we may write equation (1) (remember, without convection) as

$$C_{t+1}^{x,y,z} = \frac{1}{6}(C_t^{x+1,y,z} + C_t^{x-1,y,z} ... C_t^{x,y,z-1}) \tag{3}$$

where $C_t^{x,y,z}$ represents a discrete point in space and time. Equation (3) has been written using an ellipsis as it is a repetitive and long formula in full form. We have discretised the RHS using a second-order central spacing scheme which has an accuracy of $O(\delta x^2)$. Given the discretisations we have used, this method is known as forward time central spacing, or FTCS. The equation has an intuitive meaning: for a given point (or voxel for three dimensions) $C_t^{x,y,z}$, its value at the next time step $t+1$ is given by the average of the values of the 6 adjacent voxels to it. This method is explicit as it calculates each value sequentially. As stated earlier, it is easy to program and suits the scope of our needs for this model, but has two main drawbacks: it is only first-order accurate in time, and it has a stability condition: $\frac{6D\delta t}{\delta x^2} \leq 1$. Given our previous method of defining D, our scheme is stable.

We have discretised our distribution so we can evaluate the diffusion on a 3D grid. A 4D grid is not feasible as each step would be $O(N^4)$ operations, whilst a 3D grid is only $O(N^3)$ where N is the length of the grid. As a large N is needed for a reasonable representation of the system, we must instead have 2 separate 3D arrays representing all spatial points for t and $t - 1$ respectively. It is best to initialise the grid as an array of zeros and add the initial density conditions as zero value voxels represent a point of zero density.

When evaluating the density distribution's evolution, we can exploit the symmetry of the problem by only evaluating the evolution of positive x,y, and z. This corresponds to only 1/8th of the sphere and therefore uses less memory (more efficient). We need to define additional boundary conditions for the interior voxels of the sphere that contain at least 1 zero coordinate. As an example, consider $C_t^{i,0,k}$.We cannot evaluate its time evolution according to equation (3) because there is no $C_t^{i,-1,k}$. However, as the density diffuses symmetrically, $C_t^{i,-1,k} = C_t^{i,1,k}$. We then substitute this term into equation (3) and have a new, slightly adjusted finite difference scheme. This method works for voxels with multiple zero coordinates too.

The boundary conditions (BCs) on the exterior of the grid represent an additional assumption of our model. There is no symmetry we can use to get the values of the voxels outside our grid, so the boundary condition of the system is that all edges of the grid are fixed at zero. On a molecular level, the particles have an equal chance of moving in any direction, but their bulk velocity points radially outwards. This is not reflected at the outer boundary condition- it permanently "pulls" all particles out of the grid without letting any enter and keeps the outer voxels at 0. We will investigate this assumption's contribution to the overall error by modifying these BCs.

We consider the total domain of our model as a cube, with the outer faces as the boundary conditions (inner faces refer to the ones affected by the inner symmetry discussed earlier). We adjust each boundary voxel not on a vertex of the cube to follow the condition

$$C_t^{outer} = \frac{C_t^{outer-1}}{2} \tag{4}$$

. This means that the outer face BCs are now dynamic with the density distribution itself. The vertex BCs are still "black holes" but represent a sufficiently small part of the domain that their effect should be negligible. We ignore them as they are slightly harder to program and should not drastically affect the result. Note that our program has a slight flaw: for a time t, the boundary conditions calculated are actually C_{t-1}^{outer}, but given the very small values we will be dealing with and how slowly they change, this is also considered negligible. This flaw has been left in the code because it is much simpler to program than the true solution. The new BCs' goal is to decrease the possible sink effect.

2.2.1 Other Schemes

The main issues with our algorithm are that it is only first-order accurate in time and second-order accurate in space. Increasing the spatial resolution means it is more computationally expensive- we will quantitatively test this later. First-order accuracy in time limits our modeling of late-stage behavior. We will outline three approaches that solve these issues and explain why we are not using them:

- Radial finite difference (RFD): Without convection, the diffusing cloud has radial symmetry. We could rewrite the Laplacian operator ∇^2 in spherical coordinates, which would only leave the terms that depend on r, the radial vector. The density distribution would be a 1D array, and so we can make δr very small. The algorithm needed is described in Versypt & Braatz (2014).
- Method of lines (MOL): We write the temporal derivative as $\frac{dC}{dt}$ to form a system of ODEs. These can then be solved using a method like Runge-Kutta, which greatly reduces the error. This would be a stiff system of ODEs so implicit methods like diagonally implicit Runge-Kutta are recommended. An example is shown in Hundsdorfer et al. (2003)
- Crank-Nicholson (CN): We discretise the time step using a second-order accurate scheme. This leads to an implicit scheme where the entire system is solved simultaneously using methods like Gauss-Seidel. This is done in Dehghan (2001)

We can achieve accurate solutions by combining RFD with either MOL or CN. RFD is not used because for any situation that isn't radially symmetric (e.g convection) the symmetry disappears so the method loses its value (in the homogeneous case however it is the best spatial FDM one can use). Neither MOL nor CN are used because although they have higher orders of temporal accuracy, both lead to implicit methods. The goal of this paper is to analyse how useful explicit methods are, hence determining that an implicit method is very accurate is not particularly useful data for us.

2.3 Initialising The Diffusion

1kg of barium is detonated 190km above the Earth's surface. Assuming a uniform density distribution in a spherical volume, the number density of the barium cloud, N_0, is given by:

$$N_0 = \frac{3}{4m_{Ba}\pi R_{initial}^3} \tag{5}$$

where m_{Ba} is the mass of a barium atom and $R_{initial}$ is the initial radius of the diffusing sphere. Based on the data from the experiment, $R_{initial}$ was 3km, so $N_0 = 3.86 \pm 0.04 \times 10^{13} m^{-3}$. The error comes from our initial model: we want the boundary value of the sphere to be half that of the central value at $t = 0$, so we effectively lose $1/100$th of particles, assuming 50 grid points represent the radius. The more grid points used, the smaller the error. Because of this error, we will only measure in terms of relative density. None of our calculations directly rely on a realistic density, so we can increase the accuracy of our results by removing the error discussed previously.

2.4 Brightness

When observing the barium atoms diffuse, we are not observing its density C but rather the ability to see through the cloud. If we can see through it easily, the density is likely low or that point has a small depth. We refer to this as brightness, where the more difficult it is to see through a point in the cloud the brighter that point is. Mathematically, this can be expressed as:

$$B(x,y) = \int_{-\infty}^{\infty} C(x,y,z)dz \tag{6}$$

where B is the brightness and z is the line of sight we integrate along. We can use a Riemann sum to numerically represent this as:

$$B(x,y) = \delta z \sum_{-\infty}^{\infty} C(x,y,n\delta z) \tag{7}$$

but given the symmetries in our current array, it can be evaluated instead as:

$$B(x,y) = 2\delta z \sum_{0}^{N} C(x,y,n\delta z) \tag{8}$$

where N is the length of the array (we do not need to go to infinity). This equation represents the upper quadrant of the x and y axes. Visually, this would represent how the cloud appears from the ground.

2.5 Radius Calculation

When attempting to replicate the experimental results, we must measure our cloud's shape in terms of its radius / maximum distance from the origin with a non-zero density. This is done by calculating the first voxel from the center that satisfies $C = 0m^{-3}$, and defining the radius as the distance to the previous voxel. These numbers are all floating point values, and so due to computational inaccuracy, none of them will ever be equal to true zero. Instead, we define our condition such that $C = 0.0m^{-3}$ to two significant figures. As we are using relative density, this is a reasonable approximation. The error from the approximation will not be considered as it will likely be negligible.

2.6 Theoretical Results

To quantitatively measure if our finite difference algorithms are giving meaningful results, we need to test them against known solutions. When solving equation (1) (without the convection term), we may substitute in an ansatz $C(r,t) = R(r)T(t)$ (remember that we can write the solution in spherical coordinates). Solving using the previously specified initial and boundary conditions, we find:

$$C(r,t) = \frac{1}{2r\sqrt{\pi D t}} \int_0^{R_{initial}} x[e^{\frac{-(r-x)^2}{4Dt}} - e^{\frac{-(r+x)^2}{4Dt}}]dx \tag{9}$$

where $R_{initial}$ is the initial radius of the sphere and D is the diffusion coefficient. The derivation is shown in full by Ayers (1970). At t=0, the first term will tend to infinity while the integral will tend to 0. Theoretically, this will cancel out, but there is a risk of performing this computationally where we introduce large errors for the initial value. Noticing that we are integrating Gaussians, we can rewrite this equation in a way more suited for computation:

$$\frac{1}{2}[2 - erfc(r_+) - erfc(r_-)] + \frac{1}{r}\sqrt{\frac{Dt}{\pi}}[e^{-r_+^2} - e^{-r_-^2}] \tag{10}$$

where $r_\pm = \frac{R \pm r}{2\sqrt{Dt}}$ and erfc is the complementary error function, denoted by:

$$erfc(z) = 1 - \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{11}$$

the implementation of this function in many libraries is poor for $z < 0$, so for this condition only, we define a new equation:

$$erfc(z) = 2 - erfc(-z) \tag{12}$$

Combining equations (10),(11), and (12), we have a theoretical solution for equation (1).

Brightness can also be represented theoretically. Rewriting equation (6), we find:

$$B(\rho) = \int_{-\infty}^{\infty} C(r,t)dz = 2\int_{0}^{\infty} C(\sqrt{\rho^2+z^2},t)dz \tag{13}$$

where $\rho^2 = x^2+y^2$ and $r^2 = \rho^2+z^2$. The issue with this is that any integral with ∞ as a bound tends to cause computational inaccuracies, along with general rounding errors in the computation. It's also possible to substitute equation (9) into equation (6), which avoids the computation of $erfc$ and involves a double integral. The issue with double integrals is that they can be computationally expensive and errors introduced in the first integral will propagate through the second. Monte Carlo integration could solve the computational expense issue but would introduce further errors due to significant limitations in its accuracy.

It can be shown [G J Daniell (private communication)] that the brightness can instead be written as:

$$B(\rho) = \frac{R^3}{Dt}e^{-\frac{\rho^2}{4Dt}}\int_{0}^{\frac{\pi}{2}} e^{-\frac{R^2 sin^2\theta}{4Dt}} I_0(\frac{R\rho sin\theta}{2Dt}) sin\theta cos^2\theta d\theta \tag{14}$$

where I_0 is the modified Bessel function of the first kind of order 0, defined as:

$$I_0(x) = \sum_{k=0}^{\infty}\frac{(x^2/4)^k}{k!^2} \tag{15}$$

Due to the combination of the exponential term, sinusoidal terms, and Bessel function in equation (14), it is unlikely that Riemann sums or the trapezoidal rule will be effective. Instead, we use Simpson's rule due to its fourth-order accuracy. We also discretise the domain with 1000 points, and given that the domain is finite, the error should be negligible. If the reader is unsatisfied with this, we recommend adaptive Gaussian quadrature rather than using a higher resolution in the domain of the integral. Although purely optional, we solve for the theoretical brightness using equation (14) to find $B(x,y)$ rather than $B(\rho)$. This is because it makes it slightly easier to visually plot from a programming perspective, but it should make a negligible difference to the actual value of brightness. The only errors would come from computational rounding, but these should be small enough that they can be ignored.

2.7 Investigation of Error Origin

The errors on the FTCS density distribution are calculated from the maximum difference between it and the theoretical value at any given time.

We evaluate the respective maximum error contributions (not relative error as we are already using relative density) denoted by ε_{max} from the finite difference scheme and the boundary conditions used. We evaluate the error caused by FTCS using a resolution twice as high as the standard resolution for the spatial discretisation and evaluate the error from the boundary conditions using the modified BCs outlined earlier.

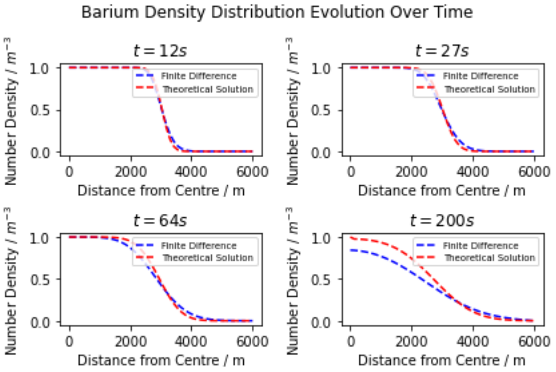


Figure 2. A plot displaying how the barium number density distribution varies over time, comparing the FTCS scheme with the theoretical solution

Table 1. Maximum relative density error at various times for different schemes

Time / s	Standard ε_{max}/m^{-3}	High Res ε_{max}/m^{-3}	Alt BC ε_{max}/m^{-3}
12	0.0616	0.0610	0.0616
27	0.0747	0.0680	0.0747
64	0.104	0.0815	0.104
200	0.178	0.109	0.178

3 RESULTS

3.1 Density Distribution

We begin by measuring how the density distribution, $C(r,t)$, evolves through time. We choose $t = 12s, t = 27s$, $t = 64s$, and $t = 200s$. The first three values are chosen simply because these are the times that photos of the cloud were taken during the experiment as seen in Figure 1 (though we have data for more than these three times), and the final value is taken because that is the final value in our range of t.

The errors between $t = 12s$ and $t = 64s$ in Figure 2 are all relatively small; the largest error occurs at approximately the spherical boundary and as time progresses it moves closer to the center of the sphere. At $t = 200s$ the maximum error has significantly increased with a value of approximately $\pm 0.17m^{-3}$. The computational solution returns a negligible error compared to this value (fourth-order accuracy) due to Simpson's rule for the integral so it is ignored. The statistical uncertainty from operations e.g. addition and division for density values is also ignored due to how much larger the present errors are.

As seen from table (1), the errors between the "black hole" BCs and the modified boundary conditions, to 3 s.f, are all zero. With the increase in spatial resolution, whilst initially there is a negligible difference, by $t = 200s$ the error is reduced by almost a factor of two. The time taken to move from t to $t+1$ in the standard scheme is $0.70 \pm 0.1s$ whereas the high-resolution scheme has the time taken $\approx 3.8 \pm 0.1s$ to 2 s.f. The errors come from an inherent instability in computers themselves: the time taken for one step may not be the same as another. The errors have been estimated but should be reliable.

3.2 Brightness Distribution

The errors from the theoretical brightness in Figure 3 are known to be small for $t > 4s$ and so are taken as zero. For $t < 4s$, due

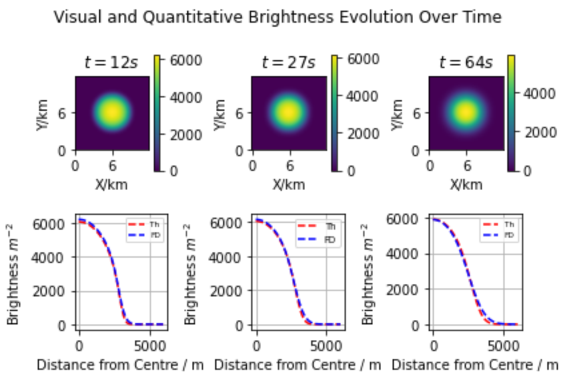


Figure 3. Each column represents a time slice. The top row represents a visual representation of the cloud with a brightness scale. The bottom row is the brightness measured radially outwards. Th means theoretical solution, FD means finite differencing solution

to the number of very large and very small values in equation (14) large errors were present for voxels not close to the center of the sphere. The errors from the Riemann sum are small as for $t = 12s$ the maximum relative error is ≈ 0.06 (to 1 s.f), a value very similar to the density. The errors also follow a similar spatial trend to density, where the largest is near the edge of the sphere and the smallest is close to the boundary or center. $t = 200s$ has not been calculated here but given the previous three t values we can assume it will follow a similar trend to the density distribution.

3.3 Comparison With Experiment

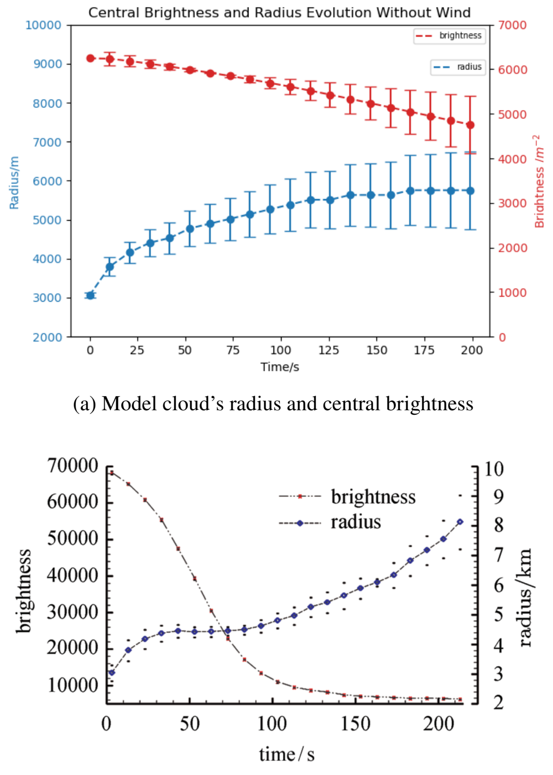
We now look quantitatively at discrepancies between our model and the actual diffusion of the cloud, namely the cloud's radius and central brightness.

The relative errors on the radius in Figure 4(a) are assumed to be equal to the maximum relative error of the density: we do not include the error from the initialisation of the diffusion as it is comparatively small. The errors on the central brightness come from the discrepancy between the model's central brightness and the experiment's. Whilst the graph is not shown here, it should be noted that for completeness we also created the same plot with the additional boundary conditions discussed earlier. There was no noticeable change between Figure 4(a) and this. The errors on the radius grow as expected, with the radius growth plateauing for $t \geq 175s$.

The central brightness errors shrink before growing again: this is due to the numerical central brightness initially being larger than the theoretical brightness and then falling to a smaller value, hence there is a point where they are equal. The central brightness falls at a roughly steady pace = $-7.25 \pm 0.8 m^{-2} s^{-1}$. The error on this term comes from the maximum central brightness error, converted into a relative error and applied to the gradient. Note that the error on the initial central brightness appears to be zero. This makes sense as initially, the only noticeable error would come from the Riemann sum when evaluating brightness.

4 DISCUSSION

The FD of the density distribution between $t = 0s$ and $t = 64s$ appears to closely resemble the theoretical results as seen in Figure 2. The maximum error approaching the center is likely because it follows the sharpest drop-off between two adjacent densities. Initially, the maximum drop-off is close to the center but the outer densities flatten faster than the central densities, so it tends to move inwards. Comparing our results from the standard FTCS scheme to the higher resolution scheme and the additional BC scheme, we can see that the error is almost entirely caused by the resolution limitations. To 3 s.f, the error between the basic scheme and the additional BC scheme is identical, so for this model it does not need to be changed as the sphere is sufficiently far away. For a larger initial radius, the basic BC scheme may cause a larger error. The improved resolution scheme almost halves the maximum error at $t = 200s$ so it is clear where our errors stem from.



(a) Model cloud's radius and central brightness

(b) Real cloud's radius and central brightness

Figure 4. Comparison of our current model cloud's evolution with experiment

Using any of the alternative schemes outlined in the methods section would achieve results much closer to the theoretical solution. These would also avoid the extensive computational time caused by the higher resolution FTCS. It is not worth considering computational errors: due to the low order of accuracy present in the FTCS scheme, the computational errors will be negligible comparatively. It can be assumed that a similar error reduction would occur with an increase in temporal resolution, although the change would be less dramatic as it is only first-order and not second-order like the spatial scheme.

Something else worth considering is that due to the large array being used, FTCS computation is quite slow (even without the resolution increase). This also presents the radial spatial scheme as attractive: without increasing the resolution, the evaluation time is much shorter than using Cartesian coordinates (assuming the problem remains symmetric, which for F region physics is unlikely). Overall, FTCS works adequately well to model density and enough to trust our results at lower values of time, but higher values of time or higher order spatial accuracy require better algorithms defined previously.

The brightness of the cloud has similar errors to the density distribution: the theoretical solution is evaluated to only have errors from computational inaccuracy which will be small due to Simpson's rule being applied in equation (14), and the finite difference brightness depends directly on the model density. The only additional source of error comes from the Riemann sum (equation (7)), which due to being a finite difference method will have better accuracy as resolution increases. Our resolution is already reasonable, so our accuracy for shorter periods is still acceptable. If the reader is unsatisfied with this, a standard numerical substitution (e.g. trapezium rule) for integration is suitable as we are only integrating in one dimension. As a reminder, there would still be a significant error from the density itself. The predicted shape of the cloud using this model also works reasonably well up to $t = 64s$. Comparing the cloud shape in Figure 3 to the clouds in Figure 1, the visualisation is reasonably accurate as the cloud has not yet deviated significantly from a sphere.

Figure 4 shows quantitatively how the cloud evolves and how the actual evolution compares to our model. Our errors are significantly larger, but that's due to the different meanings of each graph's error. The experimental errors represent an uncertainty in resolution, whilst our model's errors come from the difference between the model and the theoretical solution. Perhaps the most obvious difference is that our central brightness scale is 10 times smaller than the experimental scale. Jindong et al. (2014) do not give units so one may assume their scale is a relative brightness scale. Regardless, we can still check if our data has the same trends as theirs. Evidently for both the brightness and the radius evolution this is not the case: our brightness drops off far slower at a roughly steady pace with only $t \leq 10s$ closely resembling their brightness data. Even with the errors, our brightness data does not match experimental values. Our radial data holds form better (up to about $t = 50s$) but it too cannot match the experimental data even with errors included. This means our theoretical model and assumptions are wrong.

By interpreting the gradient of the experimental radius trend as the bulk velocity of barium atoms (their average velocity), we see that the graph experiences what appears to be an external force: the initial bulk velocity slows to near $0ms^{-1}$ at about $t = 50s$, and then increases again. This resembles the velocity of a particle moving one way, decelerating to a stop, and then accelerating in the direction of the applied force. As we are measuring radially, this change of direction is not explicit but assumed. This also tells us that adapting the initial condition to be more physically realistic e.g. a Gaussian distribution would not significantly improve our model, as there would still be no external force. Examining Figure 1 at $t = 300s$, we can see a significant asymmetry in the cloud along horizontally, most likely due to wind. This could then be modeled using equation (1) with the convection terms.

It would seem appropriate to apply the same strategy used on the homogeneous PDE to the convection equation: FTCS diffusion with a CS convection term. Using data from Zhu et al. (2020), the experimental value of $D = 2 \pm 1 \times 10^5 m^2 s^{-1}$ and the value of the horizontal wind speed is given by $v = -44 \pm 2ms^{-1}$. Note that there are two wind speeds: the meridional (flowing north to south) and zonal (east to west) so there would be an additional term in equation (1) representing the y direction, however coincidentally at 190km they have roughly the same value. The large errors on these terms come from being read off a graph. In convection equations, however, there are additional stability constraints for explicit schemes:

- $r = \frac{2v\,\delta t}{\delta x} \leq 1$
- $2R = \frac{6D\,\delta t}{\delta x^2} \leq 1$

With the size of our cloud and the value of our diffusion coefficient, these conditions cannot be met unless we reduce the size of δt to the point where our model becomes practically meaningless. An alternative to this is implicit diffusion with explicit convection: this removes the second condition entirely, with the first condition being easy to work with. It would provide second-order accuracy in space and time, though a practical demonstration of this method is beyond the scope of this paper. The much larger spread of particles due to the convection terms should hopefully bring the model closer to experimental data. Note that here the word radius is used liberally, and with convection terms refers to the distance between the origin and the furthest non-zero density voxel. For long periods this would need to be adjusted as eventually, the origin would not be even close to representing the center of the cloud, nor would gravity's effect be negligible.

It is also worth briefly considering the ionisation aspect of the model. We have assumed barium remains unionised but creation and loss terms would be present in equation (1). This would decrease the density of barium atoms (some would be turned into barium ions and electrons), perhaps an additional source of the brightness decrease although likely small as the ions are still visible. The ions would be affected by the electric and magnetic fields in the ionosphere, creating a significantly more complicated PDE to solve as the angles of these fields are not parallel or perpendicular to any of our axes the way the wind terms are. The mathematics behind this is described by Chandra (1964). This PDE would become difficult to solve using any of the previously outlined methods numerically.

5 CONCLUSIONS

This paper aimed to evaluate the usefulness of explicit finite difference schemes when modeling diffusion in the F region of the ionosphere. We began by ignoring external forces and fields, along with having the initial density of the sphere equal to a constant value (other than the sphere's edge, equal to half that of the rest of the sphere). We used an FTCS scheme and found that the accuracy was adequate; for larger times the accuracy decreased due to the scheme only being first-order accurate in time, with the largest error on the relative density being $0.2m^{-3}$ at $t = 200s$. The largest contributor to this error was the resolution of the scheme itself, as this error almost halves when we double the spatial resolution and keep the same temporal resolution. This slowed the process down by a factor of ≈ 5. The brightness had similar results: the Riemann sum method provided a reasonable level of accuracy as our brightness errors tended to be small or similar to the density error. For both the theoretical density and brightness, the computational error was ignored as it was very small compared to finite difference errors.

When comparing our radial and central brightness evolution to experimental data, it was found that ignoring meridional and zonal winds meant that the model we were using could not reasonably describe the experimental data. Explicit schemes could not be used in the modeling of the convection equation (1) due to the size of the diffusion coefficient within the F region (specifically 190km) and so for realistic systems, explicit methods are unconditionally unstable. We suggested an alternative: implicit diffusion with explicit convection. This would provide reasonable accuracy, but it too fails when considering ionisation processes, electromagnetism effects, and more.

The solution to all of this is to not use a FDM at all. The realistic models have complex geometries and boundary conditions, so it is better to use what is known as a finite element method (FEM). These involve splitting the domain into smaller subdomains where each element can take on its own shape, as opposed to the finite differencing where in our case we were only using voxels (cubes). This solves both the stability and complexity issues we were faced with when using FDM. An example of FEM being used in modeling F region physics is given by Mendillo et al. (1993). A concluding statement: explicit schemes work adequately well for basic problems, do not work for more complex problems where implicit methods must be used, and the entire class of finite difference methods must be replaced using finite element methods when considering models of actual F region physics.

REFERENCES

Abbott J. T., Austerweil J. L., Griffiths T. L., 2015, PSYCHOLOGICAL REVIEW, 122, 558

Ayers D. L., 1970, Journal of Heat Transfer, 92, 180

Baez J. C., Tweed D., 2013, Math Horizons, 21, 5

Berthelot G., Sad S., Bansaye V., 2021, Sci Rep 11, p. 14061

Boyle P., Broadie M., Glasserman P., 1997, Journal of economic dynamics and control, 21, 1267

Brin S., Page L., 1998, Computer networks and ISDN systems, 30, 107

Chandra S., 1964, Journal of Atmospheric and Terrestrial Physics, 26, 113

Dehghan M., 2001, Applied mathematics and computation, 124, 17

Green D., 2016, Procedural content generation for C++ game development. Packt Publishing Ltd

Guo H., Tang W., Liu Y., Wei W., 2010, Physical Review E, 81, 051137

Gupta R., 1998, arXiv preprint hep-lat/9807028

Hellman M. E., 1979, Scientific American, 241, 146

Hundsdorfer W. H., Verwer J. G., Hundsdorfer W., 2003, "", 33

Jindong W., Li L., Ran T., Cheng L., Yue L., Cheng B., Xie L., 2014, Chinese Journal of Space Science, 34, 837

Keller J. B., 2004, Proceedings of the National Academy of Sciences, 101, 1120

Kerr R. A., Bartol T. M., Kaminsky B., Dittrich M., Chang J.-C. J., Baden S. B., Sejnowski T. J., Stiles J. R., 2008, SIAM journal on scientific computing, 30, 3126

Martyn D., 1959, Proceedings of the IRE, 47, 147

Mendillo M., Semeter J., Noto J., 1993, Advances in space research, 13, 55

Ovaskainen O., Meerson B., 2010, Trends in ecology & evolution, 25, 643

Purwaningsih R., Arief M., Handayani N., Rahmawati D., Mustikasari A., 2018, in IOP Conference Series: Materials Science and Engineering. p. 012044

Rishbeth H., 1988, Journal of the institution of Electronic and Radio Engineers, 58, S207

Shinozuka M., 1972, Computers & Structures, 2, 855

Turchetto M., Bellingeri M., Alfieri R., Nguyen N.-K.-K., Nguyen Q., Cassi D., 2023, Mathematics, 11

Versypt A. N. F., Braatz R. D., 2014, Computers & chemical engineering, 71, 241

Wigner E. P., 1967, SIAM review, 9, 1

Zelen M., 1994, Korean Society of Preventive Medicine: Conference Proceedings, pp 250–260

Zhu X., Hu Y., Zhao Z., Ni B., Zhang Y., 2020, Earth, Planets and Space, 72, 1

6 ADDENDUM

6.1 Random Walks

Random walks are a computational method used to simulate effectively random motion. Every time an object takes a step, the probabilities of each direction it could go in are equally likely. For example, in 1 dimension, an object could go either left or right. Importantly, the length of each step the object takes and the time between each step is fixed. When the length and time between successive steps can vary, this is known as Brownian Motion.

6.1.1 Random Walks in Industry

Telecommunication networks are networks of different nodes that pass information between each other. They form the basis of most wireless communication and serve as a vital part of modern technology. An attack strategy refers to an attempt to damage or slow down the spread of information within the network. These are not always bad: non-telecommunication networks could be cancer cells in the body, so slowing the spread of disease is of course crucial and beneficial. By using random walks along the nodes of the network, researchers such as Turchetto et al. (2023) can isolate the nodes most crucial to the network, and hence the ones most useful to remove during an attack strategy.

Similarly, Google uses an algorithm developed by Brin & Page (1998) known as PageRank to identify the relative importance of a webpage. It uses a random walk to traverse through web pages linked to each other (e.g. via hyperlink) and based on the number of times it arrives at a particular webpage it assigns a relative importance value. The more times it arrives, the higher the number. Although alternative algorithms have now been designed, this is the most famous webpage ranker known.

The industry's use of random walks is not, however, limited to computer science: in finance, a model of stock prices is one of random walks. It assumes that given the number of variables that go into determining the stock price, the emergent feature of the stock's evolution resembles a random walk. This has the significant implication that the stock's future cannot be determined from historical events. Given the significance of this, many financial analysts use Monte Carlo simulations to test its validity.

6.1.2 Random Walks in Academia

Random walks have a very wide application in academia. They are frequently used in biology to study the movement of animals, as described by Berthelot et al. (2021). It is suggested that given certain pre-existing conditions, the spatial complexity of animals in all different types of habitats can be explained by them taking a random walk. It gives researchers estimates on animal abundance and density for a certain space. This is very useful in better understanding animal behaviour, and provides a remarkably simple way of doing it.

Understanding complex decision-making via random walks is not limited to animals: it also provides a way of understanding human memory recollection. Abbott et al. (2015) propose that when humans remember things, they tend to retrieve semantically similar memories as a cluster. This suggests that the retrieval of the memory is not a strategic process but rather a random "foraging" in roughly the right area. This closely resembles the process of a random walk, where memories/nodes are randomly jumped between until the correct memory is located.

Physical sciences also frequently rely on random walks. A good example is solving the diffusion equation: as demonstrated by Keller (2004), although on large scales the diffusion may have a preferred direction, microscopically a single element of the diffusing material undergoes a violent and supposedly random movement from colliding with other objects. This element's motion closely resembles a random walk, and macroscopically solutions to the diffusion equation can be found by modeling the diffusing material using a random walk process.

6.2 Monte Carlo Simulations

There are often problems in science that either have no analytical solution or one that is too complex to reasonably evaluate. A popular alternative to solving these problems is to obtain a statistical answer: one simulates the problem many times using random sampling and an approximate answer emerges. These are known as Monte Carlo simulations. An intuitive example is Monte Carlo integration, where a large number of points are generated within both an unknown and known area. By observing the ratio of points within the area to the total number of points, an estimate of the unknown area may be obtained. As with most Monte Carlo simulations, this relies on a large number of simulations: if there are only a few points generated, the accuracy of the area will be poor.

6.2.1 Monte Carlo in Industry

Structural engineering has to account for all the seemingly random effects of daily life and ensure that structures will not receive damage from them. An example of this is described by Shinozuka (1972), in which Monte Carlo simulations are run to determine a structure's response to different vibrations and excitations. Note that some of the analysis done in this paper cannot be done analytically, highlighting the importance of simulations in this field.

Monte Carlo is often used to generate risk value i.e. financial loss on a product due to certain parameters. An example of this is given by Purwaningsih et al. (2018), where the loss of the price per kilogram of poultry is simulated using Monte Carlo techniques. It uses historical data to generate random numbers that affect the production e.g. rainfall, and then re-runs the simulation many times to get the average loss value.

Monte Carlo methods are some of the most commonly used statistical modeling methods in all of finance. An example of this would be asset pricing: one simulates an asset price, stimulates it based on various probability factors, and then averages the different cash flows over different paths created by the simulation. This is often achieved using Monte Carlo integration, as the mathematics behind a process like this involves multi-dimensional integrals: often these integrals are evaluated over many domains and so it quickly becomes preferable to other numerical integration techniques. This is described in more detail in Boyle et al. (1997).

6.2.2 Monte Carlo in Academia

Lattice QCD is a numerical method of solving the interactions between quarks and gluons in hadrons, as described in Gupta (1998). Due to the large and complex integrals needed to extract meaningful values from the theory, the integrals are instead evaluated over a grid/lattice using Monte Carlo integration. Whilst it is still expensive and often requires supercomputers to evaluate, the Monte Carlo method of integration allows approximations of the integrals to be made.

Another use of Monte Carlo simulations comes from biology: often it is impossible to simulate reactions involving enzymes, cells, etc. at an atomic level, but Monte Carlo simulations are particularly suitable as they are less spatially dependent. An example is given by Kerr et al. (2008), where the diffusion and solution processes are simulated on cellular membranes. It achieves highly accurate results and is not particularly computationally expensive.

One of the most powerful techniques using MC is known as Markov Chain Monte Carlo (MCMC). It allows for parameters to be estimated based on how well these parameters describe a previous set of data. An example is shown in Baez & Tweed (2013), where MCMC is applied to global simulations of climate science data. Similar to the previous example, it is too computationally expensive to simulate the problems with physical accuracy, but MCMC techniques allow for a system's evolution through time to be estimated on relatively accurate parameters e.g. how clouds interact with the ocean. The parameters are randomly sampled and repeatedly adjusted until they comfortably describe the posterior dataset.

6.3 Random Numbers

Random number sequences follow two conditions: the values are uniformly distributed over a defined interval, and future ones cannot be predicted from previous ones. Computationally, random numbers are extremely difficult to produce due to the second requirement. For that reason, programs that utilise random numbers use what are known as "pseudo-random numbers", where a sufficiently complex algorithm produces what appears to follow the two conditions. Random numbers are a vital part of science, and we'll explore some of the ways they're used now.

6.3.1 Random Numbers in Industry

Procedural generation refers to data generation algorithmically, rather than manually. It speeds up and diversifies gameplay as it usually relies on random numbers. As an example, rather than generate an entire open world for a player, one can generate a small amount of content and then as the player explores the bounds of the pre-generated content, new procedurally generated content of the map can appear. The randomly generated number may refer to the height of a mountain or the number of branches on a tree. This is described in detail in Green (2016).

Random numbers also play a vital role in healthcare, with a particular focus on clinical trials. During clinical trials it is vital that when different patients are assigned to certain roles or groups, there is as little human bias in the assignment as possible. Random numbers help significantly with this, as although they are never completely random, the algorithms are sufficiently complex that the desired effect of randomness is achieved. This is described in Zelen (1994).

One of the most important industrial uses of random numbers is cryptography. Cryptography aims to encode a message that can only be decoded by the intended recipient. A common cryptography method is the "lock and key" method, where the encrypter and recipient both have a one-time use key to unlock the encryption. These keys are often numbers and to prevent someone from being able to guess the number, the key is generated using a random number algorithm. This makes it significantly harder to hack an encryption or predict the value of the key in a future encryption, as explained in Hellman (1979).

6.3.2 Random Numbers in Academia

Stochastic models refer to models that depend on random variables. These are popular modeling methods given that they do not need to be deterministic. The random variables are of course generated by random numbers. These models are often found in modeling ecological processes such as how the extinction rate of an animal's population depends on both biotic and abiotic factors. The parameters that control these factors are generated by random number generators, simulating the inherent randomness of large scale biology, as described in Ovaskainen & Meerson (2010).

When describing complex nuclei, it is often difficult to rely solely on the physics of their interactions. In particular, the distributions of their energy levels are of great difficulty to mathematically replicate. A solution to this is what is known as a random matrix: by generating a matrix of random numbers the statistical properties of complex nucleon's energy levels can be better replicated. The numbers within the matrices are of course generated using random number generators. Whilst the mathematical detail is too complex to explain here, it is shown in detail by Wigner (1967).

Matrices aren't the only physical application of random numbers. When taking any measurement, there will always be "noise", which refers to a background uncertainty in your measurement. An informal example of this is talking to a friend outside; despite listening to them, the wind, cars, and all other background features will be present underneath the friend's speech. To simulate this inherent randomness, random numbers are fantastic as they accurately represent the feature we want: inherent unpredictability. The opposite is also true, as the randomness of noise can be used to generate random numbers. An example is described by Guo et al. (2010).

This paper has been typeset from a TEX/LATEX file prepared by the author.