Computational Physics

Project 7: Diffusion Equation & Entropy

Due: April 9th 2018

George Davila[1]*

1 Department of Physics, University of Central Florida, 4111 Libra Drive, Physical Sciences Bldg. 430, 32826, Orlando, United States

Abstract: Here we discuss project #7. We review the context of the problem, its relation to real physical systems, and the methodology by which we tackled its solution. All figures are appended at the end of the paper.

1. Introduction

Entropy is one of the most important, profound and useful concepts in all of physics.The conceptual foundations of entropy can be built up from an information science perspective as a simple consequence of statistics. And yet it is pervasive throughout physics. Of course we have the 2nd Law of thermodynamics

$$dS \geq 0 \text{ (closed system)} \tag{1}$$

The 2nd Law of Thermodynamics is one of the very few laws consistent throughout all of physics, from materials science to relativistic plasma dynamics to quantum gravity. It even—and perhaps most especially—holds when we consider then entropy of the universe as a whole. Recent results in quantum gravity [4] even suggest that much of what we know to be classical general relativity can be derived as a corollary of the 2nd Law applied to quantum entanglement entropy:

$$S = -\text{tr}(\rho \ln \rho) \quad (\rho \text{ is the density matrix here}) \tag{2}$$

also commonly called the von Neumann or Renyi entropy. It's essential the negative of the expectation value of the operator $\ln \rho$ in the mixed state formalism. For quantum pure states (or in the eigenbasis) this looks very

* *E-mail: GDavila@knights.ucf.edu*

much like a standard thermodynamic entropy (asides from the k_B constant)

$$S = -\sum_j \lambda_j \ln \lambda_j \tag{3}$$

And indeed, even highly quantum systems are subject to the 2nd Law, whereas they manage to differ and often surpass classical systems in a great number of other ways.

Since entropy is one way to measure informational complexity, not simply reserved for the study of physical systems, we may perform a study of entropy on simulated systems.

Given a Hilbert space $\mathcal{H}$ with dimension $\mathcal{N}$ that fully describes the system, we then have the following relationship for the informational entropy $\mathcal{S}$

$$\dim\mathcal{H} = \mathcal{N} = e^{\mathcal{S}} \tag{4}$$

A description which is not limited to physical systems. A random walker simulation can be assigned such a Hilbert space description. Consider a 1d simulation of a single walker on a unit lattice that can be anywhere in a box of given length L (perhaps L just being how far it can walk in a given time times 2). Then we have 1 spin-like degree of freedom (walker there or not there) per site for L sites so we have a Hilbert space of maximal size

$$\dim\mathcal{H} = \mathcal{N} = e^{\mathcal{S}} = 2^L \tag{5}$$

And we can, of course, constrain the Hilbert space and the entropy by tracking the actual motion of the walker.

2. Analytics

Remark 2.1.
We frame our discussion in terms of walkers. Although we don't necessarily need to track walkers to implement the codes, we deal with a system that effectively models a system of random walkers. So speaking in terms of walkers allows us to contextualize the underlying processes a bit more effectively.

2.1. Diffusion

(This subsection follows the analytics for Project 6, follow the same theory)

Multiple particles are placed in a volume and allowed to undergo random walks. For this type of system, density is a good quantity to look at. While we can't say much about individual trajectories, we can make predictions about the density of the system in general time regimes. At the beginning, in the low time regime, it should hover around the initial density for a bit while slowly dissipating. As the walkers have more time to

move about the system will become more and more diluted and the density should tend to 0. We recover these behaviors in the equations below.

Lets review—very briefly—some of the fundamental theory following Einstein's 1905 paper [5] (Einstein uses f in the paper but we use ρ below) . We can expand the density as

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t+\tau) = \int_{-\infty}^{+\infty} \rho(x+\Delta,t)\cdot\varphi(\Delta)\,\mathrm{d}\Delta \tag{6}$$

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t)\cdot\int_{-\infty}^{+\infty}\varphi(\Delta)\,d\Delta + \frac{\partial\rho}{\partial x}\cdot\int_{-\infty}^{+\infty}\Delta\cdot\varphi(\Delta)\,\mathrm{d}\Delta \tag{7}$$

$$\rho(x,t) + \tau\frac{\partial\rho(x)}{\partial t} + \cdots = \rho(x,t) + \frac{\partial^2\rho}{\partial x^2}\cdot\int_{-\infty}^{+\infty}\frac{\Delta^2}{2}\cdot\varphi(\Delta)\,\mathrm{d}\Delta + \cdots \tag{8}$$

Rearrange the last equation as

$$\tau\frac{\partial\rho(x)}{\partial t} = \frac{\partial^2\rho}{\partial x^2}\cdot\int_{-\infty}^{+\infty}\frac{\Delta^2}{2}\cdot\varphi(\Delta)\,\mathrm{d}\Delta + (\text{higher order terms}) \tag{9}$$

Represent the remaining integral as some function D (it will soon be apparent why we choose this symbol):

$$\tau\frac{\partial\rho(x)}{\partial t} = \frac{\partial^2\rho}{\partial x^2}\cdot D + (\text{higher order terms}) \tag{10}$$

Minus the higher order terms this is not simply a Diffusion equation with a diffusion coefficient D. And if we take D to be constant, it simply becomes the heat equation [6]. Then, starting from a time $t = 0$, the density of the system will evolve in a Gaussian fashion:

$$\rho(x,t) = \frac{N}{\sqrt{4\pi Dt}}e^{-\frac{x^2}{4Dt}} \tag{11}$$

Take the large time limit

$$\rho(x, t\to\infty) = 0 \tag{12}$$

Moreover, in the low time regime, the density decays fairly slowly. So the expected behaviors are recovered. The crossover regime is $x^2 \sim 4Dt$, so in order to prevent the system from dispersing too quickly we could make D very small.

This tells us much about how we might expect our coffee equation to behave. We should expect a fairly symmetric and fairly Gaussian diffusion of our simulated particles. But it is important to note that our systems use a relatively small number of particles $N \sim 10^2$, so they should be expected follow these rules to a rough degree, but by no means exactly. Of course, if we ran an $N \sim 10^{23}$ simulation we would almost certainly see a near-exact adherence to any degree of error we might be reasonably concerned with.

2.2. Entropy

Per unit time t each of n initial walkers can take one step on a 2d grid. For n non-interacting walkers (so that two or more can occupy the same site on the lattice) each with individual Hilbert space $\mathcal{H}_i$, we have a total separable Hilbert space $\mathcal{H}$

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \mathcal{H}_3 \otimes \cdots \otimes \mathcal{H}_{n-1} \otimes \mathcal{H}_n \tag{13}$$

But we deal with walkers with identical Hilbert spaces $\mathcal{H}_w \equiv \mathcal{H}_1 = \mathcal{H}_2 = \mathcal{H}_3 = \cdots = \mathcal{H}_{n-1} = \mathcal{H}_n$, so

$$\mathcal{H} = (\mathcal{H}_w)^{\otimes n} \tag{14}$$

And then the max size of a single walkers' Hilbert space $\mathcal{H}_w$ is given as

$$\dim\mathcal{H}_w = (4 \text{ directions})^{(1 \text{ step per unit time})\cdot t} = 4^t \tag{15}$$

Since each walker has 4 possible moves it can make. Then the total Hilbert space is size

$$\dim\mathcal{H} = n \cdot 4^t \tag{16}$$

Is the maximum possible size of the Hilbert space corresponding to the maximum possible informational entropy at a given time for a given number of walkers.

If we alter this by instead having n_i walkers added at time t_i, $i \in \{0, \ldots, k\}$ s.t. $t_0 = 0$, until a final time t then the walkers contribute to the Hilbert space in a similar way:

$$\dim\mathcal{H} = n_0 4^t + n_1 4^{(t-t_1)} + n_2 4^{n_2(t-t_2)} + \cdots + n_{k-1} 4^{\left(t-t_{k-1}\right)} + n_k 4^{(t-t_k)} \tag{17}$$

Giving us a *maximal* possible entropy

$$\mathcal{S}_{\max} = \ln\left[n_0 4^t + n_1 4^{(t-t_1)} + n_2 4^{n_2(t-t_2)} + \cdots + n_{k-1} 4^{\left(t-t_{k-1}\right)} + n_k 4^{(t-t_k)}\right] \tag{18}$$

Which is the absolute maximum entropy for an unbounded walk under the given scenario. Obviously this upper bound can become extraordinarily large for relatively small time steps (e.g. $4^{100} \approx 1.6 \cdot 10^{60} \implies \mathcal{S}_{\max} = \ln\left(n \cdot 4^{100}\right) \approx \ln(n) + 138$). So, for purposes of practicality, it is necessary to constrain the entropy. There are a few ways we can constrain the entropy

1. Bound the walk to some area, or so that the density or some other property is conserved (or obeys some well-defined relation).

2. Constrain where the walkers start from.

3. Regard the walkers as indistinguishable from one another. So when 2 walkers run into each other their Hilbert spaces collapse as $2^t \times 2^t \to 2 \cdot 2^t$ from then on forward.

4. Regard grid points as either occupied or unoccupied, don't care about multiple occupancy.

5. Update the available info. This effectively collapses the Hilbert space.

This project effectively applies constraints $\#2, 3, 4$. This is why our measured entropies are significantly lower than this upper bound.

3. Computations

3.1. Diffusion Equation Computations

We generate the Diffusion equation plots in *Mathematica*, as the computer algebra software allows us to better manage piece-wise functions. These plots are in the file named *DiffusionEqnCompPhysics.pdf*. A Fortran 95 form of these simulations can be seen in the Project 6 report. We append these codes as Fig. 6 and Fig. 7 for reference.

We considered 5 distinct cases, where we changed the shape of the initial distribution $\rho(x, t = 0)$ each time. We scale the plots so that any unnecessary constants are taken to be 1. The initial distributions used are:

1. Gaussian distribution $\rho(x, 0) = \exp(-x^2)$ (pg. 2 of *DiffusionEqnCompPhysics.pdf*)

2. Uniform distribution $\rho(x, 0) = 1$ for all x (pg. 3 of *DiffusionEqnCompPhysics.pdf*)

3. Piecewise distribution $\rho(x, 0) = 1$ for all $|x| \leq 1/2$, 0 otherwise (pg. 4 of *DiffusionEqnCompPhysics.pdf*)

4. Piecewise distribution $\rho(x, 0) = 0$ for all $|x| \leq 1/2$, 1 otherwise (pg. 5 of *DiffusionEqnCompPhysics.pdf*)

5. Dirac delta distribution $\rho(x, 0) = \delta(x)$ (pg. 6 of *DiffusionEqnCompPhysics.pdf*)

The outputs for each case are as follows:

- Analytic solution for the initial distribution
- 2D plot showing evolution at various time steps, displaying how each of these evolve in 1D
- 3D plot showing the solution in 2D space

3.2. Entropy Computations

For the entropy simulations, the codes used & the simulation plots produced are shown in the figures appended.

A simulation with no time steps ($nt = 0$) would be trivial, since the code we constructed in class is designed to measure entropy at sufficiently large nt. The code designed in class doesn't yield any results for an input of $nt = 0$, nor should it.

Properties:

1. Time-dependence - Varying numbers of times steps nt
2. 'Random walk' (we deal with a system analogous to a bunch of random walkers)

We use & show in the figures appended a variety of time steps:

1. $nt = 100$ - Fig. 2
2. $nt = 400$ - Fig. 3
3. $nt = 700$ - Fig. 4
4. $nt = 900$ - Fig. 5

4. Discussion & Conclusions

Looking at the diffusion plots, the solutions behave exactly as me might expect. They undergo relatively slow,symmetric, decay.

Looking at Figs. 2, 3, 4, and 5, it is clear that their entropies reflect a similar underlying dynamics. It's simply the scale that changes between plots. We can tell that the entropies are logarithmic not just from the shapes of these plots, but also the fact that

This reflects the fact that, for all of these instances, the underlying Brownian motion is similarly stochastic.

These plots also show that no major *emergent* behavior comes into effect for Brownian motion at these scales. We should expect this since the particles don't interact and the motion is random. If, however, we allowed the particles to 'interact' in some way—such as not letting them occupy the same points—we would see a notable decrease in the entropy compared to the cases we analyzed here. This would also very likely make the plots more dependent on scale. Longer running simulations would have more interactions, corresponding to decrease from maximal entropy. To the contrary we might not see much interaction for low time scales. For example, if we placed interacting walkers at a distance such that they could only reach each other after 100 time steps, the subsequent $nt = 100$ plot would look very much like it would for similarly situated non-interacting walkers.

References

[1] S. Stolbov, Computational Physics Class Notes

[2] B. Ydri, *Computational Physics: An Introduction to Monte Carlo Simulations of Matrix Field Theory*, arXiv:1506.02567 [hep-lat] (June 2015)

[3] K. Anagnostopoulos, *Computational Physics Volume 2: A Practical Introduction to Computational Physics and Scientific Computing* pg. 91 (Aug. 2014)

[4] ChunJun Cao, Sean M. Carroll, Spyridon Michalakis, "Space from Hilbert Space: Recovering Geometry from Bulk Entanglement," Phys. Rev. D 95, 024031 (2017).

[5] A. Einstein, "On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat" PhD thesis (1905). 1956 translation by R. Furth & A.D. Cowper.

[6] Weisstein, Eric W. *Heat Conduction Equation.* From MathWorld–A Wolfram Web Resource.

Figure 1. **One of the codes used, entitled *ent.f*, that we wrote in class.**

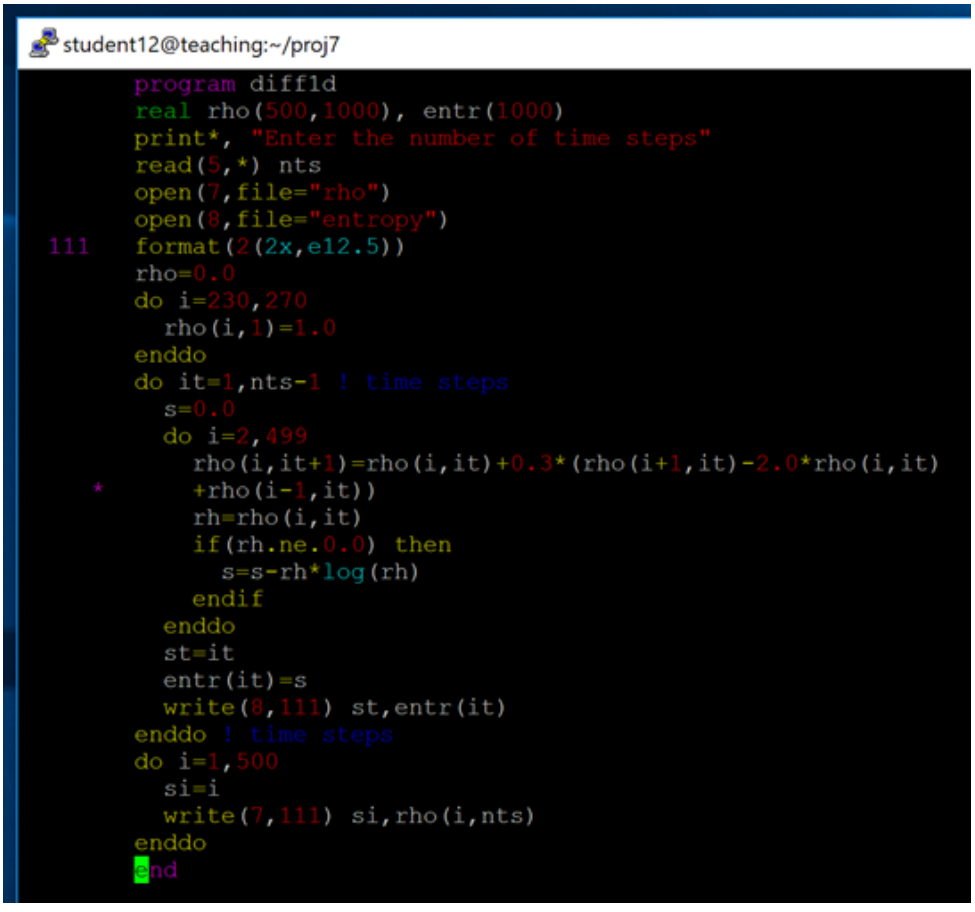

Figure 2. **Simulation for time-dependent random walk case. Number of time steps is $nt = 100$.**

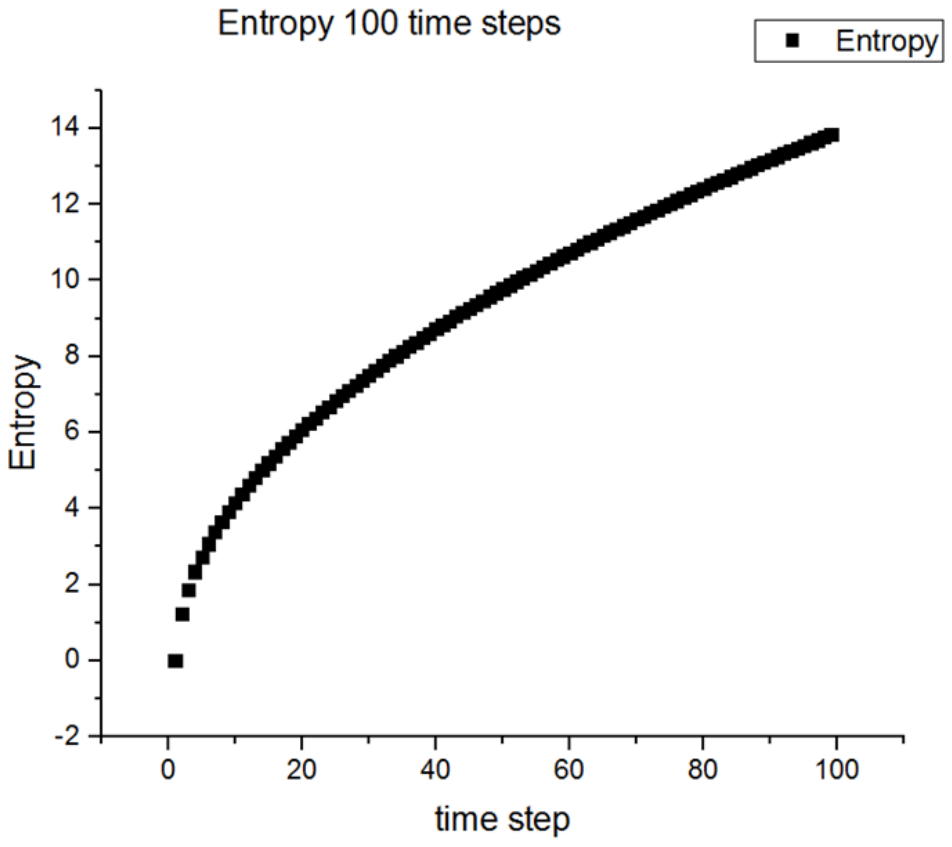

Figure 3. **Simulation for time-dependent random walk case. Number of time steps is $nt = 400$.**

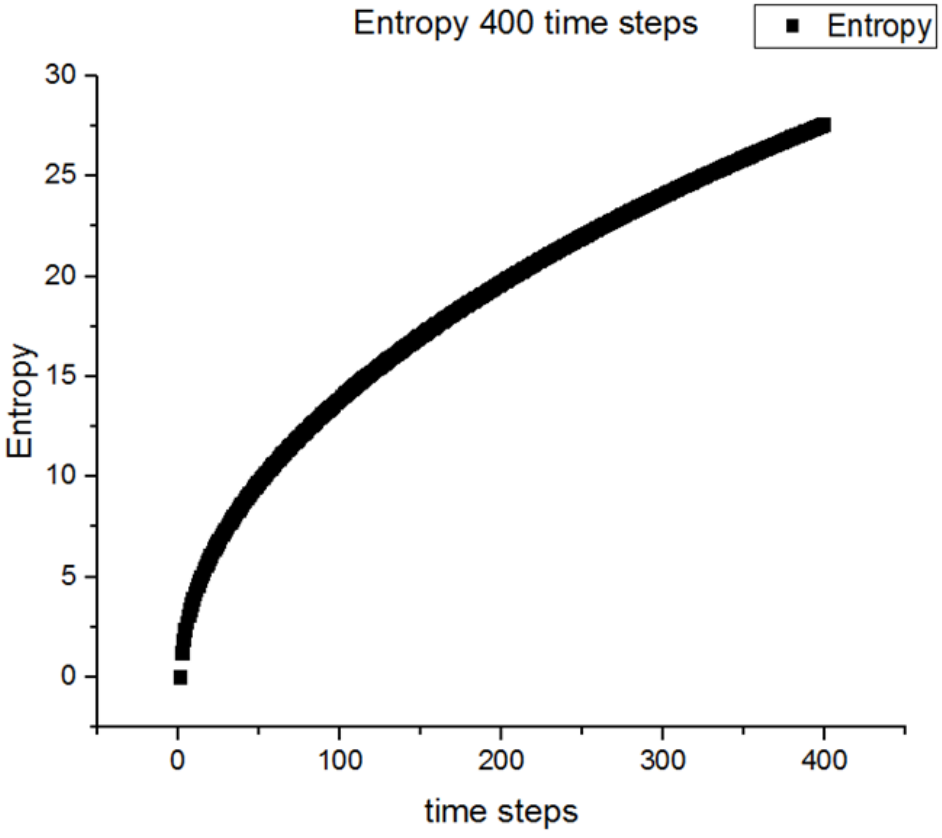

Figure 4. **Simulation for time-dependent random walk case. Number of time steps is** $nt = 700$.

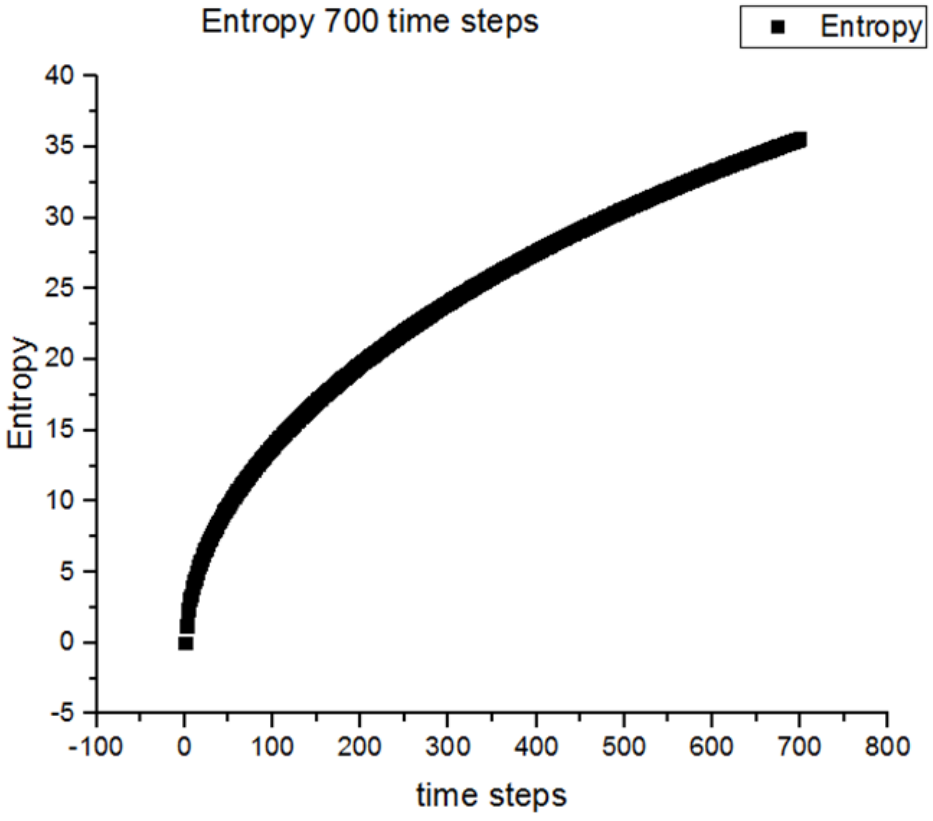

Figure 5. **Simulation for time-dependent random walk case. Number of time steps is $nt = 900$.**

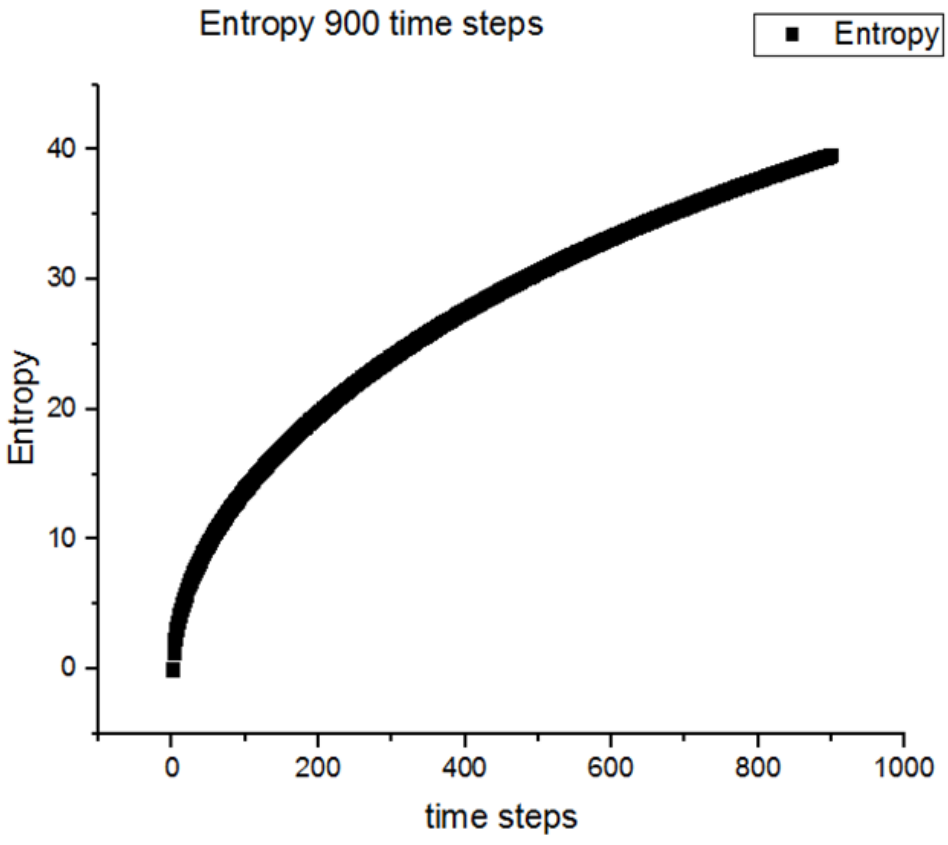

Figure 6. Project 6 code, entitled *coffee_notimestep.f*, that we wrote in class. For full details see project 6 report.

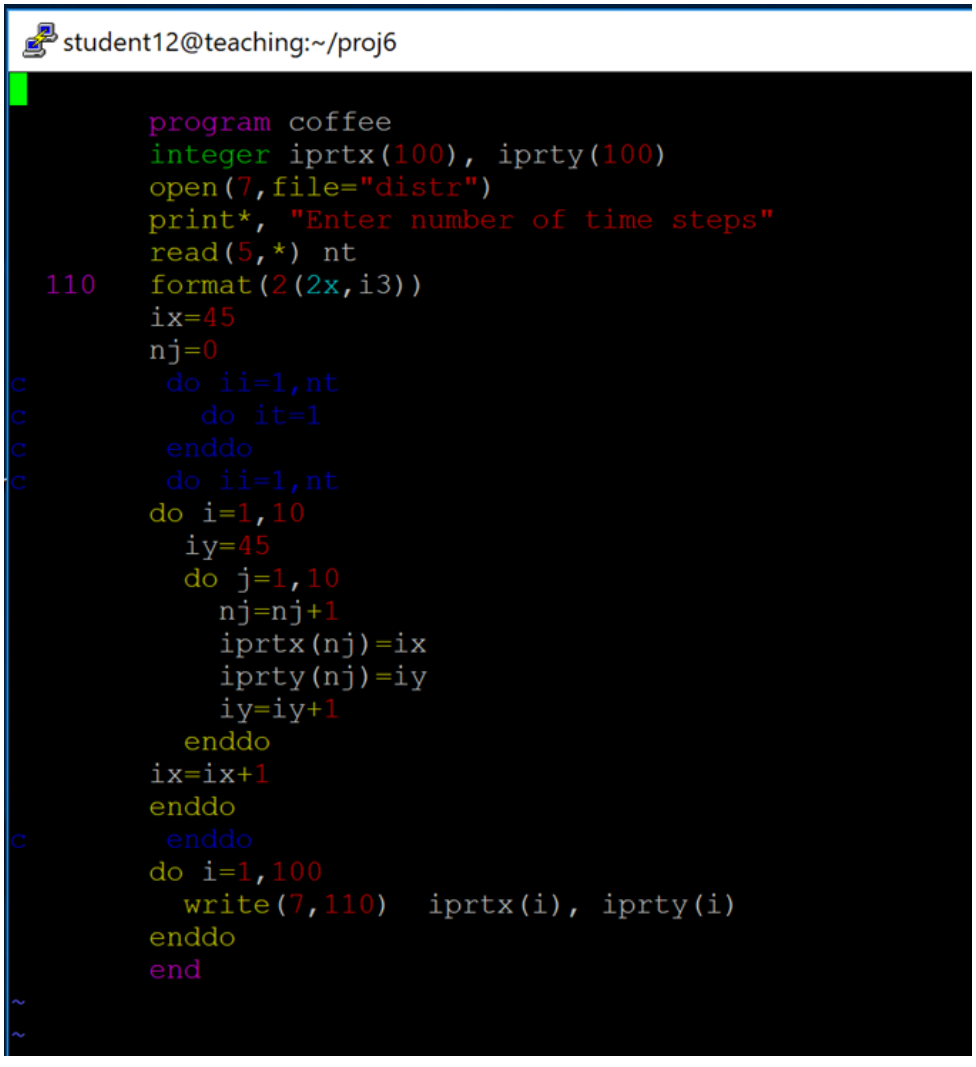

Figure 7. Project 6 code, entitled *coffee.f*, that we wrote in class. This code uses random walks, which can be seen by the inclusion of the *rand* function. The other code *coffee_notimestep.f* has no such pseudo-random number generating function in it. For full details see project 6 report.

student12@teaching:~/proj6

```
      program coffee
      integer iprtx(100), iprty(100)
      integer part_index , posx0 , posy0
cc    empty line
      open(7,file="2Dsnapshot")
  110 format(2(2x,i3))
      print*, "Enter number of time steps"
      read(5,*) nt
!  110  format(2(2x,i3))
      posx0 = 45
      posy0 = 45
      do i = 1,10
        do j = 1 , 10
          part_index = (i-1) * 10 + j
          iprtx(part_index) = posx0 + (j-1)
          iprty(part_index) = posy0 + (i-1)
        enddo
      enddo
cc    Make particles move in random directions
      do it =1 , nt + 1
        r1 = rand(0) !generates random number
        nj = int(r1*100)
        r2 = rand(0)
        if(r2.lt.0.5) then !"If r2 less than 0.5"
          if(r2.gt.0.25) then
            iprtx(nj)=iprtx(nj) + 1 !moves right if random 0.25<r2<0.5
          else
            iprtx(nj)=iprtx(nj) - 1 !moves left if random 0<r2<0.25
          endif
        else
          if(r2.gt.0.75) then
            iprty(nj)=iprty(nj) + 1 !moves up if random 0.75<r2<1.0
          else
            iprty(nj)=iprty(nj) - 1 !moves down if random 0.5<r2<0.75
          endif
        endif
      enddo
cc    Save the data to the file
      do i= 1 , 100
        write(7,110) iprtx(i) , iprtx(i)
      enddo
      end
~
~
```