Fermi Project Update

Ethan Savitch

June 2020

Contents

1 Introduction

Goal: make the probe range and velocity normally distributed.
Method: Add a local variable called technology to the program, use random walks to have technology be normally distributed around the technological capabilities of the abiogenesis seed. Then make probe range and velocity functions of this new technology variable.

2 Technology

2.1 Ornstein-Uhlenbeck (OU) Process

We use an Ornstein-Uhlenbeck[1] process to calculate the random walk for technology:

$$\boxed{dx_t = \theta(\mu - x_t)dt + \sigma dW_t} \tag{1}$$

μ is the location that everything will eventually end up at. If $\mu = 0$, then their is an restoring force towards the origin, that increases with increased distance from the origin. The higher θ is, the quicker μ is reached.[2]

$$x_t = x_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{-\theta(t-s)} dW_s \tag{2}$$

[1]https://en.wikipedia.org/wiki/Ornstein-Uhlenbeck_process
[2]We have used the change of variables: $s = e^{2\theta t}$

x_0 is the initial location. It follows that the first moment, or mean of x_t is given by:

$$E[x(t)] = \overline{x}(t) = x_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) \tag{3}$$

It can similarly be shown that:

$$Cov[x(t), x(s)] = \frac{\sigma^2}{2\theta}\left(e^{-\theta|t-s|} - e^{-\theta|t+s|}\right) \tag{4}$$

It follows that:

$$Var[x(t)] = Cov[x(t), x(t)] = \frac{\sigma^2}{2\theta}(1 - e^{-2\theta t}) \tag{5}$$

2.1.1 Long-Term Gaussian Behavior ($t >> 1$)

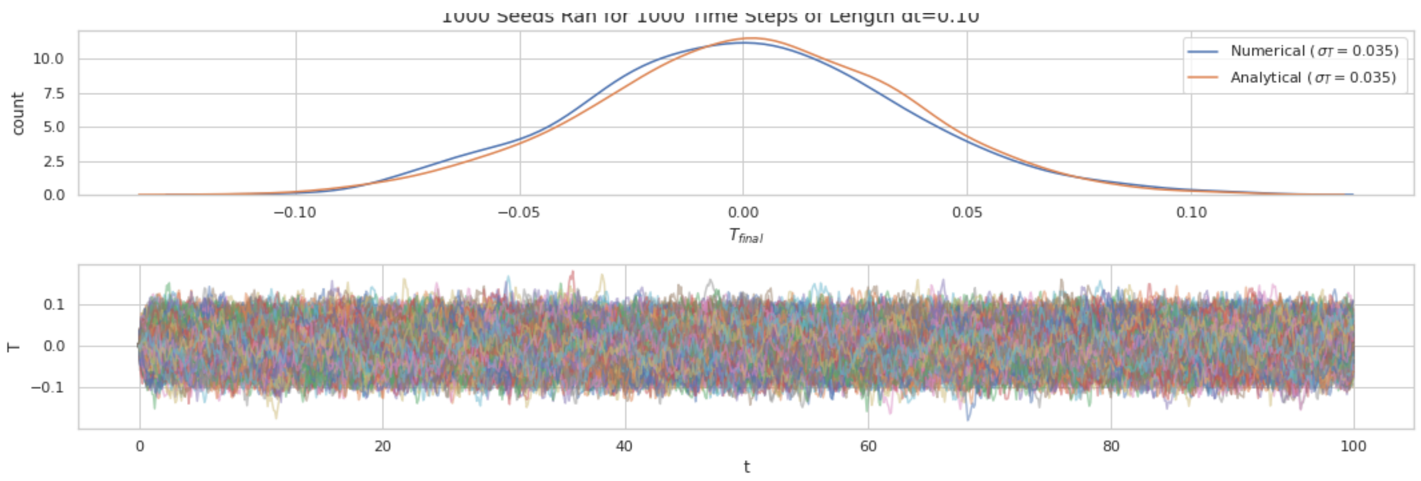
$$\boxed{E[x(t >> 1)] = \mu} \tag{6}$$

$$\boxed{Var[x(t >> 1)] = \frac{\sigma^2}{2\theta}} \tag{7}$$

2.2 Euler-Maruyama Method

We can use a numerical method called the Euler-Maruyama method to simulate the OU process.

```
#-------------------------------------time (t)-------------------------------------------
num_seeds = 1000
ti = 0            #initial time
tf = 100          #final time
N  = 1000         #number of time steps
#-----------------------------------Technology (T)------------------------------------
iTech   = 0      #value to start from
mu      = 0      #value to converge to
sigma   = 0.05   #randomness; directly proportional to width of gaussian
theta   = 1      #rate of convergence; inversely proportional to width of gaussian
#-----------------------------------Initialize------------------------------------------------------
dt = float(tf-ti)/N #length of each time step
timeList = np.arange(ti, tf, dt)
techList = np.zeros(N)
techList[0] = iTech                                  #all seeds have the sdame initial position
fTechList = [] #final value of technology
#--------------------------------Calculate Analytically--------------------------------------------
mean  = mu  #mean value for large t
stdev = sigma/np.sqrt(2*theta) #width as a normal gaussain
techListAnalytic = np.random.normal(loc=mean, scale=stdev, size=num_seeds)
#--------------------------------Calculate
    Numerically-------------------------------------------------------
fig, (ax2,ax1) = plt.subplots(figsize=(15,5), nrows=2, ncols=1)
for _ in range(num_seeds):
    for i in range(1, len(timeList)): #i starts at 1
        t = (i-1) * dt                             #current time
        tech = techList[i-1]                       #current technology value
        dW = np.random.normal(0, np.sqrt(dt))      #Gaussian Noise (mean=0, stdev=sqrt(dt))
#        dW = np.random.uniform(-1,1) # Uniformly Sampled Noise U(-1,1)
        #---------------------------------------------------------------------------------------
        push  = sigma * dW                         #displacing force
        pull  = theta*(mu-tech)*dt                    #restoring force
        techList[i] = tech + push + pull                 #next position
    fTechList.append(techList[-1])                        #array keeping track of final values
#-----------------------------------Plot Output-------------------------------------------------------
    ax1.plot(timeList, techList, alpha=.5)
    ax1.set(xlabel='t', ylabel='T')
sns.kdeplot(fTechList, ax=ax2, label=fr'Numerical ($\,\sigma_T = {np.std(fTechList):.3f}$)') #
    Numerically Calculated
sns.kdeplot(techListAnalytic, ax=ax2, label=fr'Analytical ($\,\sigma_T = {stdev:.3f}$)') # Analytically
    Calculated
ax2.set(xlabel=r'$T_{final}$', ylabel='count')
ax2.legend();
plt.tight_layout()
plt.suptitle(f"{num_seeds} Seeds Ran for {N} Time Steps of Length dt={dt:.2f}", y=1.01);
```



3 Probe Velocity ($v(T)$) as Function of Technology (T)

So general idea here is that total (specific) kinetic energy will scale with technology, then solving for velocity will give $v(T)$. Since the units of specific kinetic energy are the same as velocity squared, we can write the constant of proportionality in terms of the (approximate) initial probe velocity. I now make the following assumptions:

1. Technology starts from $T = 0$
2. When $T = 0$, the initial probe velocity is $v(0) \equiv v_0$ and $v_0 << c$. Thus, when technology is zero, kinetic energy is purely non-relativistic ($K = \frac{1}{2}mv^2$).
3. Specific kinetic energy scales exponentially with technology.[3]

With these assumptions, I can write the scaling relation like:

$$\boxed{Specific\ Kinetic\ Energy \equiv K/m \propto e^{2T} = \left(\frac{v_0^2}{2}\right) e^{2T}} \tag{8}$$

3.1 Non-Relatavistic Kinetic Energy

In this case, kinetic energy is given by:

$$K = \frac{1}{2}mv^2$$

[3]the factor of two helps clean up the solutions

Thus, specific Kinetic Energy is:

$$K/m = \frac{v^2}{2} = \left(\frac{v_0^2}{2}\right) e^{2T}$$

Solving for v and taking the positive solution yields our equation for non-relativstic probe velocity as a function of technology:

$$\boxed{v(T) = v_0\, e^T}$$

3.2 Relativistic Kinetic Energy

In this case, kinetic energy is given by:

$$K = m_0 c^2 \left(\frac{1}{\sqrt{1-(v/c)^2}} - 1\right) \tag{9}$$

Thus, specific relativistic kinetic energy is:

$$K/m_0 = c^2 \left(\frac{1}{\sqrt{1-(v/c)^2}} - 1\right) = \left(\frac{v_0^2}{2}\right) e^{2T} \tag{10}$$

Solving for v and taking the positive solution yields our equation for non-relativistic probe velocity as a function of technology:

$$\boxed{v(T) = \frac{c v_0 e^T \sqrt{4c^2 + \left(v_0 e^T\right)^2}}{2c^2 + \left(v_0 e^T\right)^2}} \tag{11}$$

- $v(T) = Probe\ Velocity$
- $T = Technology$
- $c = Speed\ of\ Light = 2.99 * 10^{10}\ cm/s$
- v_0=*Approximate Initial Probe Velocity, assuming an initially non-relativistic kinetic energy*
  (ie: $v(0) = v_0$ iff $v_0 << c$)

4 Probe Range ($r(T)$) as Function of Technology (T)

We can derive the equation for probe range directly from our previously derived expression for relativistic velocity. We want the only relevant input parameter here to be the maximum time a probe can spend in space, which we can call the proper time t_0. If we call the probes velocity v and we define:

$$\gamma = \frac{1}{\sqrt{1-(v/c)^2}} \tag{12}$$

then from relativity it follows that the dilated time a probe will be in space as observed on the planet which sent this probe out will be:

$$t = t_0 \gamma \tag{13}$$

Additionally, if as observed from this planet the destination of the probe is a distance r_0 away, then in the frame of the probe the contracted distance to destination is given by:

$$r = \frac{r_0}{\gamma} \tag{14}$$

We proceed by solving for gamma in the above two equations and equating:

$$\gamma = \frac{r_0}{r} = \frac{t}{t_0} \tag{15}$$

So solving for the contracted probe range yields:

$$r = t_0 \left(\frac{r_0}{t}\right) \tag{16}$$

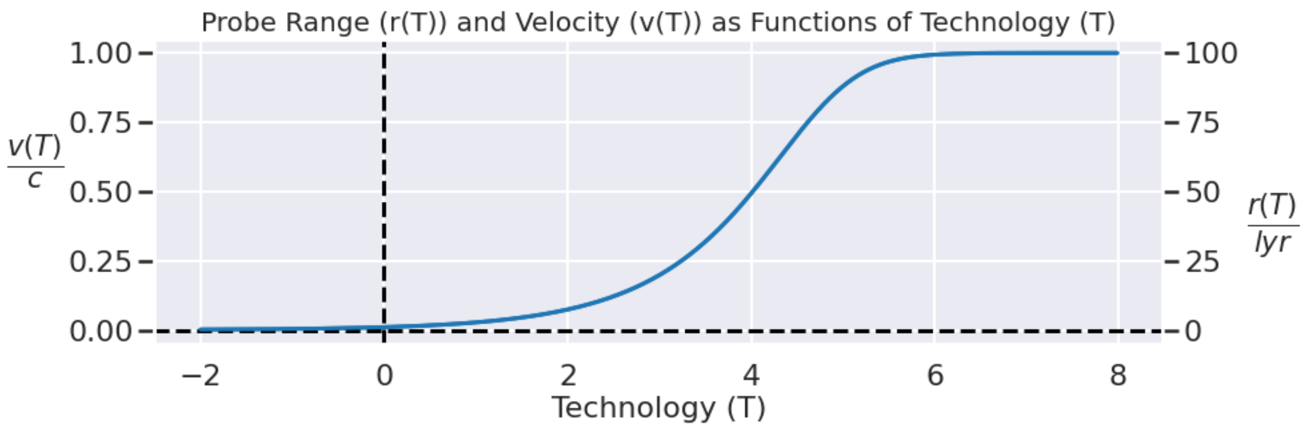
Finally, the last step is to notice that r_0/t is the distance the probe travelled over the time it took to travel that in the frame of planet which sent the probe out, thus this is the same as our velocity calculated in equation 11. Thus, our final probe range as a function of technology becomes:

$$\boxed{r(T) = t_0 v(T) = t_0 \left(\frac{c v_0 e^T \sqrt{4c^2 + \left(v_0 e^T\right)^2}}{2c^2 + \left(v_0 e^T\right)^2}\right)} \tag{17}$$

- $v(T)$ is the probe velocity as shown in equation 11
- t_0 is the maximum time a probe can stay in space

4.1 Example: $t_0 = 100yr,\ v_0 = 0.01c \Longrightarrow r_0 = v_0 t_0 = 1lyr$

If we assume that initially, when $T = 0$, that probes have velocities approximately 1% the speed of light and the maximum time that a probe can stay in space is 100 years then $t_0 = 100yr$ and $v_0 = 0.01c$. Additionally, for simplicity I let $c = yr = 1$. In this case the plots for $r(T)$ and $v(T)$ are shown below.



5 Next Steps

1. Implement all of this in the actual fortran model. All of the above was modeled using python in order to play around with different ideas.
2. Determine what additional parameters we want to have normally distributed.