

Fermi-Pasta-Ulam-Tsingou Problem

Simon Austin Lee Ian May
Jack Baskin School of Engineering
University of California, Santa Cruz
siaulee@ucsc.edu, imay1@ucsc.edu

<https://github.com/Simonlee711/Research/tree/master/Fermi-Pasta-Ulam-Tsongu>

Abstract

The **Fermi-Pasta-Ulam-Tsingou** problem in physics is an apparent paradox that given a system of point masses connected by springs, it will almost exactly form the same periodic behavior known as the Fermi-Pasta-Ulam-Tsingou recurrence, rather than an expected ergodic behaviour. This behaviour came by surprise because they expected the system to *thermalize* or reach thermal equilibrium in just a short amount of time. Yet the system appears to evade the ergodic hypothesis and this explanation remains a hot topic in research today. Therefore, in this project we are building a numerical analysis program that will help us visualize linear and nonlinear forms of the Fermi-Pasta-Ulam-Tsingou problem, to help us visualize this unique behaviour.

1 Preface

In the year 1953, Enrico Fermi, John Pasta, Stanislaw Ulam, and Mary Tsingou conducted computer simulation experiments of a vibrating string that included non-linear terms. Based on some initial assumptions they were expecting the bahvior of this system to have an ergodic properties. And for those who are unfamiliar an ergodic behavior is when a system has the property of when given sufficient time, they impringe on all points in a given space. [1] So while these mathematicians expected some sort of ergodic behavior, they actually ended up getting a *quasi-period* [2] behavior. The following behavior was demonstarted given specific intial values k.

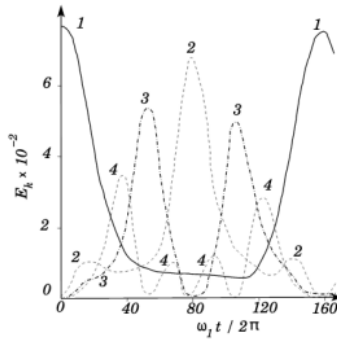


Figure 1: The Quasi-Periodic Behavior of the Fermi-Pasta-Ulam-Tsingou Problem

Overall the FPUT experiment conducted in 1953 was quite the momentous experiment because it showcased some of the complexities a nonlinear system can have in terms of its behavior. Also in the context of scientific computing, this experiment was one of the first computer simulations in analyzing these systems.

2 Equation Breakdown

In this section we will discuss the mathematics of **Figure 1**, and also explain both the linear and nonlinear cases of what the FPUT experiment should do. Like we mentioned in the previous section, we are interested in the behavior of a system of point masses connected by springs (represented as a particle of strings), and how they move given an initial configuration. This system can be seen as a coupled system of second order differential equations of the form:

$$\begin{cases} \ddot{x} = f_i = K(x_{i+1} - 2x_i + x_{i-1})(1 + \alpha(x_{i+1} - x_{i-1})) \\ x_i(0) = x_i^0 \\ \dot{x}_i(0) = v_i^0 \end{cases} \quad (1)$$

In **equation 1**, K is the spring constant, α is the strength of the nonlinearity, and $x_i(t)$ is the deviation of the i^{th} mass from its equilibrium position. Additionally x_i^0 is the initial displacement of the i^{th} mass, and v_i^0 is the initial velocity of the i^{th} mass. We assume that all masses are normalized to one. These parameters and this system works for an infinite systems of point masses. However we can determine the size of the system and truncate that system to N total masses, so $i = 1, \dots, N$

In order to solve this particular equation second order ODE, we must find a numerical method to solve this. Since computers do not perform calculus, there are methods constructed to help solve these with computation. Therefore in order to advance solutions through time we are able to take the x_i^n and x_{i-1}^{n-1} in order to get the next solution. This mechanism is what we call the leapfrog method because we use previous and current solutions to generate a future solution. This solution can be calculated using the following:

$$x_i^{n+1} = f_i = K(x_{i+1} - 2x_i + x_{i-1})(1 + \alpha(x_{i+1} - x_{i-1}))$$

It is important to know that this equation that resembles how we advance our solutions through time has a linear and non linear component. In the linear component we set our α to be equal to 0. Doing so we will have a normal behavior where the string of masses will simply oscillate up and down without any surprises.

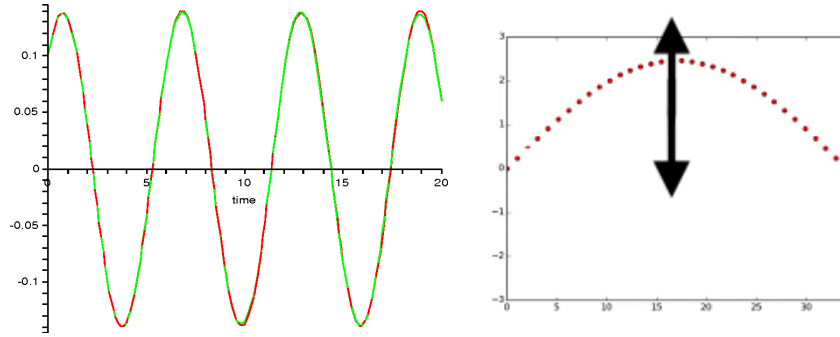


Figure 2: Oscillating behavior as time moves forward, and representation of string of masses as time moves forward

There is nothing obsolete about this behavior, as long as our equation stays linear, the behavior of the masses on a string will remain consistent. However let's revisit if we include an α value that is not equal to 0. As mentioned in the Preface, Fermi, Pasta, Ulam, and Tsingou thought that the behavior would not follow any certain pattern. However as they ran the simulation they were shocked to find that there was a pattern within this non-linear system. As to why this occurs remains an active part of research but it is odd but amazing how this system in particular behaves.

3 FPUT implementation

In the previous section we discussed how these systems were solved numerically through computation. Now that we know the governing equation for this leapfrog method, there are a few architectural designs that we wish to see. The following x_{i+1}^n , x_i^n , and x_{i-1}^n can describe the indices of an array and we want to show a high dimensional graphic of what is taking place in this algorithm.

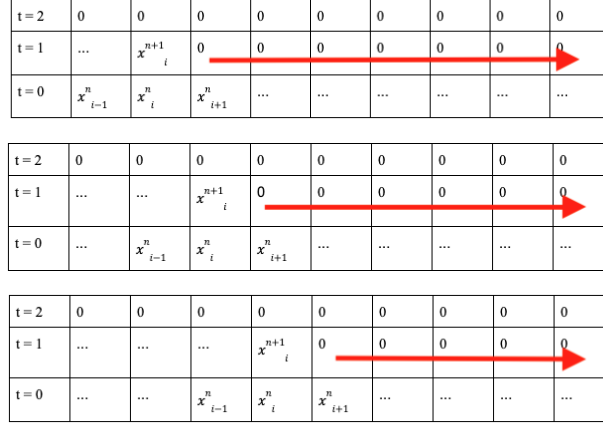


Figure 3: How the leapfrog method architecture fills in solutions as it advances in time

In this graphic we can see how the x^n array influences what the solutions of the x^{n+1} arrays will be. And it is also important to note that the solutions get filled from the second index all the way to $N - 1$ indices as there is no way to calculate a x^n_{i-1} on the left side and x^n_{i+1} on the right side. That is the purpose of the dummy masses that we initialized in **Equation 1**. So through this representation we can see specifically how the new solution is generated as we advance through time and how these cells obtain their values.

In our implementation we can cycle through these arrays by allocating a brand new x^{n+1} array of size N and use a for loop that resembles space to iterate within the interval $1 < x < N - 1$ (in a 0 indexed language). After going through the space loop we can follow a specific pattern to continue to advance in time. The pattern can be seen as follows:

$$\begin{aligned} x^{n-1} &= x^n \\ x^n &= x^{n+1} \end{aligned}$$

What this particular means is that x^{n-1} is now timestep 1 and x^n is time step 2. This process repeats frequently as long as our time loop keeps iterating.

4 Results

Since the Fermi-Pasta-Ulam-Tsingou problems are still an active field of research, part of this research involves experimenting to see what results we can generate with different number of masses, as well as capturing snapshots at specific time intervals. Specifically we test with $N = 1, 8, 16, 32$ in terms of different numbers of masses and test at $t = \frac{T_f}{4}, \frac{T_f}{2}, \frac{3T_f}{4}, \text{and } T_f$ time intervals. The following plots are generated showing these different mass numbers and time conditons.

4.1 Linear-System

Lets first examine these two set of plots of the following system.

$$\begin{cases} \ddot{x} = f_i = K(x_{i+1} - 2x_i + x_{i-1}) \\ x_i(0) = x_i^0 \\ \dot{x}_i(0) = v_i^0 \end{cases} \quad (2)$$

Our first plot as shown in **Figure 4**, shows our system if we have $N = 1$ mass with no non-linear component. As expected our graph does nothing special as our system simply moves up and down through time. However lets begin to play around with this N variable. Within the subplots displayed in the top right from top to bottom we have the following values for N : 8, 16, 32. Within the subplots we have two columns split into a time graph at points $t = \frac{T_f}{4}, \frac{T_f}{2}, \frac{3T_f}{4}, \text{and } T_f$ time intervals, as well as tracking the mass $i = N/2$ at all time intervals. Through some intial analysis we are not suprised with the overall behavior of $i = N/2$. As expected it oscillates as these

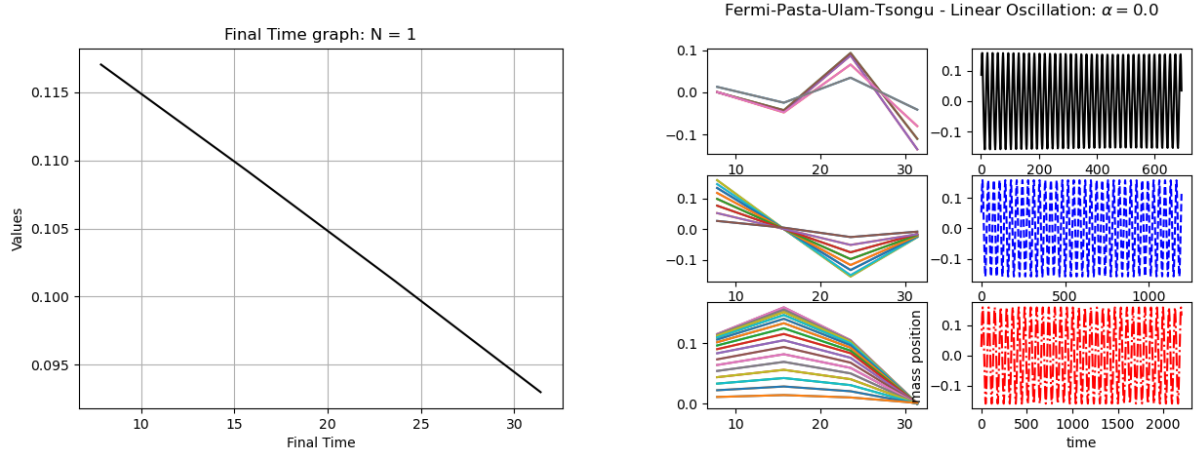


Figure 4: A plot with 1 mass, as well as a set of subplots of the linear case

sets of string of masses move forward in time. However as we can see in the $N = 16, 32$ values, we begin to lose the oscillating structure. My thinking is that these systems move up and down so fast that they construct this weird behavior seen in these graphs. Though very odd, their is also a beauty that we can appreciate from this simulation.

4.2 Non-Linear Systems

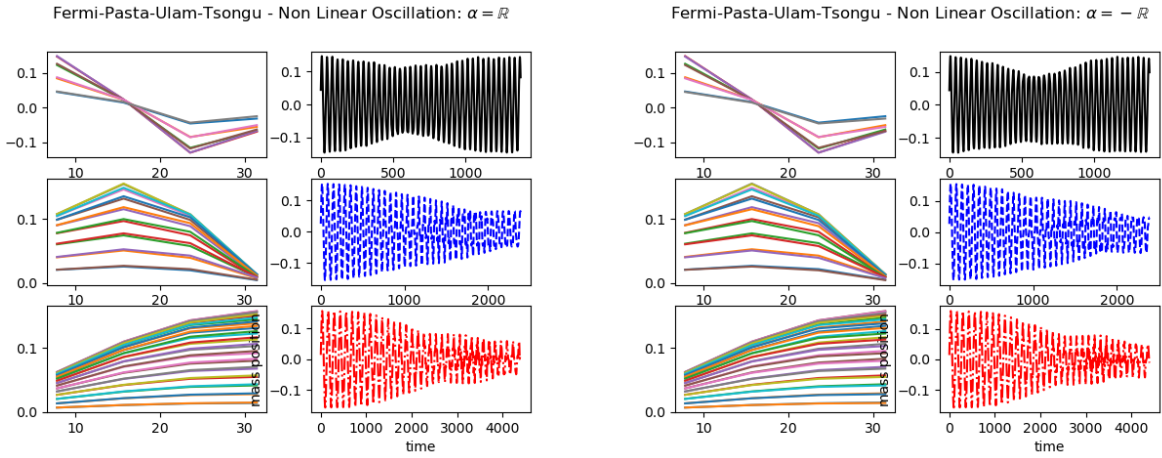


Figure 5: A set of subplots that show different values of N and how the non-linear constant α affects the system

Next we introduce the α constant which was previously described to be the strength of the nonlinearity. What this constant does is introduce a new component to our system as previously shown in **Equation 1**. By adding the $(1 + \alpha(x_{i+1} - x_{i-1}))$, there is no surprise that it will affect the overall behavior of the system. One of the big factors that change within this non-linear system is actually the amount of time steps we take. In a linear system we calculated the time step to be the following:

$$M \geq T_f \sqrt[2]{K} \quad (3)$$

However in a non-linear system we calculate the amount of time steps introducing a new variable C where $C < 1$ which will satisfy us due to the linear version no longer being sufficient for computation. In the following graphs we use the value $C = 0.5$ and it is used to calculate the time step in the following equation:

$$M = \left\lceil \frac{T_f \sqrt[2]{K}}{C} \right\rceil \quad (4)$$

So as shown in **Figure 5**, we see that this system has a "squeeze" affect as for the system with 8 masses. Some implications we can make is that the 16 mass system, as well as the 32 mass system is probably likely going to behave in the same way if our graphs were extended out further. So although the nonlinear constant hardly had an affect on the $t = \frac{T_f}{4}, \frac{T_f}{2}, \frac{3T_f}{4}, \text{and } T_f$ time interval graphs, we can see noticeable differences coming from the center mass graph. Another interesting behavior can be seen with the two graphs side by side who have the similar time interval graphs but when it comes to the center mass graph, we can see that system is reflected across the horizontal axis. As we continue our future tests this behavior remains constant.

4.3 Playing around with the C

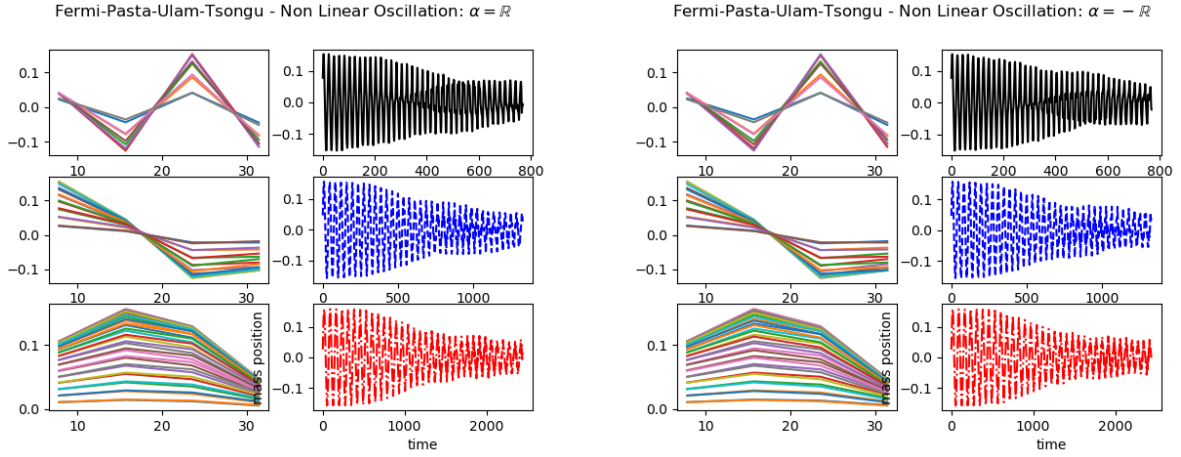


Figure 6: A set of subplots that show the system if we manipulate the system to $C = 0.9$

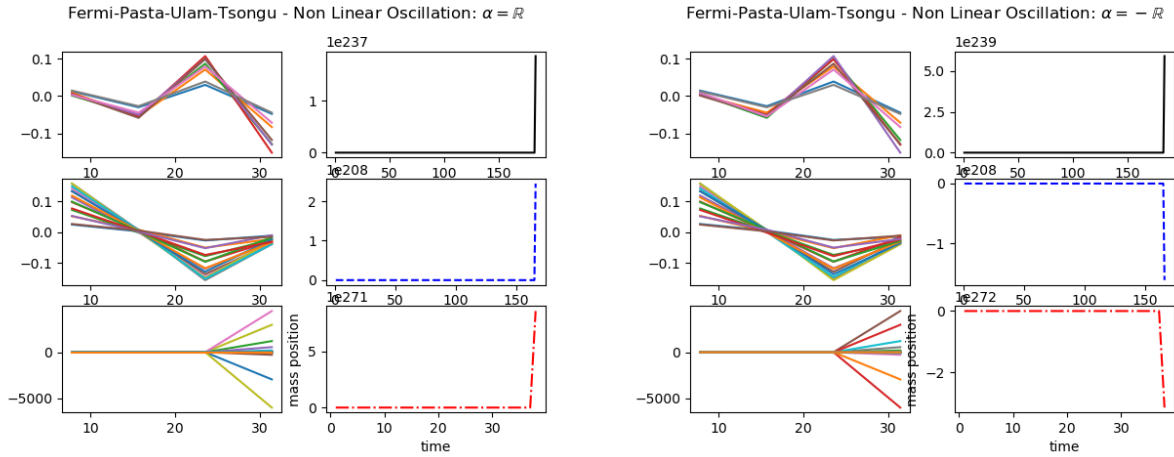


Figure 7: A set of subplots that show that the system breaks when the value reaches $C = 0.99$

Our last set of experiments that we will run involves this C variable. As it approaches the value 1.0, the solution actually gets ruined. So through some quantitative analysis I found a median value for C and tried to see how the

behavior of the graphs were affected by this variable. Lets begin with **Figure 6** and the set of time interval graphs. We can immediately already see that these graphs in particular are drastically different than our original graphs. We can see as well that all three systems with the different N amount of masses behave differently as well. In our first system we can see that the graphs have apparent local maxima and minima, while the second system has more of a point of intersection. And our last system resembles a figure that almost looks like some surface. In the center mass graph we can see that the oscillations also look very similar.

As we take a look at our final **figure 7**, we can see we have officially broken our system. I was able to do this by setting the value inbetween the interval $0.98 < C < 1.0$. This appears to make total sense because when we were working with the non-linear systems we made it clear that we could no longer use the same time-stepping equation as we did for the linear systems. So as we reach closer and closer to 1.0, we slowly begin to see the time-stepping equation to resemble the previous linear time-stepping equation so it breaks our code.

5 Comments

Through this whole assignment I became more interested in the topic itself. So while I have provided all my findings, and analysis, I wanted to speak shortly about this system and thermalization [3]. I briefly mentioned in the abstract that the system may reach thermalization which is a process in which mutual interactions cause physical bodies to reach thermal equilibrium. By reaching this equilibrium we are able to see the energy of a system to maximize its entropy [4]. Entropy in physics is described as when a physical system hits a state of disorder, or uncertainty. Physicists have been able to calculate this through the equation $S = k_b \ln \Omega$ [5]. This all gets tied back to the Fermi-Pasta-Ulam-Tsingou equation through how the non-linear system behaves. Although the entropy is high for these systems because it has a quasi-period behavior, what we have made an emphasis throughout this paper is that this system has some repeating sequence that cycles over and over and over. With this purpose in mind, I have appreciated the beauty of this system for I have no real scientific reasoning as to why this ODE is the way it is.

6 Conclusion

The Fermi-Pasta-Ulam-Tsingou equations are one of the most unique systems of ODE's to be discovered to this day. With its unique behavior that is not totally understood, we were able to construct a set of computational tools that allowed us to analyze some of its peculiar behaviors. While I have no academic hypotheses for why this system is the way it is, I thought it was rather interesting to be able to compute this ODE myself. In addition to this research project that we have just conducted, I believe we should give high praise to the four mathematicians who worked with such a system in the 50's. With computational tools being more of a focal point today, it was impressive to see that nearly 70 years ago, Mary Tsingou in particular was able to use scientific computing as a tool to study this equation. So as the power of computation continues to rise exponentially, I am highly curious to continue the mysterious wonders of our natural world.

7 Acknowledgement

I just wanted to thank both GSI Ian May, and TA Chris DeGrendele for all their help throughout the Fall 2021 term. I was not only able to grasp a lot of useful computational tools from the both of them, but I got to know two and their interests in research. I am lucky to have spent time learning from two incredible kind people who have made me become more interested in the field of scientific computing as well. Through this unique pandemic university experience they have always made themselves available to help, and I cannot thank them enough for helping me learn more on my young academic journey. Thanks again for a great quarter and as Ian always says Cheers!

References

- [1] D.V. Anosov (2001) [1994], "Ergodic theory", Encyclopedia of Mathematics, EMS Press

- [2] The meteorological glossary: 2d ed. 1930. Meteorological Office, Great Britain. "Certain phenomena which recur more or less regularly but without the exactness of truly periodic phenomena are termed quasi-periodic."
- [3] Zabusky, N. J.; Kruskal, M. D. (1965). "Interactions of solitons in a collisionless plasma and the recurrence of initial states". *Physical Review Letters*. 15 (6): 240–243. Bibcode:1965PhRvL..15..240Z. doi:10.1103/PhysRevLett.15.240
- [4] "Collisions and Thermalization". *sdphca.ucsd.edu*. Retrieved 2018-05-14.
- [5] Wehrl, Alfred (1 April 1978). "General properties of entropy". *Reviews of Modern Physics*. 50 (2): 221–260. Bibcode:1978RvMP...50..221W. doi:10.1103/RevModPhys.50.221.