American Journal of Computational Mathematics, 2025, 15(1), 58-65
https://www.scirp.org/journal/ajcm
ISSN Online: 2161-1211
ISSN Print: 2161-1203

Impact of k x^n Force on Potential Oscillations

Haiduke Sarafian

University College, The Pennsylvania State University, York, PA, USA
Email: has2@psu.edu

How to cite this paper: Sarafian, H. (2025) Impact of k x^n Force on Potential Oscillations. *American Journal of Computational Mathematics*, **15**, 58-65.
https://doi.org/10.4236/ajcm.2025.151003

Received: February 19, 2025
Accepted: March 17, 2025
Published: March 20, 2025

Abstract

It is common sense to assume, under the influence of modified Hooke law, that a spring-mass system should oscillate. A systematic numeric analysis proves otherwise. We have proven that the mentioned modified force subject to k x^n for *even* n integers fails to produce oscillations. In contrast, the same format for odd n integers is conducive to harmonic oscillations. For the latter case, the impact of the chosen odd n values on the oscillation periods is mathematically identified. For selected cases, the corresponding oscillations are graphed. The analysis is based on applying a Computer Algebra System (CAS), *Mathematica* [1]-[3].

Keywords

1D Nonlinear Forces, Period Prediction, Harmonic Oscillations, *Mathematica*

1. Introduction

A thorough literature search shows that the character of a 1D elastic rubber band, *i.e.*, a spring in a spring-mass system, is confined to springs subject to a linear character, *i.e.*, the compressed or elongated length. This results in a harmonious oscillation with a formulated period embedded in the equation of motion. The equation of motion of the mass is subject to a linear dynamic ordinary differential equation with the built-in period. It is a curious question if one asks why only the linear character is preferred. Years back, the author proposed a nonlinear format conducive to different types of oscillations and fabricated a spring subject to the cubic character [4] [5]. Somewhat motivated by this proposal, other types of position-dependent elastic media (springs) have been considered in this current study.

Nowadays, utilizing a Computer Algebra System (CAS), specifically *Mathematica*, provides a forum to analyze this issue. Boldly, this article claims that without this CAS, the analysis of this issue would have been unaddressed. Mathematical problems have been addressed in this article, yet the actual fabrication of the proposed springs is unclaimed and left to the community of interested readers.

This article is composed of three sections. In addition to the Introduction, in Section 2, Formulation and Analysis, systematically, the dynamic equation of motions for the chosen k x^n force shows that the only functions subject to the odd integers powered are conducive to oscillations. Applying a CAS, the oscillations are graphed. Utilizing the graphs, the period of the oscillations is identified not only by using the graphs but also by identifying the roots of the oscillating functions. It is explicitly shown that the compression (elongation) lengths are embedded in the solution of the ODE, as initial conditions play a crucial role. Depending on the initial condition, some ODEs become stable while others are chaotic. This issue is overlooked in the literature. For the sable cases, a trend line is identified as capable of predicting a period for *any* odd large in odd n. Two practical examples are given.

This section includes selective oscillation functions, which are accomplished because of the tremendous power of the CAS, *Mathematica*, utilized.

2. Formulations and Analysis

Customarily, when working with a rubber band (an ideal spring), the statement reads, [... elongating and/or compressing a perfect spring by xxx units from its equilibrium, the attached mass to the spring oscillates.] Rehashes this sentence call clarifications. Because one of the objectives of this work is to work with non-ideal springs, namely springs that are subject to forces characterized by F(x) = k x^n. A plot of these functions for various odd n values is shown in **Figure 1**. The functions have a common crossing point at x = 1. Although a constant quantity of k usually controls the springs' stiffness, this doesn't alter the mentioned feature.

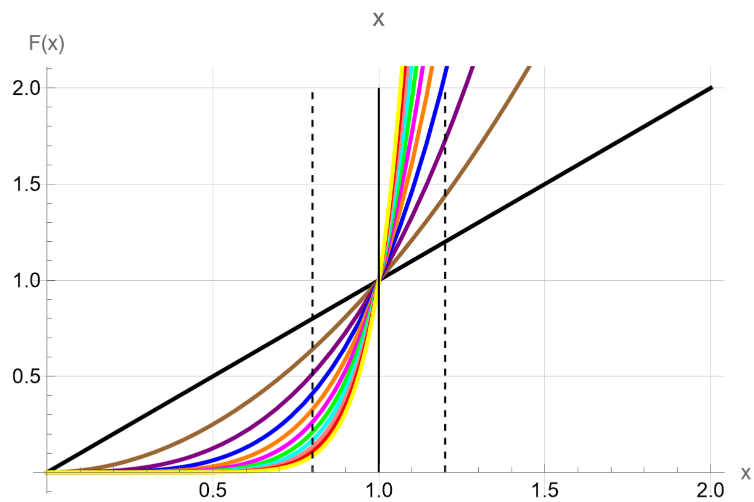

Figure 1. Plot of x^n, between 0 <= x <= 2.0 for various odd integers n. The black line displays the ideal spring (n = 1). The other curves departing from the straight, slanted line correspond to n = 2, 3, ...,11 on descending curves.

Figure 1 includes two vertical parallel dashed lines. These are drawn at about x = 1. It also consists of a solid vertical line that passes through the pivotal x = 1 point. As shown, the only curve at this point that sustains its slope is the straight black slanted line corresponding to the ideal spring. All the other curves that pass this point exhibit different slopes. The mentioned curves with abscissa less than unity have slopes less than their extensions with abscissa larger than unity. This is a crucial observation because this is the reason that the statement "a spring is elongated (compressed) by xxx units" should be qualified. Meaning if the spring is a non-ideal, then the xxx units should be spelled out, as it should read "lesser or longer than the unity." The impact of this statement numerically is justified in the follow-up paragraph.

This subsection begins with the oscillating character of an ideal spring, k x. For the graph scaling, the k is set at 10 units; there is nothing special about this selection. The equation describing the movement of a block of mass m on a horizontal frictionless surface attached to one end of this spring with its other end fastened to a stationary support is

$$kx(t)+m\ddot{x}(t)=0 \tag{1}$$

where the double-dots are the second derivative w/time. To be consistent with the rest of the study cases, we avoid symbolically solving this trivial ODE. We consider two sets of initial conditions to emphasize the point of interest. Both sets share one of the initial conditions, *i.e.*, the block begins the movement with zero speed; one is initially positioned at a place less than unity, and the other is beyond unity, see Figure 1. The corresponding *Mathematica* code reads.

This code is generic and written for the k x^n force. Setting n = 1 gives information about the case at hand. As mentioned, the stiffness is set at k = 10 units and block mass m = 1 units for simplicity. The code's third line sets the character of the force. The output of the code is the display of the oscillations. The elements of the initial condition in the fourth line correspond to the initial condition of interest.

```
Clear[n,sol1,sol2,testcase11,testcase12,period1,period2,roots1,roots2]
eqn[n_]=x''[t]+10.0 x[t]^n;
n=1;tmax=15; (* only Odd intergers are allowed*)
initial={{x[0]==0.8,x'[0]==0},{x[0]==1.1,x'[0]==0}};
sol1[n_]=NDSolve[{eqn[n]==0,initial[[1]]},x,{t,0,tmax}];

plotn1=Plot[50x[t]/.sol1[n],{t,0,tmax},PlotStyle->{Dashing[0.01],c[[n]]},
GridLines->Automatic,PlotRange->All,AxesLabel->{"t","x"},ImageSize->500];

testcase11=Plot[50x[t]/.sol1[n],{t,0,tmax},PlotStyle->{Dashing[0.01],c[[n]]},      GridLines->Automatic,PlotRange->All,AxesLabel->{"t","x"},ImageSize->500]
```

```
sol2[n_]=NDSolve[{eqn[n]==0,initial[[2]]},x,{t,0,tmax}];
plotn2=Plot[50x[t]/.sol2[n],{t,0,0.5tmax},PlotStyle->c[[n]], GridLines->Automatic,PlotRange->All,AxesLabel->{"t","x"},ImageSize->500];

testcase12=Plot[50x[t]/.sol2[n],{t,0,0.5tmax},PlotStyle->c[[n]], GridLines->Automatic,PlotRange->All,AxesLabel->{"t","x"},ImageSize->500]

{tabn[[n]],Show[{plotn1,plotn2}]}
```

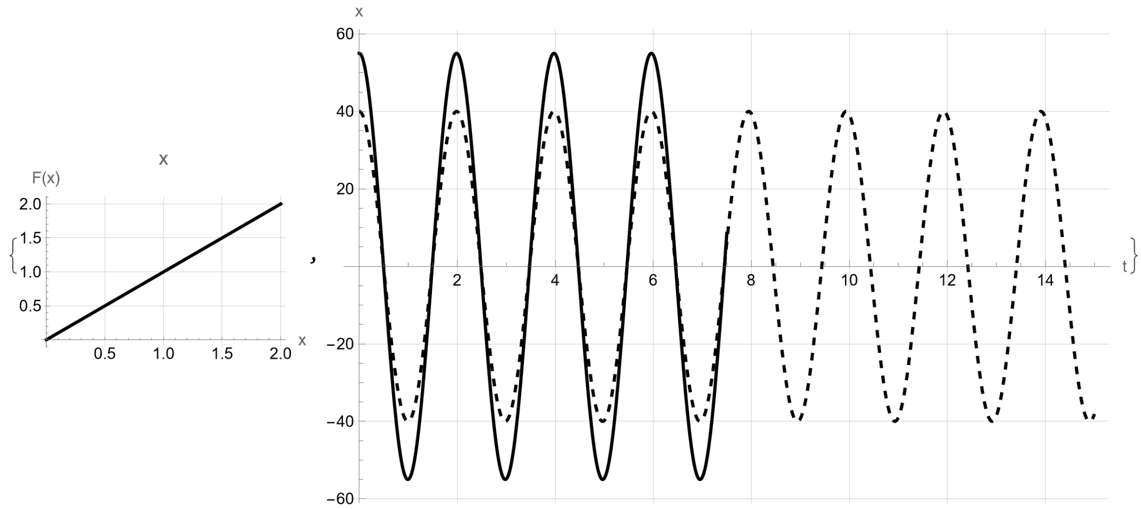

Figure 2. A display of oscillations of the block under the influence of an ideal spring is shown on the right panel. The dashed and the solid curves are associated with the mentioned initial conditions. The left panel displays the character of the applied force.

The left panel of **Figure 2** displays the applied force on an ideal spring, and the right panel is the impact of the mentioned force on the oscillations of the massive block. The panel embodies two curves. The dashed curve corresponds to the initial position of the block placed on the left side of the x = 1 shown in **Figure 1**. The solid curve occurs when the block is placed on the right side x = 1, with an initial elongation longer than unity. As shown, these two curves have the same oscillation period. Their oscillation heights are intentionally set at different values, making the curves distinguishable. The point is that the slope of the linear force shown in **Figure 2** is the same as the pivot; therefore, it is irrelevant to the initial position of the block and results in the same oscillation character. This feature is unique for the ideal spring. This may be the unexplained reason that literature passively mentions a block is either pulled and/or compressed from its relaxed position!

As mentioned in the previous paragraph, the difference between these curves, the dashed and the solid, is due to the impact of the initial conditions. The point is that the character of the force displayed on the left panel shows the curve has varying slopes, especially about the pivot position x = 1, depending on where the block begins its movement; on the left of x = 1 gives the dashed curve, and on the right of x = 1, it provides the solid curve result oscillations with period.

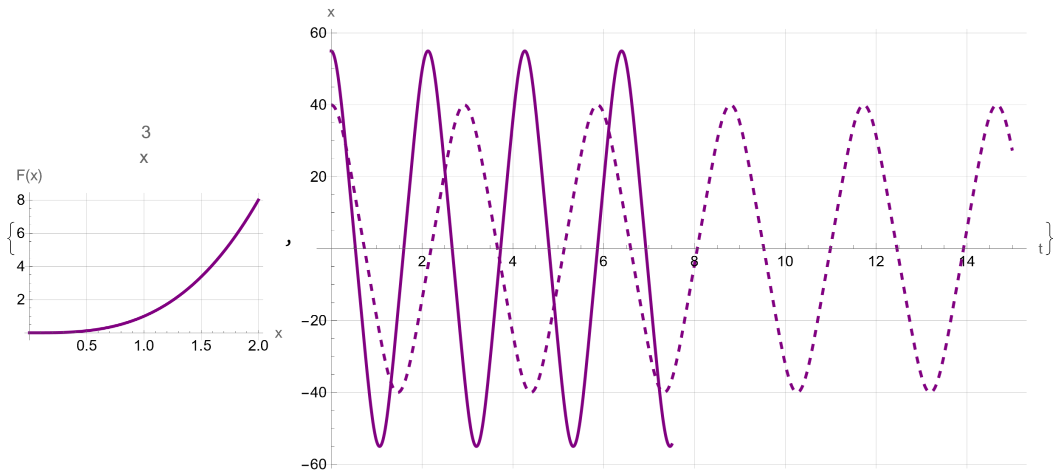

Figure 3. The description of this plot is the same as **Figure 2**. The difference is that the applied force is k x^3, as displayed on the left panel.

As mentioned in the abstract, oscillations of the block come about from forces subject to k $x^{n=\text{odd integers}}$. At this stage in the article, one might wonder why *even integer* powers have been ignored/skipped. With the code in hand, one may try running the code with an even integer, say n = 2. The output is not only not oscillatory but is chaotic. The output is not included, avoiding clattering. The interested reader may easily confirm the claim!

Now, one may systematically exercise, as the author did, displaying the character of the oscillations under the influence of the character of the forces for the rest of the cases, *i.e.*, n = 5, 7, ..., 11. The following paragraphs embody the results.

Finally, this segment is closed by displaying the result of n = 11. This is selected as a typical representative example for n > 5. A detailed study shows that for the high values of n, *i.e.* n's larger than 5, the periods associated with the two different initial conditions mentioned are drastically different. The utilized code is designed to handle this issue. Here is the output related to n = 11.

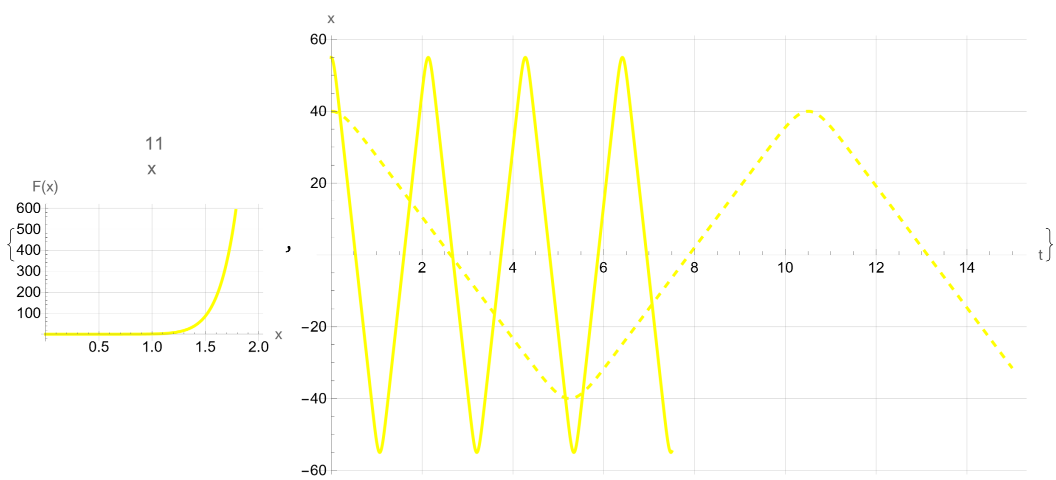

Figure 4. The description of this plot is the same as **Figure 2**. The difference is that the applied force is k x11, as displayed on the left panel.

The right panel depicts the difference mentioned. The dashed curve has a more extended period, while the solid curve almost has a compatible period compared to the previous solid cases. The forthcoming paragraph discusses this feature.

Although the period of a chosen applied force and its associated oscillations may be read off from the graph displayed with reasonable accuracy, for more accuracy, one may search for the roots of the oscillations, *i.e.*, the abasia of the intersection coordinate. The following code accomplishes the task.

Here, two consecutive roots for n = 11 are identified. The difference of the calculated roots yields the associated period.

If[n>5,Plot[50*x*[t]/.sol2[*n*],{*t*,0,3},PlotStyle->*c*[[*n*]],GridLines->Automatic,PlotRange->All,AxesLabel->{"t","x"},ImageSize->500],Null]

The roots of the solid curve in Figure 3 are,

roots1={FindRoot[x[t]/.sol1[n],{t,2.7}],FindRoot[x[t]/.sol2[n],{t,13.2}]}

{0.5,2.52},{0.7,3.7}

{{t->2.62646018454372`},{t->13.359729886451856`}}

The roots of the dashed curve in Figure 3 are,

roots2={FindRoot[x[t]/.sol2[n],{t,0.5}],FindRoot[x[t]/.sol2[n],{t,2.68}]};

{0.5,2.52}

period1={n,{(t/.roots1[[2,1]])-(t/.roots1[[1,1]])}}

period2={n,{(t/.roots2[[2,1]])-(t/.roots2[[1,1]])}}

{0.5,2.52}

The roots of the solid and the dashed curves, respectively, are,

{11,{10.7333}}

{11,{2.13756}}

This procedure has been repeated for n = 1, 3, 5, 7, 9, 11. The results are stored in the **data1** listing. The data is graphed; see Figure 4.

data1={{1,1.98},{3,2.99},{5,3.907},{7,5.75},{9,6.248},{11,10.73}}

Noticing the first overlapped data. This is unique to ideal springs, indicating the relevance of the initial condition. The lesson learned is that when non-ideal springs are used, one must carefully consider how to refer to their use.

The displayed data in Figure 5 motivates fitting the data. With trial and error, a reasonable function is identified. The code and procedure are given. The plot of the fitted function and the data is shown in Figure 5.

{a m^3+b m^2+d m+e/.fit1,a $\sqrt{m}$ +b m^2+d m+e/.fit2}

{a m^3+b m^2+d m+e/.fit1,a $\sqrt{m}$ +b m^2+d m+e/.fit2}

With this fitted function in hand, one may predict the oscillation period without calculation, e.g., for n = 13 and 14, see Figure 6.

{16.585,2.03989}

{25.8363,1.92031}

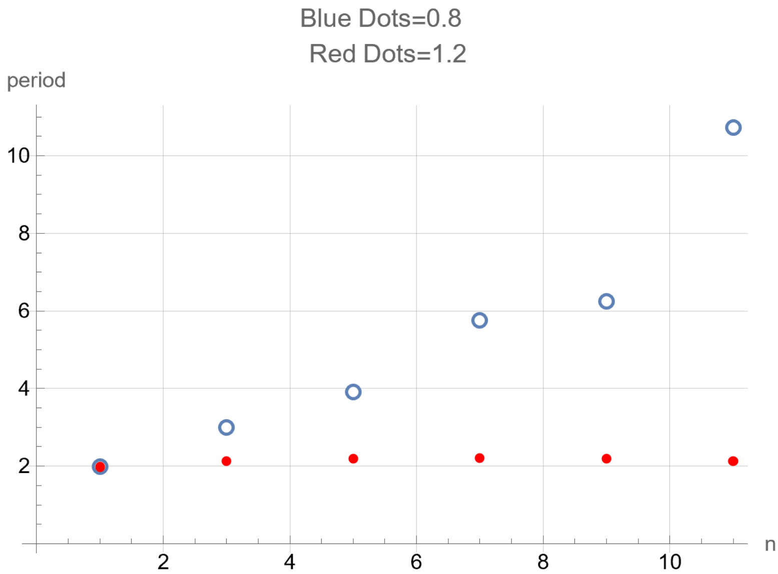

Figure 5. Display of the periods vs. the ordered power of k x^n. The open circles correspond to the first initial, and the dots correspond to the second type of initial conditions.

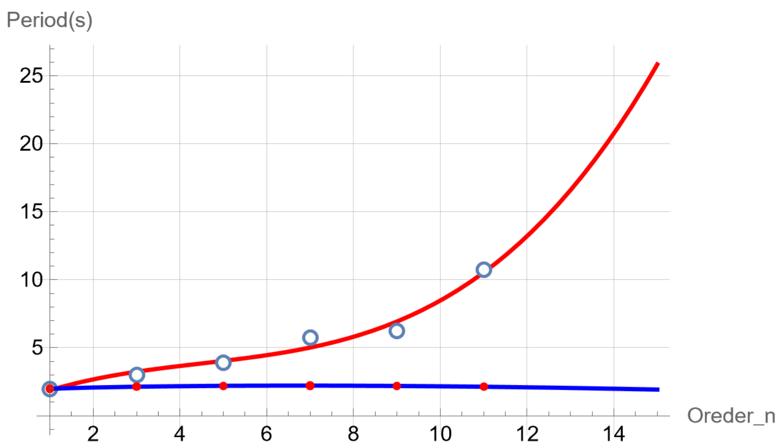

Figure 6. Data and the fitted function. The horizontal axis is the order of the non-ideal spring, k x^n. The vertical axis is its associated period in seconds.

The data is collected for n = 1, 3, 5, ..., 11. These are stable oscillations with periods 2.03 s and 1.92 s, respectively.

4. Conclusions and Comments

In this short note, we brought out an overlooked physics problem. In short, the oscillation movement of a spring mass traditionally is confined to the character of an ideal spring, the one subject to a linear force, F(x) = k x. We have argued that oscillations should not all be confined to a linear force. As such, family forces subject to k x^n are considered. Although naturally, one might assume any integer n should result in oscillations, this is a false assumption. Numerically, the resulting ODE yield oscillations are shown only for odd integer n. The even integer n yields no oscillation but chaos. As mentioned, this is proven numerically. Analytically, except for k x^3, cases with n > 3 fail analytic solutions. Currently, 1) efforts are being made to address the issue, 2) The same is valid for k $x^{n=even}$, and 3) Fabrication of the mentioned springs in the text is left to interested readers.

Another point is that the accumulated data stored in **data1** has been fitted with a reasonable function, lacking physical interpretation. *I.e.*, one may think about the oscillation period of an ideal spring subject to trivial ODE, m $\ddot{x}$ (t) + k x(t) = 0 with a built-in angular speed, $\omega = \sqrt{\frac{k}{m}}$, that yields T = $(2\pi)/\omega$ compatible with the graph shown in **Figure 2**, T = 1.98 s. Efforts are on the way to explore a formulation similar to the letter for n > 1.

Finally, it is a resourceful reference that interested readers may consult with [6].

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] (2024) Mathematica V14.1. http://Wolfram.com

[2] Wolfram, S. (2003) The Mathematica Book. 5th Edition, Cambridge University Publications, Cambridge.

[3] Sarafian, H. (2019) Mathematica Graphics Examples. 2nd Edition, Scientific Seasrch Publishing, Wuhan.

[4] Sarafian, H. (213) Linear, Cubic and Quintic Coordinate-Dependent Forces and Kinematic Characteristics of a Spring-Mass System. *World Journal of Mechanics*, **3**, 265-269. https://doi.org/10.4236/wjm.2013.36027

[5] Sarafian, H. (2010) Static Electric-Spring and Nonlinear Oscillations. *Journal of Electromagnetic Analysis and Applications*, **2**, 75-81. https://doi.org/10.4236/jemaa.2010.22011

[6] Sarafian, H. (2023) Haiduke Sarafian's Collective Articles 2020-2023. Scientific Research Search Publishing, Wuhan