CONVERGENCE ANALYSIS OF THE MODIFIED CHEBYSHEV-PICARD ITERATION ALGORITHM

A Thesis

by

MASON CHAPPELL MOORE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	John L. Junkins
Co-Chair of Committee,	Manoranjan Majji
Committee Members,	Freddie Witherden
Head of Department,	Srinivas Rao Vadali

August 2021

Major Subject: Aerospace Engineering

ABSTRACT

The topic of this work is the analysis of the ability of Modified Chebyshev-Picard Iteration (MCPI) to converge on an accurate solution. The convergence analysis includes a discussion of MCPI and a brief explanation of the derivation of first and second order MCPI. The limitations and benefits of MCPI are also discussed. MCPI has historically used relative error to evaluate if the algorithm has converged to a solution. This work presents a means of evaluating the absolute error of the solution. The Absolute Error Analysis is derived for first and second order MCPI and the difference between it and relative error is discussed using an example problem. The different means of evaluating convergence are then used to evaluate how changes to MCPI affect convergence of a nonlinear oscillator. The convergence is evaluated by inspecting the error profiles of the converged solution, the number of iterations necessary to converge to the solution, and the maximum final time which converges to a solution. MCPI convergence is then analyzed when using two-body motion for three predefined orbital trajectories. The convergence is analyzed according to the error profiles of each trajectory, the number of iterations necessary to converge for each trajectory, and the maximum final time at which each trajectory can converge to a solution. The use of first and second order MCPI is discussed along with the use of Cartesian Coordinates and Orbital Elements and their affect on the convergence of MCPI. The three trajectories are then analyzed with the introduction of J2 and drag perturbations, both individually and combined effects. The perturbed trajectories are evaluated using Multi-segment MCPI to extend the convergence window of MCPI. The variation in MCPI variables used to compare with the number of iterations necessary to converge and the error profiles of each segment to evaluate the efficacy of Multi-segment MCPI as a numerical integration tool. The entirety of the MCPI code usage was developed for universal implementation and is available to the public.

DEDICATION

To my wonderful wife, Hannah.

Thank you.

I love you.

ACKNOWLEDGMENTS

I would like to thank my wonderful wife, Hannah. Thank you for following me out to Texas with less than two months notice. You have helped me push myself to be better and worked with me through any personal struggles I have had over the years. You have helped me keep my head on straight and kept my confidence up when I was feeling quite low. I am grateful for our years together and excited for the upcoming years.

Thank you to my parents and sister. Throughout my life, you have provided me with numerous examples of the importance of education and personal growth. From an early age, you have each taught me the value of striving to do my best and learning all that I can. Whether it was driving me to and from events or helping me with homework, you have always been there for me.

Thank you Dr. Junkins. You have given me the opportunity to learn a great deal about myself. You have given me a wonderful chance to learn from the best and brightest about engineering and mathematics. I am not staying in Texas for as long as originally planned, but I am grateful for the guidance and advice that you have offered me during this time. I frequently write "own it" on papers around me as a reminder that it is critical to own the material I work with and not let it own me.

Thank you Dr. Majji. Working with you has been a blessing. I know that I would not have gotten through this thesis without your help and guidance along the way. Thank you for your instruction in the classroom and in one on one discussion.

Thank you to Chris, Joe, Patrick, Sandeep, and Vishala for help as I worked to learn optimal control theory, MCPI, or on homework problems. I'm glad to have been able to work with each of you.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor John L. Junkins and Manoranjan Majji of the Department of Aerospace Engineering and Professor Freddie Witherden of the Department of Ocean Engineering.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University.

NOMENCLATURE

a	Semi-Major Axis
C_d	Drag Coefficient
e	Eccentricity
GEO	Geostationary Earth Orbit
h	Angular Momentum
i	Orbital Inclination
J2	J2 Earth Gravitational Perturbation Parameter
LEO	Low Earth Orbit
m	Mass of Spacecraft
M	Number of Points Sampled in MCPI
MCPI	Modified Chebyshev-Picard Iteration
MEE	Modified Equinoctial Elements
N	Maximum Order of Chebyshev Polynomials Used
OE	Orbital Elements
r	Radial Distance
R_e	Radius of the Earth
RK	Runge-Kutta
RK4	Runge-Kutta Fourth Order
S	Integration Matrix
S	Surface Area of Spacecraft
SGEO	Super Geostationary Earth Orbit
t_0	Initial Time

t_f	Final Time
T_n	Chebyshev Polynomial of Order n
W	Weight Matrix
ϵ	Non-Linear Parameter in Duffing Oscillator
Ω	Right Ascension of Ascending Node
ω	Argument of Perigee
Φ	Matrix of Chebyshev Polynomials
τ	Time in Chebyshev Node Space
θ	True Anomaly

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

This thesis will analyze the convergence of Modified Chebyshev-Picard Iteration (MCPI) in various ways. MCPI is a numerical integration scheme which utilizes Chebyshev Polynomials and Picard Iteration to approximate the dynamics of a system. The approximated dynamics are then integrated to result in the states of the system at each point in time. The tool was developed by Dr. Junkins and Dr. Bai. The analysis should help provide further understanding into what contributes to MCPI's ability to converge. It is known that MCPI has a time window in which it is able to converge. However, this window is not explicitly defined, nor is it clearly understood what settings should be used to maximize the efficiency of MCPI.

MCPI has used Relative Error Analysis in the past to analyze whether the algorithm has converged upon a solution. Relative Error Analysis compares the current iteration's solution to the prior's and calculates the change. If the change is less than a defined boundary, the algorithm will report the solution as converged. This paper will derive a new error analysis tool referred to as Absolute Error Analysis. Absolute Error Analysis uses a comparison between the dynamics of the system and of the estimation to define whether or not the solution is converged. The significance of this is discussed along with example problems to find the benefits and drawbacks of using Absolute Error Analysis.

The paper will analyze the use of MCPI with a non-linear oscillator. The analysis will include variations to the non-linearity of the oscillator along with variations to the parameters in MCPI. The final time of the MCPI segment will also be varied to see how these parameter changes affect the ability for MCPI to be able to converge and how Absolute Error Analysis evaluates the problem.

MCPI will also be analyzed for various problems in astrodynamics. Differential equations needing to be integrated is a very common issue within the astrodynamics world. This paper will analyze the use of MCPI to evaluate the two-body problem and then will add the perturbing forces of J2 and Drag, both individually and combined. These variations on astrodynamics will be analyzed using three predefined orbits, a low earth orbit, a super geostationary orbit, and an

eccentric orbit. These problems will be evaluated with variations in the number of sampled points used in MCPI, referred to as M, the acceptable tolerance for MCPI to report convergence, and the length of the segment

The majority of the paper will analyze second order equations using second order MCPI. There will be a comparison between using first order MCPI to solve second order equations. The first order MCPI will also be used with first order problems. The paper includes a comparison between the use of Cartesian Coordinates and Orbital Elements with respect to J2 perturbation in MCPI.

The paper uses created MATLAB code to most efficiently use MCPI in first or second order and easily change the variables with MCPI while maintaining default values. The code also includes the creation of Multi-segment MCPI, an easy to use function which repeats MCPI for a given number of segments to broaden the convergence window. Multi-segment MCPI was used in analysis of the perturbed astrodynamics problems and an analysis into how to best use it is included.

The analysis into MCPI is valuable to any person or organization that routinely integrates differential equations. The process of numerical integration can be time consuming and lead to inaccuracies. There are tools which are useful in numerically integrating, but the opportunity to use a more efficient tool is welcomed among physicists, engineers, and mathematicians. Among those who would benefit from a more efficient tool are those working with astrodynamics. Whether it be finding the optimal control for a two point boundary value problem or a Monte Carlo simulation of the trajectory of a body, there are times when hundreds if not thousands of integrations must be performed. Any improvement in the integration tool is extremely valuable for these purposes.

MCPI has the potential of converging to a solution of equal or greater accuracy than traditional numerical integration methods. Even if this improvement has a one percent improvement on prior performance, when coupled with the number of integrations necessary, that can save a significant amount of time. By saving time, it can allow the estimations to be analyzed and applied quickly. However, to utilize this potential time benefit, MCPI must converge to a solution. This paper will provide insight into how to best ensure that the algorithm will converge thus saving time in the calculations.

Part of the reason that MCPI is able to converge in more quickly than other integration schemes is by lowering the number of points necessary to converge and thus lowering the number of evaluations of the function that must be performed. The limitation of points also leads to a lower amount of space necessary to store the data. MCPI invites the potential for smaller computers to compute the states when given initial conditions. MCPI also requires fewer evaluations of the function at each point. This paper will discuss the proper number of points necessary to converge for various example problems.

1.1 Organization

Chapter Two will begin by defining numerical integration and a commonly used tool for numerical integration. The need for improved methodology will then be discussed. A nontechnical derivation of first and second order MCPI follows along with definitions of Chebyshev Polynomials and Picard Iteration. The limitations and benefits of MCPI will be discussed in the following section. An example problem will be presented and the definition of the convergence measures will be given.

Chapter Three will present the definition of Relative Error Analysis along with a discussion of the use of it with respect to MCPI. The definition of Absolute Error Analysis will then be given. Then the first and second order uses of Absolute Error Analysis will be derived and any limitations to their usages will be discussed. The two error analysis tools will then be compared using a simple example and discussed according to the convergence measures.

Chapter Four discusses the application of MCPI to the Duffing Non-linear Oscillator problem. The problem is defined and then the problem is varied in final time, M, acceptable error tolerance and the non-linearity parameter used in the duffing oscillator. Each variation is analyzed according to the convergence measures as defined in Chapter Two. Each subsection of the chapter includes an explanation of how the variation leads to a better understanding of MCPI's convergence.

Chapter Five discusses the use of MCPI within astrodynamics. Three orbits are analyzed throughout the chapter using different dynamics. The three orbits are a Super GEO orbit, a LEO circular orbit, and an eccentric orbit. The initial analysis into these orbits is done using two body

motion in Cartesian Coordinates. The distinction between using first or second order MCPI is discussed. The final time, value of M, and error tolerance is then varied and the measures of convergence are analyzed.

Chapter Five continues with the analysis of the same three orbits when perturbations are introduced. The perturbations analyzed are J2, Drag, and the combination of the two. Due to the necessity for a larger convergence window to see the effects of the perturbations, Multi-segment MCPI is created and discussed. The new form of MCPI is then used with each of the perturbed orbits and its ability to converge is discussed by varying the values of the final time of each segment, the value of M for each segment, and the error tolerance. The convergence measures are discussed with respect to each variation in each perturbed orbit trajectory.

Chapter Six includes a summary of paper and includes important conclusions. The discussion of each conclusion will include further discussion into the challenges related to MCPI and any further research will can be done.

1.2 Contributions

The primary contributions given by this thesis include the creation of a MATLAB program that increases the usability of MCPI. The code includes the ability to easily change MCPI and function variables in the same style as the use of ODE45. The program is universally applicable and can be applied to first and second order MCPI. The output of the program include the time, states, error history, error profile, number of iterations, and a convergence flag.

The derivation of the Absolute Error Analysis with respect to MCPI is a considerable contribution offered by this thesis. The derivation includes use in first and second order MCPI and the functionality of the Absolute Error Analysis is included in the previously mentioned MATLAB code. Along with the derivation of the Absolute Error Analysis, an analysis into the means of evaluating convergence when compared with Relative Error Analysis is included. Historically, the use of Relative Error Analysis has been the primary means of evaluating convergence and so an understanding into the benefits and limitations of each means of evaluating convergence is necessary.

Included in the thesis is a discussion of how the change of parameters within MCPI affect the ability for the algorithm to converge. The discussion of how changes in segment length, number of sampled points, and acceptable tolerance of the algorithm provides further insight into how to assure that MCPI will be able to converge on a solution as needed. And ensure that the converged solution is reached at maximum efficiency.

The creation of the Multi-segment MCPI code, which is included in the MATLAB program, is an important contribution of this paper. By including the use of Multi-segment MCPI, the convergence window of MCPI increases greatly while potentially decreasing the number of points necessary for each segment. The use of Multi-segment MCPI nearly eliminates the limitation of the unknown limited convergence window of MCPI. The thesis includes an analysis into how Multi-segment MCPI can be most efficiently used with respect to the perturbed trajectories of the three orbits. The variation of the parameters used in Multi-segment MCPI offers a further understanding into what potential limitations and benefits of Multi-segment MCPI may exist.

2. MODIFIED CHEBYSHEV-PICARD ITERATION

2.1 Numerical Integration

Integration in the process by which the known dynamics of a system can be used to find the states at given times. Given the initial or final position of an object and the velocity and acceleration dynamics, the position of the object at each time is known. The change of state dynamics are defined as a function of the time and the state at that point in time as shown in equation 2.1, where x is the state which is changing over time. For example, x may be the position in meters where $f(t, x(t))$ would be the calculation of the velocity in meters per second.

$$f(t, x(t)) = \dot{x} \tag{2.1}$$

Some functions have known integrals that can be expressed explicitly, as shown in equation 2.2. The ability to solve for these integrals explicitly allows for the states at each point in time to be solved directly.

$$\sin t = \int \cos(t) dt \tag{2.2}$$

There are various functions that cannot be exactly integrated easily due to the complexity of the problem. Some functions may be impossible to integrate exactly due to unknown integrals. An example of these functions is Newton's fundamental equation of gravitation when perturbations are introduced as shown in equation 2.3.

$$F = G\frac{m_1 m_2}{r^2} + P \tag{2.3}$$

where F is the force due to gravity, G is the universal gravitational constant, m_1 is the first body's mass, m_2 is the second body's mass, r is the distance between the two bodies, and P is some perturbing force.

For equations like these, an estimation of the integral can be found using various numerical methods. Numerical methods estimate the integral by solving for the function evaluation at various points and using the result to estimate the true dynamics of the system and the integral.

These numerical methods can be extremely accurate when the step size between evaluations is small enough. However, as the step size decreases, the computational cost increases. As smaller steps are taken towards a fixed end point, more steps are necessary. As more steps are included, the number of evaluations of the function increases and the time necessary to receive a solution increases as well.

2.1.1 Runge-Kutta Methodology

A commonly used numerical integration method is the Runge-Kutta methodology. The most common variation of this methodology is generally known as "RK4". This Runge-Kutta version is a fourth order estimation of the desired function integration. [1] As shown in equation 2.4, the initial value at each time step is $\mathbf{y}_0$, the time step is given as h, and the estimated value after the time step is given as $\mathbf{y}$. This scheme solves for the states after a time stop of h as represented by $\mathbf{y}$.

$$\mathbf{y} = \mathbf{y}_0 + \frac{h}{6}(\mathbf{k}_0 + 2\mathbf{k}_1 + 2\mathbf{k}_2 + \mathbf{k}_3) \tag{2.4}$$

where the values of $\mathbf{k}_i$ are coefficients defined in equation 2.5 [1]. Here $\mathbf{f}(t, \mathbf{y})$ represents the function being integrated and the change of state of $\mathbf{y}$ over the time step.

$$\begin{bmatrix} \mathbf{k}_0 \\ \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{f}(t_0, \mathbf{y}_0) \\ \mathbf{f}(t_0 + \frac{h}{2}, \mathbf{y}_0 + \frac{h}{2}\mathbf{k}_0) \\ \mathbf{f}(t_0 + \frac{h}{2}, \mathbf{y}_0 + \frac{h}{2}\mathbf{k}_1) \\ \mathbf{f}(t_0 + h, \mathbf{y}_0 + h\mathbf{k}_2) \end{bmatrix} \tag{2.5}$$

The function is evaluated four times for each time step of RK4. If there are 100 steps made in the integration, 400 function evaluations are necessary. This number of evaluations increases for higher order Runge-Kutta algorithms. For example, five evaluations are necessary at each time

step for a fifth order Runge-Kutta method. By increasing the number of evaluations, the accuracy of the estimation increases without decreasing the time step value but the number of evaluations increases.

One of the most commonly used integrators in MATLAB is ODE45. ODE45 is based on an explicit Runge-Kutta (4,5) formula. ODE45 offers the ability to integrate a custom function over a given time range with adaptive step-size and the ability customize options such as error tolerance. MATLAB has other integration options however, ODE45 tends to give satisfactory results in satisfactory time. Whereas other methods may be more accurate but they require more time, or offers results quickly with lower accuracy.

2.2 Need for Improved Methodology

While ODE45 offers generally satisfactory results, it would be desirable to be able to receive more accurate results in less time. Thus by decreasing the time necessary, the results can be more quickly used in the following steps of whatever method required the results. This may be a simulation of spacecraft dynamics or any other estimation of values.

One way of minimizing the computational cost of an estimation would be to compute the estimate in parallel. However, due to the sequential nature of an integration, this can be difficult. The entire purpose of the integration may be to find the trajectory of a spacecraft and therefore the values along the trajectory must be found in series and can not be found in batches of values.

2.3 Derivation of Modified Chebyshev-Picard Iteration

Modified Chebyshev-Picard Iteration (MCPI) is an integration method that takes advantage of the benefits of Chebyshev Polynomials and Picard Iteration to create a more computationally efficient estimation of the integral of a function. MCPI is restricted by unknown, a priori, time span restrictions. However, when within the time span restrictions, MCPI can be a more efficient method of obtaining a solution than previously stated methods like ODE45.

MCPI was developed by Dr. Junkins and Dr. Bai. [2] Recent work extended this approach for effective solution initial and boundary value problems. [3] [4]

2.3.1 Chebyshev Polynomials

Chebyshev Polynomials are a series of polynomials used to approximate a series of values as shown in equation 2.6. The Chebyshev Polynomials as listed as $T_i(t)$ which when multiplied by some series of coefficients, $\boldsymbol{\alpha}$, approximate the value of $x(t)$. The Chebyshev Polynomials are dimensionless, as seen by the evaluation of the polynomials using $\cos$, which results in dimensions of x and α being the same. In the example of x being positions with units of meters, α would also be in the units of meters.

$$x(t) = [T_0(t), T_1(t), ...T_N(t)]\boldsymbol{\alpha} \tag{2.6}$$

In MCPI these values are the values of functions. However, Chebyshev Polynomials can approximate any series of values. They are not bound to linear or nonlinear equations. To properly approximate the values, the order of the polynomials may need to be increased.

The values of each Chebyshev Polynomial is calculates as shown in equation 2.7. The recursive relationship of the polynomials allows the Polynomials to be evaluated at each node point in an increased order easily.

$$\begin{aligned} T_0(\tau) &= 1 \\ T_1(\tau) &= \tau \\ T_n(\tau) &= 2\tau T_{n-1} - T_{n-2} \text{ for } n \geq 2 \end{aligned} \tag{2.7}$$

Additionally, the integral and derivative of each polynomial can be explicitly expressed irregardless of the function being approximated. The evaluation of the integral and derivative can be done once at the beginning of the algorithm and will remain true throughout the algorithm, thus requiring less evaluations and lowering the computational cost.

Cosine sampling, as shown in equation 2.8, is used to select the node points of the Chebyshev Polynomials on the interval of -1 to 1. Cosine sampling is used because it decreases the Runge Phenomenon and contributes to the Orthogonality of the matrix of Chebyshev Polynomials.

$$\tau = -\cos\frac{j\pi}{M} \text{ for } j = 0, 1, 2, ..., M \tag{2.8}$$

The Runge Phenomenon is the result of uniform sampling that leads to inaccuracy near the end points. In estimation, the estimation of a point has a reliance on the points nearby. When there are not sufficient data points nearby, a satisfactory estimate of the point can be difficult to find. For data points near the ends of the time span, initial and final times, there are little to no data points on other side of them to provide support for the estimation.

By using cosine sampling, there is an increased number of points near the end points and a decreased number of points near the center of the time span. This increased number of points leads to higher accuracy at the endpoints. The decreased number of points in the center will lead to lower accuracy in the center of the time span but that lower accuracy is generally acceptable. Commonly, when integrating forward, there is a focus on the accuracy at departure and arrival. While the accuracy in the center of the trajectory will need to stay within required bounds, the accuracy at the end points is frequently more important. Numerical methods of integration are commonly used in Two Point Boundary Value Problems where the explicit goal is to match the initial and final values to the given values. Inaccuracy at endpoints here would be inexcusable.

A benefit of Chebyshev Polynomials is that the integral of the Polynomials is well defined. Therefore, when integrating numerically, the integral can be defined as a predetermined function of time at each node. This saves time computationally because this integral is dependant on time only and not the function which is being integrated. The integrated polynomials can be defined once at the start of the function and the resulting matrix is used multiple times during the evaluation without need for repeated calculation.

2.3.2 Picard Iteration

Picard Iteration is a numerical integration scheme where an initial estimate of the states over the trajectory is used to find the correct values of the states. Equation 2.9 shows the evaluation of the state according to some estimated value of the state. The new value of $x^i(t)$ is then used to

solve for x^{i+1} and so on. The state is iterated over and corrected with each iteration until the step size of the iteration is within a required tolerance.

$$x^i(t) = x_0 + \int_{t_0}^{t} f(s, x^{i-1}(s))ds \tag{2.9}$$

Picard iteration allows a simple estimation of the states to converge to the true solution. Picard Iteration can only converge within a given time span. This time span is not generally known beforehand because it varies due to the equation being integrated. The integration scheme also requires the integral of the function to be known to evaluate it at each state and create the new estimation of the states.

The number of iterations necessary to converge varies according to the initial estimate of the system. A common initial estimate of the states is to set all states equal to the initial values or to set all non-initial states to zero. The practice of setting the estimate equal to zero can cause a possible division by zero if using relative error where the current iteration is compared with the prior iteration values.

2.3.3 Chebyshev-Picard Iteration

The ability to explicitly evaluate the integrals of the Chebyshev Polynomials means that when combined with Picard Iteration, the function's unknown explicit integral can be approximated and used to converge to a solution. This evaluation can be applied to functions of any order. However, this thesis will only discuss the derivation and convergence evaluation of the first and second order MCPI.

The derivation that follows is based on the work of Junkins and Bai. [2]

2.3.3.1 First Order MCPI

By defining the dynamics of the First Order MCPI as the function stated in equation 2.1, the function can be evaluated using Chebyshev Polynomials as shown in equation 2.10. The dimension of $\mathbf{g}$ is the state value. Whereas $\mathbf{f}$ would be, for example, meters per second, when multiplied with time to result in an evaluation of $\mathbf{g}$ that would simply be in terms of meters.

$$\mathbf{g}(\tau, \mathbf{x}(\tau)) = \omega_2 \mathbf{f}(t, \mathbf{x}(t)) = \Phi \boldsymbol{\beta} \tag{2.10}$$

The time domain is transformed from t to τ to ensure that the function is being evaluated from -1 to 1. This time domain is required to keep the orthogonality property of the Chebyshev Polynomials. This transformation is done using ω_2 which is equal to

$$\omega_2 = \frac{t_f - t_0}{2} \tag{2.11}$$

The matrix Φ in equation 2.10 is a matrix of the $N + 1$ Chebyshev Polynomials evaluated at each of the $M + 1$ nodes as shown in equation 2.12. Where the value of each polynomial is evaluated according to equation 2.7.

$$\Phi = \begin{bmatrix} T_0(\tau_0) & T_1(\tau_0) & \dots & T_N(\tau_0) \\ T_0(\tau_1) & T_1(\tau_1) & \dots & T_N(\tau_1) \\ \vdots & \vdots & \ddots & \vdots \\ T_0(\tau_M) & T_1(\tau_M) & \dots & T_N(\tau_M) \end{bmatrix} \tag{2.12}$$

The values of M and N are not by definition the same. However, for this thesis the value of M and N will be equivalent throughout even when the value of M is stated to change, the value of N will change to the same value.

Then using a weighted least-squares fit, the value for the coefficients $\boldsymbol{\beta}$ in equation 2.10 can be found. Because the value of M is equal to the value of N, the least squares fit is not entirely necessary. The value of $\boldsymbol{\beta}$ can be found using the inverse of Φ. However, depending on the value of M and N, this square matrix may be very large (more than 40x40) and the inverse can be computationally costing to evaluate. By using a weighted least-squares fit, a weight matrix can be used to make Φ orthogonal or so that $\Phi^T W \Phi$ is a diagonal matrix. Thus, the evaluation of the integral is simple and fast. Equation 2.13 shows the least squares evaluation and equation 2.14 shows the diagonal weight matrix that takes advantage of the orthogonality stated earlier.

$$\boldsymbol{\beta} = (\Phi^T W \Phi)^{-1} \Phi^T W \mathbf{g} \tag{2.13}$$

$$W = diag([1/2, 1, 1, \ldots, 1, 1, 1/2]) \tag{2.14}$$

The value of $\boldsymbol{\beta}$ does not vary with time and so when integrated with Picard Iteration along with the Chebyshev Polynomials, the polynomials are the only ones that change during the integration. The integration of the polynomials is known as S and the matrix can be shown to be equivalent to

$$S = \begin{bmatrix} 1/4 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & -1/2 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1/4 & 0 & -1/4 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 & \frac{1}{2(N-2)} & 0 & -\frac{1}{2(N-1)} \\ 0 & 0 & \ldots & 0 & 0 & 0 & \frac{1}{2(N-1)} & 0 \\ 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \frac{1}{2N} \end{bmatrix} \tag{2.15}$$

The resulting equation for first order MCPI is

$$\mathbf{x}^i(\tau) = \mathbf{x}_0 + S\boldsymbol{\beta} \tag{2.16}$$

2.3.3.2 Second Order MCPI

The derivation of second order MCPI follows the same general principles as stated in first order MCPI. The base equation being integrated does change to the equation shown in 2.17. This equation must be integrated twice instead of once.

$$\ddot{x} = f(t, x(t), v(t)) \tag{2.17}$$

The Picard Iteration necessary to solve for the states is shown as seen in equation 2.18.

$$\mathbf{x}^i(\tau) = x_0 + \int_{-1}^{\tau}(v_0 + \int_{-1}^{s}(\mathbf{f}(q, \mathbf{x}^{i-1}(q), \mathbf{v}^{i-1}(q))dq)ds) \tag{2.18}$$

The first integration of equation 2.17 solves for the first order change of state, stated as v_0, in the same manner as the first order MCPI. However, the resulting solution is not $\mathbf{v}^i(\tau)$ but instead is represented as $\boldsymbol{\beta}^i$. Second order MCPI actually doesn't require that the velocity at each time be evaluated in order to solve for $\mathbf{x}^i$. However, the evaluation of $\mathbf{v}^i$ is simply calculated as shown in equation 2.20.

$$\boldsymbol{\beta}^i = \mathbf{v}_0 + S(\Phi^T W \Phi)^{-1}\Phi^T W \mathbf{f}\omega_2 \tag{2.19}$$

$$\mathbf{v}^i(t) = \Phi(\tau)\boldsymbol{\beta}^i \tag{2.20}$$

Using $\boldsymbol{\beta}^i$ in nearly the same way that $\boldsymbol{\beta}$ was used in equation 2.16, the coefficients related to the states are calculated. Equation 2.21 shows the addition of ω_2 in order to transform the results into the correct time domain. The resulting coefficient $\boldsymbol{\alpha}^i$ is then multiplied with the Chebyshev Coefficients to evaluate each state in equation 2.22.

$$\boldsymbol{\alpha}^i = x_0 + S_2\beta^i\omega_2 \tag{2.21}$$

where S_2 is a condensed version of S found in equation 2.15.

$$\mathbf{x}^i(t) = \Phi(\tau)\boldsymbol{\alpha}^i \tag{2.22}$$

2.4 Discussion of Limitations and Benefits of MCPI

MCPI, like all numerical integration methods, has limitations that are unique to itself along with benefits that are unique. As the understanding of the limitations and benefits increase, the utility of MCPI can increase. This increased utility will lead to the ability for MCPI to evaluate more complex problems with either a more accurate solution in the same time or receiving a solution of equal accuracy in a shorter time span.

2.4.1 Limitations of MCPI

Picard Iteration has an undefined convergence window for each problem. While there have been studies into the relationship between the eigenvalues and the ability for MCPI to converge there are not rules by which a convergence window can be defined for all equations. The inability to define convergence windows limits the utility for MCPI to be used because it cannot be applied equally to all problems. For example, in an optimal control trajectory, MCPI may convergence to the trajectory for some of the control options but may not converge for others. It is unknown a priori and leads to the use of other numerical methods.

Because cosine sampling is non-uniform sampling, there will be larger gaps between nodes than in a uniform sampling. These large gaps in the center of the interval can lead to missed events or values. For example, if a trajectory crashes into the earth or goes outside of a predefined bound in those gaps, the event may not trigger and thus fail to properly evaluate the trajectory. Similarly, if the trajectory passes near the moon, or any 3rd body, but only during one of those gaps, the gravitational effect of the moon using conic sections could be ignored.

In the center of the interval, the trajectory is known to be less accurate due to cosine sampling. The potentially large gaps between nodes in the center of the interval also make it difficult to properly match the dynamics of the system. If the system is linear or close to linear between these points, the gap is not a concern. However, if the system is extremely nonlinear, this gap may be too large and require more points to converge, thus requiring more evaluations and increasing the computational cost. An example of this is that of a Low Earth Orbit (LEO) evaluated in Cartesian Coordinates, where if the gap is too large, the spacecraft may miss large sections of the orbit and getting the dynamics to properly match would be difficult.

Due to the possible inaccuracy of MCPI, primarily in the center of the interval, the number of points necessary is not always clear. It has been found that for some functions there is a clear benefit in increasing the number of points up until it essentially reaches a plateau. Up until this plateau, the number of iterations necessary to converge decreases as the number of sample points increases. The optimal number of samples would be at the edge of that plateau. At this point,

the number of iterations is minimized and the number of function evaluations is also minimized. However, this plateau is not explicitly defined and changes according to each system.

2.4.2 Benefits of MCPI

Chebyshev Polynomials have explicit defined integrals which are only dependent on the node in the time domain. This means that by defining the number of nodes and order of the polynomials, the polynomials, their derivatives, and their integrals are known and constant throughout the algorithm. Because these matrices, along with the integration matrices S and S_2, are constant throughout, the only evaluations for each iteration are the evaluations of the function once at each node point. This limits the number of function evaluations and reduces the computational time.

There are times where an initial guess of the states can be made for a system with the knowledge that the true states will not vary a large amount from this estimate. In MCPI, this initial estimate can be given and MCPI can converge faster than if given a "cold start" or estimating all states as 0 or as their initial values. An example of this would be with a perturbed trajectory. If the unperturbed orbit is known, it can be given as the initial estimate which will quickly converge to the truth. However, for a RK4 method, no initial estimate is given, simply the initial states and the dynamics. Thus having a greater understanding of the problem cannot be utilized.

2.4.3 Need for Further Understanding

By gaining a deeper understanding of how MCPI converges for various problems, the limitations of MCPI can be avoided and the benefits can be maximized. While this understanding is not an explicit equation for optimal convergence, working the process in relation to various examples is an effective means of gaining a deeper understanding into MCPI. Specifically, working through various options of MCPI can offer the ability to solve the problem more efficiently in the future.

2.5 Duffing Oscillator Example

A simple example of a nonlinear problem that can be solved in MCPI is the Duffing Oscillator defined as

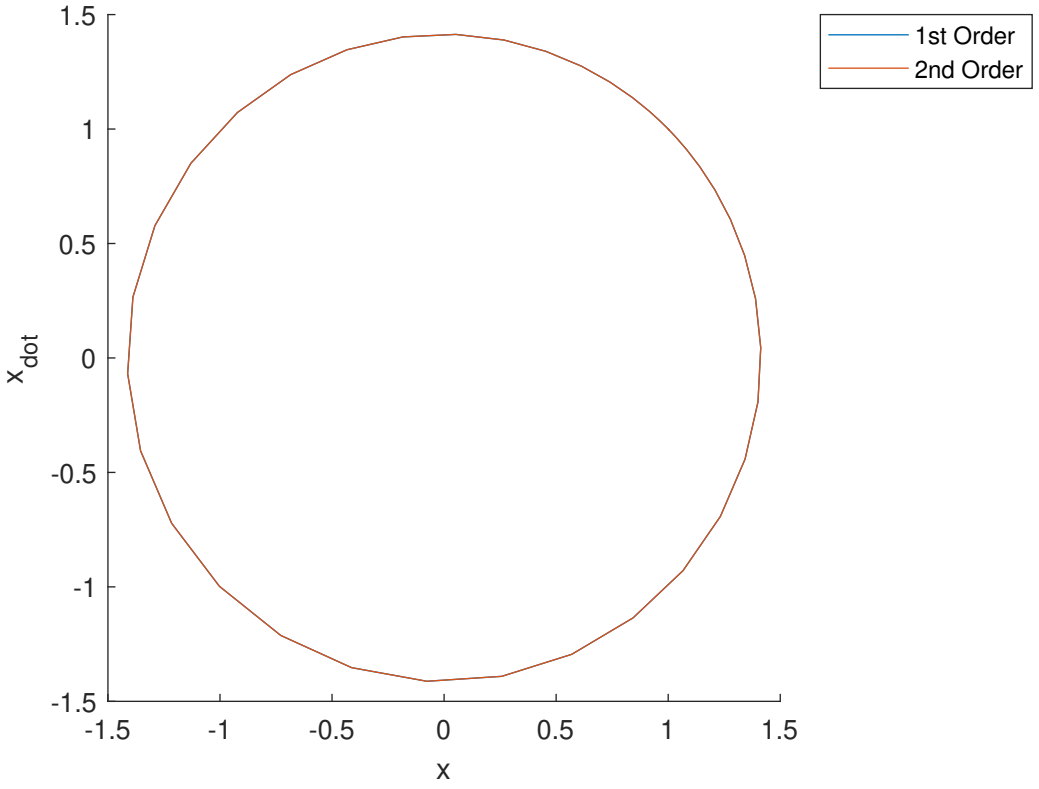

Figure 2.1: Phase Portrait of Duffing Oscillator Evaluated for First and Second Order MCPI

$$\ddot{x} = -x - \epsilon x^3 \tag{2.23}$$

where ϵ is the coefficient related to non-linearity.

The duffing oscillator can be solved using either first or second order MCPI. By using an initial value of the position and velocity of 1, figure 2.1 shows the resulting phase portrait. In this example, the time span was from 0 to 2π which is approximately the period of the oscillator with the non-linearity parameter set as $\epsilon = 1e-3$.

In this example, the value of M and N was set to 50 and the acceptable tolerance was set to $1e-8$.

2.5.1 Convergence Measures

To evaluate the converge of the solution shown in figure 2.1 there are various means of evaluating convergence or means of understanding what will converge in both 1st and 2nd order MCPI. These convergence measures will be used to evaluate convergence in Chapters 4 and 5 as well.

2.5.1.1 Error Profile

The algorithm will define a solution as converged if the maximum value in the error profile is less than the required tolerance. By default the error tolerance is defined as $1e-8$ but can be changed according to the requirements of the estimation. In later chapters, the definition of error is given and the effect of changing the acceptable tolerance is evaluated.

Figure 2.2 shows the error profile of the first and second order solutions. Any point at which the line is discontinuous is a point where the error was evaluated as exactly 0. If the algorithm evaluated that the error at a point was less than $1e-20$, this value was reported as 0 to limit memory usage.

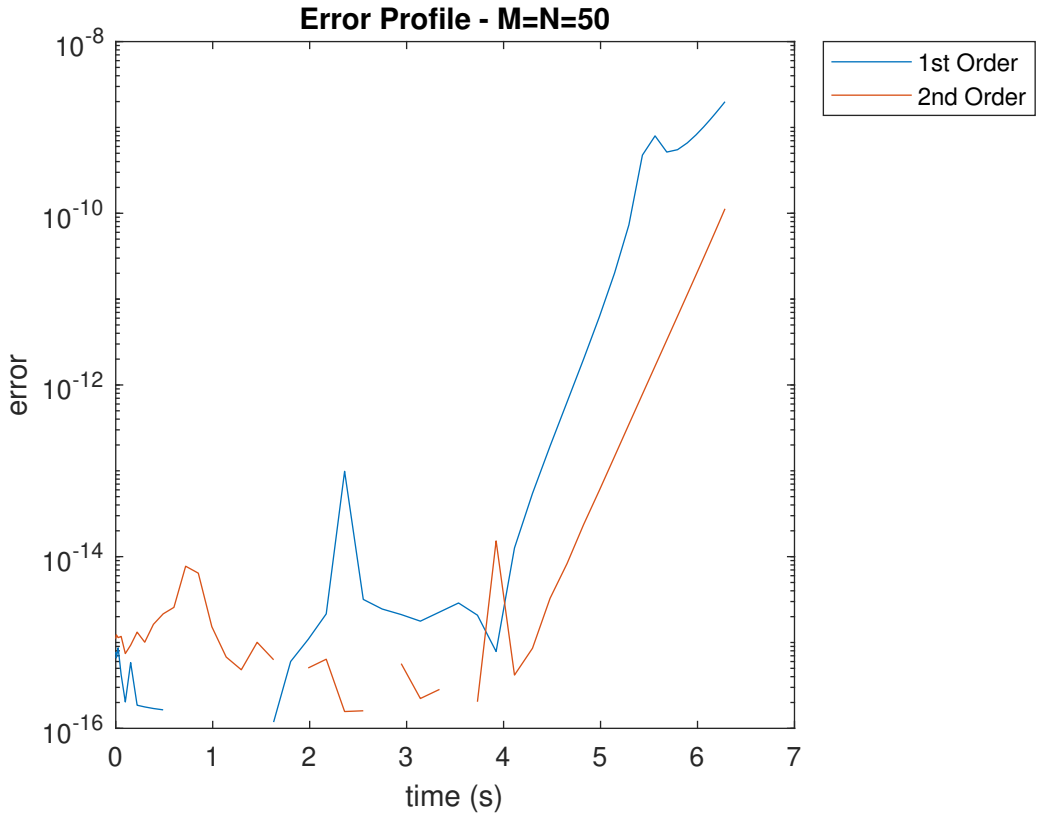

Figure 2.2: Error Profile of Duffing Oscillator

The figure illustrates that the error for both first and second order MCPI are approximately machine zero ($1e-16$) at the beginning of the trajectory and both errors began to increase in the second half of the trajectory but stayed within the allowable tolerance.

2.5.1.2 Iterations to Converge

For values that converge, an important measure is how quickly they are able to converge. If a solution requires a large number of iterations (more than 100) to converge, then MCPI may not be the most effective tool for the problem.

Figure 2.3 shows the maximum error in the system for each MCPI method at each iteration. Both evaluations started with the same initial estimate of the trajectory of which the entire trajectory was the initial states. This is the default estimation for the MCPI algorithm.

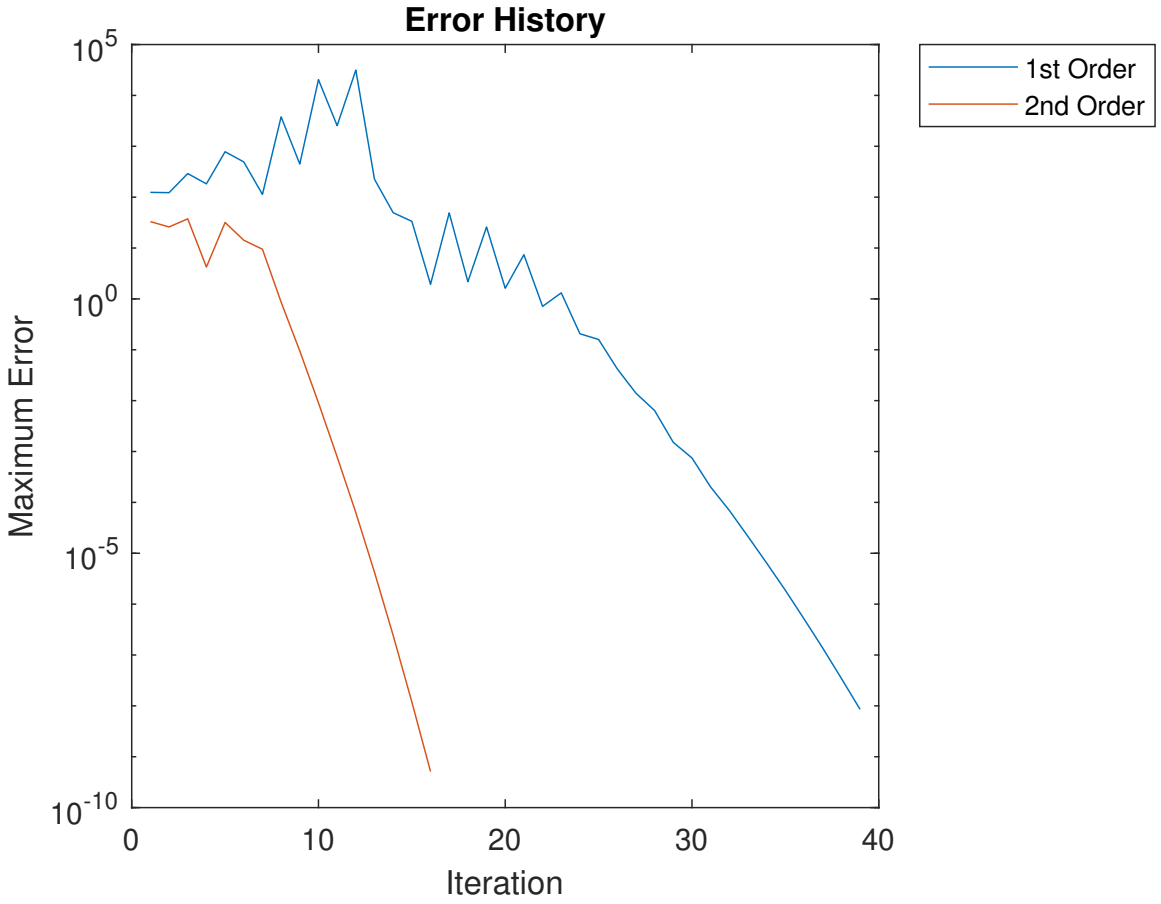

Figure 2.3: Error History of Duffing Oscillator

The second order MCPI began with a lower error and continued to improve resulting in a converged solution in 16 iterations. The first order MCPI increased in its maximum error for the first approximately 12 iterations before slowly converging to a solution in 39 iterations. While this disparity of 16 to 39 iterations is not a consistent ratio for iterations necessary to converge, it is consistent that second order MCPI is generally able to converge in fewer iterations than first order MCPI.

2.5.1.3 Maximum Final Time

A primary limitation of MCPI is the lack of definition of what problems will be able to converge in a given time span and what is the maximum time span over which a given problem is able to

converge. The measure of Maximum Final Time is used in the following chapters when making variations to the problem to evaluate how the variation affects the ability for MCPI to converge. For example, as the values of M and N increase linearly, does that correlate with a linear increase in the maximum final time? If not, what is the relationship?

3. ERROR ANALYSIS

To understand if a solution has been found using any method, a means of evaluating convergence must be decided on. This error evaluation may be based on the variation from the truth or the variation from prior iterations. The error evaluation may also be based on the current states or the current dynamics. It is rare that the true values of the system can be used as the measurement for error analysis. Historically, the relative analysis is used in MCPI.

3.1 Relative Error Analysis

Relative error analysis is the evaluation of convergence based on the prior iteration values of the states or derivatives of the states. MCPI has previously used the relative evaluation of the states to evaluate convergence.

$$\mathbf{RelativeError}\% = \frac{|\mathbf{x}^i - \mathbf{x}^{i+1}|}{|\mathbf{x}^i|} \tag{3.1}$$

The relative error as shown in equation 3.1 does show properly if the Picard iteration is converged to a solution. However, it does not properly show that the converged solution is the correct solution. It is reasonable that there are not sufficient data points to converge to the true solution and so the system will converge to the best solution that it can. However, if the program reports that the solution is converged and the solution is not at the true solution, it would require inspection to verify if the solution is true.

The need for inspection to confirm the solution means that the algorithm cannot be trusted in an automated system or complex system. This means that a new means of evaluating convergence may be necessary. By evaluating the dynamics of the system to the true dynamics at each point, the convergence evaluation is more reliable.

3.2 Derivation of Absolute Error Analysis

The absolute error is defined as the error between the solution's dynamics and the true dynamics of the states. By comparing the dynamics at each level, the evaluation of convergence is more strict and more reliable. It will not report as converged for every converged solution but only for the true to dynamics converged solution.

$$\mathbf{AbsoluteError}\% = \frac{|\mathbf{f}(\mathbf{x})^i - \mathbf{Comp}^i|}{|\mathbf{f}(\mathbf{x})^i| + \zeta} \tag{3.2}$$

where $\mathbf{f}(\mathbf{x})^i$ is the evaluation of the function at each state in the iteration i and $\mathbf{Comp}$ is the estimation of the function at each state according to MCPI. The value ζ is a small value to avoid any potential evaluation problems when the value of $f(x)$ is very close to zero. For the majority of the thesis this value is set to zero, unless otherwise stated.

If the value of the absolute error is within the defined error bounds, the system has converged to the true solution according to the dynamics. By default, the error bounds in the code are set to $1e-8$ but can be changed. An analysis of how the change in error tolerance affects various problems is found in chapters 4 and 5.

3.2.1 First Order Derivation

Using the defined relationship between the state at each time and the chebyshev polynomials shown in equation 3.3, the dynamics of the system can be derived for a comparison.

$$\mathbf{x}(t)^i = \sum_{n=0}^{N} \beta_n T_n(t) \tag{3.3}$$

where N is the order of the MCPI, β_n is the coefficients of the solution found with MCPI, and $T_n(t)$ are the corresponding chebyshev polynomials.

The value of β_n is constant with time and therefore by taking the derivative of equation 3.3, the resulting equation is found as

$$\mathbf{f}(\mathbf{x})^i = \frac{d\mathbf{x}^i}{dt} = \sum_{n=0}^{N} \beta_n \frac{dT_n}{dt} \tag{3.4}$$

Here the known derivatives of the chebyshev polynomials gives the predefined relationship of

$$\frac{dT_n}{d\tau} = nU_{n-1} \tag{3.5}$$

This means that the summation is no longer from zer to N but now from one to N because when $n = 0$ then the value of U_{n-1} is undefined and multiplied by a zero.

The derivative of the chebyshev polynomials is with respect to the adjusted time span and not seconds. To account for this, the resulting derivative needs to be divided by the time constant ω_2.

The value of U_n is defined as chebyshev polynomials of the second kind. [5] Where just as the chebyshev polynomials of the first kind, T, they can be defined as a recurrent series as shown as

$$U_0(\tau) = 1 \tag{3.6}$$

$$U_1(\tau) = 2\tau \tag{3.7}$$

$$U_2(\tau) = 4\tau^2 - 1 \tag{3.8}$$

$$U_{n+1}(\tau) = 2\tau U_n(\tau) - U_{n-1}(\tau) \tag{3.9}$$

The evaluation of the function at each time is then found to be

$$\mathbf{f}(t) = \frac{d\mathbf{x}^i}{dt} = \frac{1}{\omega_2} \sum_{n=1}^{N} \beta_n n U_{n-1}(\tau) \tag{3.10}$$

Here, the value of $\mathbf{Comp}$ is calculated. Thus the absolute error can be calculated as a comparison between $\mathbf{Comp}$ and $\mathbf{f(t)}$.

$$\mathbf{AbsoluteError}(t_i) = \frac{|\mathbf{f}(t_i, x_i) - \frac{1}{\omega_2} \sum_{n=1}^{N} \beta_n n U_{n-1}(\tau_i)|}{|\mathbf{f}(t_i, x_i)| + \zeta} \tag{3.11}$$

This result is the absolute error of a first order system.

3.2.2 Second Order Derivation

MCPI is also efficient for a second order system. The derivation of the second order system follows the same methodology as the first order system.

$$\mathbf{f}(t, \mathbf{x}) = \frac{d^2\mathbf{x}^i}{dt^2} = \sum_{n=0}^{N} \beta_n \frac{d^2 T_n}{dt^2} \tag{3.12}$$

where the value of of the second derivative of the chebyshev polynomials is found as

$$\frac{d}{d\tau}\frac{dT_n}{d\tau} = \frac{d}{d\tau} n U_{n-1} = \frac{n((n+1)T_n - U_n)}{\tau^2 - 1} \tag{3.13}$$

The value of $\frac{n((n+1)T_n - U_n)}{\tau^2 - 1}$ is undefined when τ is at the end points of -1 and 1. [6] It can be proven that the value can be approximated as

$$\frac{d^2 T_n}{d\tau^2}|_{x=-1} = \frac{n^4 - n^2}{3} \tag{3.14}$$

$$\frac{d^2 T_n}{d\tau^2}|_{x=1} = (-1)^n \frac{n^4 - n^2}{3} \tag{3.15}$$

Again to calculate the value of $\mathbf{Comp}$ for the second order system, the value must be divided by ω_2.

The absolute error evaluation of the second order system is calculated as

$$\mathbf{AbsoluteError}(t_i) = \frac{|\mathbf{f}(t_i, x_i) - \frac{1}{\omega_2^2}\sum_{n=1}^{N} \alpha_n \frac{n((n+1)T_n - U_N)}{\tau_i^2 - 1}|}{|\mathbf{f}(t_i, x_i)| + \zeta} \tag{3.16}$$

In both first order and second order absolute error evaluation, there is a level in inaccuracy to the dynamics at the end points. This inaccuracy is due to the fact that MCPI is correcting the states of the system and not the dynamics. At the end points the dynamics are not necessary to match. Because of this the end point dynamics are excluded when evaluating convergence. This exclusion should only apply when there are sufficient points that the end points are not isolated. If the end

points have sufficient points for support ($M > 20$), then it the dynamics error at the endpoints is negligible. This is primarily due to the fact that the trajectory as a whole is what is of concern and not necessarily the end points. For a Two Point Boundary Value Problem, this error evaluation may be problematic. However, this type of problem is not evaluated in this thesis.

3.2.3 Comparison of Relative and Absolute Error Analysis

While Relative Error Analysis evaluates whether or not the states are converging to a solution, the Absolute Error Analysis evaluates whether or not the dynamics of the converged solution match the true dynamics at each state. This means that at the minimum Relative Error Analysis requires two iterations. If the initial estimate is accurate, then the second iteration will not change the states by more than the tolerance and the program will report that it is converged and quit.

If the initial estimate of the Absolute Error Analysis is accurate enough, the program will require one iteration instead. This is because the dynamics at that solution are sufficient and it does not need to be compared to a secondary solution. This behavior of one less iteration for Absolute Error Analysis continues forward in general through the examples. It is not always true but is a general estimate for how many iterations may be necessary to converge.

Figures 3.1 and 3.2 show the results of a simple time and position dependant function as defined by equation 3.17.

$$\dot{x} = \cos t + p * x \tag{3.17}$$

where p is a small variation from the base cosine equation. When p is at or near 0, the equation for the change of x is simply $\cos t$ but the introduction of p allows for a slightly more complex problem to be evaluated. First Order MCPI is used to solve this equation.

Both of these evaluations use the vales from table 3.1 as the default MCPI values. However, figure 3.1 uses a p value of $1e-1$ whereas figure 3.2 uses a p value of $1e-2$. These values are one magnitude of order of difference to illustrate the different convergence evaluations of MCPI.

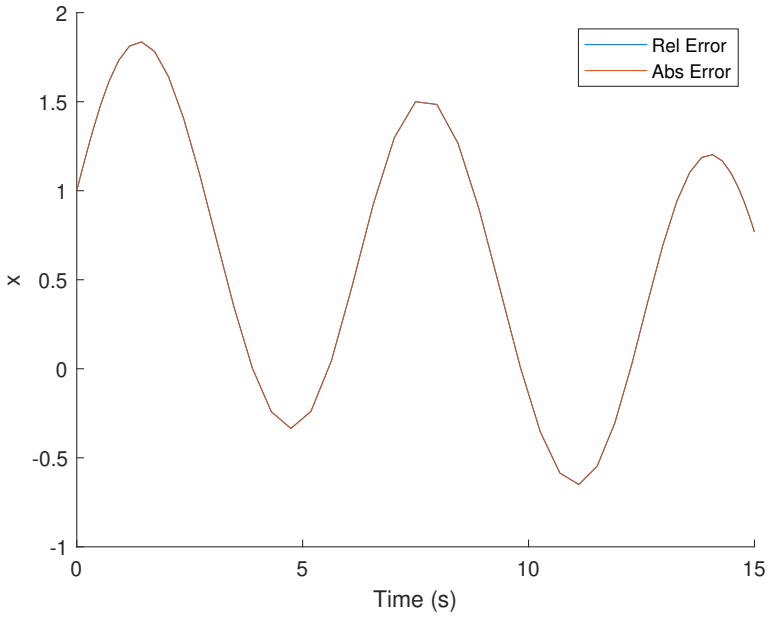

Figure 3.1: Example Problem when $p = 1e - 1$

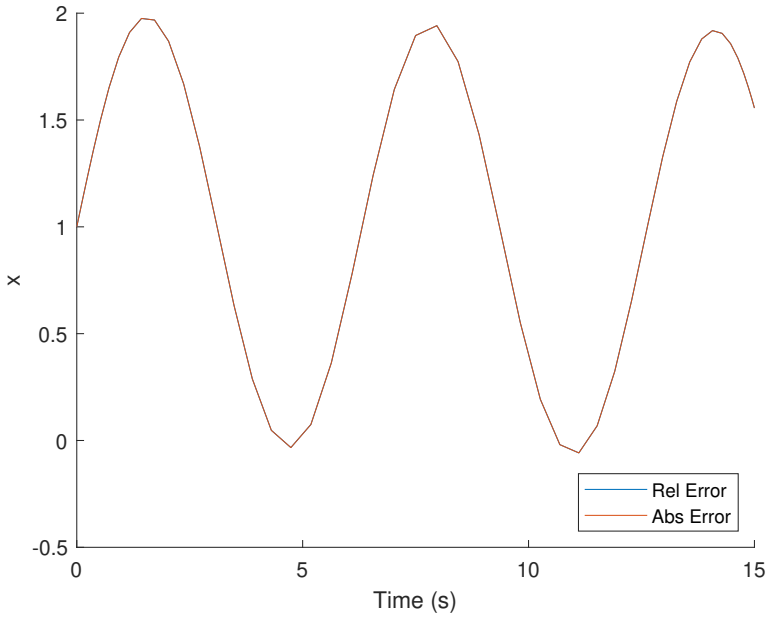

Figure 3.2: Example Problem when $p = 1e - 2$

Table 3.1: Default MCPI Values for Cosine Example Problem

Variable	Value
M	50
N	50
tol	$1e-8$
x_0	1
t_0	0 seconds
t_f	15 seconds
$iter_{max}$	100

3.2.3.1 Convergence Iterations

Figure 3.3 shows the error history of first order MCPI when the problem has the perturbation value of $p = 1e-1$ for both means of evaluating convergence. The MCPI algorithm has not changed, only the definition of convergence has.

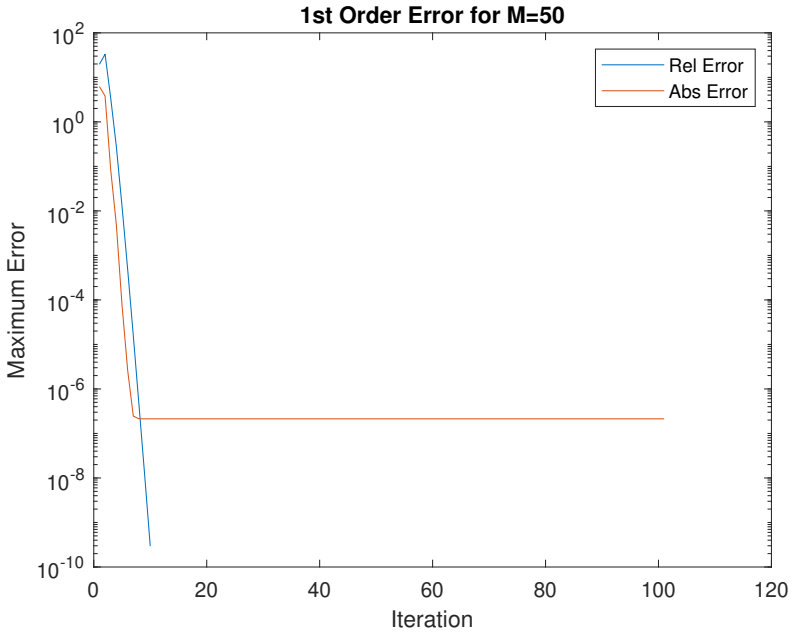

Figure 3.3: Error History of Example Problem when $p = 1e-1$

The problem convergence to a solution in approximately 10 iterations according to Relative

Error Analysis. At that point, the states simply do not change very much. However, Absolute Error Analysis does not report the solution as converged at all. In figure 3.3 there is little discernible difference between the two solutions. However, the Absolute Error Analysis not converging means that the converged solutions does not accurately represent the dynamics defined by equation 3.17 within the assigned error tolerance.

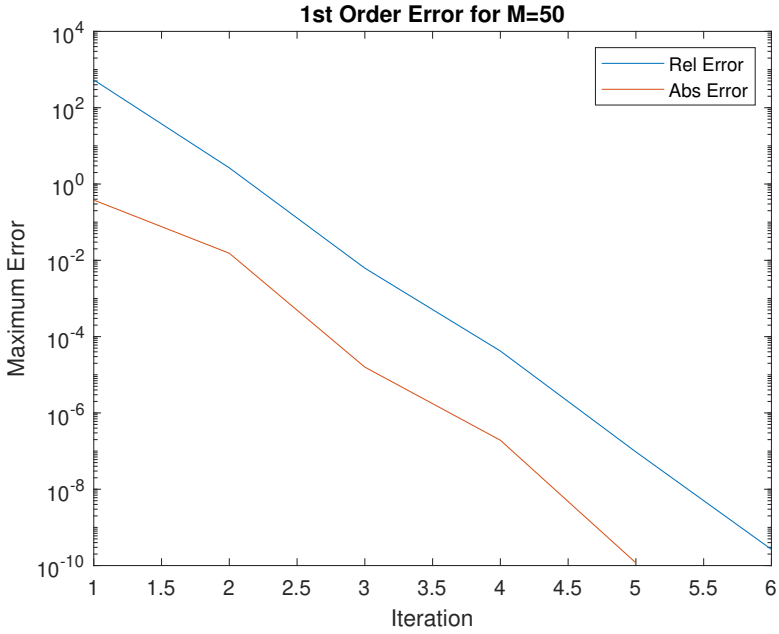

Figure 3.4: Error History of Example Problem when $p = 1e - 2$

Figure 3.4 shows that for this less perturbed equation both means of evaluating error converge to the solution quickly. It is important to note that in this example the Absolute Error Analysis has a smaller maximum error by approximately 2 orders of magnitude each iteration. This leads to an convergence evaluation one iteration sooner than offered by Relative Error Analysis. One iteration may not seem significant but that is 50 less evaluations of the given functions.

3.2.3.2 *Error Profile*

In the example problem, when $p = 1e-1$ the Relative Error Analysis reports the solution as converged. Figure 3.5 shows that for the maximum error for this solution which used approximately 10 iterations, is in the order of magnitude of approximately $1e-10$ which is well below the required acceptable tolerance of $1e-8$. However, the Absolute Error Analysis appears to have 4 different points were the error is more than the allowable tolerance level. These error spikes would likely be mitigated by increasing the value of M or decreasing the final time value.

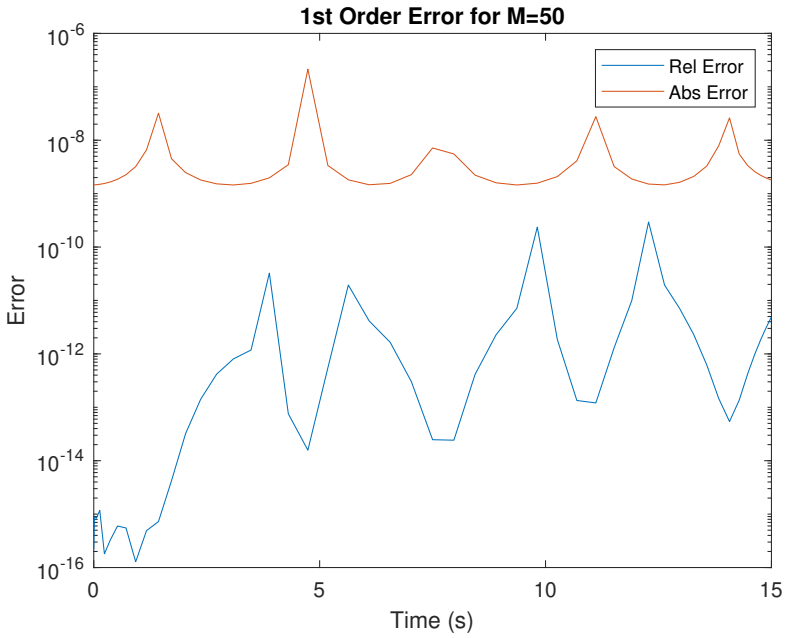

Figure 3.5: Error Profile of Example Problem when $p = 1e-1$

It is clear that although the points may have converged to a solution this solution is not a reliable solution given the required tolerance. The dynamics of the converged solution do not adequately match the dynamics of the true solution.

Figure 3.6 shows the error history as when both methods were able to converge to a solution. Here, all of the values were less than $1e-9$ and the majority of the points were less than $1-e10$.

Even though the Absolute Error was able to converge in one less iteration, there does not appear to be a loss in reliability of the convergence evaluation.

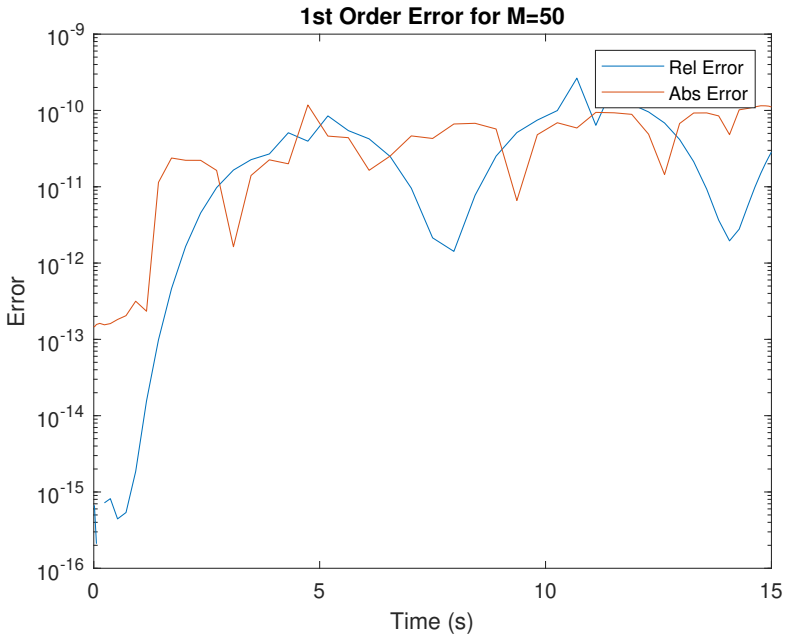

Figure 3.6: Error Profile of Example Problem when $p = 1e - 2$

As noted in the above example, there are situations in which Relative Error Analysis will report a converged solution that is not properly representative of the dynamics of the system. It is because of this that the use of Absolute Error Analysis is a preferred evaluation means when the exact solution is needed.

The two means of evaluating convergence will continue to be analyzed in Chapters 4 and 5.

4. APPLICATIONS TO NONLINEAR OSCILLATORS

A common use of numerical methods is the integration of nonlinear oscillators. These oscillators may have difficult or complex integrals and are used frequently in physics and mathematics. By gaining an understanding into how nonlinear oscillators are able to converge in MCPI, the ability for the tool to be used across various fields increases.

This thesis will use the Duffing Oscillator as the example of nonlinear oscillation. Equation 4.1 defines the duffing oscillator.

$$\ddot{x} = -x - \epsilon x^3 \tag{4.1}$$

The base values for each MCPI variable and Duffing Oscillator variable is given in Table 4.1. The final time of π seconds was selected because it is the value of one half of the period when ϵ equals zero. The value of $\epsilon = 0.9$ was selected so that the non-linearity term in equation 4.1 would be large but not the dominate term in the equation.

Table 4.1: Base Variable Values for Nonlinear Oscillation

Variable	Base Value
M	50
N	50
Tol	$1e-8$
$Iter_{Max}$	100
t_0	0 seconds
t_f	π seconds
ϵ	0.9

In this chapter, the values in table 4.1 will be varied and the effect that the variation has on the ability of MCPI to converge will be discussed.

4.1 Variation in Final Time

Given the unknown maximum convergence window of the base problem, the problem is analyzed with variations in the final time with the remaining variable set as the base values.

The final time will be varied from 1 second to 2π seconds in 0.25 second increments. The value of 2π was selected as the final time because it is equal to one period of the oscillation when ϵ is equal to zero. The period of the oscillation when ϵ is equal to 0.9 is equal to roughly 4.55 seconds

4.1.1 Error Profile

Figure 4.1 shows the error profiles using Absolute and Relative Error Analysis. In the Absolute Error Analysis, it is clear to note the large error values for the final nodes (when $\tau = -1$ or 1.) As discussed in Chapter 3, this error variation is expected due to the derivative of the Chebyshev Polynomials at these points being undefined. The algorithm uses an approximation of the terms that is not included in the maximum error calculation when evaluating convergence.

Both error evaluations were not able to converge to a solution with a final time value of larger than 3.25 seconds for the given MCPI variable values.

The absolute error in figure 4.1 remains at approximately machine zero for as long as necessary before approximately the second half of the trajectory when it begins to slowly increase. This means the absolute error is being held down at machine zero for a little bit longer with each Picard iteration until the final "trail up" is within the required tolerance.

The relative error in each time span has the same spike at approximately 0.4 seconds in. The error for each time span then decreased. Then the error began to slowly increase until its maximum error was within the bounds. The error spike at 0.4 seconds is possibly due to the fact that at approximately this time, the trajectory crosses the 0 axis in velocity. Thus small variations in the trajectory appear to be larger errors. This is the cause of the spikes just after 2.5 seconds as well.

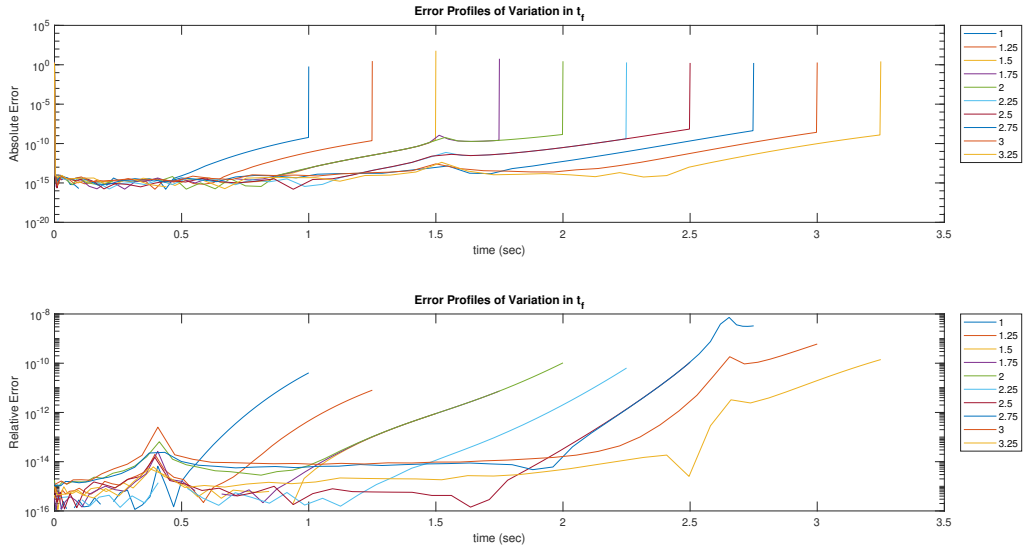

Figure 4.1: Error Profile of Duffing Oscillator With Variations in Final Segment Time

4.1.2 Iterations to Converge

As expected, the Absolute Error Analysis is able to converge in fewer iterations as shown in figure 4.2. In general, the difference in iterations is one. However, both analysis methods had the same maximum final time of 3.25 seconds. This was tested for step sizes as small as 0.001 seconds and throughout, the two analysis methods kept the same maximum final time as the other. The relationship of Absolute Error Analysis requiring approximately one iteration less held true.

When the number of iterations is equal to zero in figure 4.2 MCPI was not able to converge to a solution.

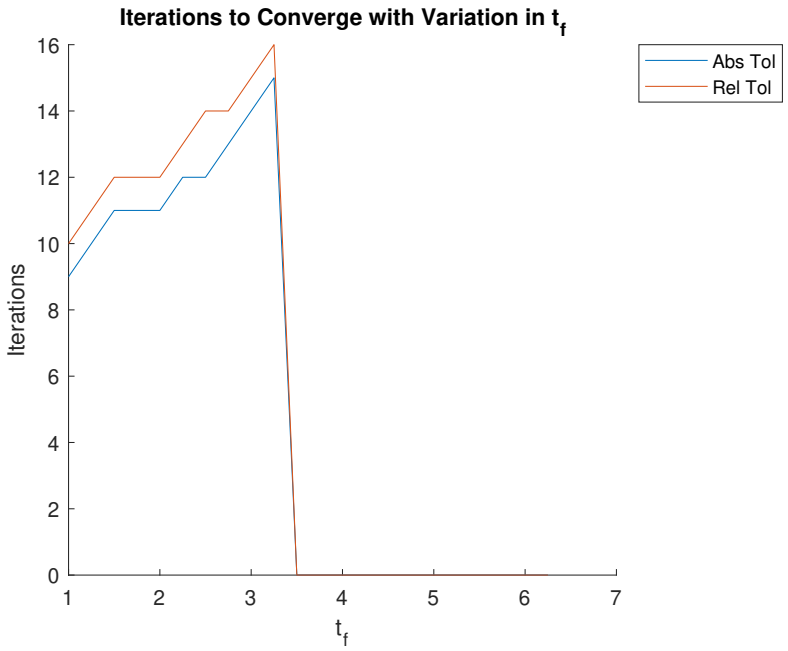

Figure 4.2: Iterations Necessary to Converge with Variation in Final Time of Duffing Oscillator

4.2 Variation in M and N

As the number of sampled points in the MCPI algorithm changes, the ability for the algorithm to converge changes rapidly. For a nonlinear oscillator, there needs to be enough points to capture the motion and ensure that the solution converges properly. However, as stated previously, as the number of sample points increases, the amount of time necessary to evaluate the problem increases as well.

The value of M and N was varied from 10 to 200 at increments of 5. To evaluate the maximum final time in relation to each M value, the value of the final time was varied from 1.5 to 4 seconds in 0.1 second increments. It was verified that by setting the maximum value at 4 seconds, this was not a restriction on the resulting maximum final values.

4.2.1 Error Profile

Figure 4.3 shows the error profiles for each solution when the final time was set to π seconds. Each line represents the error when the value of M and N were set to the corresponding value in

the legend.

The first item of note in the absolute error subplot is that the error profile when M and N equal 35 is distinct from the other trajectories. The other trajectories all remained at approximately machine zero for the same amount of time before following the same error profile as it increased towards the end of the trajectory. It is clear that there is no additional accuracy gained by increasing the value of M more than the 60. This relationship holds for the other error profiles that are not included in the figure to improve readability of the plot.

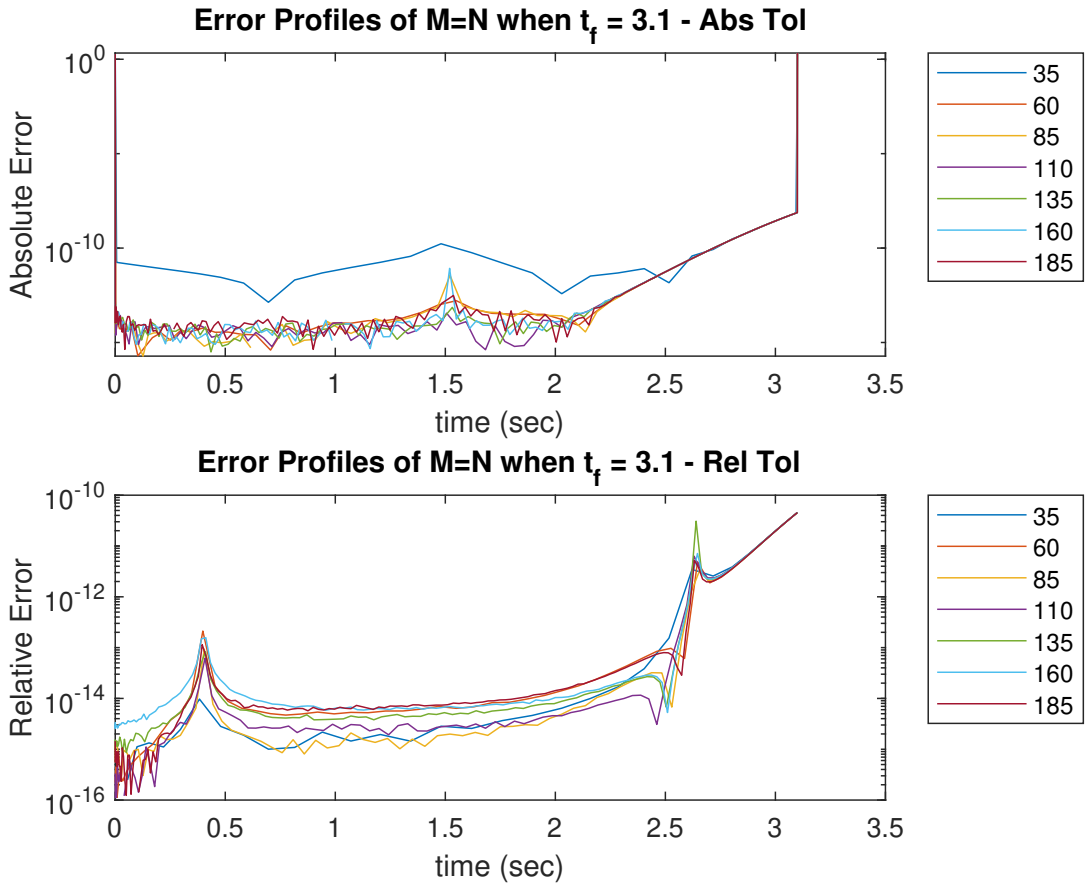

Figure 4.3: Error Profile of Duffing Oscillator With Variations in Number of Sample Points

4.2.2 Iterations to Converge

Figure 4.4 shows the number of iterations necessary to converge using Absolute and Relative Error Analysis. When the value of M is less than 50, the Absolute Error Analysis is not able to converge for all trajectories. As the final time value is decreased, the number of points necessary to converge also decreases. For Relative Error Analysis, the algorithm does report conversion for $M = 10$ although it requires more iterations to converge than when the value of M is larger.

Using both means of evaluating convergence, and for each final time period, there is a clear plateau in the number of iterations necessary to converge on the solution. For relative error, the ramp down to the plateau is clear while this ramp down is not clear for most of the profiles when using absolute error. This means that for the duffing oscillator, the number of iterations is near constant with respect to M so long as the algorithm can converge.

In general, the number of iterations to converge to the solution continues to be smaller for the Absolute Error Analysis. However, this variation in necessary iterations decreases as the final time decreases. As the segment becomes shorter, the two methods converge in similar time.

When the number of iterations is equal to zero in figure 4.4 MCPI was not able to converge to a solution.

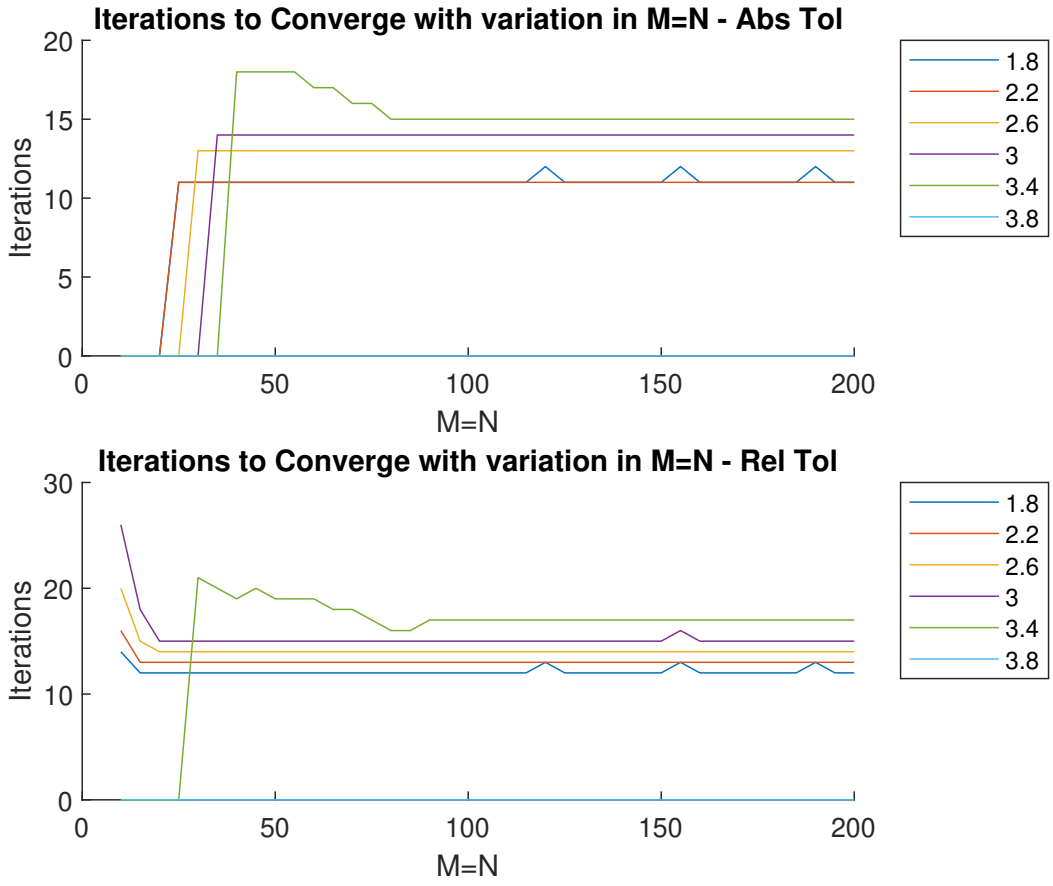

Figure 4.4: Iterations Necessary to Converge with Variation in Sample Points of Duffing Oscillator

4.2.3 Maximum Final Time

The maximum final time when using Relative Error Analysis is near constant when changes in M are made, as shown in Figure 4.5. The change in maximum final time from a value of $M = 10$ to $M = 200$ is approximately 0.25 seconds. However, as seen in figure 4.4, there is a considerable decrease in the number of iterations when M is varied from 10 to approximately 20.

When $M \leq 30$ Absolute Error Analysis is not able to converge, meaning that the converged solution according to Relative Error Analysis is not representative of the true system. The two measures converge on a maximum final time at approximately $M = 40$ with a maximum final time of 3.4 seconds.

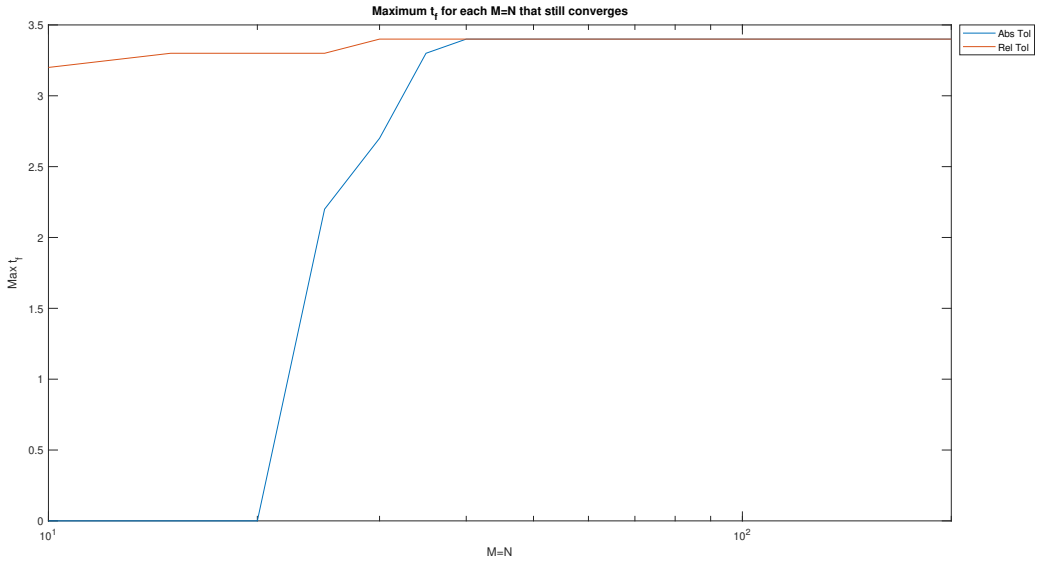

Figure 4.5: Maximum Convergence Window of Duffing Oscillator with Variation in Sample Points

For maximizing the final time, the most efficient value of M is 40 points. This is with the plateau of where increasing the value of M does not change the number of iterations necessary to converge. Thus, the value of M when solving the Duffing Oscillator with $\epsilon = 0.9$ is best set to 40.

4.3 Variation in Error Tolerance

The value for which MCPI reports the series as converged is defined as the error tolerance. As the value of error tolerance decreases, the algorithm is expected to require more iterations to converge to a solution. By increasing the tolerance, the algorithm is able to report a converged solution in fewer iterations. The decrease in iterations corresponds with a less accurate solution to the problem. The different values of error tolerance are expected to have a direct relationship with the error profiles and iterations necessary to converge on a solution.

The value of error tolerance is varied from $1 - e1$ to $1e - 13$ in 13 logarithmic steps. All other variables in MCPI and the duffing equation are set to their base values defined by table 4.1.

4.3.1 Error Profile

Figure 4.6 shows the error profiles for each of the given error tolerances. As noted with variation in the final time value, the Absolute Error Analysis keeps the error at approximately machine zero for as long as necessary before the error profile slowly climbs to the required tolerance level. The first approximately 0.25 seconds is nearly the exact same for all 13 error profiles. This means that as the the algorithm does each iteration, it is pairing the next several nodes to machine zero. Then the next few nodes in the next iteration and so on.

The relative error is fairly constant within the trajectories as can be seen in figure 4.7. There is a significant plateau when using Relative Error Analysis in which the maximum error goes from greater than $1e-1$ to $1e-10$ in a single iteration. This is why there only appears to be 4 error profiles in the relative error subplot of figure 4.6.

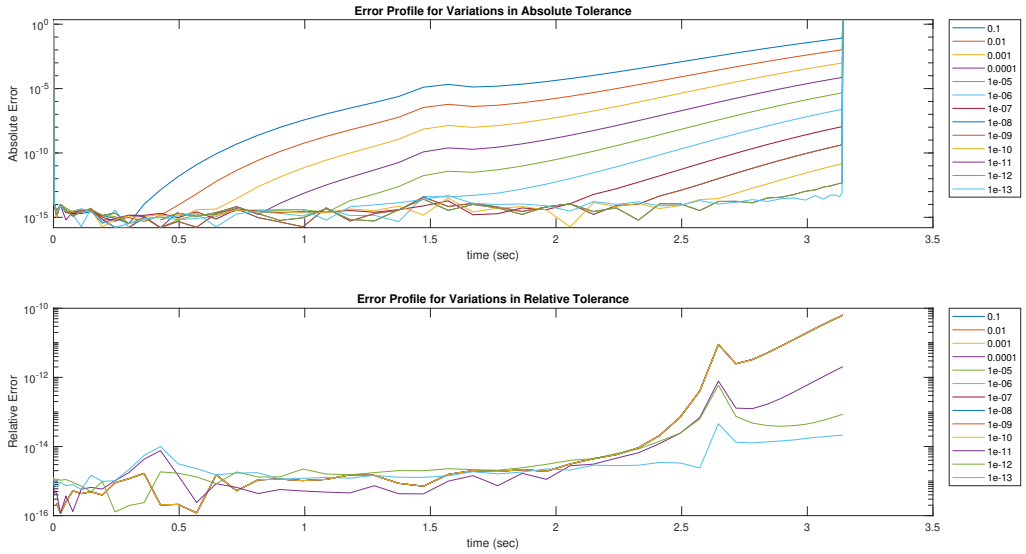

Figure 4.6: Error Profile of Duffing Oscillator With Variations in Acceptable Tolerance

4.3.2 Iterations to Converge

Figure 4.7 shows the number of iterations necessary to converge with the varied values of acceptable error tolerance. The plateau of iterations necessary to converge on the solution when using relative tolerance while the absolute tolerance converges in fewer iterations is interesting. It may mean that the states are oscillating between iterations. After inspection of the algorithm, it was confirmed that the number of iterations necessary to converge with variation in tolerance did plateau at 16 iterations for any tolerance value between 1 and $1e-10$. This is because between iteration 8 and iteration 15, there was a singular point that was fluctuating around zero, leading to consecutive iterations where the maximum error was 1. This plateau is not likely to remain for all other values of M or final time.

The variation in iterations necessary using absolute error is a nearly linear plot when the x axis is logarithmic. At all times this line stay below the number of iterations necessary to converge using relative error. As expected, as the value of acceptable tolerance increases, the number of iterations necessary to reach said tolerance decreases.

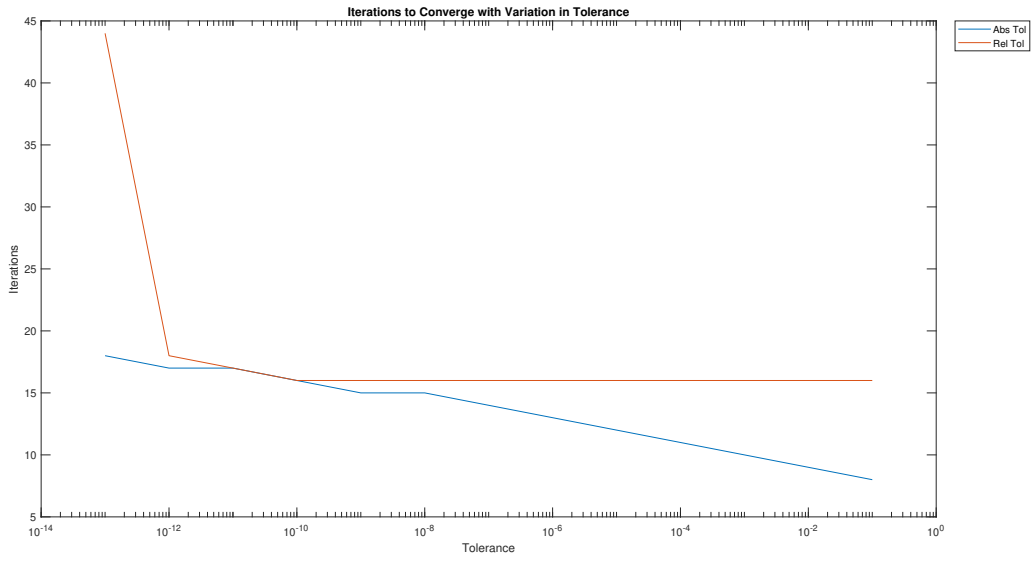

Figure 4.7: Iterations Necessary to Converge with Variation in Acceptable Tolerance of Duffing Oscillator

4.4 Variation in Oscillation Non-Linearity

The ability for MCPI to converge on a solution depends on the equation itself along with the variation in time span and MCPI values. By varying the value of ϵ, MCPI's ability to converge with near linear oscillations can be analyzed. It is expected that as the value of ϵ approaches machine zero, the algorithm will be able to converge in fewer iterations.

The value of ϵ was varied according to table 4.2. The value of $1e-15$ was used to represent machine zero. The value was not set exactly to zero for plotting purposes. The value of the final time was also varied from 2 to 25 seconds at 0.25 second increments to evaluate the maximum time window in which MCPI can converge.

Table 4.2: Variations in ϵ

ϵ_{var}	0.9	0.5	0.1	0.05	0.01	0.005	0.001	0.0005	0.0001	$1e-5$	$1e-8$	$1e-15$

Figure 4.8 shows the phase portrait for each of the variations in ϵ. For values of less than approximately $1e-2$, there is little difference in the visual appearance of the phase portrait. However, there may be a difference in how easily MCPI is able to converge.

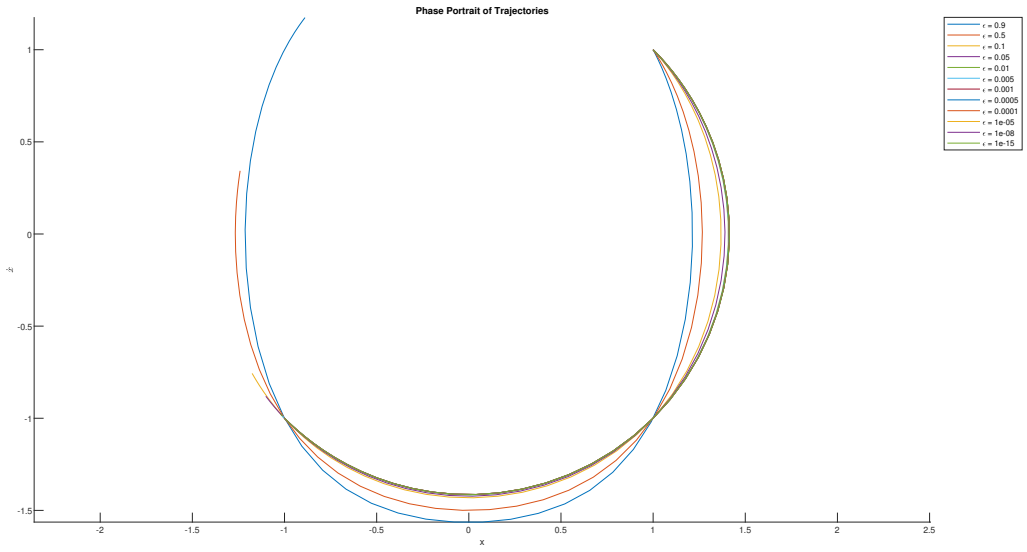

Figure 4.8: Phase Portrait of Duffing Oscillator for π seconds with variations in ϵ

4.4.1 Error Profile

Figure 4.9 shows the absolute and relative error for the converged solutions when the final time value is set to 6 seconds. As with the variation in M, there is not a large amount of variation in how the error profile looks even with significantly different value for ϵ. For absolute error, the error profiles each remain at machine zero for approximately the same amount of time and increase nearly identically.

The relative error profiles have a unique situation in which almost all values of ϵ have an error spike at approximately 4 seconds. This is again a point in which the velocity passes through 0 and the node point is very close to the 0. This leads to spikes in the relative error values.

In figure 4.9, when the error is calculated as less than machine zero, it is saved as $1e-20$.

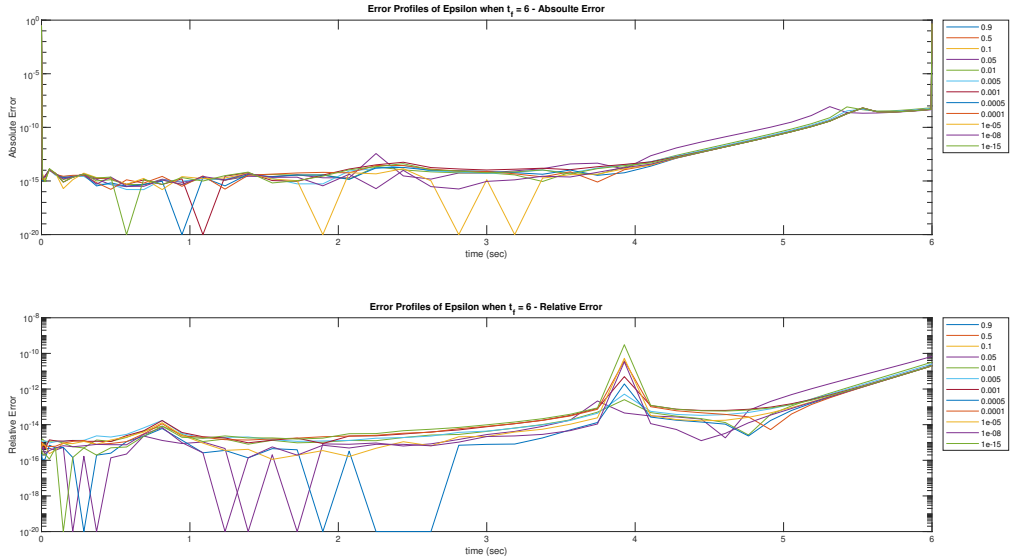

Figure 4.9: Error Profile of Duffing Oscillator With Variations in Non-linearity

4.4.2 Iterations to Converge

Figure 4.10 shows the number of iterations necessary to converge. It is important to note that when the iteration's value goes to zero, that means that the algorithm was not able to converge within the maximum iterations. Refer to table 4.1 for the maximum iterations and table 4.2 for the values of ϵ which are tested.

The relationship of Absolute Error Analysis requiring fewer iterations than Relative Error Analysis holds true as the value of ϵ is held. As the value of ϵ decreases, the ability for the problem to converge in both error analyses increases. The Relative Error Analysis requires more iterations to converge which for the lowest values of ϵ the number of iterations necessary surpasses the maximum iterations allowable.

The value of ϵ has little effect on the number of iterations necessary to converge so long as the algorithm is able to converge. This relationship holds true for both absolute and relative error. There is an outlier where the final time is 14 seconds and ϵ is equal to $1e-8$ where it has a lower number of iterations necessary than when ϵ is equal to $1e-15$ which should converge in the same

or fewer iterations. This specific example is because when the value of ϵ is so close to machine zero, there are nodes that end up passing closer or farther from zero. There are repeated iterations where the maximum value is reported as zero due to the fluctuation about zero.

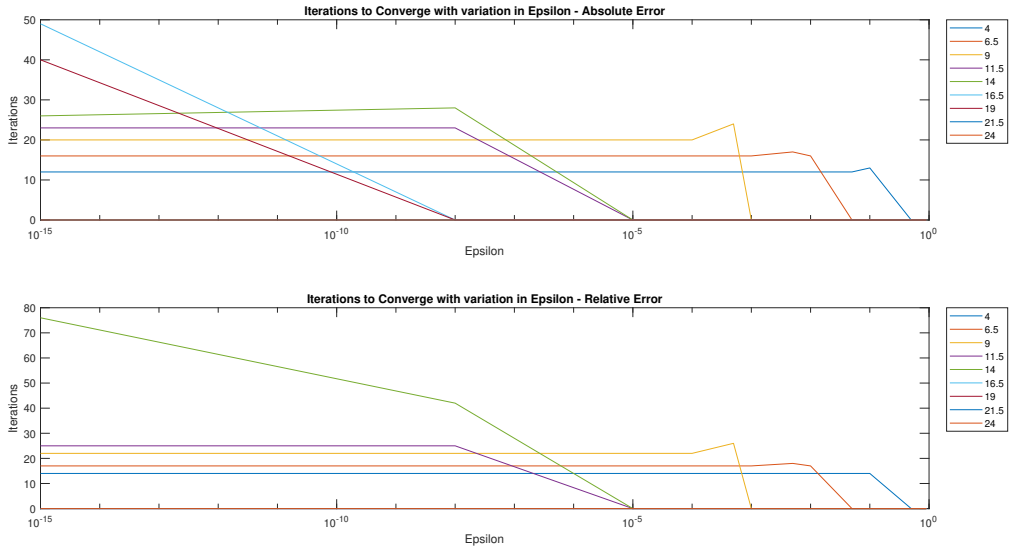

Figure 4.10: Iterations Necessary to Converge with Variation in Non-linearity of Duffing Oscillator

4.4.3 Maximum Final Time

Figure 4.11 shows the maximum final time at each value of ϵ. As the value of ϵ changes from near machine zero to 0.9, the maximum final time does not vary significantly between the two means of evaluating convergence. For every value of ϵ other than $1e-15$, the two methods had the same maximum time span. The Relative Error Analysis was not able to converge for the larger values of final time due to the maximum iterations constraint. Because Absolute Error Analysis is able to converge in fewer iterations, it was not constrained by the maximum iterations for these values and was able to converge.

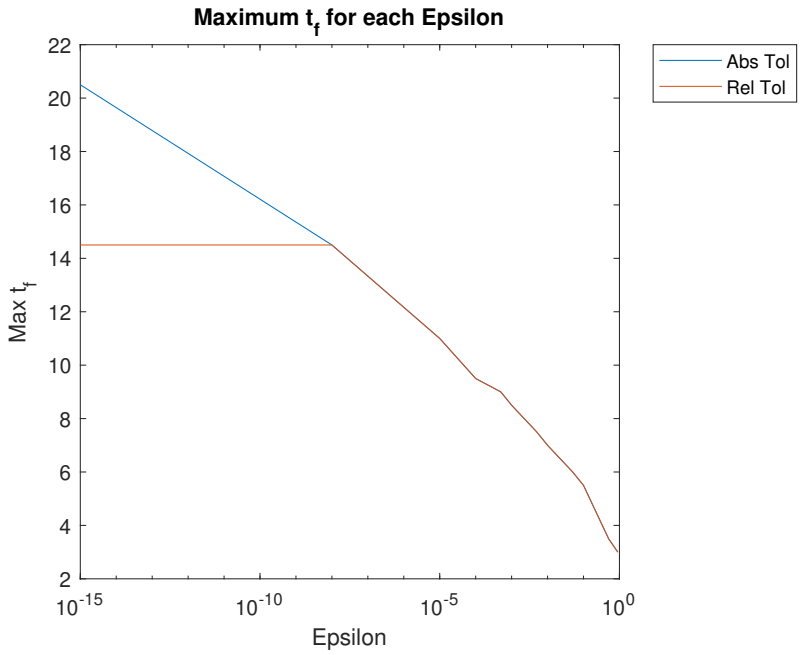

Figure 4.11: Maximum Convergence Window of Duffing Oscillator with Variation in Non-linearity

5. APPLICATIONS TO ASTRODYNAMICS

Numerical integration methods are commonly used in astrodynamics due to the inability to explicitly integrate the perturbed fundamental equation for orbital motion as listed as equation 2.3. This perturbation can be various things, using a more accurate gravitational field, spacecraft drag, solar radiation, third body effects, thrust, etc. The use of a numerical integration scheme in orbital dynamics requires a reliable convergence pattern and accurate results. This chapter will address how variations in orbital parameters and perturbations affect the ability for MCPI to converge.

Three orbits will be analyzed in the chapter. Their initial positions are listed in table 5.1 as orbital elements. The Super GEO orbit is a slow moving circular orbit with no inclination where the effects of perturbations are expected to be small. The LEO orbit is a near earth orbit where the effects of J2 and Drag are expected to have a considerable effect. The Eccentric Orbit was selected to see the effect of adding eccentricity and inclination to the orbit. These orbits can be seen in figure 5.1.

Each of these orbits will be analyzed throughout the chapter with variations in final time, M, acceptable tolerance and with the addition of J2 and drag perturbations. The base values for each variable is given in table 5.2. Also listed are the values used to scale the distance and time units (DU and TU) and the gravitational parameter used.

Table 5.1: Definition of The Three Orbits Used in Convergence Analysis

	a (km)	e	i (deg)	Ω (deg)	ω (deg)	θ (deg)
Super GEO	100,000	0	0	0	0	0
LEO	8,000	0	0	0	0	0
Eccentric Orbit	40,000	0.3	15	15	60	0

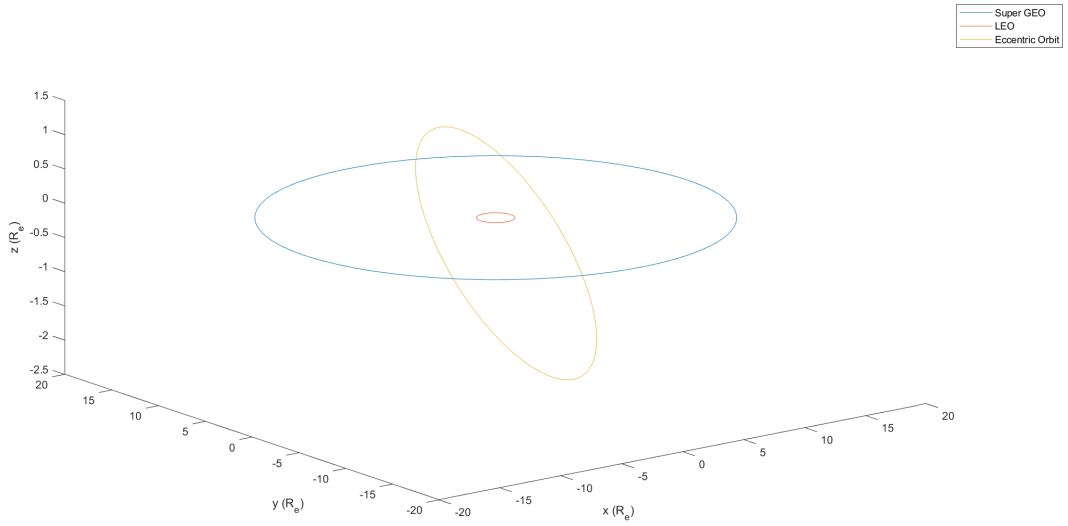

Figure 5.1: Visualization of the Three Orbits Used in This Chapter

Table 5.2: Base Variable Values for Astrodynamics Application

Variable	Base Value
t_f	1 Orbital Period
M	50
N	50
Tol	$1e-8$
$Iter_{Max}$	100
DU	6,378.135 km
TU	806.8 s
μ	398,600.4415 km^3/s^2

The initial analysis of the trajectories will be done with unperturbed motion. The analysis will be done primarily in Cartesian Coordinates and in both first and second order MCPI. The ability to converge using Cartesian Coordinates compared to Orbital Elements is discussed when a

perturbation is introduced. The preferred order of MCPI is analyzed in the first section.

5.1 Variation in Final Time

To analyze the ability of MCPI to converge for each orbit, the length of the MCPI segment was varied from 10% of an orbit to five orbits increasing in increments of 25% of an orbit. However, due to complications in the ability for MCPI to converge, the value of $zeta$ in equation 3.2, is changed to be equal to $1e-3$ instead of the base value of 0. This change will be explained more profoundly in subsection 5.1.2. All other variables are set to their base value.

Using Cartesian Coordinates, MCPI can be used with first or second order equations. To understand which order equation to use, an initial analysis was done between the two orders. This resulted in separate figures for each orbit using each order of MCPI and error analysis method.

5.1.1 Error Profile

By comparing figures 5.2 and 5.3, the difference in how the different orders of MCPI apply to each orbit and each means of evaluating convergence. Upon first inspection of the figures, it appears that second order MCPI has more consistent error profiles between the different time windows. However for most most of the subplots in figure 5.3, the subplot is not zoomed in as much as the corresponding subplot in 5.2. The reason for this different in appearance is that the first and last nodes have the increased inaccuracy in second order MCPI.

The second order MCPI does not converge as easily as the first order MCPI. This is best seen comparing the number of converged trajectories in the Relative Error Analysis error profiles of the Super GEO and LEO trajectories.

It is of note that MCPI appears to handle the SGEO and LEO trajectories very similarly in first order MCPI. Whereas the eccentric orbit is more difficult to converge on a solution. This difficulty in converging is especially unique with respect to the Absolute Error Analysis of the eccentric orbit. Using absolute error in first and second order MCPI has difficulty reaching convergence.

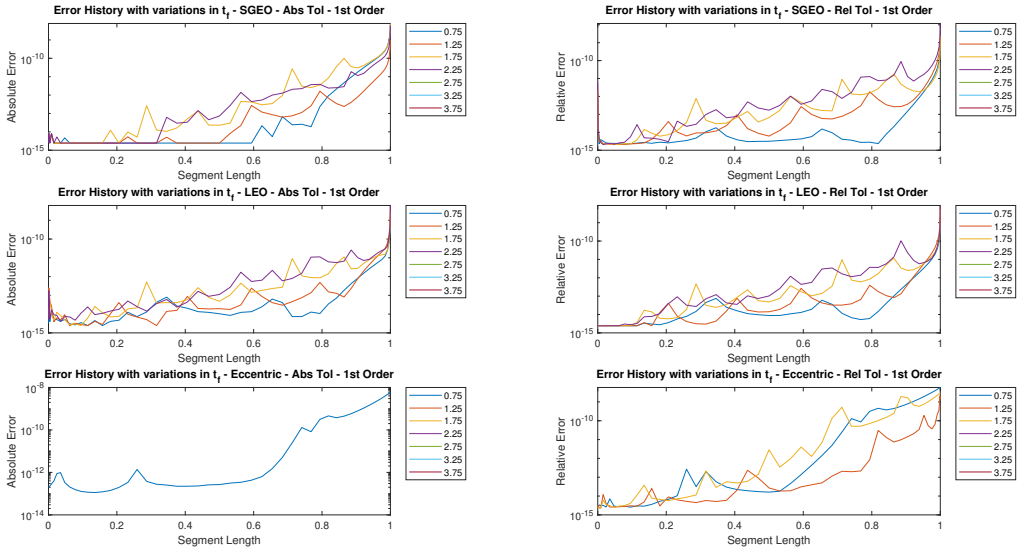

Figure 5.2: Error Profile of Each Orbit Using First Order MCPI with Variation in Final Segment Time

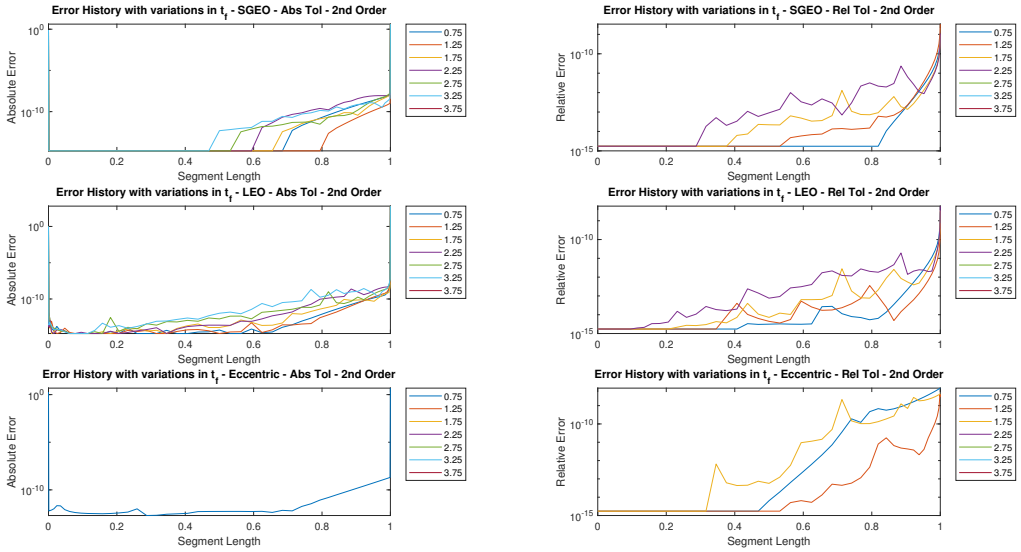

Figure 5.3: Error Profile of Each Orbit Using Second Order MCPI with Variation in Final Segment Time

5.1.2 Iterations to Converge

MCPI has a difficultly converging to a solution when the number of revolutions is a multiple of 0.5 orbital periods and this difficulty increases when greater than one orbit when the value of ζ is equal to zero. The original analysis of the system with variation in final time had a step size of 25% of an orbital period and a ζ value of zero. As seen in figure 5.4, this leads to no converged solution for all trajectories when using Absolute Error Analysis for segments with a length of 1.5, 2, 2.5, 3, and 3.5 Orbits. The algorithm is able to converge at the time values directly before and after these segments for most of the trajectories when the time step was decreased to a value of 5% of an orbit. .

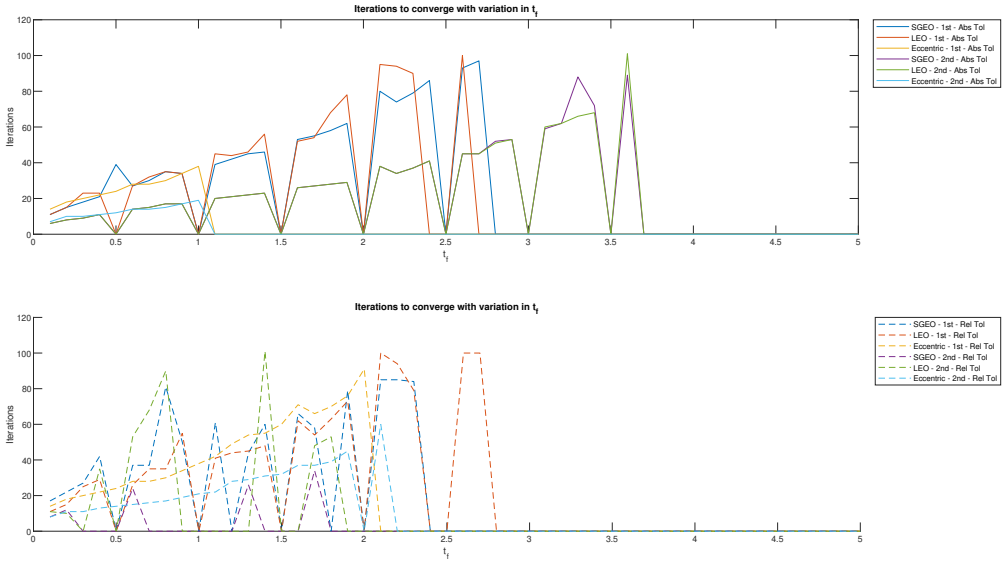

Figure 5.4: Iterations Necessary to Converge with First Order and Second Order MCPI for Each Orbit when $\zeta = 0$

The reason that the algorithm does poorly when the orbit uses multiples of 0.5 orbital periods, is because the trajectories happen to have points that are extremely close to zero when no perturbation is introduced. For example, when the orbit is of length 2 and a half orbital periods. Figure 5.5

shows the error profile and states of the LEO trajectory when the final time is equal to 2.5 orbital periods. When the time is equal to 1.25 orbital periods, the values of x and v_y are both equal to values of approximately $1e-14$. This leads to the error in y and v_x to be equal to values in the order of magnitude of $1e0$. The true value of the solution is exactly zero for this trajectory, leading to near impossibility of the system evaluating the value of each state as zero.

The code was equipped with a fail safe with this particular issue in mind. When the value of a state and its dynamics error is less than a given tolerance, the algorithm will specify that the error at that node will be defined as equal to zero. However, this did not account for potential error in corresponding states. For example, the error in x will lead to this same error in v_x. This error can be avoided by changing the tolerance which triggers this event and by insuring that the algorithm accounts for the corresponding states/velocities.

The additional adjustment to the algorithm is the inclusion of the ζ variable in the denominator when calculating the error. By setting ζ equal to $1e-3$, the fact that $f(x)$ is nearly exactly zero, is accounted for. There is no longer a division by $1e-14$ causing an explosion of the error evaluation. The numerator still requires a value that is sufficiently small but it is no longer dominated by the denominator when the denominator value is extremely small.

This small value of ζ does not largely contribute to the evaluation of error when the value of $f(x)$ is greater than ζ and therefore is does not change the convergence for the majority of the trajectory.

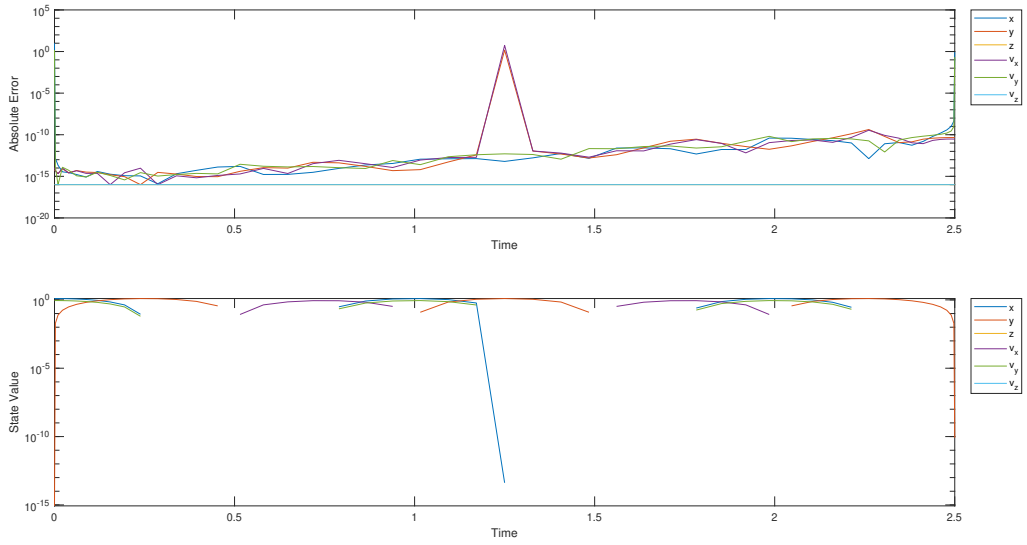

Figure 5.5: Error Profile and States of the LEO Trajectory in Two Body Motion when the Final Time is 2.5 Orbital Periods when $zeta = 0$

By changing the value of ζ to a nonzero number, the evaluation of error is consistent and does no error for these near zero values. This eliminates the potential difficulty causing the system not to converge. Figure 5.6 shows the new number of iterations necessary to converge to a solution with the varied value of ζ. When using Absolute Error Analysis, the algorithm is able to converge at every step size for final time. This improved behavior continues for first and second order MCPI.

Upon inspecting figure 5.6, it is noted that for each trajectory the second order MCPI is able to converge for less iterations than first order MCPI. For the eccentric orbit, each order of MCPI has the same maximum final time for convergence. For the circular orbits, LEO and SGEO, the second order MCPI is able to converge for final time of approximately one orbital period more. For these two trajectories, the difference in the maximum final time appears to be due to a set maximum number of iterations. The first order MCPI would likely be able to converge for those values of 3.5 orbits if it was allowed to perform more than 150 iterations.

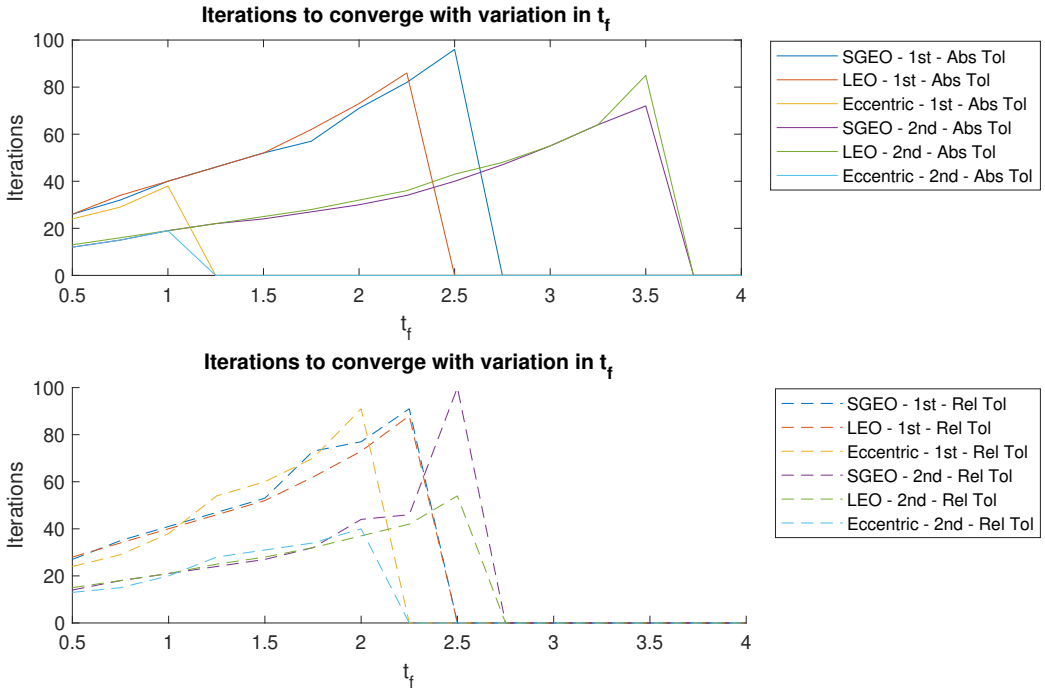

Figure 5.6: Iterations Necessary to Converge with First Order and Second Order MCPI for Each Orbit when $\zeta = 1e-3$

5.2 Variation in M and N

As noted in Chapter 4, the variation in the number of sampled points used in MCPI is essential in the understanding of MCPI convergence. Each of the trajectories is analyzed in various values of M, which N being equivalent to the value of M.

The variation in M is from 20 to 200 with a step size of 10. The final time is varied from 0.25 to 15 orbital periods with a step size of 25% of an orbital period. This step size was selected because it matches the analysis from the prior subsection. This variation in final time is necessary to find the maximum final time for each value of M for each orbit.

Because the second order MCPI was noted in the former subsection to have better convergence as defined by number of iterations and maximum final time, all remaining analysis in this section will be done with second order MCPI.

5.2.1 Error Profile

Figure 5.7 shows the error profile for each trajectory. The final time used in each orbit was 1.05 orbital periods. It can be noted that there is an error spike in each error profile at approximately one orbital period. For SGEO and LEO, these spikes are at almost exactly one orbital period.

In general, it is not easy to note the error profiles for the Absolute Error Analysis subplots due to the inaccuracy spike at the first and last nodes. Although even with this inaccuracy, it is clear that for the SGEO and LEO trajectories, there is little variation in the absolute value error profiles regardless of M with the exception of M being equal to 30.

For each trajectory, the relative solution had difficulty converging, as noted with the SGEO trajectory, the Absolute Error Analysis was able to converge for most of the values of M while the Relative Error Analysis was only able to converge for one solution.

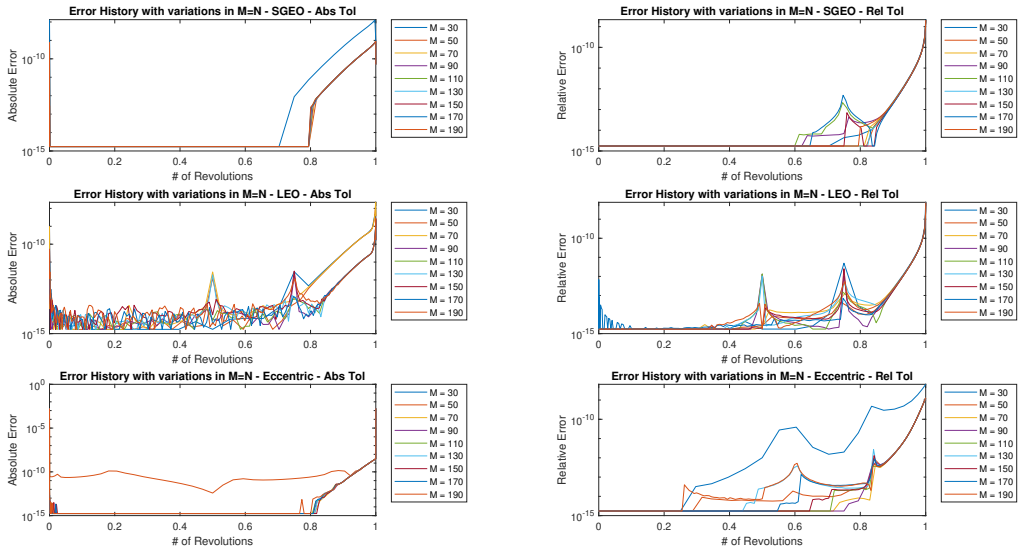

Figure 5.7: Error Profiles of Each Orbit with Variation in the Number of Samples

5.2.2 Iterations to Converge

As noted in the previous section, the use of Absolute Error Analysis led to more converged solutions than the use of Relative Error Analysis. This is best noted in figure 5.8 where the subplots associated with absolute error have clear visible lines with a correlation between M and the number of iterations necessary. The behavior of the SGEO and LEO trajectories end up being essentially identical. The same behavior as noted in prior sections is again visible, where there is a clear behavior that as the value of M is increased there is a point of least returns where the number of iterations no longer decreases by increasing the value of M. For the trajectory of length 1.25 orbital periods, this plateau begins to take place even before the M value of 20 that was used as the initial value.

The variation in the relative error analysis convergence continues as noted previously. The trajectories with the smallest final time did allow for some convergence but for values greater than one orbital period, the SGEO and LEO trajectories were not able to converge consistently. The eccentric orbit trajectory was able to converge for the smaller final times consistently. The subplot for the eccentric orbit Relative Error Analysis in figure 5.8 actually shows that for some of these orbits the main restriction in the ability to converge is the number of iterations. This subplot also had the unique behavior of increasing the number of iterations as the value of M was increased. This unique behavior is due to the the fact that the converged solution does not accurately represent the dynamics. The solution ends up going towards infinity in the x direction, so while it converged on a solution, this solution is not reliable.

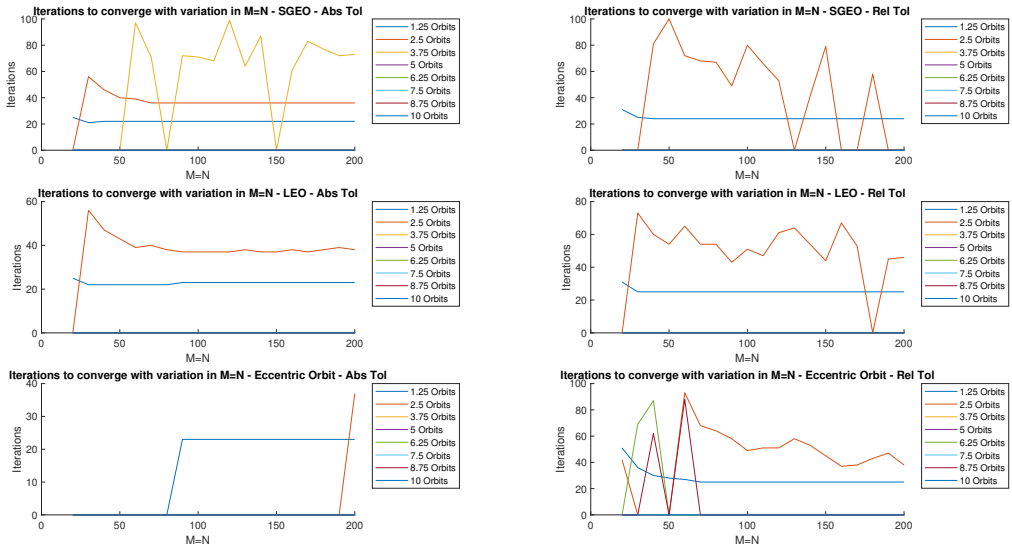

Figure 5.8: Iterations Necessary to Converge with Variation in Number of Samples for Each Orbit

5.2.3 Maximum Final Time

Figure 5.9 shows the maximum final time of each trajectory using MCPI. It is expected that as the value of M increased that the maximum final time will be able to increase. This behavior holds primarily true for most of the lines. However, when the eccentric orbit is evaluated with Relative Error Analysis, the maximum final time increases initially before reaching a peak, making a small step up, and then finally making a sizeable step down to a consistent plateau. This is due to the false convergences reported by the algorithm.

The behavior of the SGEO and LEO trajectories for each error analysis is very similar. For Absolute Error Analysis, there are only minor fluctuations in the two's maximum final time. For the use of Relative Error Analysis, there is more variation but that variation is predominantly around the inability of each to converge. Their converged solutions were consistently similar.

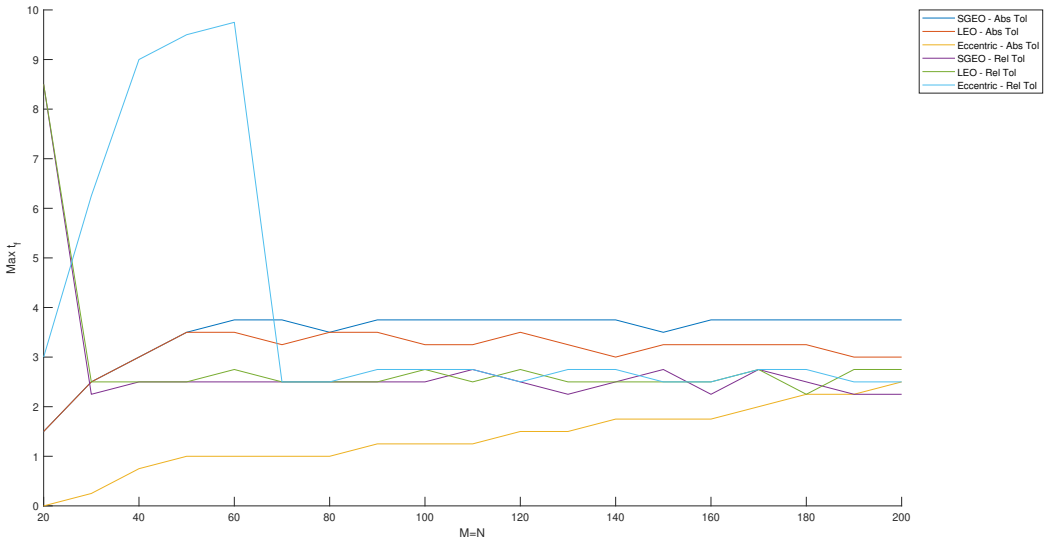

Figure 5.9: Maximum Convergence Window for Each Orbit with Variation in Number of Samples

5.3 Variation in Error Tolerance

As the acceptable tolerance of MCPI is varied, it is expected that the ability to converge will be inversely affected. As the tolerance value increases, the number of iterations necessary to reach a converged solution will decrease. This behavior is expected to remain true for all three orbits and for each means of evaluating convergence.

The value of error tolerance is varied from $1 - e1$ to $1e - 13$ in 13 logarithmic steps. The final time variation was from 25% to 15 orbital periods with a step size of 25% of an orbital period. All other variables in MCPI are set to their base values defined by table 5.2.

5.3.1 Error Profile

The behavior of each of the absolute error profiles in figure 5.10 behave exactly as expected. They each have an error of approximately machine zero at first and with each iteration holds onto that machine zero error for slightly longer until that results of an acceptable tolerance.

As with other variations, there is a difficulty of evaluating convergence for the SGEO and LEO orbits when using Relative Error Analysis. This remains consistent with the difficulty as noted in

prior sections.

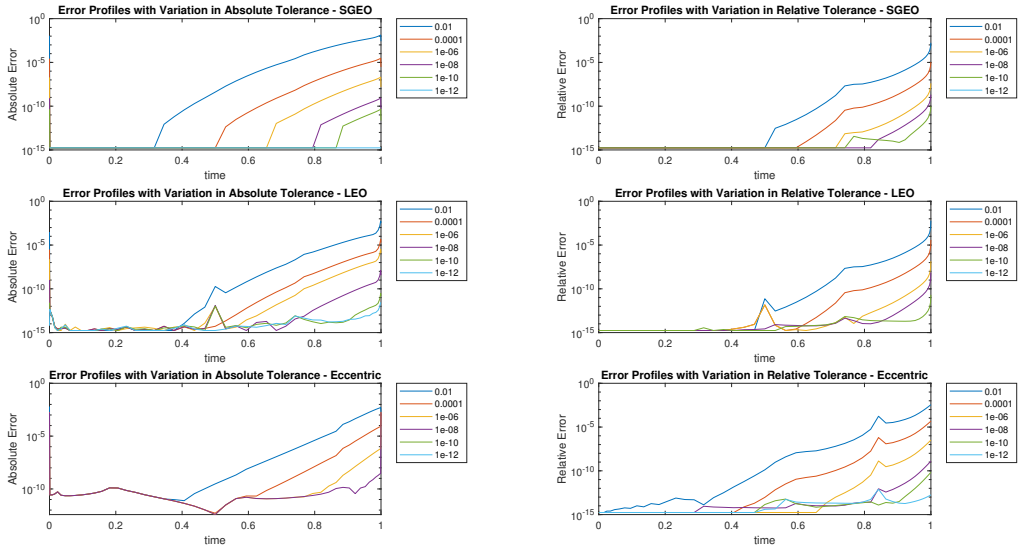

Figure 5.10: Error Profiles of Each Orbit with Variations in Acceptable Tolerance

5.3.2 Iterations to Converge

As previously stated, as the value of acceptable tolerance increases the number of iterations necessary either decreases or remains the same for all trajectories and for all means of evaluating convergence. This behavior is noted in figure 5.11 in all six subplots. When the number of iterations is equal to zero, the algorithm was not able to evaluate convergence.

Plots of note is the eccentric subplots where it is shown that The ability to converge is very limited but not by the number of iterations. None of the converged trajectories needed more than 40 iterations but for trajectories larger than 0.25 orbital periods, convergence is limited.

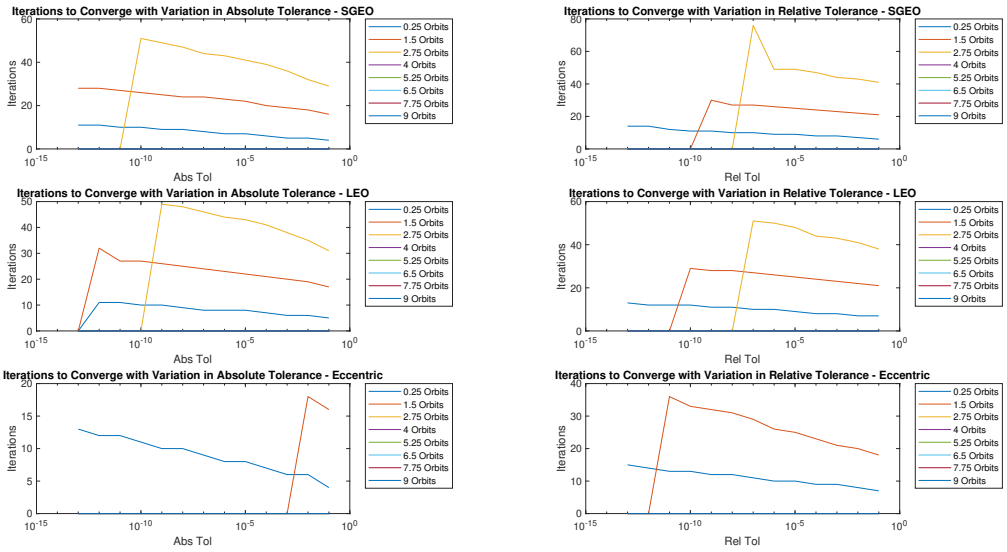

Figure 5.11: Iterations Necessary to Converge with Variations in Acceptable Tolerance for Each Orbit

5.3.3 Maximum Final Time

As noted in the section 5.2, the trajectory that is able to converge for the largest value of final time is the eccentric orbit using relative tolerance. Here the number of iterations is no longer a constraint as the acceptable tolerance is varied as shown in figure 5.12.

The expected behavior of each other trajectories holds true. When the value of the acceptable tolerance decreases, the convergence window shrinks. There are no clear violations of this behavior in any of the six lines. The SGEO trajectory using relative tolerance had the same difficulty in evaluating convergence as previously discussed.

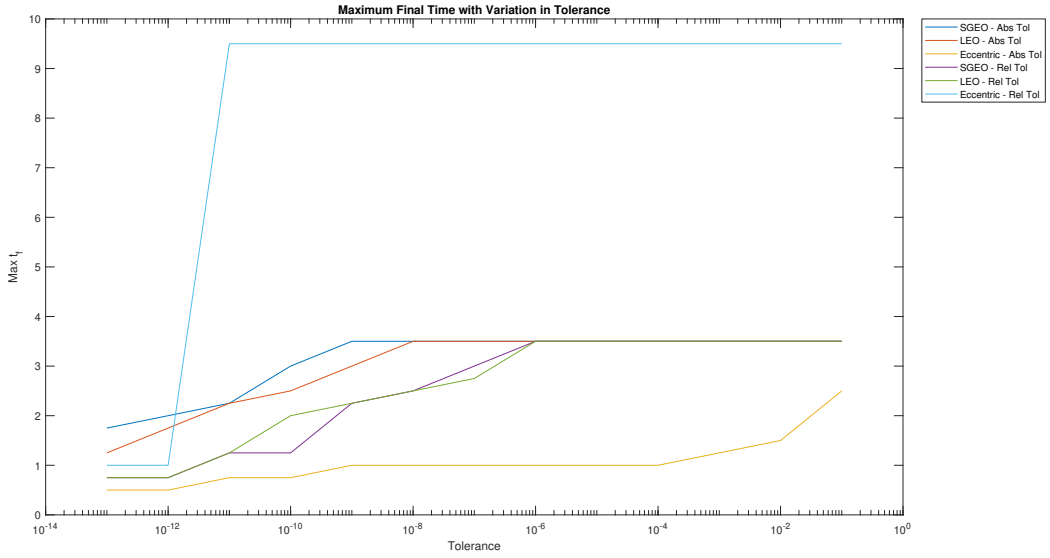

Figure 5.12: Maximum Convergence Window for Each Orbit with Variation in Acceptable Tolerance

5.4 J2 Perturbation

Having gained an understanding of how the unperturbed two-body motion is able to converge using MCPI, the perturbations to the system can be included. The first perturbation to be included is J2 Perturbation. J2 is the perturbation which accounts for the fact that the earth is not a perfect sphere. Due to the oblate shape of the earth, there is a variation in the gravitational field.

Equation 5.1 is the resulting dynamics when J2 is accounted for in Cartesian Coordinates [1]. This is the dynamics that will primarily be used in this section. Second order MCPI will be used and the system will be evaluated using Absolute and Relative Error Analysis. The value of μ is given above and the value of J_2 is equal to $1.082645e-3$ and the Earth's Radius, R_e, is set equal to 6,378.135 km [7]. These values will remain constant throughout the chapter.

$$\begin{aligned}
\ddot{x} &= -\frac{\mu}{r^3}x + \frac{3\mu J_2 R_e^2}{2r^5}\left(\frac{5z^2}{r^2} - 1\right)x \\
\ddot{y} &= -\frac{\mu}{r^3}y + \frac{3\mu J_2 R_e^2}{2r^5}\left(\frac{5z^2}{r^2} - 1\right)y \\
\ddot{z} &= -\frac{\mu}{r^3}z + \frac{3\mu J_2 R_e^2}{2r^5}\left(\frac{5z^2}{r^2} - 3\right)z
\end{aligned} \tag{5.1}$$

The perturbing force of J2 can be accounted for in any coordinate frame. A common frame used in astrodynamics is Orbital Elements. These six elements describe the position and velocity of the orbit within the defined frame with the additional benefit that there is only one "fast" variable. This "fast" variable is the only variable that changes quickly over the course of an orbital period. In fact, for an unperturbed orbit, θ is the only variable that changes and the others are held as constants.

The addition of J2 leads to non-constant values for the other five orbital elements and a perturbation to the change of θ. These changes of elements are noted in equation 5.2. The perturbing effect of J2 is listed in short as the variables p_r, p_s, and p_w. The value for each of these equations is calculated using equation 5.3. [7]

$$\begin{aligned}
\dot{h} &= rp_s \\
\dot{e} &= \tfrac{h}{\mu}\sin\theta p_r + \tfrac{1}{\mu h}((h^2+\mu r)\cos\theta + \mu e r)p_s \\
\dot{i} &= \tfrac{r}{h}\cos(\theta+\omega)p_w \\
\dot{\Omega} &= \tfrac{r\sin(\theta+\omega)}{h\sin i}p_w \\
\dot{\omega} &= -\tfrac{1}{eh}(\tfrac{h^2}{\mu}\cos\theta p_r - (r+\tfrac{h^2}{\mu})\sin\theta p_s) - \tfrac{r\sin(\theta+\omega)}{h\tan i}p_w \\
\dot{\theta} &= \tfrac{h}{r^2} + \tfrac{1}{eh}(\tfrac{h^2}{\mu}\cos\theta p_r - (r+\tfrac{h^2}{\mu})\sin\theta p_s)
\end{aligned} \tag{5.2}$$

$$\begin{aligned}
p_r &= -\tfrac{3\mu J_2 R_e^2}{r^4}(1 - e\sin^2\theta\sin^2(\theta+\omega)) \\
p_s &= -\tfrac{3\mu J_2 R_e^2}{r^4}(\sin 2(\theta+\omega)\sin^2 i) \\
p_w &= -\tfrac{3\mu J_2 R_e^2}{r^4}(\sin 2i\sin(\theta+\omega))
\end{aligned} \tag{5.3}$$

The effect of J2 perturbation is not the dominating force, as noted by the definition of perturbation. Because of this, the effect of J2 is not easily noted in a single orbital period. However, as noted previously, MCPI may not be able to easily converge for a final time of more than one orbital period and is almost assuredly not going to converge for a final time of hundreds of orbital periods. Because of this a variation in the use of MCPI must be applied. For the remainder of this chapter Multi-segment MCPI will be used. Multi-segment MCPI uses a defined time span as a segment and a given number of segments and repeats MCPI for each segment using the final value

of the previous segment as the initial value of the current segment. The time span can be varied to account for the most efficient value for convergence.

For this paper, the number of segments in Multi-segment MCPI is set to 10. The expected time span for each segment is one orbital period. However, this value will be adjusted according to the results of the variation in final time analysis. Because the use of a single time span is to be used for all segments, it is assumed that this value will hold true even as the trajectory is perturbed and the maximum number of segments which can converge is near infinite. For this purpose, the maximum final time with variations in M and Tolerance will not be found.

5.4.1 Variation in Final Time

The variation in final time when analyzing J2 with Multi-segmented MCPI is from 50% of an orbital period to two orbital periods with a step size of 25% of an orbital period. The avoidance of the multiples of 50% of the orbital period is no longer relevant due to the perturbed states not occurring at exactly zero. The value of ζ is set to zero.

5.4.1.1 Error Profile

Figure 5.13 shows the error profile of each of the three trajectories using the two convergence analysis tools. It is clear to see where the error of each segment is added to the error of the previous segments. The error profile of each segment is uniform for each trajectory with little variation from one segment to the next. This shows that the use of Multi-segmented MCPI does not diminish in power as more segments are added.

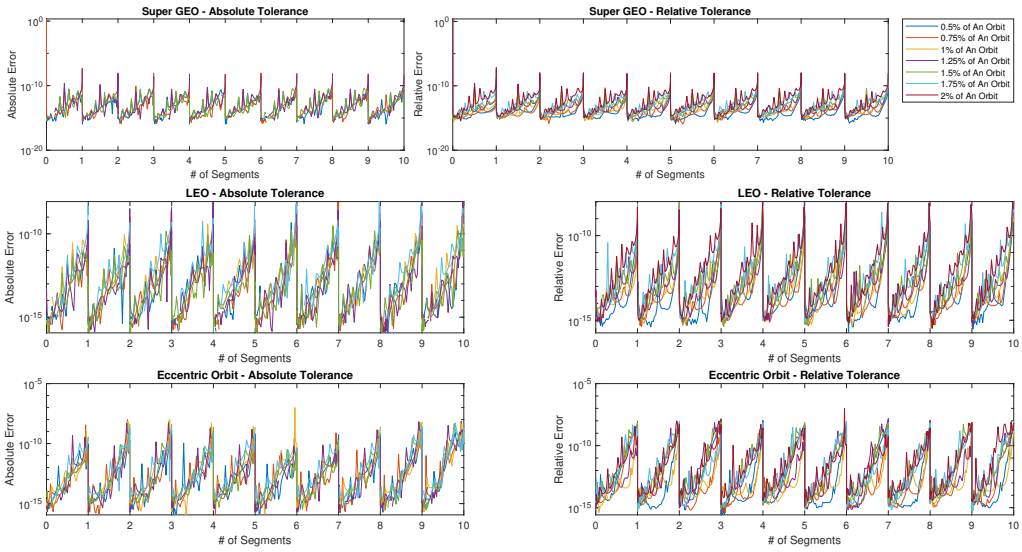

Figure 5.13: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Segment Lengths

5.4.1.2 Iterations to Converge

Figure 5.14 shows the number of iterations necessary to converge for each segment in Multi-segment MCPI. Due to having a fixed time span for each segment, the number of iterations when the segment did not converge is set to the maximum number of iterations and not defined as zero. This is best evident in the SGEO and Eccentric Orbit subplots where the number of iterations for the larger segments is equal to 100.

For the shorter segments, the Multi-segmented MCPI is able to converge on a solution at a consistent rate regardless of the number of revolutions. This consistently is visible in the LEO and SGEO trajectories. The Eccentric Orbit appears to be difficult for MCPI to converge on for segments of more than one orbital period. Due to this and the consistency of the SGEO and LEO orbits, one orbital period will be used as the segment length for the remainder of this segment.

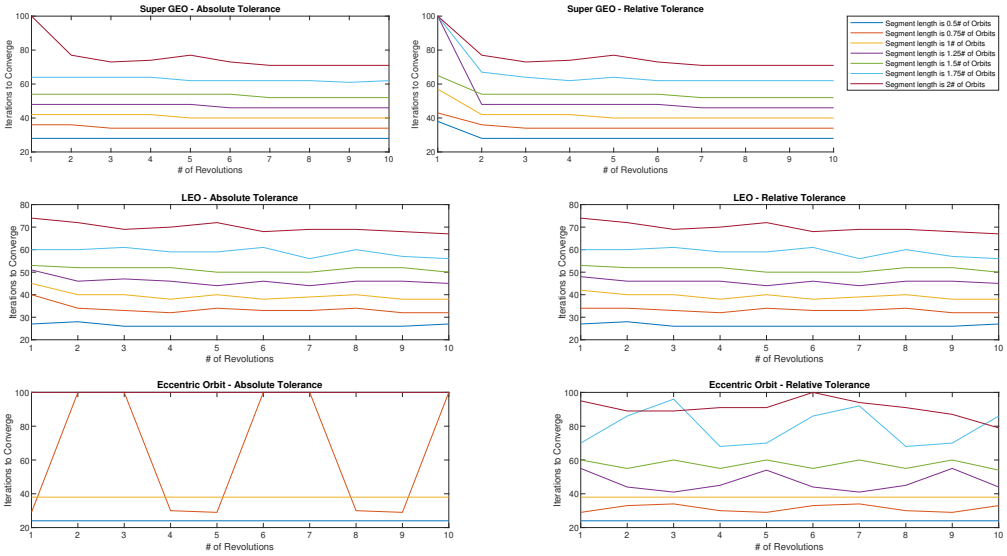

Figure 5.14: Iterations to Converge for Each Orbit over Ten Segments with Varied Segment Lengths

5.4.2 Variation in M and N

As noted in previous sections, the variation in the number of sampled points in MCPI has a critical effect in the ability for MCPI to converge and the number of iterations in which it can converge. The effect of the J2 perturbing force on the Multi-segmented MCPI with variations in M was analyzed for M values of 20 to 200 with a step size of 20. The smaller step size was not necessary due to the analysis done earlier in this chapter when discussing the two-body problem.

5.4.2.1 Error Profile

Upon inspecting figure 5.15, it is visually clear where the value of M was too low to converge for any of the segments. When the LEO trajectory is analyzed with Absolute Error Analysis, it is able to converge for nearly every value of M except when M is equal to 20. This same behavior appears for the Eccentric Orbit where the M values of 20 and 40 are not able to converge but the difference between the values is clear in the error plots.

When using Relative Error Analysis, the value of M is less significant in converging to a

solution. However, the accuracy of said solution is in question because if it would not converge for Absolute Error Analysis, the dynamics of the system are not correct, within the tolerance.

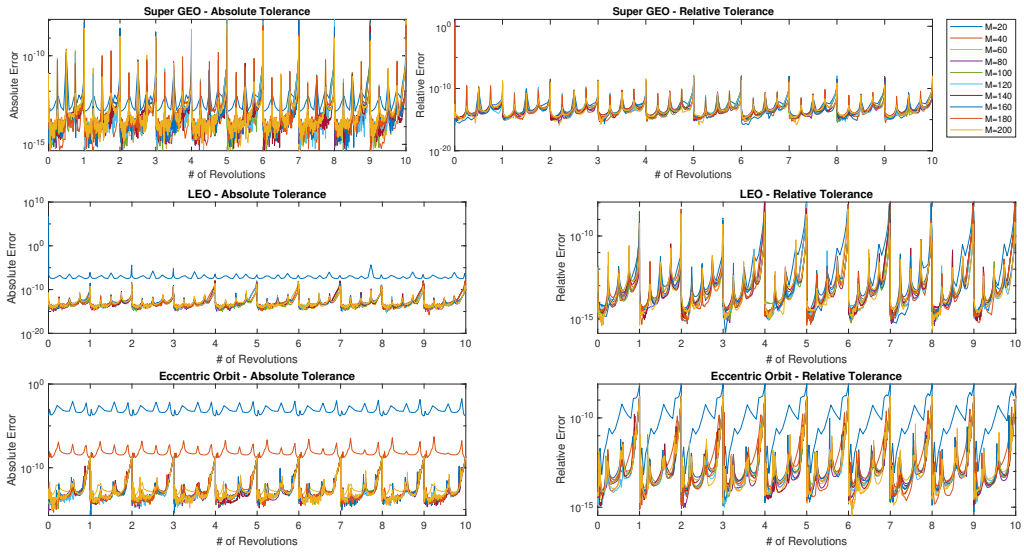

Figure 5.15: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Samples per Segment

5.4.2.2 Iterations to Converge

Figure 5.16 follows the same relationship as figure 5.14 where the non-converged solutions will have 100 iterations as the result. This is clear in the LEO and Eccentric subplots as was noted in the prior subsection. However, this can be deceptive because the SGEO subplot with respect to relative tolerance appears to have a value start at 100 but that value is actually 99, meaning that it was able to converge.

An additional item of note for many of the figures is that the value of M has a plateau of when the change in M changes the value of the number of iterations necessary. For the Eccentric orbit, it would appear that the step size of M is large enough that the number of iterations necessary effectively are sorted into two straight lines. These subplots are not empty.

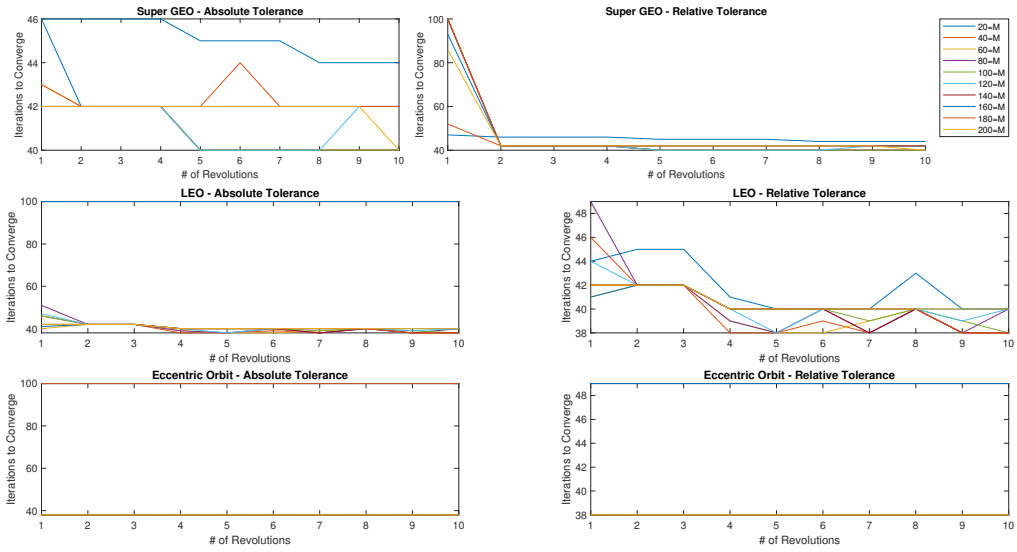

Figure 5.16: Iterations to Converge for Each Orbit over Ten Segments with Varied Samples per Segment

5.4.3 Variation in Error Tolerance

An important analysis into how well Multi-segment MCPI can be used for J2 and other perturbed solutions is how the effect of changing the tolerance of the solution may change the resulting solution. There is the possibility of compounding error driving the solution farther and farther from the true value at the final time. Because of this, it is expected that the error tolerance in Multi-segmented MCPI is required to remain low.

The value of error tolerance is varied from $1 - e1$ to $1e - 13$ in 13 logarithmic steps. All other variables in MCPI are set to their base values defined by table 5.2.

5.4.3.1 Error Profile

As noted in previous sections, there is a clear pattern of behavior for the error profiles as the values of tolerance are changed. This remains constant as the values of tolerance are changed in Multi-segment MCPI as noted by figure 5.17. There is not a clear variation from the pattern as the segment number increases. There is a minor variation from the pattern for the error profile of

the SGEO trajectory using Relative Error Analysis. Here there are several of the trajectories that converged on the same number of iterations, leading to a near equivalent error profile. However, this did not continue for the future segments which no longer converged in the same number of iterations leading to separate solutions.

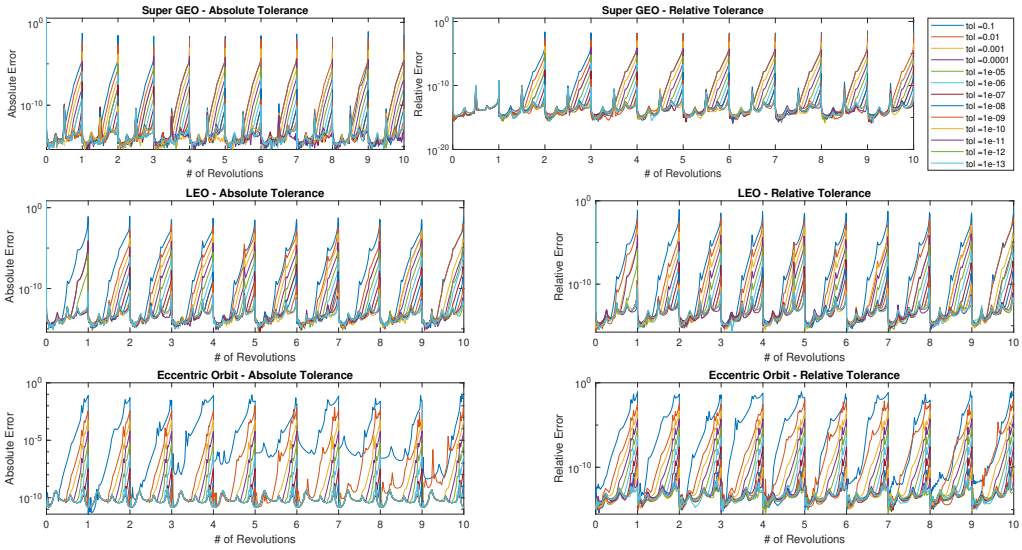

Figure 5.17: Error Profile of Each Orbital Trajectory over Ten Segments with Variations in Accepted Tolerance

The acceptable tolerance is varied and the resulting final states of each segment are varied. The reliability of the solution comes into question when the error tolerance becomes too large. Figure 5.18 shows the values of the final states according to each tolerance value and each error analysis method.

In general, once the error tolerance is less than $1e-6$ the solution is almost exactly the solution of a smaller tolerance like $1e-13$. The compounded error only begins to take effect for values of tolerance greater than $1e-6$. However even for error tolerance of $1e-1$, the overall error is commonly in the second or third significant digit. It is assumed that this would continue to vary more as the number of segments increases.

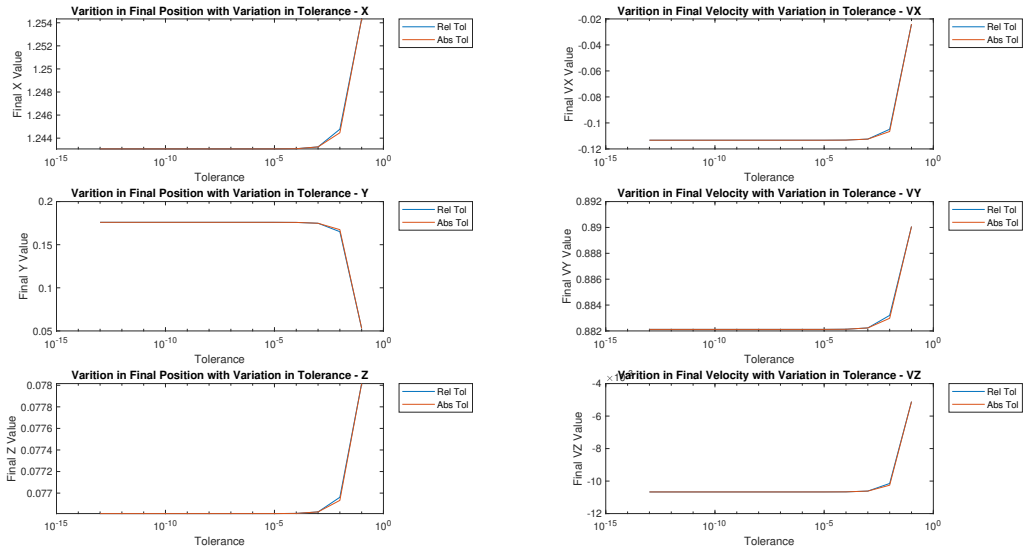

Figure 5.18: Variation in the Final States when Multi-Segment MCPI has Variation in Acceptable Error Tolerance when Solving J2 Perturbed Motion

5.4.3.2 Iterations to Converge

Figure 5.19 shows the variation in iterations necessary to converge. Again for a non-converged solution, the number of iterations is equal to the maximum number of iterations. This is not set to zero, because the 100th iteration solution is used for the next segment.

It is of note for the majority of the trajectories, the number of iterations for each tolerance level remains close to constant or decreases over time. It is worth mentioning that the Eccentric Orbit when using absolute tolerance and a tolerance level of $1e-9$ had a consistent increase in iterations after the 6th segment until it was not able to evaluate convergence any more. This makes sense that as an eccentric orbit becomes slightly more eccentric, the ability to converge behaves almost like a cliff. This behavior is consistent with other discussions of MCPI.

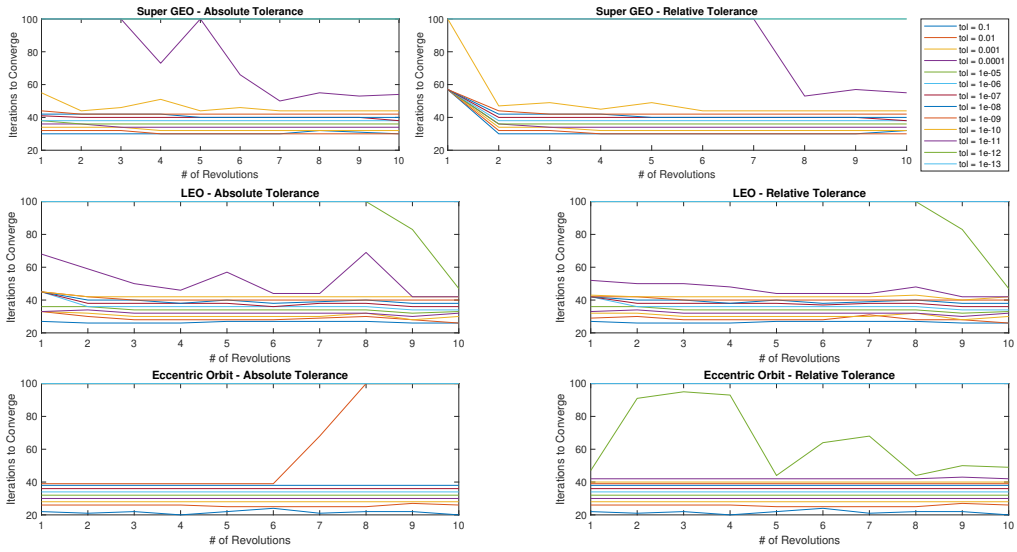

Figure 5.19: Iterations to Converge for Each Orbit over Ten Segments with Variations in Accepted Tolerance

5.4.4 Comparison between Orbital Elements and Cartesian Coordinates

As mentioned previously, the ability to converge with these perturbations has been primarily analyzed as a second order system in Cartesian Coordinates. The use of Orbital Elements would likely lead to improved convergence. This analysis will not result in the change of the analysis from Cartesian Coordinates to the use of Orbital Elements in order to remain consistent.

The benefit of Orbital Elements is the use of five "slow" variables and one "fast" variable. This benefit can be found in other coordinate frames like Modified Equinoctial Elements (MEEs). Because this benefit is expected to be replicated, the use of MEEs will not be analyzed in this thesis. While possible that MEEs have a separate convergence relationship with MCPI, the analysis of this relationship is not in the scope of this thesis.

The comparison between Cartesian Coordinates and Orbital Elements will be done with a variation in M as well to see if there is a notable difference in the change of M with each coordinate system. The value of M is varied from 20 to 200 at steps of 20.

5.4.4.1 *Error Profile*

To simplify the discussion of the coordinate systems, only the LEO trajectory is included in figure 5.20. The behavior of the LEO error profiles was consistent with the behavior of the other two trajectories.

It is noted that for the LEO trajectory, the value of M equal to 20 was too low for both coordinate systems when using Absolute Error Analysis but not for the use of Relative Error Analysis. The average error for the non-converged solutions is of the same magnitude regardless of the coordinate system.

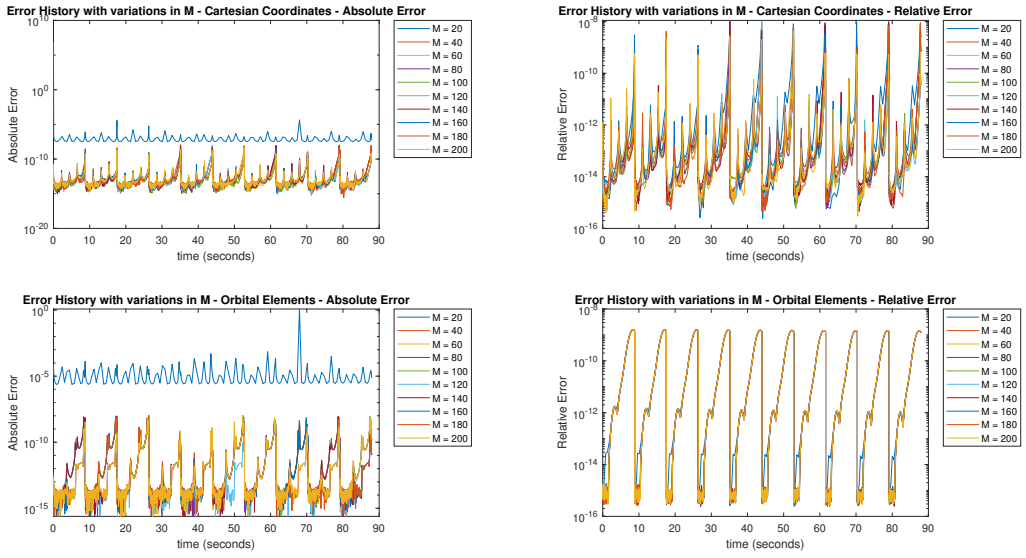

Figure 5.20: Error Profile of Each Orbital Trajectory over Ten Segments in Cartesian and Orbital Elements

5.4.4.2 *Iterations to Converge*

The significant benefit of the use of Orbital Elements is the reduction of the number of iterations necessary to converge to a solution. Using either Absolute or Relative Error Analysis, the converged solution was found in approximately ten iterations regardless of the value of M. The

value of M changed the number of iterations necessary, although not by a large margin when using Cartesian Coordinates. The number of iterations necessary for Cartesian Coordinates was consistently between 40 and 50 iterations for the converged solutions.

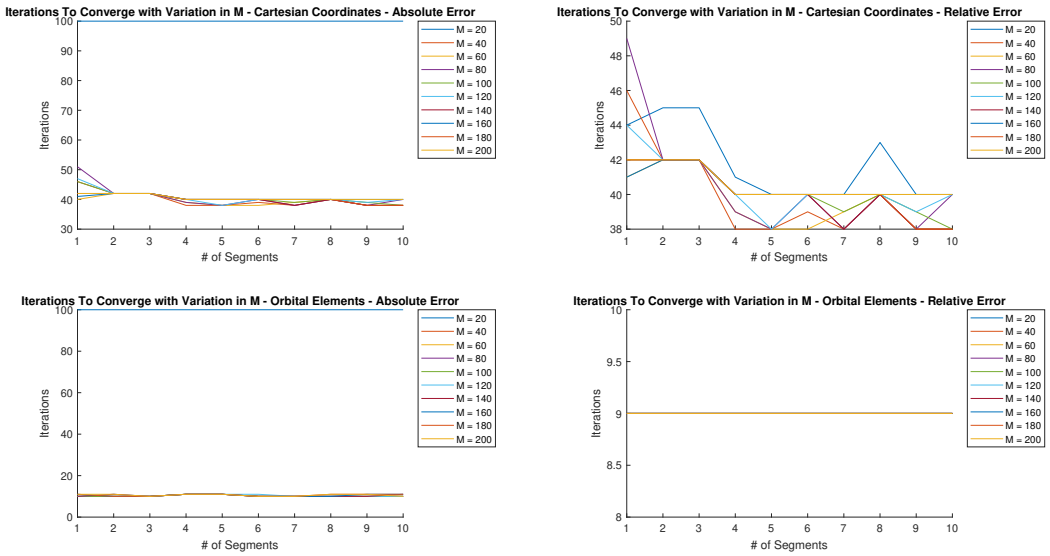

Figure 5.21: Iterations to Converge for Each Orbit over Ten Segments in Cartesian and Orbital Elements

5.5 Drag Perturbation

Another critical perturbation in astrodynamics is Drag. The equation used to calculate drag is listed as equation 5.4, [1] where the constant values are listed in table 5.3. These values will remain constant throughout the chapter. The values were selected because they roughly represented the drag associated with the International Space Station and the drag coefficient of a flat plate. [8]

$$\mathbf{d} = -\frac{1}{2}\rho S C_d v \mathbf{v} \tag{5.4}$$

The value of ρ in equation 5.4 is the density of the atmosphere around the spacecraft. There are various ways to estimate the density of the atmosphere from a simple exponential function to

Table 5.3: Base Values Used in Drag Equations

Variable	Base Value
S	0.01 km^2
C_d	2
m	400,000 kg

complex calculations approximating the composition of the atmosphere, the temperature, and the pressure at an altitude. For this paper the value of ρ will be approximated using an estimation of the temperature as a function of altitude and using that estimated temperature to estimate the density as noted by equations 5.5, 5.6, and 5.7. [9]

$$T = -131.21 + 0.00299(r - R_e) * 1000 \tag{5.5}$$

$$p = 2.488(\frac{T + 273.1}{216.6})^{-11.388} \tag{5.6}$$

$$\rho = \frac{p}{0.2869 * (T + 273.1)} \tag{5.7}$$

While the drag dynamics may change according to the supersonic speeds, equation 5.4 is an estimation of the drag effect that is acceptable to find how drag affects MCPI convergence. The analysis of supersonic drag is not in the scope of this thesis.

Multi-segment MCPI is used with second order MCPI and Cartesian Coordinates throughout this section and the remainder of the chapter. Ten segments will be used in each use of Multi-segment MCPI. The analysis will be done for all three trajectories, although it is expected that the SGEO orbit will have little variation in its results from unperturbed two-body motion. At the altitude of the SGEO, there should be very little atmosphere to cause drag.

5.5.1 Variation in Final Time

As with the analysis of the J2 perturbation, an analysis into the convergence window of Multi-segment MCPI is crucial in gaining an understanding of how to best use MCPI. The analysis is done from an initial value of 50% an orbital period to two orbital periods with a step size of 25%

of an orbital period. It is expected that there will be little to no issue with time spans that are multiples of 50% of an orbital period due to the perturbation. It is possible that this issue does occur in the SGEO trajectory due to the small amount of perturbation.

5.5.1.1 Error Profile

Nearly all of the error profiles in figure 5.22 show that the figures converged. The exceptions to the rule are the SGEO Relative Error Analysis and the Eccentric Orbit in both analysis methods. However, in general, each of the trajectories have uniform error profiles, there is little sign that variation in final time changes the error profile to extreme extents.

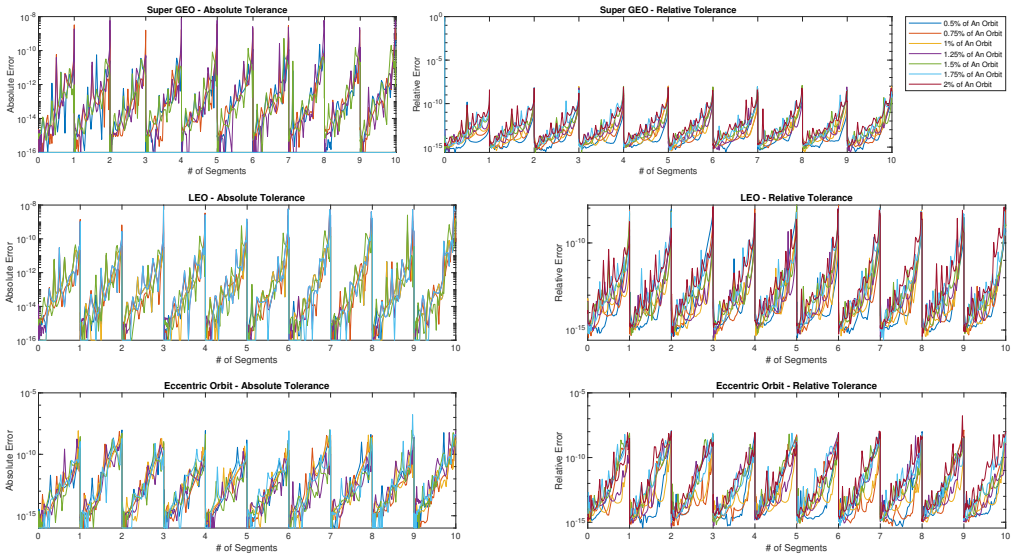

Figure 5.22: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Segment Lengths

5.5.1.2 Iterations to Converge

As stated in the prior subsection, there are concerns with the convergence of the SGEO Relative Error Analysis for when the final time is 175% of an orbital period along with the eccentric orbit in both analysis methods, as seen in figure 5.23. For the eccentric orbit the relative tolerance is only

not able to converge at the very end of the trajectory for the longest segment length, two orbital periods.

The figures show that Multi-segment MCPI, in general, works well with the segment length of one orbital period. It will be the segment length for the rest of this section.

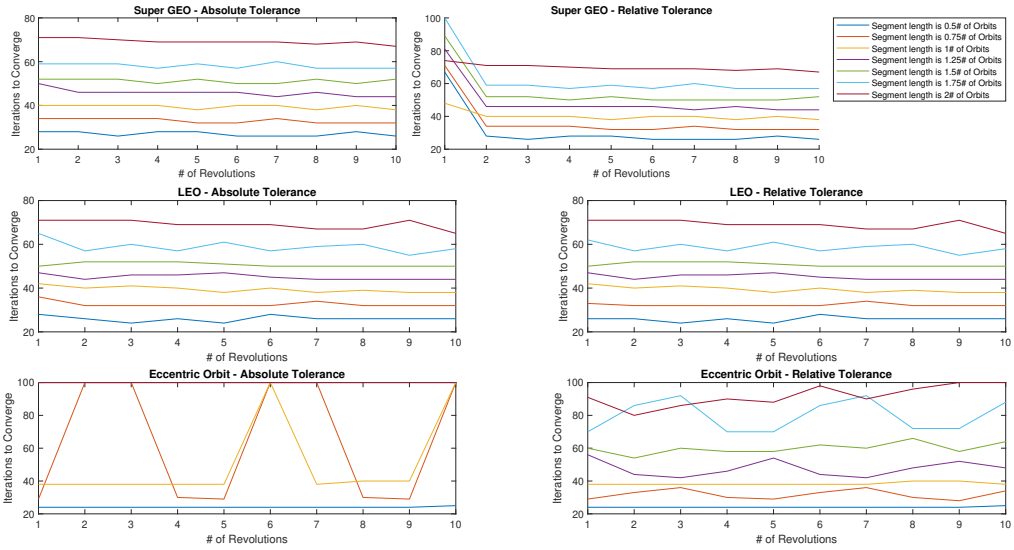

Figure 5.23: Iterations to Converge for Each Orbit over Ten Segments with Varied Segment Lengths

5.5.2 Variation in M and N

This subsection will discuss the variation in the value of M in the same manner as was discussed with respect to J2. The same variation in M will be analyzed with a minimum value of 20 to a maximum of 200 and a step size of 20. The Multi-segment MCPI will have a segment time span of one orbital period.

5.5.2.1 Error Profile

It is clear in figure 5.24 that for the majority of the trajectories an M value greater than 20 is not going to have a considerable difference in the resulting error profile. The eccentric orbit does

require a value of M of more than 40 to fully converge for this segment length. Thus the confidence in the M value of 50 that is used by default appears to be secure. While for some trajectories an M value of 30 is appropriate, 50 tends to apply well to all problems.

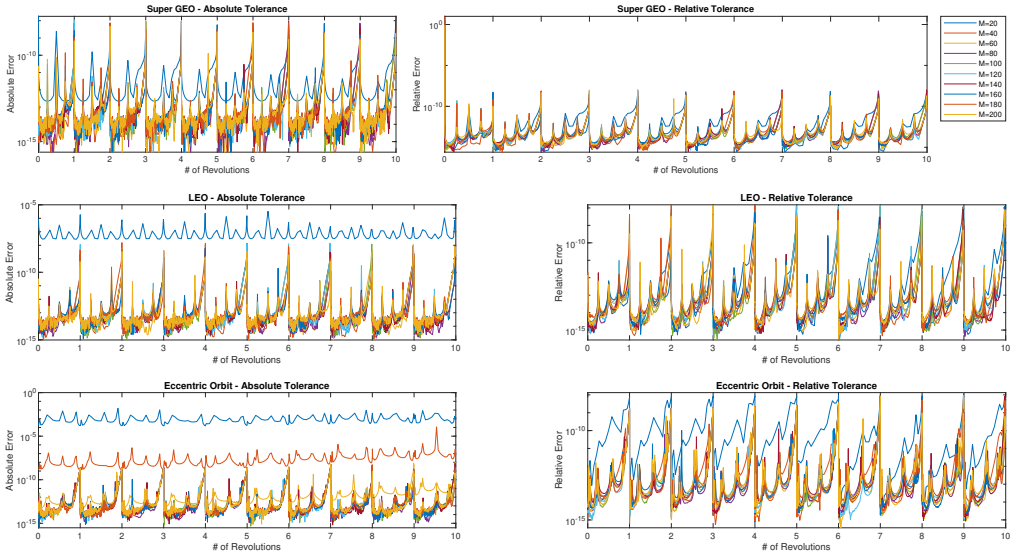

Figure 5.24: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Samples per Segment

5.5.2.2 Iterations to Converge

As stated with the analysis of J2, the non-converged segments have an iteration count of 100 as shown in figure 5.25. Here it is clear that for nearly every trajectory, the initial segment could not converge to a solution with 20 sampled points. However, for some it was able to give a convergence for the later iterations, meaning that the initial segments is the most difficult to find convergence. This holds true for almost all trajectories. the exception being the eccentric orbit. Here the orbit starts to reach a eccentricity that is large enough to cause the algorithm trouble in measuring convergence.

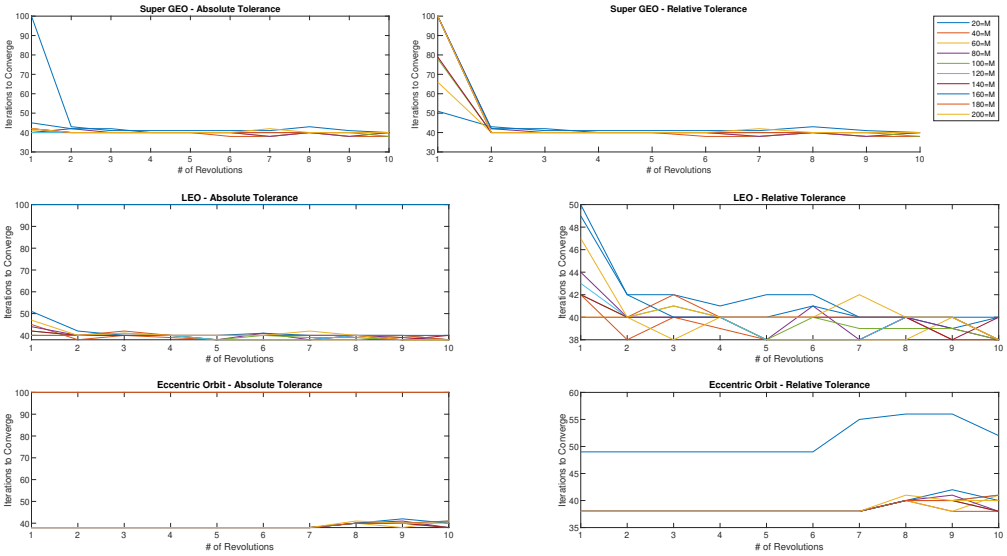

Figure 5.25: Iterations to Converge for Each Orbit over Ten Segments with Varied Samples per Segment

5.5.3 Variation in Error Tolerance

The variation in the error tolerance of the drag perturbed motion is expected to result in very similar results to the J2 perturbation. None of the trajectories are low enough to experience major drag effects (less than 100 km altitude). Thus the small perturbation of drag is expected to change the tolerance convergence in a small amount.

The value of error tolerance is varied from $1 - e1$ to $1e - 13$ in 13 logarithmic steps. All other variables in MCPI are set to their base values defined by table 5.2.

5.5.3.1 Error Profile

The changes in error tolerance for the drag perturbed motion, the error profiles are shown in figure 5.26, result in similar error profile relationships as previously found. Again, the first segment tends to have a similar error profile despite the error tolerance requirement, as seen in the SGEO relative error subplot. Then as the segments continue, there is a small variation in the error profiles of the segments. All of the figures keep roughly the same relationship as previously stated in that

the error tends to stay at close to machine zero before ramping up to the acceptable tolerance.

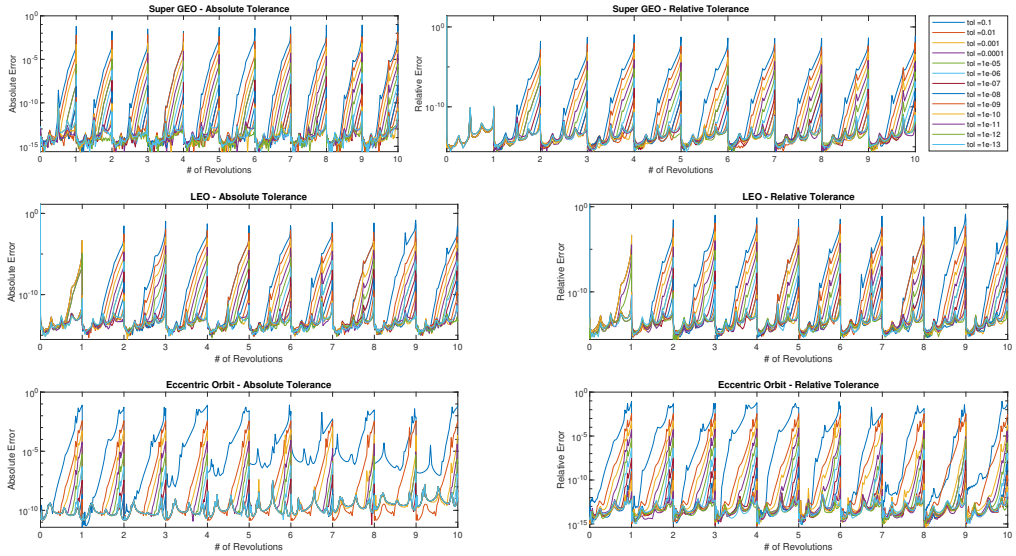

Figure 5.26: Error Profile of Each Orbital Trajectory over Ten Segments with Variations in Accepted Tolerance

As with the J2 perturbed motion, the drag perturbed motion can result in a significantly different resulting end point if the value of acceptable tolerance changes. Figure 5.27 shows the final states of the LEO trajectory with each variation in tolerance. Similar to the J2 trajectory, the error is in the second or third significant digit for the largest errors. However, it appears that for drag perturbed motion, the error tolerance can be as large as $1e-4$ and still converge to approximately the same solution as was found with $1e-12$. This would vary as more segments are added, but it appears that the drag perturbed motion is less sensitive to the error tolerance.

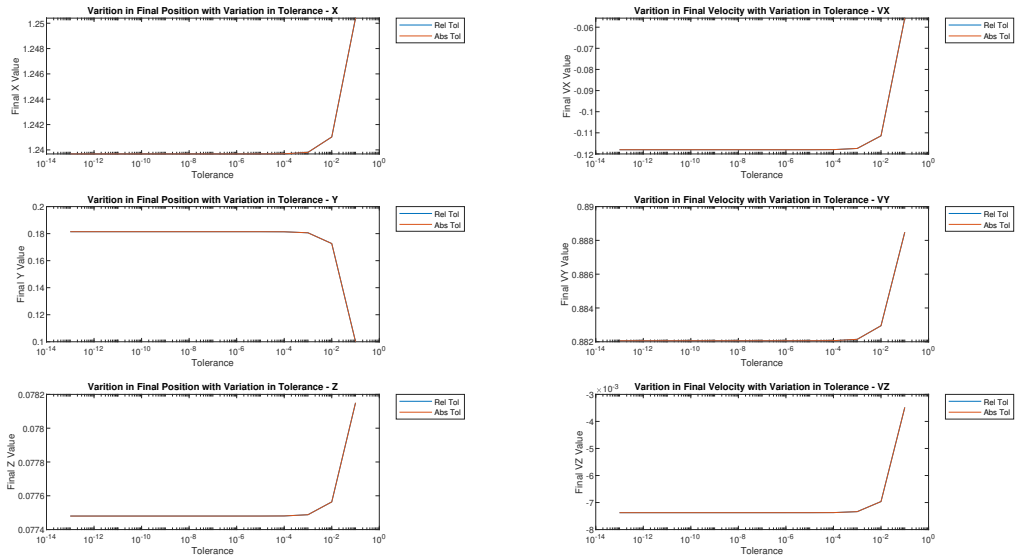

Figure 5.27: Variation in Final Cartesian States when the Acceptable Tolerance is Varied in Drag Perturbed Motion

5.5.3.2 Iterations to Converge

As previously noted, the first segment appears to have similar results regardless of the error tolerance given. For the LEO trajectory, there are groupings of error tolerances that all converge in the same number of iterations. These groups disperse with each segment and each tolerance has a consistent number of iterations necessary. Figure 5.28 also shows that for every trajectory and means of evaluating convergence, the error tolerances of $1e-12$ and $1e-13$ do not have consistent convergence. This lack of convergence suggests that for drag perturbed motion, a minimum error tolerance of $1e-10$ would be a limitation.

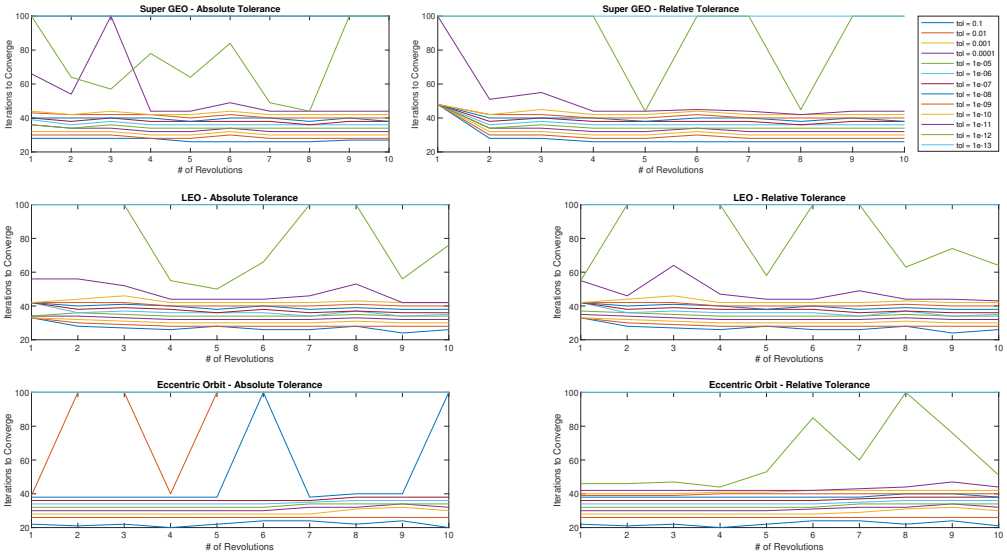

Figure 5.28: Iterations to Converge for Each Orbit over Ten Segments with Variations in Accepted Tolerance

5.6 J2 and Drag Combined Perturbation

After analyzing the convergence of Multi-segmented for J2 and Drag individually, the combined perturbation force will be analyzed. It is expected that that this combined perturbed motion will be more difficult to achieve convergence for the LEO trajectory and Eccentric Orbit. It is expected that the SGEO trajectory will behave very similarly as with each individual perturbation. At a radius of 100,000 km, the perturbation forces are expected remain considerably smaller than necessary to change the convergence patterns.

Multi-segment MCPI will be used for this section in connection with second order MCPI and a combination of the dynamics expressed in equations 5.1 and 5.4. Ten segments of length of one orbital period will be used unless otherwise stated. The values expressed in tables 5.2 and 5.3 will hold true in the definition of the dynamics.

5.6.1 Variation in Final Time

As with prior analysis, the segment length will be varied from a final time of 50% of an orbital period to two orbital periods with a step size of 25% of an orbital period. Further definition of the variation is explained in prior sections.

5.6.1.1 Error Profile

Figure 5.29 shows the variation in the error profile with the variation in the segment lengths. As with the variation when using J2 or drag perturbations alone, it is shown that the error profiles remain consistent with each segment. There is no longer the similar number of iterations that occurs in the first iteration but the algorithm is still able to converge to very similar error profiles for each trajectory.

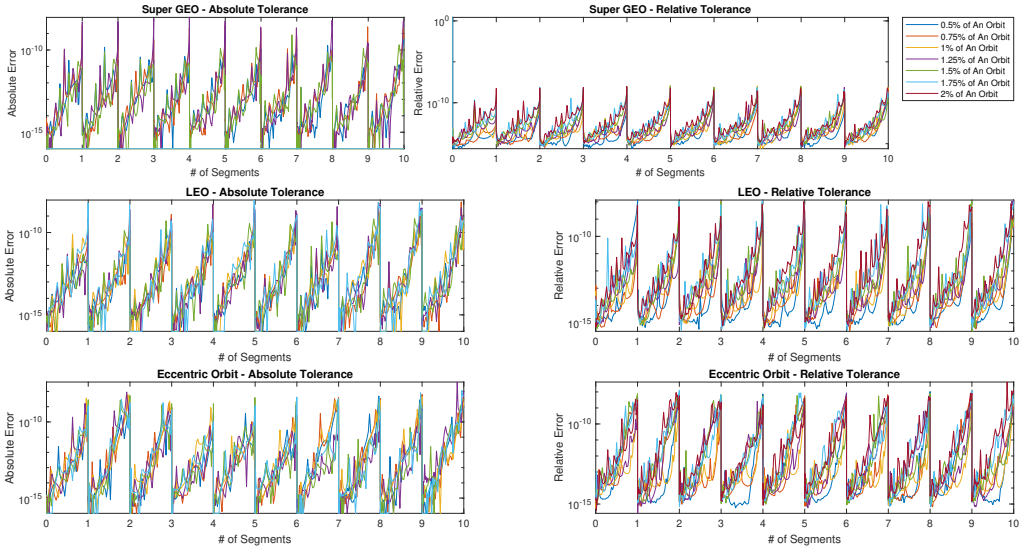

Figure 5.29: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Segment Lengths

5.6.1.2 Iterations to Converge

Figure 5.30 has a very similar behavior as noted in figures 5.14 and 5.23. The LEO trajectory is able to converge without concerns but the eccentric orbit and SGEO have more difficult converging on a solution. The combination of J2 and drag effects does not seem to have a profound impact in the ability of MCPI to converge on a solution. It is of note that the profiles of the SGEO for figure 5.30 is nearly identical to that of 5.23. Which likely means that the drag force has more of an effect than J2. However the figures are very similar to the J2 plots.

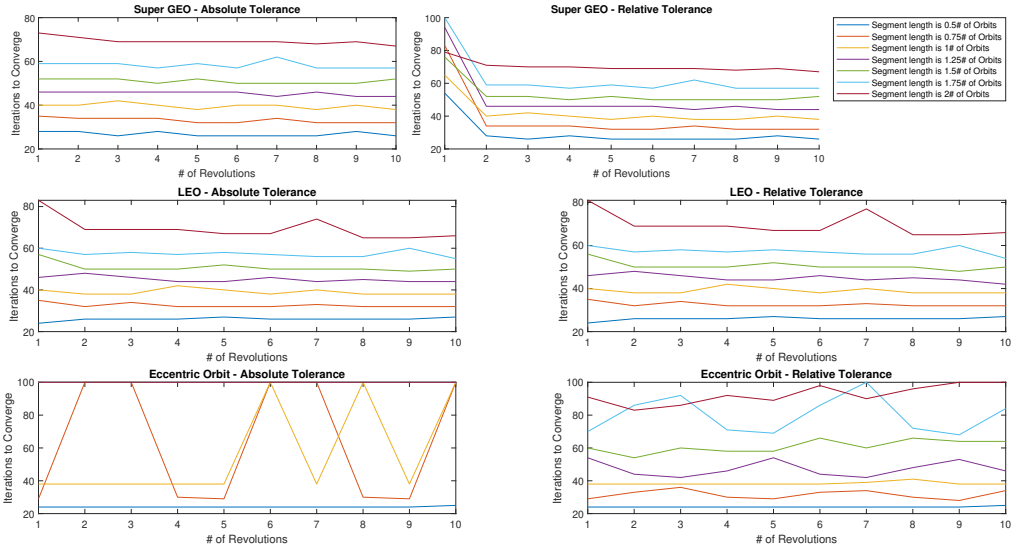

Figure 5.30: Iterations to Converge for Each Orbit over Ten Segments with Varied Segment Lengths

5.6.2 Variation in M and N

The variation of the number of sampled points in each MCPI segment is varied from 20 to 200 once again with the step size of 20. This variation in connection with the two combined perturbations is expected to have a similar effect as the prior perturbations, while it is possible that the combined perturbations require a higher number of sampled points to converge to a solution,

the low level of both perturbing forces likely correlates with the same number of required points. Especially because the number of point being around 40 or 50 has been fairly consistent among non-linear problems.

5.6.2.1 Error Profile

Figure 5.31 shows the resulting error profiles with variation in M. It is of note that in the eccentric orbit's subplots, the error profile associated with M equal to 60 is not the same as the error profiles of larger M values. This distinction means that using an M value of more than 60 is necessary to receive the optimal error profile. While not visually clear, this $M = 60$ error profile does not stay below $1e{-}8$ and therefore does not converge. It appears that the other two trajectories are able to converge with M values of 40 while meeting the required tolerance.

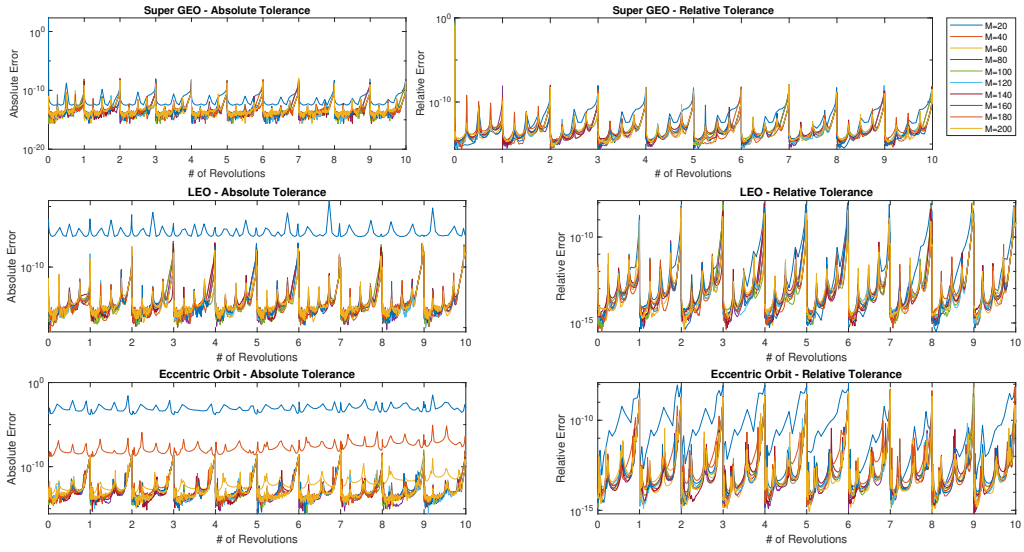

Figure 5.31: Error Profile of Each Orbital Trajectory over Ten Segments with Varied Samples per Segment

5.6.2.2 *Iterations to Converge*

As with the previous sections, the first section of the Multi-segment MCPI appears to have the most difficult time converging to a solution. For example the LEO trajectory using Absolute Error Analysis as shown in figure 5.32 has a large amount of variation in the number of iterations necessary for the first iteration before having consistent but slightly increasing number of iterations for later segments.

This figure also shows the most later segments not being able to converge due to variation in M. In figures 5.16 and 5.25, there is not the consistent growth in number of iterations necessary as seen in each subplot or the number of trajectories that are quickly unable to converge in later segments. By combining the perturbing forces, it appears that the solutions are more sensitive to convergence than previously seen.

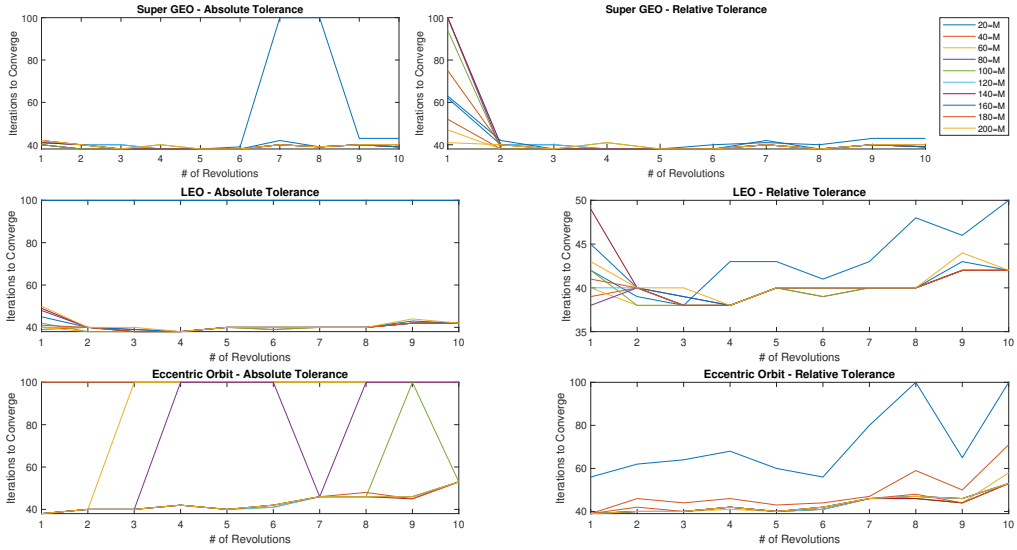

Figure 5.32: Iterations to Converge for Each Orbit over Ten Segments with Varied Samples per Segment

5.6.3 Variation in Error Tolerance

As with the prior perturbing discussions, it is expected that as the acceptable error tolerance is decreased, the number of iterations necessary will increase even until it is not able to converge within the maximum iterations. It the overall error in the final position will be very similar to the error in the other two perturbed analyses. Both had perturbed solutions in approximately the same way so it is not expected that the perturbing forces will in a way cancel each other out in the error of the final segment.

The value of error tolerance is varied from $1 - e1$ to $1e - 13$ in 13 logarithmic steps. All other variables in MCPI are set to their base values defined by table 5.2.

5.6.3.1 Error Profile

As with the other solutions, there is a consistent trend in how the error profile behaves with respect to the acceptable error tolerance as shown in figure 5.33. It is of note that the absolute error profile of the eccentric orbit does have a distinct error profile when the tolerance is set to 0.1. It does not require the same accuracy in the dynamics ans so after the third segment, the absolute error does not go below roughly $1e - 7$ as opposed to the standard of going to machine zero and slowly rising up. This means that the expected error of the tolerance equal to 0.1 is larger than seen in the J2 or Drag perturbed solutions.

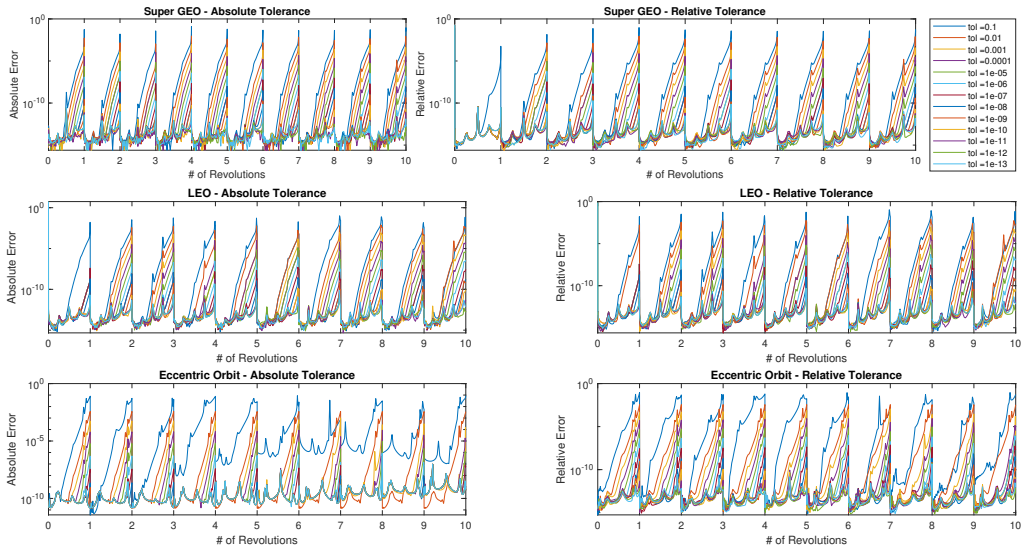

Figure 5.33: Error Profile of Each Orbital Trajectory over Ten Segments with Variations in Accepted Tolerance

The resulting error in the LEO orbit with the larger tolerance values does have a change of approximately one order of magnitude as seen in figure 5.34. Whereas the former solutions had errors in the second or third significant digit, the combined perturbation resulted in a error in the first or second significant digit. It is still expected that an error tolerance of $1e-4$ should result in a solution with a significantly small error when compared to the solution the using an error tolerance of $1e-13$. It should be noted that for the smallest values of tolerance, the Multi-segment MCPI was not able to converge on a solution according to the given tolerance.

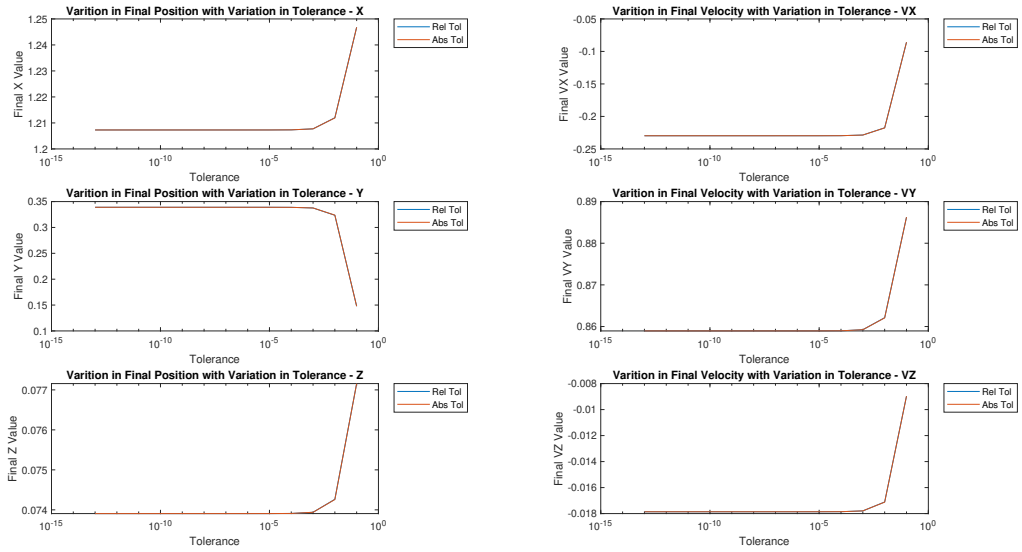

Figure 5.34: Variation in Final States with Variations in Acceptable Tolerance with J2 and Drag Perturbations

5.6.3.2 Iterations to Converge

As noted in the previous segment, for the solutions where the value of the tolerance was smaller than $1e-10$, all of the trajectories were not able to converge on a solution within the maximum number of iterations as can be seen in figure 5.35. The solution of the eccentric orbit with Absolute Error Analysis was not able to converge when the error tolerance was smaller than $1e-6$. This likely results in a final result that is not accurate to the dynamics which with an increased number of segments would become more inaccurate.

There was once again a grouping of solutions in the first segment when solving for LEO and SGEO. These tolerances quickly diverged in the number of iterations necessary but as noted in figure 5.34, had similar final results.

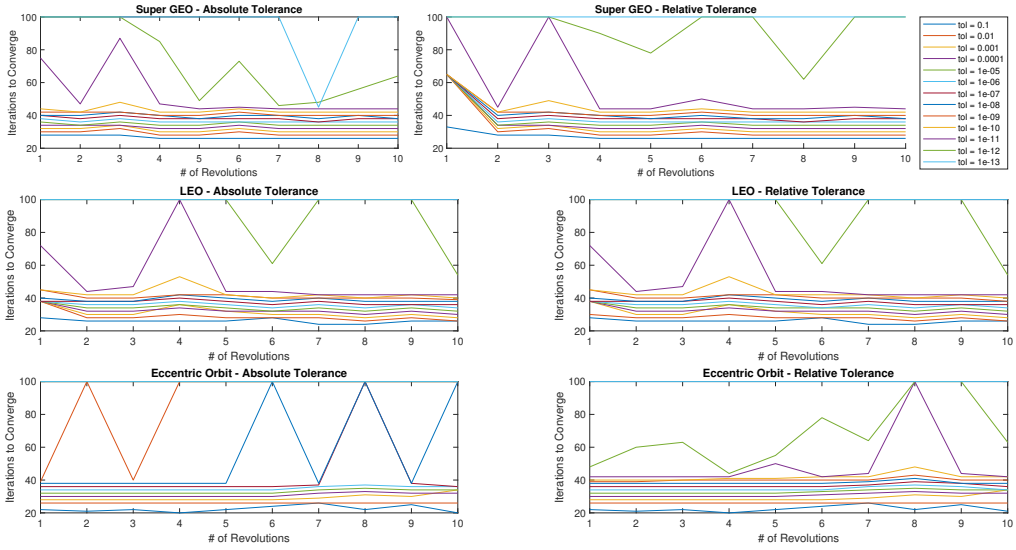

Figure 5.35: Iterations to Converge for Each Orbit over Ten Segments with Variations in Accepted Tolerance

6. SUMMARY AND CONCLUSIONS

Chapter Two discussed the need for an improved methodology when numerically integrating differential equations. A non-technical derivation of MCPI was included. The limitations of MCPI included the unknown convergence window for each problem. The contributing factors into the convergence window at this point are not clearly defined. There is the potential for inaccuracy in MCPI due to the non-uniform sampled data points, especially in the center of the interval where the points are the most spaced out. The benefits of MCPI include the decreased number of data points and the decreased number of function evaluations at each data point. Thus lowering the computational cost of the algorithm.

The algorithm was analyzed using a simple Duffing Oscillator problem with no variation in the problem itself. The convergence measures of Error Profile and Iterations to Converge were defined. Error History showing the error at each node according to the converged solution. This figure can show where the algorithm is having the most difficult time converging and where the solution is most reliable. The figure discussing the number of iterations necessary to converge shows how the function is able to evaluate the function and how a variation in the parameters effects the ability to converge. The maximum final time measure of convergence was defined in this chapter as well. The measure is a value report because the number of iterations necessary is not a linear or consistent relationship with the final time. It is not uncommon to have a converged solution at a low number of iterations for one final time and after a small increase in the final time not be able to converge on a solution.

Chapter Three demonstrated the two means of evaluating convergence, Absolute and Relative Error Analysis. The Absolute Error Analysis was derived for first and second order MCPI and compared with Relative Error Analysis using a simple first order sample problem. It was found that Absolute Error Analysis resulted in a converged solution in fewer iterations when the Absolute Error. The converged solutions had similar error profiles for both means of evaluating convergence. This signified that the Absolute Error Analysis was a preferred means of evaluating convergence

for this problem.

The variable ζ was added to the denominator of the evaluation of Relative and Absolute Error Evaluation. This value can remain zero for the majority of problems but can be set to a nonzero value that is less than zero to avoid concerns with the states or function evaluations being equivalent to machine zero.

The sample problem was also analyzed using a variation to the problem which made the solution slightly more complex. This resulted in a converged solution using Relative Error Analysis but not for Absolute Error Analysis. This shows that the convergence measure for Absolute Error Analysis is more stringent and is a better metric for evaluating convergence. Instead of simply evaluating whether a solution was reached, it is able to evaluate whether the solution represents the correct dynamics.

Chapter Four discussed the use of second order MCPI with variations in the duffing oscillator. It was noted by varying the final time of the MCPI segment, that the Absolute Error Analysis and Relative Error Analysis had the same maximum final time for the given problem but that the Absolute Error Analysis was able to converge in fewer iterations. The Absolute Error Analysis also keeps the error at approximately machine zero for as long as necessary before causing a trail up that has a final value with the defined error bounds.

As the value of M was varied, while keeping N equivalent to M, it was noted that the error profiles for M values greater than 40 were all fairly equivalent. There is a clear benefit in restricting the number of sample points because the increase in M did not correspond with a decrease in the number of iterations or an decrease in the error profile. There is a trade off point where the algorithm is able to converge for a minimum value of M while minimizing the number of evaluations of the function at each node point is necessary. The Relative Error Analysis is able to converge for a similar maximum final time regardless of the value of M. However, some of these converged solutions do not accurately represent the dynamics of the system as noted by the fact that the maximum final time for Absolute Error Analysis is smaller. The two means of evaluating convergence met for the same maximum final time for larger values of M.

As the error tolerance of the algorithm is varied, it is noted that the Absolute Error Analysis profile continues to hold onto the error at machine zero for as long as necessary until the increase in later errors is with the defined bounds. For all values of acceptable tolerance, the first few points are reliable due to the similarity with the initial states that are given. Again, it was noted that the number of iterations necessary for Relative Error Analysis was always greater than or equal to the number of iterations necessary for Absolute Error Analysis.

As the non-linearity of the dynamics was varied, the ability for the algorithm to converge varied as expected. There was very little difference in the final error profiles with the exception that the relative error profile had a spike in error at a specific time for all of the converged trajectories. This spike in error was due to the node being close to a velocity state equal to zero. Because of that, any minor variation in the state lead to changes in the velocity which resulted in variations in the given states. The increased sensitivity at this point limited the ability of the system to converge for some of the values of ϵ.

Chapter 5 discussed the application of standard MCPI on the two body problem for each of the three orbits. The three trajectories were selected to see how the proximity to the earth and eccentricity of the orbit affect the MCPI convergence capabilities. The initial variation in final time also was analyzed using first and second order MCPI to evaluate whether or not the convergence issues were unique to one order. It was noted that the second order MCPI leads to converged solutions in fewer iterations than when first order MCPI is used for both Absolute and Relative Error Analysis.

It was also noted that for the unperturbed trajectories, there are convergence issues whenever the segment length is a multiple of 50% of the orbital period and ζ is equal to zero. This was due to the center node falling on a multiple of 25% of the orbit. Which for the circular orbits, so long as it is starting at an initial value of zero for one of the position, results in a value of zero for one of the position values. This value of zero or near zero, makes it more difficult to evaluate the error. In response the value of ζ was increased to a nonzero value of $1e-3$ to increase the value of the denominator when evaluating the error. This resulted in consistent convergence regardless of of

the length of the segment, while remaining in the convergence window.

Using the new variation in the value of ζ, the value of M was varied to evaluate the effect of M in the convergence of the two body problem. The Absolute Error Analysis had an easier time converging a solution for most of the problems, specifically for the problems where the value of final time are less than five orbital periods. It was also noted that the MCPI algorithm was not able to easily evaluate convergence for the eccentric orbit trajectory. The eccentric orbit was however able to converge for a final time of 14.5 orbital periods when using Relative Error Analysis while unable to converge for final times of less iterations when the value of M is between 40 and 80. However was unable to converge for larger values of M. This behavior was unique and due to inaccurate convergence when using Relative Error Analysis

The acceptable error tolerance was varied for the two body problem next with the same irregular time variation as previously mentioned primarily to find the maximum final time for each tolerance value. Again, the Relative Error Analysis had a difficult time finding converged values because of the frequency of points where a node has a state value equal to zero or near zero. The same relationships as noted previously are present in this variation of the error tolerance. There is a limitation of tolerance levels for which the algorithm is able to converge and for the smallest tolerance values, the maximum final time is greatly restricted.

The three orbits where then used to analyze perturbed motion with the J2 and drag perturbations applied. Due to the slow impact of each perturbing force, the window of convergence would need to be greatly increased. To account for the larger convergence window, Multi-segment MCPI was created. The code for Multi-segment MCPI is included in the code package with the MCPI for general purpose use. Each perturbing force was analyzed for ten segments and the segment length is one orbital period.

The J2 perturbing force was analyzed without other perturbing forces first to see if the perturbing force had a clear negative effect on the convergence abilities. The result was that the 50% orbital period time spans no longer had convergence problems due to the center node corresponding with a value at exactly zero. The perturbed solution resulted in improved convergence for

these values. In future uses of MCPI, if the dynamics are overly simplistic, a perturbing force can actually help avoid these anomalies in convergence.

It was noted that the variation in final time for each segment lead to similar results in the number of iterations necessary for each segment. While the first segment had more difficulty converging on a solution, the later segments would commonly converge in roughly the same number of iterations as each other. The exception was the eccentric orbit which had more sensitivity to the segment length and resulted less consistent iterations necessary.

As the value of M was varied it because clear that for M values of greater than 50, there is little difference in the error profiles and in the number of iterations necessary to converge. When the value of M is below 50, the algorithm is not able to converge for all of the trajectories. As noted previously a default value of 50 for M has been generally applicable for all problems. The variation in M should be evaluated from a central value of 50.

The acceptable error tolerance was varied within Multi-segment MCPI which can result in significantly different results according to the number of segments and the tolerance given. As noted with the J2 perturbed problem, there was little variation in the final states for the tolerance values of less than $1e-6$ although there was a significant difference in the number of iterations necessary to reach the tolerance for each segment. Using a tolerance of $1e-8$ as a default offers a solution which will not significantly vary from the solution using a more stringent tolerance criteria.

The use of Cartesian Coordinates as opposed to Orbital Elements was discussed with a variation in the value of M. The two coordinate systems were analyzed for the LEO trajectory. It was shown that the two coordinate systems require roughly the same value of M for the same time span and the use of orbital elements reduces the number of iterations necessary to converge by a factor of four. The algorithm was also converging at roughly the same number of iterations for Absolute Error Analysis and Relative Error Analysis when using Orbital Elements. The use of orbital elements is recommended in future use of perturbed motion in astrodynamics due to the fact that five of the variables are slow and the initial value of each element is an accurate estimate for the value at each

time.

The system was analyzed using the drag perturbation instead of J2 to see any clear differences. This perturbation, like the J2 perturbation had a larger impact on the LEO trajectory than the other two trajectories. The same variations of final time, M, and error tolerance where used to evaluate the ability to converge with the drag perturbation included. The final results were very similar to the results obtained from the J2 perturbation. The drag perturbation did lead to more difficulty in evaluating convergence with each final time value for the eccentric orbit. Even leading to the one orbital period trajectory having difficulty converging on a solution. As the value of M, and also N, were changed it was noted that the same limitation of M equal to 50 applies as well to each of these trajectories.

When the value of the acceptable tolerance was varied for the drag perturbed motion, a unique result happened with the eccentric orbit using Absolute Error Analysis. The error profile when using an error tolerance of $1e-1$ does not return to approximately machine zero after the 3rd segment. The error introduced by the relaxed error tolerance because to present itself as error in the dynamics of the initial value estimate as well as the rest of the states. This resulted in significantly more erroneous final states as was noted. Again, the error tolerance of less than $1e-6$ does not lead to significantly different results.

The final analysis was the analysis into the combined J2 and drag perturbing effects. It was found that in general, the convergence was affected by the drag perturbing forces. Within the variations of final time, M, and error tolerance, the error profiles and number of iterations necessary for convergence followed the trends of the drag effected results. This is due to the large amount of drag used as the perturbing force. The spacecraft has a larger drag force than the international space station and therefore greatly affected by drag.

6.1 Challenges

The challenges of this paper included deciding which problems to analyze and what variations can be made to analyze it. By decreasing the step size of any variable, potentially outlying situations can be found and the general trend of a system can be found. However, this can and did lead

to a congestion of data that can be difficult to visualize succinctly.

The three orbits selected for analysis in the astrodynamics problems provided some insight but further variation to the trajectories would be beneficial. This would include an analysis into how each orbital element changes the convergence criteria. However, by varying six parameters the amount of trajectories to analyze with variation in final time, M, and accepted tolerance and variation in two-body and perturbations, would lead to a large number of trajectories which the author did not have enough time to work through.

An additional challenge of the paper was that the relationship between convergence and not being able to converge can be a fine line. This line may be lead by outlying values. Which when analyzed, do not lead to clear and succinct definitions of what is necessary for the system to converge. This relationship holds true for the use of Absolute Error Analysis and Relative Error Analysis.

It is also possible for a converged solution using Relative Error Analysis that is not a true solution to the system. This inaccuracy of converged solutions may lead to inaccurate assumptions and conclusions if the solutions is not visually inspected or verified with Absolute Error Analysis.

6.2 Further Study

Among the possibilities for further research into the convergence of MCPI, the common factor is an analysis into more problems in which MCPI or a numerical integration scheme may be used. As previously stated, analysis into the variation of orbital elements would be beneficial in the understanding of how the problems affect MCPI. In astrodynamics, the use of MCPI to solve using Modified Equinoctial Elements (MEEs) would be a valuable addition. The expansion of more precise gravitational fields, three-body dynamics, and n-body dynamics would greatly benefit from further analysis.

In addition to these applications of MCPI, the use of Multi-segment MCPI in Optimal Control Theory offers clear benefits. It is common to solve the problem a nonlinear solver with a numerical integration scheme. This can easily lead to thousands of integrations and millions of evaluations of the function. The use of MCPI could lead to dramatically improved efficiency of finding solutions.

The standard use of MCPI has an issue of where any unknown variation in the optimal control may lead the problem to be outside of its convergence window. By minimizing the final time and using multiple segments, there is an assurance that the problem is likely to converge quickly and reach a result.

REFERENCES

[1] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., 1999.

[2] X. Bai, *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*. PhD thesis, Texas A&M University, 2010.

[3] Bai and Junkins, "Modified chebyshev-picard iteration methods for solution of initial value problems," *The Journal of Astronautical Sciences*, vol. 59, no. 1, pp. 335–359, 2012.

[4] Bai and Junkins, "Modified chebyshev-picard iteration methods for solution of boundary value problems," *The Journal of Astronautical Sciences*, vol. 58, no. 4, pp. 615–642, 2011.

[5] T. O. line Encyclopedia of Linear Sequences, "Chebyshev polynomials of the second kind." Available at `https://oeis.org/A053117` (accessed: 05.30.2021).

[6] T. J. Rivlin, *Chebyshev Polynomials, From Approximation Theory to Algebra and Number Theory*. Dover Publications, 2020.

[7] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier, 2014.

[8] P. F. G. S. J. Stark, *Spacecraft Systems Engineering*. John Wiley & Sons, 2011.

[9] G. R. C. NASA, "Earth atmosphere model - metric units." Available at `https://www.grc.nasa.gov/www/k-12/airplane/atmosmet.html` (accessed: 05.15.2021).

APPENDIX A

MCPI CODE PACKAGE

The code used to run MCPI can be found at https://github.com/gelfmoore/MCPI

This code can be used for first and second order MCPI, single segment or multi-segment MCPI, and with Absolute Error Analysis or Relative Error Analysis.

The variables within MCPI that can be changed with the options struct are:

- *M* The number of sample node points
- *N* The maximum order of Chebyshev Polynomials
- *tol* - The acceptable tolerance for evaluating convergence
- *itermax* - The maximum number of iterations before the algorithm reports the solution as not converged
- *conv_eval* - Set to 1 when using Relative Error Analysis and 2 when using Absolute Error Analysis
- *errtol* - The minimum value of the state or the function before evaluating the error as *errtol*
- *display* - Set to 1 to turn on display in MCPI. Set to 0 to suppress display
- *order* - Set to 1 to use first order MCPI. Set to 2 to use second order MCPI
- *NumSeg* - The number of segments used in Multi-segment MCPI.
- *errparameter* - Value added to the denominator of the error evaluation.