Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 2-5, 2015, Boston, MA, USA

DETC2015-47333

MULTI-POINT MULTI-HARMONIC COLLOCATION WITH CONTINUATION TO COMPUTE BRANCHES OF NONLINEAR MODES OF STRUCTURAL SYSTEMS

Sean J. Kelly
Graduate Student[1]
Engineering Mechanics Program
University of Wisconsin-Madison
534 Engineering Research Building
1500 Engineering Drive
Madison, Wisconsin 53706
sjkelly486@gmail.com

Matthew S. Allen
Associate Professor
Department of Engineering Physics
University of Wisconsin-Madison
535 Engineering Research Building
1500 Engineering Drive
Madison, Wisconsin 53706
msallen@engr.wisc.edu

Hamid Ardeh
Graduate Student
Mechanical Engineering
University of Wisconsin-Madison
534 Engineering Research Building
1500 Engineering Drive
Madison, Wisconsin 53706
hamid.ansari.ardeh@gmail.com

ABSTRACT

Fast numerical approaches have recently been proposed to compute the undamped nonlinear normal modes (NNMs) of discrete and structural systems, and these have enabled remarkable insights into the nonlinear behavior of more complicated systems than are tractable analytically. Ardeh et al. recently proposed the multi-point, multi-harmonic collocation (MMC) method, which finds a truncated harmonic description of a structure's NNMs. The MMC algorithm resembles harmonic balance but doesn't require any analytical pre-processing nor the computational expense of the alternating time-frequency method. In a previous work this method showed significantly faster performance than the shooting method and it also allows the possibility of truncating the description of the NNM to avoid having to compute every internal resonance. This work presents a pseudo-arc length continuation version of the MMC algorithm which can compute a branch of NNMs and compares its performance with the shooting/pseudo-arc length continuation algorithm presented by Peeters and Kerschen in 2009. The algorithms are compared for two systems, a two degree-of-freedom (DOF) spring-mass system and a 10-DOF model of a simply-supported beam.

INTRODUCTION

Nonlinear modes are a natural extension of linear modes to nonlinear dynamic systems, and can provide a wealth of insight into the dynamic responses that a system may exhibit. This work focuses on undamped nonlinear normal modes (NNMs) using the definition proposed by Kerschen, Vakakis and others (see, e.g. [1]). Undamped NNMs form the backbone of the forced, steady-state response of the damped nonlinear system, showing how much the resonance frequency will shift with increasing force/response amplitude [1] (also see [2] for an energy balance technique that connects the force amplitude to the response amplitude near the NNM). They are also related to invariant manifolds in the state space, so a free response that is initiated near the manifold tends to decay along the manifold

[1] M.S. Completed Dec. 2014. Currently: Applied Mechanics Analyst, Cummins Emission Solutions, 1801 Highway 51-138, Stoughton, WI 53589 USA, sean.kelly@cummins.com.

unless a bifurcation is encountered [3-5]. For a system with heavy or nonlinear damping these concepts are guaranteed using the damped nonlinear mode framework developed by Shaw and Pierre [6]; this work focuses on undamped NNMs, which provide a good approximation for the damped manifolds if the damping is light and linear. NNMs also provide insight into bifurcations [1], internal resonance [1], period lengthening [5], and other complicated phenomena. They were also recently shown to provide insight into the frequency smearing that occurs in the power spectra of randomly excited nonlinear structures.

It is important to note, that while NNMs provide tremendous insight into a system's dynamic response, linear superposition does not hold so they cannot yet be used to reduce the cost required to compute the response of the system. This is an important point since some reasoning that holds with linear systems and linear modes proves erroneous for nonlinear systems. On the other hand, NNMs can be used to provide an initial guess for the steady-state response [2] and some promising work has been performed to explore superposition of NNMs. For example, Ardeh recently proved that over a local region a certain class of nonlinear systems accept a bi-linear superposition law [7]. He demonstrated that this law provides an excellent approximation to an arbitrary response of a 2DOF system over a wide range of energy and work is ongoing to use these concepts to more efficiently compute the response of nonlinear structures and to gain new insights into nonlinear response.

While NNMs have been computed analytically for simple systems for several decades, beginning with Rosenberg's seminal work in 1960 [8, 9], interest has increased in recent years thanks to computational algorithms that can quickly and relatively robustly compute NNMs from systems with several DOF. Peeters and Kerschen recently presented a fast and robust computational algorithm [10] that uses shooting together with a pseudo-arclength continuation algorithm to iterate on the transient response of a system until it is periodic over one cycle of the NNM. The algorithm employs a fast Newmark integration routine [11] that integrates the equations of motion and their Jacobian simultaneously and thus obtains the response and the Jacobian of the shooting function simultaneously at a significantly reduced computational cost. This has made it possible to compute the NNMs of structures with hundreds of degrees-of-freedom [12]. Even then, nonlinear modal analysis can be expensive, especially for higher order systems where many intricate internal resonances may be exist, each requiring a fine step size and considerable computational effort to follow.

Ardeh recently presented a new algorithm, dubbed Multi-point Multi-harmonic Collocation (MMC) that uses the equation of motion and its Jacobian to compute a Fourier series approximation to the NNM. The algorithm does not require any analytical pre-processing that is used in the harmonic balance method where the assumed harmonic solution is substituted into the equation of motion and the expressions obtained are integrated over one period to obtain algebraic expressions for the resulting amplitudes of the response harmonics. In the multi-harmonic balance algorithm this integral is replaced by an FFT/IFFT [13] greatly increasing the cost of the algorithm. In fact, the MMC algorithm does not integrate the response over any time interval, so it has the potential to dramatically reduce the cost required to compute the periodic responses of a system. In their initial work, Ardeh and Allen showed cases where the MMC algorithm was orders of magnitude faster than shooting [14]. Also, because MMC uses a Fourier series description, one may be able to truncate the description to avoid computing higher order internal resonances.

While the original work on MMC only presented the equivalent of a shooting technique, this work implements the MMC algorithm within a pseudo-arc length continuation framework in order to compute branches of nonlinear normal modes using MMC. This allows one to compare the speed of the algorithm to other approaches in a more meaningful setting, and to test the algorithm in a wide range of settings both near and far from bifurcation points. The results so far show that MMC could be a promising compliment to existing continuations techniques (such as the algorithm by Peeters and Kerschen [10]).

The rest of the paper is organized as follows. The following section briefly derives the MMC algorithm and explains how pseudo-arc length continuation was implemented. The next section compares the performance of the algorithm to that by Peeters and Kerschen [10] for two systems, a 2DOF system that has been studied extensively in other works [1, 9, 10] and a 10 DOF reduced order model for a simply-supported beam that exhibits geometric nonlinearity due to bending-membrane coupling.

REVIEW OF MMC AND IMPLEMENTATION OF PSEUDO-ARC LENGTH CONTINUATION

Review of MMC

The MMC algorithm is valid for undamped dynamic systems with smooth nonlinearities, and the equation of motion for this class of system can be written as follows.

$$\ddot{x} = f(x), \quad x \in \Re^n \tag{1}$$

The response of interest is periodic, so it can be readily expanded in a Fourier series as follows, where the frequency ω or period T is not known a priori.

$$\ddot{x}(t) = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} + \sum_{k=1}^{N} \left\{ \begin{bmatrix} A_{1k} \\ \vdots \\ A_{nk} \end{bmatrix} \cos(k\omega t) + \begin{bmatrix} B_{1k} \\ \vdots \\ B_{nk} \end{bmatrix} \sin(k\omega t) + \right\}$$

$$\omega = \frac{2\pi}{T}, T \in \Re^+ \tag{2}$$

One can solve for the unknown period and the coefficients A_{jk} and B_{jk} that define the periodic response using a Newton-Raphson approach. Note that Ardeh & Allen also presented a steepest descent algorithm in [14] but that algorithm is not employed in this work. The Newton Raphson scheme is implemented by defining a set of M collocation points that are distributed between 0 and 2π in phase as follows, $\phi_{i+M} = 2\pi + \phi_i, i = 1,...,M$, where $\phi_{i+M} \in [0,2\pi]$ and $\phi = \omega t$. To facilitate the derivation that follows, we define a copy of the assumed solution, $\bar{x}$, in which time is scaled so the interval becomes 2π and then $\bar{x}(\phi_k) = x(\frac{\phi_k}{2\pi}T)$. Inserting the harmonic description into Eq. (1) and then defining the difference function, δ, at each point.

$$\delta_k(\bar{x}, x) = \ddot{\bar{x}}(\phi_k) - f(\bar{x}(\phi_k)) = 0, k = 1,...,2M \tag{3}$$

One can then use the Newton Raphson procedure to find an update to the coefficients, that drives the difference function towards zero. Specifically, denoting the vector of coefficients at the qth time step as C^q,

$$\begin{aligned} C^{(0)} = \Big[& c_1 \left| A_{11} ... A_{1N} \right| B_{11} ... B_{1N} \right| \\ & ... \left| c_n \left| A_{n1} ... A_{nN} \right| B_{n1} ... B_{nN} \right| \Big]_0^{\mathrm{T}} \end{aligned} \tag{4}$$

we seek updates, ΔC and $\Delta\omega$, that drive the difference function towards zero.

$$\begin{aligned} C^{(q+1)} &= C^{(q)} + \Delta C^{(q)} \\ \omega^{(q+1)} &= \omega^{(q)} + \Delta\omega^{(q)} \end{aligned} \tag{5}$$

To compute these updates, the Jacobian for difference function was derived, as will be summarized below, and is used to form an overdetermined linear system that provides the update to the coefficients and period that drives the difference function towards zero.

$$\begin{bmatrix} \Im_{C^q}^{\bar{\delta}_1^q} & \Im_{\omega^q}^{\bar{\delta}_1^q} \\ \vdots & \vdots \\ \Im_{C^q}^{\bar{\delta}_{2M}^q} & \Im_{\omega^q}^{\bar{\delta}_{2M}^q} \end{bmatrix} \begin{bmatrix} \Delta C^q \\ \Delta\omega^q \end{bmatrix} = \begin{bmatrix} \bar{\delta}_1^q \\ \vdots \\ \bar{\delta}_{2M}^q \end{bmatrix} \tag{6}$$

The harmonic solution is readily differentiable, and this facilitates computation of the Jacobian. First, the following are defined,

$$\Gamma(v^T) = \begin{bmatrix} v^T & 0_{1\times(2N+1)} & \vdots & 0_{1\times(2N+1)} \\ 0_{1\times(2N+1)} & v^T & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{1\times(2N+1)} \\ 0_{1\times(2N+1)} & \cdots & 0_{1\times(2N+1)} & v^T \end{bmatrix} \tag{7}$$

and

$$\begin{aligned} \alpha_m^T &= \Big[0 \quad \cos(\phi_m) \quad 2^2\cos(2\phi_m) \quad ... \quad (N)^2\cos(N\phi_m) \\ &\quad \sin(\phi_m) \quad 2^2\sin(2\phi_m) \quad ... \quad (N)^2\sin(N\phi_m) \Big] \\ \beta_m^T &= \Big[1 \quad \cos(\phi_m) \quad ... \quad \cos(N\phi_m) \\ &\quad \sin(\phi_m) \quad ... \quad \sin(N\phi_m) \Big] \\ \gamma_m^T &= \Big[0 \quad -\sin(\phi_m) \quad -2^3\sin(2\phi_m) \quad ... \quad -(N)^3\sin(N\phi_m) \\ &\quad \cos(\phi_m) \quad 2^3\cos(2\phi_m) \quad ... \quad (N)^3\cos(N\phi_m) \Big] \\ \lambda_m^T &= \Big[0 \quad -\sin(\phi_m) \quad ... \quad -N\sin(N\phi_m) \\ &\quad \cos(\phi_m) \quad ... \quad N\cos(N\phi_m) \Big] \\ \phi_m &= \frac{m}{M}(2\pi) \in [0, 2\pi), m = 1,...,M. \\ \phi_{m+M} &= 2\pi + \phi_m \end{aligned} \tag{8}$$

and one can then show that the Jacobian of δ is given by the following.

$$\begin{aligned} \bar{x}_m^{(k)} &= \Gamma(\beta_m^T)\bar{C}^{(k)} \\ \delta_m^{(k)} &= \omega^2\Gamma(\alpha_m^T)\bar{C}^{(k)} + f(\bar{x}_m^{(k)}) \\ \Im_C^{\bar{\delta}_M^{(k)}} &= \omega^2\Gamma(\alpha_m^T) + \Im_x^f(\bar{x}_m^{(k)})\Gamma(\beta_m^T) \end{aligned} \tag{9}$$

$$\begin{aligned} \Im_\omega^{\bar{\delta}_M^{(k)}} = \Big[& 2\omega\Gamma(\alpha_m^T) + \\ & 2\omega\frac{\pi m}{M}\Gamma(\gamma_m^T) + \frac{2\pi m}{\omega M}\Im_x^f(\bar{x}_m^k)\Gamma(\lambda_m^T) \Big]\bar{C}^{(k)} \end{aligned} \tag{10}$$

For further details and discussion the reader is referred to [14].

Pseudo-arc length continuation MMC

MMC was implemented within the pseudo-arc length continuation framework developed by Peeters et al. and detailed in [10]. The implementation follows that in [10] very closely, so only an overview will be presented here. The algorithm begins at a linear normal mode (LNM) at a low energy (small displacement). The null vector of the Jacobian above is then found and this defines the direction in which the coefficient vector , C^q, can be increased without increasing δ (or the error in the approximation of the solution to Eq. (1) by $x(t)$ in Eq. (2). A step is taken using the initial stepsize defined by the user and the Newton Raphson procedure is used to find a new solution. As explained in [10], a constraint is added to the Newton procedure to force the corrections to be perpendicular to the prediction vector. The same stepsize control algorithm outlined in [10] is used, which defines the next step, $s_{(j)}$ as

$$s_{(j)} = \left(\frac{P^*}{P_{(j-1)}} \right) s_{(j-1)} \tag{11}$$

where P^* , the desired number of steps in the Newton routine, is a parameter chosen by the user and $P_{(j-1)}$ is the number of steps in the previous Newton iteration.

In the implementation of MMC, this was augmented by limiting the change in stepsize over each iteration. Specifically, after computing s(j) as given above, the following conditions were applied.

$$R = \frac{s_{(j)}}{s_{(j-1)}} \tag{12}$$

$$s_{(j)} = \begin{cases} R_{\min} s_{(j-1)} & r < R_{\min} \\ R_{\max} s_{(j-1)} & r > R_{\max} \\ R & \text{otherwise} \end{cases} \tag{13}$$

where $R_{\min}$ and $R_{\max}$ limit the maximum percent change in the stepsize in each iteration. In the studies that follow these parameters were set as shown in Table 1 to assure that the step-size didn't change more than 50% between prediction steps.

Table 1: Step Size Control Parameter Values

Parameter	MMC	NNMcont
$s_{\min}$	1e-8	1e-6
$s_{\max}$	10	10
$R_{\min}$	1.5	n/a
$R_{\max}$	.5	n/a
N^*	10	3

Comments on period lengthened solutions

NNMs have unique properties not present in their LNM counterparts. If ω_o is the fundamental frequency of a NNM, then $\omega = \omega_o / n$ is also a solution for an integer $n > 1$. These additional solutions are referred to as period-lengthened solutions as their periods are $T = nT_o$ $T = nT_o$. Additionally, solution branches exist that connect branches stemming from the continuation of any of the linear normal modes (LNMs), or their period-lengthened solutions. These branches stem from the main branches at bifurcation points.

An interesting aspect of the MMC algorithm is that it is capable of finding the period-lengthened solutions and the connecting branches that previous continuation-based algorithms would not typically detect. The number of harmonics retained in $x(t)$ has considerable effect on which solution the algorithm may determine.

NUMERICAL EXPERIMENTAL SETUP

In order to benchmark the effectiveness of the MMC algorithm, it was compared to the algorithm presented by Peeters et al. in [10], and which will here be referred to as NNMcont. The computed NNM solutions, error of the solutions, and required computation time will be compared between the two algorithms. Also, the parameters in the MMC algorithm, such as the number of harmonics retained in the MMC displacement solution (N) will be investigated for their effects on the results.

The Continuous MMC and NNMcont algorithms were tested under similar conditions in order to compare speed and accuracy of the methods. The algorithms were run to obtain the NNMs of a 2-DOF system described by the nonlinear equations of motion.

$$f(x) = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{Bmatrix} -0.5x_1^3 \\ 0 \end{Bmatrix} \tag{14}$$

The NNMs were determined for various energy levels and number of solutions. The computational time required for each algorithm to determine the sets of NNMs was measured using Matlab's CPU time measurement tool. The error for each method was determined by integrating the 2-DOF system from the initial conditions of the NNMs up to the computed period. The integration was performed by Matlab's ode45 function with a relative tolerance of $1x10^{-9}$ to minimize error introduced through integration and focus on error due to the computed initial conditions and period. The error was computed by measuring the differences between initial and final displacements, scaled by the initial displacements.

$$\varepsilon^H = \frac{\|x(0) - x(T)\|}{\|x(0)\|} \tag{15}$$

Both algorithms could perform at faster speeds than reported in this paper. The NNMcont algorithm used a graphical user interface which plotted updated frequency vs. energy solutions at each step which was not easily disabled. Since this likely required non-negligible computational time, the MMC algorithm was programmed to also plot updated values at each step so that this added time cost was equivalent for both algorithms. The error computation using ode45 was not included in the reported computational times.

FREQUENCY RESULTS

The fundamental frequency of the NNM that extends from the first LNM of the system is shown in Fig. 1.

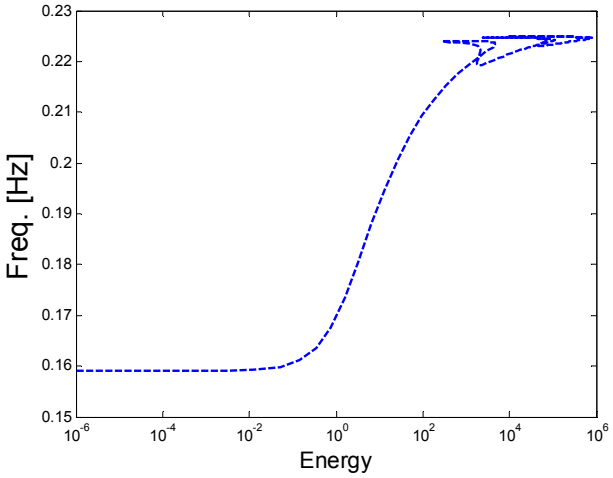

Fig. 1: Frequency vs. Energy Curve Extending from First Linear Normal Mode.

The results shown in Fig. 1 were generated using NNMcont with 300 integration steps (N_{int}=300) and the shooting periods set to half of the full period since the solution is symmetric. MMC delivered nearly identical results for frequency along the backbone until the solution reached the internal resonances. Within the internal resonances, the number of harmonics kept within the MMC algorithm (N) greatly affected the NNM computed by MMC. The effects of the number of harmonics retained in the displacement function were studied for the 2-DOF system by varying *N* from 5 to 30. The resulting frequency vs. energy curves were determined and plotted against the results computed by NNMcont. NNMcont always stays on the branch continued directly from the LNM since a departure would require a large change in the period, and is therefore a benchmark to determine when MMC has left this main branch.

The results show that significant differences begin to occur in the vicinity of the first three internal resonances. It is important to note that the solutions exist at discrete points, so some of the differences that are seen, such as those when N=25, are simply due to the MMC algorithm jumping to a new point. In other cases, such as when N=15, MMC finds a new branch that was missed by NNMcont, and then there are also cases, such as for N=5, where the solution from MMC does not seem to be on a valid branch of NNMs.

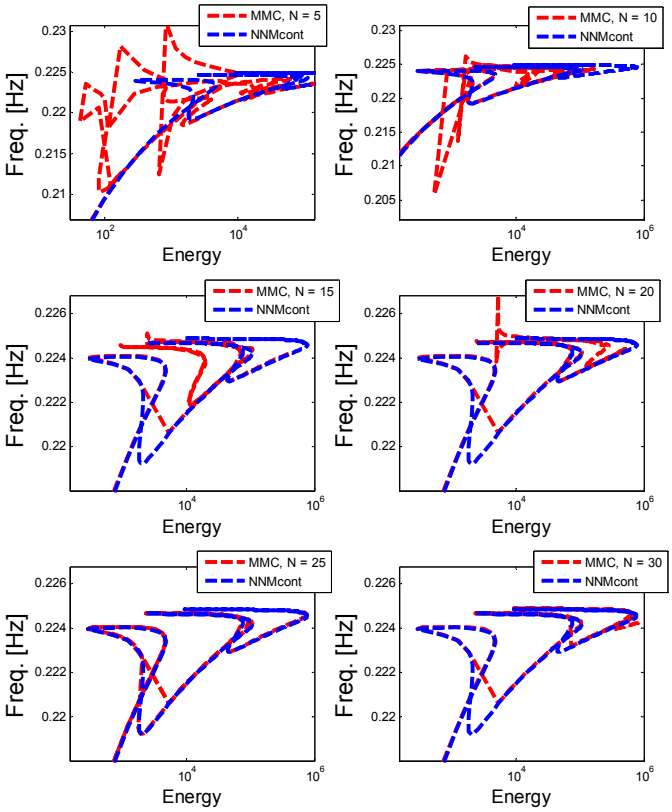

Fig. 2: Frequency vs. Energy Solutions Computed by the MMC and NNMcont algorithms. The number of harmonics retained in MMC was varied from *N*=5 to *N*=30.

As more harmonics are kept in the MMC algorithm, MMC becomes more likely to stay on the main frequency branch. With 30 harmonics kept, MMC matches the NNMcont solution through all three internal resonances. When the number of harmonics is reduced to 25, MMC still computes the same values as NNMcont, but it follows a strange path wrapping around the first internal resonance twice. With N = 20, MMC computes the same solution as NNMcont within the first two resonances, but jumps either to a different branch or to an erroneous solution before the third. Next, when N = 15, MMC appears to find a valid branch of a period-lengthened solution. Period-lengthened solutions will be discussed in more detail in the following sections.

When only 5 or 10 harmonics are included in the MMC algorithm, the results greatly differed from NNMcont. Two factors are thought to contribute to the differences that have been observed: the step-sizes used and the algorithm's radius of convergence.

As was shown previously in Table 1, both algorithms used similar values for the maximum and minimum step size, however MMC also placed constraints on the ratio R that NNMcont did not include. Although similar parameters were used for both algorithms, when MMC used N =5 and N = 10 the two algorithms ended up having very different step sizes.

For MMC the desired number of iterations was set higher at N^*=10 iterations, because otherwise it tended to take much smaller steps than NNMcont making the comparison difficult. With NNMcont N^*=3 was used and this seemed to give a good balance between the number of steps and the computational effort.

Step size control alone does not fully explain why large differences can be seen in the region of internal resonances for low values of N in MMC. For both algorithms, if too large of a step is taken such that the prediction location is far off from a NNM, they will not converge and the step size will be cut. However, MMC is converging at all the locations shown in Fig. 2. This is due to the nature of the MMC algorithm and the way in which it measures error. In MMC, error is measured by:

$$\varepsilon_k^{MMC} = \frac{\left\| \ddot{x}_k - f_k \right\|}{\left\| x_k \right\| (N\omega)^2 (n\sqrt{M})} \tag{16}$$

k = Collocation point index

n = Number of degrees of freedom

M = Number of collocation points

The error of the solution is then taken as the maximum of the error over all collocation points. In essence, MMC needs to only satisfy the equation of motion at each collocation point. In other words, it is $\ddot{x}$ that the error is measuring for MMC, which is very different than the error measurement ε^H for NNMCont in (15). It seems that this error measurement causes MMC to accept solutions that NNMcont would deem invalid. In fact, it is likely that at all the locations where MMC disagrees with NNMcont, MMC has found period-lengthened solutions. This occurs when MMC's computed initial displacement and velocity conditions actually correspond to a period that is an integer multiple of the computed period. For instance, the error ε^H measured by (15) at the computed period for each algorithm is shown in Fig. 3.

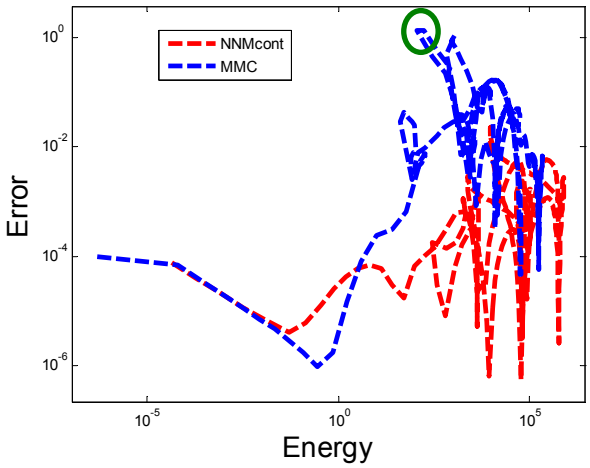

Fig. 3: Error of solutions for N=5 in MMC and N_{int}=75 in NNMcont.

The maximum error in MMC is 1.36 (136%), an unacceptably high value. However, it is difficult to interpret these error measurements because it could be legitimate error in the MMC algorithm if N=5 harmonics are simply not enough to accurately capture the dynamics of the system or it could also be that MMC has found a period lengthened solution. This can be checked by measuring error at integer multiples of the computed period. For example, the initial conditions computed at the location indicated with a green circle in Fig. 3 were integrated to integer multiples of the computed period using ode45. Then the error was computed and the result is shown in Fig. 4.

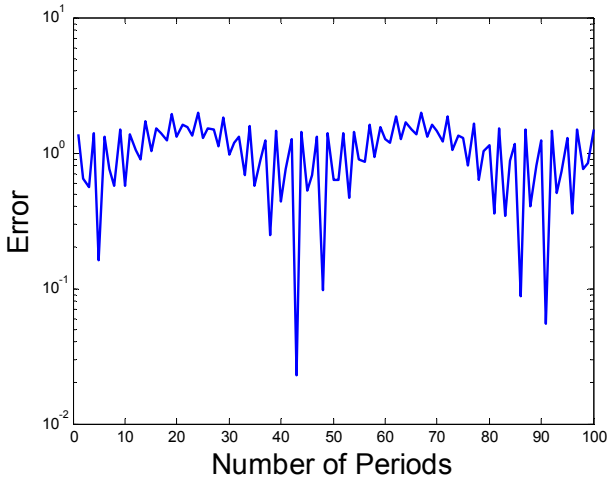

Fig. 4: Error of the MMC solution with N = 5 measured by integrating the EOM from the computed initial conditions at MMC's location indicated with a green circle in Fig. 3

When the NNM is integrated to 43 periods, a massive dip in error occurs, down to ε^H=0.0227 occurs. This indicates that the computed initial conditions may in fact correspond to a

period of 43 times the computed period. The ability of MMC to compute these other solutions can be seen as a benefit or fault of the algorithm. These period-lengthened solutions are equally valid and are physically realistic, and are difficult to compute with NNMcont or other algorithms (because starting guesses are difficult to obtain and, for NNMcont, because the time span over which the system must be integrated is long). However, other than integrating the equations of motion with ode45, which would be very expensive for large systems, there is not currently a way to determine whether the solutions that MMC has found are period-lengthened or erroneous.

TIMING AND ERROR COMPARISON

Since MMC is capable of finding different solutions than NNMcont, and these different solutions may be interpreted as error when in fact they are simply period-lengthened solutions, timing and error analysis will be performed only within the regions where the two algorithms have computed the same solutions.

As shown in Fig 5, the region in which the two algorithms compute the same results depends on the value of N in MMC. Therefore, three different timing and error analyses were completed for the MMC algorithm. The computational time and error were computed for MMC for N = 5, N = 15, and N = 30 up through the locations where the MMC solution begins to diverge from NNMcont. These end points are shown in Figs. 5-6.

As shown Figs. 5-6, the comparison between NNMcont and MMC with N=5 is only performed up the backbone before the first internal resonance is reached. NNMcont and MMC with N=15 are compared just past the first internal resonance, and for N=30 the algorithms are compared through all three internal resonances. The error through these three regions was compared against NNMcont's solutions computed with three values of integration points (N_{int}): 75, 200, and 300. The resulting errors are shown in Fig. 7.

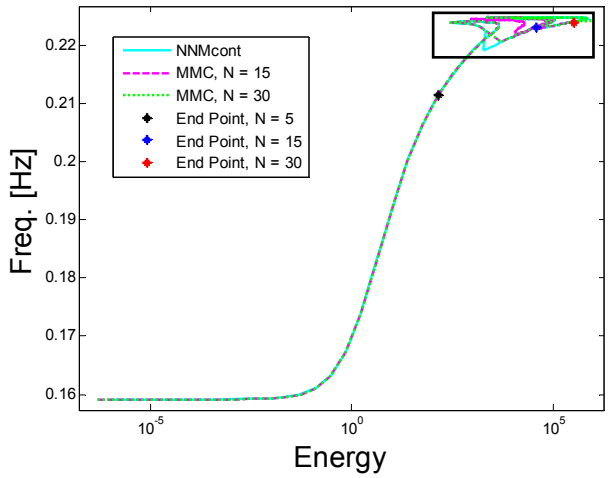

Fig. 5: Location of end points to which error and speed comparisons were made between MMC and NNMcont.

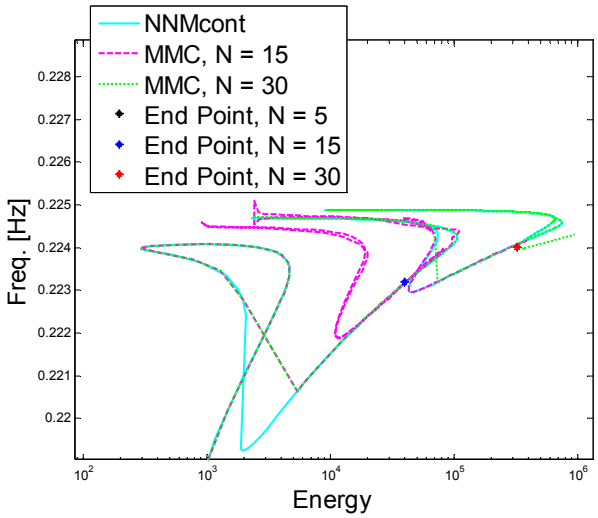

Fig. 6: Zoom view of region shown in Fig. 6.

The columns in Fig. 7 correspond to the same value of N in MMC, while the rows correspond to the same value of N_{int} in NNMcont. When N=5, the error throughout the entire region is quite low, with MMC giving a little more accurate results than NNMcont for N_{int}<300. The error increases for both algorithms starting near energy levels equal to 1, and the increase is more dramatic for MMC.

When N=15, again the MMC and NNMcont error curves follow the nearly identical paths at very low energy levels. However, the error in MMC continues to decrease past an energy level of 1 and up until the first internal resonance, while the error in NNMcont increases significantly.

Within the first internal resonance, error becomes chaotic and difficult to interpret. To compare the errors within the internal resonance region, the average error within the internal resonances was determined, and is shown in Table 2. Data for N=5 is not shown since these results were not analyzed through an internal resonance. The values in Table 2 for N=15 are the average error within the first internal resonance (excluding data before and after), and for N=30 the average error includes all three internal resonances.

Table 2: Average error in the region dominated by internal resonances

	N = 15	N = 30
MMC	6.70×10^{-4}	1.04×10^{-3}
NNMcont, N_{int} = 75	5.27×10^{-4}	1.73×10^{-3}
NNMcont, N_{int} = 200	7.23×10^{-5}	1.53×10^{-4}
NNMcont, N_{int} = 300	4.22×10^{-5}	5.93×10^{-5}

MMC's average error within the internal resonances for both N=15 and N=30 closely match the error for NNMcont with N_{int}=75. These error values are much higher than both

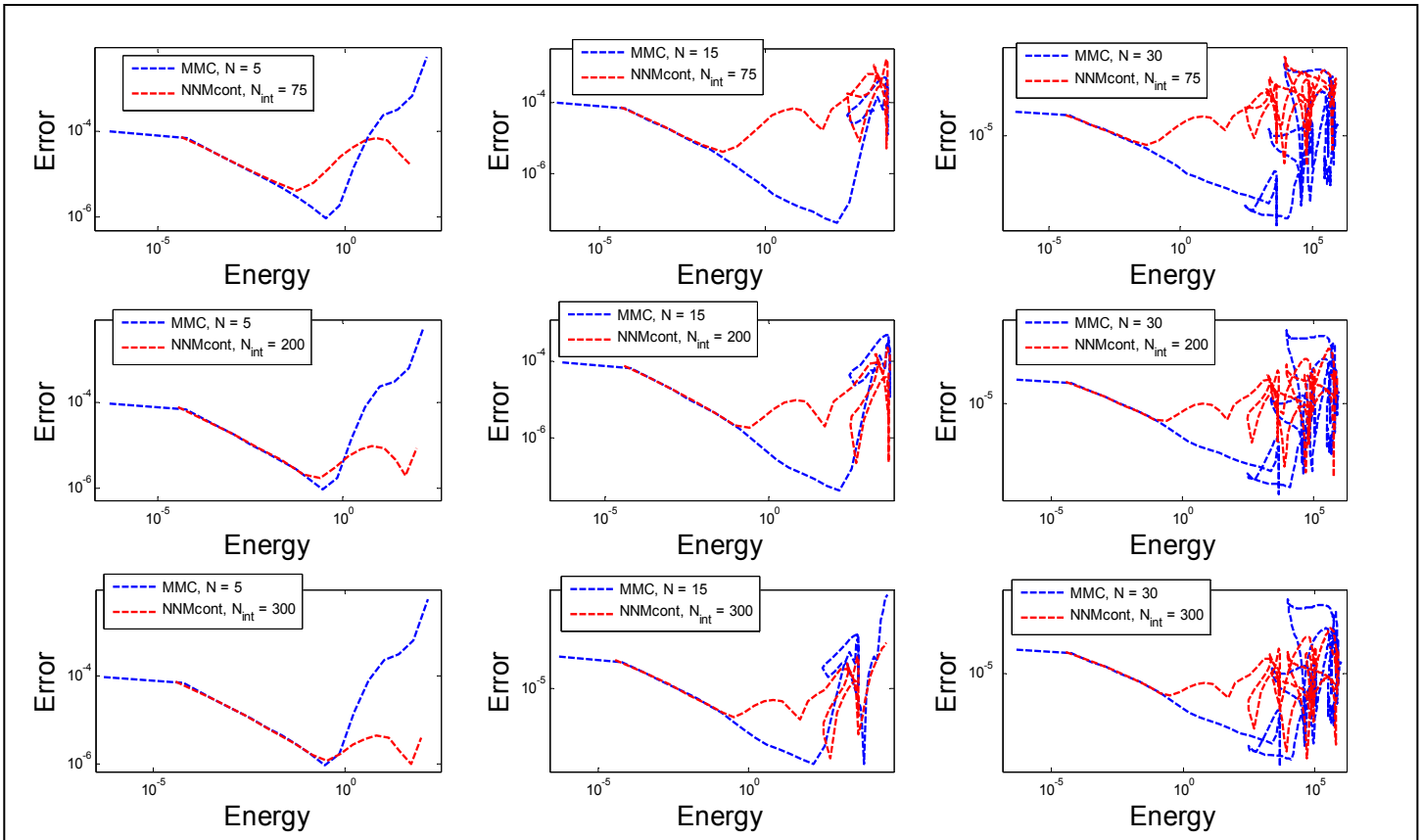

Fig. 7: Error Comparison between MMC and NNMcont in regions in which the same solutions were computed.

algorithms' errors previous to the internal resonances, but would likely be acceptable for many engineering applications.

The total computational time in seconds required to calculate the NNMs for both algorithms was determined using Matlab's timing features. Similar to the error comparison, a timing comparison between MMC and NNMcont has only been performed in regions where they computed the same solution. Again, values of N=5, 15, and 30 were used in MMC while values of N_{int}=75, 200, and 300 were used in NNMcont. The total time required for each algorithm to compute the NNMs is shown in Table 3.

Table 3: Total computational time [seconds] required for each algorithm to compute the solutions within regions indicated in Fig. 6.

Maximum Energy:	**Pt. 1**	**Pt. 2**	**Pt. 3**
MMC, N = 5	1.02	n/a	n/a
MMC, N = 15	-	6.59	n/a
MMC, N = 30	-	-	69.05
NNMcont, N_{int} = 75	5.60	58.43	1644.03
NNMcont, N_{int} = 200	7.90	74.25	1764.32
NNMcont, N_{int} = 300	9.67	81.93	2736.44

MMC provides large savings in computational time when compared to NNMcont. Regardless of the number of integration points used in NNMcont, MMC greatly outpaced it in computational time by at least one order of magnitude. Based on the structure of the algorithms, it would be assumed that the computational time would increase linearly with the number of solutions (N_s), i.e. $T_{CPU} = rN_s$.

The linear rate (r) states the average computational time in seconds required per solution. Comparing these linear rates for each algorithm is more revealing than the total computational time since the two algorithms required different numbers of solutions to reach the same point in the frequency vs. energy plot. The linear rates that correspond to the computational times in Table 3 are shown in Table 4.

Table 4: Computational time per solution [seconds per solution] for each algorithm.

Maximum Energy:	**Pt. 1**	**Pt. 2**	**Pt. 3**
MMC, N = 5	0.060	n/a	n/a
MMC, N = 15	-	0.101	n/a
MMC, N = 30	-	-	0.138
NNMcont, N_{int} = 75	0.350	0.749	3.288
NNMcont, N_{int} = 200	0.494	0.952	3.529
NNMcont, N_{int} = 300	0.604	1.050	5.473

As expected, more computational effort is required for MMC as N increases since the size of the system of equations solved for by MMC increases with N. As expected NNMcont's linear rate increases as more integration points are used, indicating the increase in time required to integrate the equations for each solution. The values of r for NNMcont increase across rows, despite the same number of integration points being used. This indicates that the solutions become more expensive at higher energy, most likely due to the fact that the continuation step size must be cut and thus many more iterations are performed per step.

As shown in Fig. 7 and Table 2, MMC and NNMcont have similar errors when only 75 integration points are used NNMcont, so the corresponding values of r may be the fairest way to compare the two algorithms. As shown in Table 4, the approximate linear rate of MMC is about one order of magnitude lower than NNMcont for N=5, 15, and 30.

As noted earlier, the average error values shown in Table 2 may be slightly misleading as the error for MMC and NNMcont are largely equal when N_{int}=200 and 300, except for a few locations of higher error in MMC. If these timing comparisons are also considered valid, than the computational savings of MMC are even greater, as the approximate linear rate for NNMcont with N_{int}=300 is approximately 1.5 times larger than for N_{int}=75. Although it is difficult to find an exact way to compare computational times between the two algorithms, it is clear that MMC does provide savings of about an order of magnitude when compared to NNMcont.

APPLICATION TO 10-DOF SYSTEM

A timing analysis was also performed on the 10 DOF non-linear model of a geometrically nonlinear beam. The beam used is the simply-supported model of one subomponent that was described in [15, 16]. A 10 DOF model for the beam was created from a larger finite element model using the Implicit Condensation and Expansion (ICE) method [17, 18]. The MMC and NNMcont algorithms were used to compute the first three NNMs of the 10 DOF system. For each mode, MMC and NNMcont computed solutions to the same maximum energy. The time required to determine the NNMs is shown in Table 5.

Table 5: Total computational time [seconds] required for each algorithm to compute the NNMs of a 10 DOF system

	Mode 1	Mode 2	Mode 3
MMC	101.0	216.2	215.0
NNMcont	79.7	242.8	49.3

Surprisingly, MMC required more time to compute the first three NNMs for the 10 DOF system than NNMcont. The results were interrogated and the authors found that the MMC algorithm was taking much smaller step sizes than NNMcont. Because the MMC algorithm has a different radius of convergence than NNMcont, smaller step sizes had to be used to keep MMC from jumping onto other branches of solutions. More work is needed to optimize the step size control strategy for MMC so that it does not take such unnecessarily small steps. In any event, a perhaps the time per step is a more fair comparison, and this is shown in Table 6. (On the other hand, it could be that the topology of the problem that MMC solves is less smooth so that smaller step sizes will always be required; further work is needed to investigate this.)

Table 6: Computational Time [seconds] Per Number of Solutions for each Algorithm to Compute the NNMs of a 10 DOF System

	Mode 1	Mode 2	Mode 3
MMC	0.3	1.8	1.4
NNMcont	1.4	3.9	2.9

Despite taking longer total time to compute solutions to the same energy level, MMC typically required less than half the computational time per step. Comparing the rates shown in Table 6 with those shown in Table 4, it can be seen that the computational time per solution for MMC increased by about an order of magnitude. This is larger than expected, since early results [7, 14] suggested that the increase was sub-linear with the number of degrees of freedom.

CONCLUSIONS

An analysis of the MMC algorithm's performance has been completed by studying the computed NNMs, error in the NNMs, and the computational time for a 2 degree of freedom system. The results were compared to those computed by NNMcont [10]. Additionally, a brief timing comparison between the two algorithms for a 10 DOF system was performed.

The results showed that the number of harmonics used in MMC had a dramatic effect on the solution that MMC generates. When low numbers of harmonics were used, MMC seemed to converge to valid solutions, however at higher energy levels they often seemed to be period-lengthened solutions. In applications where the period lengthened solutions are of interest this could be seen as a strength of the algorithm; in other applications this would be a nuisance.

It proved more difficult to use MMC in a continuation framework than was expected, and this was thought to be caused by MMC's larger radius of convergence. While NNMcont tends to fail to converge if the initial guess is poor (and hence the stepsize is cut and a solution is attempted at a closer point), MMC often converged to a new solution that may be far from the initial guess. As a result, it was more challenging to keep MMC on the same solution branch. Additionally, there are differences in the way error is measured by MMC since it operates on accelerations and this introduces additional complications.

In spite of these challenges, the computational time and error of the two algorithms were compared over regions where they behaved similarly. In these comparisons, MMC and NNMcont tended to have similar errors on average (with the settings used here), although the maximum error for MMC was usually larger than that for NNMcont. For the 2 DOF system, MMC offered large computational time savings. In fact MMC usually took about an order of magnitude less time to compute the same solutions as NNMcont, even when errors were similar. For the 10 DOF system, MMC required more total computational time, but required only about half as much time per step. These results are not fully understood yet. Computation time for numerical integration is known to scale with N_{DOF}^2 or more, and since MMC had been previously found to scale sub-linearly with N_{DOF} [14] and so one would expect to see the performance gap between MMC and NNMcont increase rather than decreasing. The Matlab implementation of MMC will be improved and these issues will be explored further in future works.

REFERENCES

[1] G. Kerschen, M. Peeters, J. C. Golinval, and A. F. Vakakis, "Nonlinear normal modes. Part I. A useful framework for the structural dynamicist," *Mechanical Systems and Signal Processing,* vol. 23, pp. 170-94, 2009.

[2] R. J. Kuether, L. Renson, T. Detroux, C. Grappasonni, G. Kerschen, and M. S. Allen, "Nonlinear Normal Modes, Modal Interactions and Isolated Resonance Curves," *Journal of Sound and Vibration,* vol. Submitted Nov. 2014, 2015.

[3] M. Peeters, G. Kerschen, and J. C. Golinval, "Dynamic testing of nonlinear vibrating structures using nonlinear normal modes," *Journal of Sound and Vibration,* vol. 330, pp. 486-509, 2011.

[4] H. A. Ardeh and M. S. Allen, "Instantaneous Center Manifolds and Nonlinear Modes of Vibration," presented at the ASME 2012 International Design Engineering Technical Conferences, Chicago, IL, 2012.

[5] H. A. Ardeh and M. S. Allen, "Investigating Cases of Jump Phenomenon in a Nonlinear Oscillatory System," presented at the 31st International Modal Analysis Conference (IMAC XXXI), Garden Grove, CA, 2013.

[6] S. W. Shaw and C. Pierre, "Normal modes for non-linear vibratory systems," *Journal of Sound and Vibration,* vol. 164, pp. 85-124, 1993.

[7] H. A. Ardeh, "Geometrical Theory of Nonlinear Modal Analysis," PhD, Mechanical Engineering, University of Wisconsin-Madison, Madison, WI, 2014.

[8] R. M. Rosenberg, "Normal modes of nonlinear dual-mode systems," *Journal of Applied Mechanics,* vol. 27, pp. 263–268, 1960.

[9] A. F. Vakakis, "Non-linear normal modes (NNMs) and their applications in vibration theory: an overview," *Mechanical Systems and Signal Processing,* vol. 11, pp. 3-22, 1997.

[10] M. Peeters, R. Viguie, G. Serandour, G. Kerschen, and J. C. Golinval, "Nonlinear normal modes, part II: toward a practical computation using numerical continuation techniques," *Mechanical Systems and Signal Processing,* vol. 23, pp. 195-216, 2009.

[11] O. Bruls and P. Eberhard, "Sensitivity analysis for dynamic mechanical systems with finite rotations," *International Journal for Numerical Methods in Engineering,* vol. 74, pp. 1897-1927, 2008.

[12] G. Kerschen, M. Peeters, J. C. Golinval, and C. Stephan, "Nonlinear modal analysis of a full-scale aircraft," *AIAA Journal of Aircraft,* vol. 50, pp. 1409-1419, 2013.

[13] T. M. Cameron and J. H. Griffin, "Alternating frequency/time domain method for calculating the steady-state response of nonlinear dynamic systems," *Journal of Applied Mechanics, Transactions ASME,* vol. 56, pp. 149-154, 1989.

[14] H. A. Ardeh and M. S. Allen, "Multi-harmonic Multiple-point Collocation: An Iterative Method for Finding Periodic Orbits of Strongly Nonlinear Oscillators," *Journal of Computational and Nonlinear Dynamics,* vol. (Submitted Dec. 2013), 2014.

[15] R. J. Kuether and M. S. Allen, "Nonlinear Modal Substructuring of Systems with Geometric Nonlinearities " presented at the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston, MA, 2013.

[16] R. J. Kuether and M. S. Allen, "Craig-Bampton Substructuring for Geometrically Nonlinear Subcomponents," presented at the 32nd International Modal Analysis Conference (IMAC XXXII), Orlando, Florida, 2014.

[17] R. W. Gordon and J. J. Hollkamp, "Reduced-order Models for Acoustic Response Prediction," Air Force Research Laboratory, AFRL-RB-WP-TR-2011-3040, Dayton, OH2011.

[18] J. J. Hollkamp and R. W. Gordon, "Reduced-order models for nonlinear response prediction: Implicit condensation and expansion," *Journal of Sound and Vibration,* vol. 318, pp. 1139-1153, 2008.