



2011-01-28

A 3-D Chain Algorithm with Pseudo-rigid-body Model Elements

Robert Parley Chase

Brigham Young University - Provo

Robert H. Todd

See next page for additional authors

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Mechanical Engineering Commons](#)

Original Publication Citation

Chase, R.P. Jr., Todd, R. H., Howell, L.L., Magleby, S.P., "A 3-D Chain Algorithm with Pseudo-rigid-body Model Elements," *Mechanics Based Design of Structures and Machines*, Vol. 39, No. 1, pp 142–156, 2011.

BYU ScholarsArchive Citation

Chase, Robert Parley; Todd, Robert H.; Howell, Larry L.; and Magleby, Spencer P., "A 3-D Chain Algorithm with Pseudo-rigid-body Model Elements" (2011). *All Faculty Publications*. 1552.

<https://scholarsarchive.byu.edu/facpub/1552>

Authors

Robert Parley Chase, Robert H. Todd, Larry L. Howell, and Spencer P. Magleby

A 3-D CHAIN ALGORITHM WITH PSEUDO-RIGID-BODY MODEL ELEMENTS

Running Head: 3-D Chain Algorithm with PRBM Elements

Robert P. Chase Jr., Robert H. Todd, Larry L. Howell, Spencer P. Magleby

Mechanical Engineering Department, Brigham Young University, Provo, UT 84602

ABSTRACT

A chain algorithm element is created from pseudo-rigid-body segments and used in a chain calculation that accurately predicts the force deflection relationship of beams with large 3-D deflections. Each chain element is made up of three pseudo-rigid-body models superimposed on each other acting orthogonally in relation to each other. The chain algorithm can predict large displacements and the force-deflection relationship of lateral torsional buckled beams significantly faster than the finite element method. This approach is not intended to compete with finite element analysis, but rather is a supplement tool that may prove particularly useful in the early phases of design when many analysis iterations are required. The 3-D chain algorithm is demonstrated and compared to the finite element analysis for the nonlinear large-deflection, post-buckling path of a flexible beam undergoing lateral-torsional buckling.

I. INTRODUCTION

During design, non-planar compliant mechanisms are often modeled in two dimensions using rectangular or radial coordinates. These 2-D models have generally been used because they are easy to set up, visualize, and understand. 3-D modeling is possible by using finite element methods, but the combination of nonlinearities and 3-D geometry or optimization makes it difficult to use in designs where many analysis iterations are required. The availability of design-friendly, 3-D models could stimulate the development of a variety of useful 3-D mechanisms. This paper introduces a 3-D chain algorithm with pseudo-rigid-body model elements that accurately predicts 3-D large deflections and the force-deflection relationship of beams where the cross-sectional product of inertia is zero.

Many 3-D mechanisms do not lend themselves to single link pseudo-rigid-body models that act in one plane. Lateral torsional buckling, illustrated in Figure 1, is an example of a 3-D deflection that cannot be described with a 2-D model.

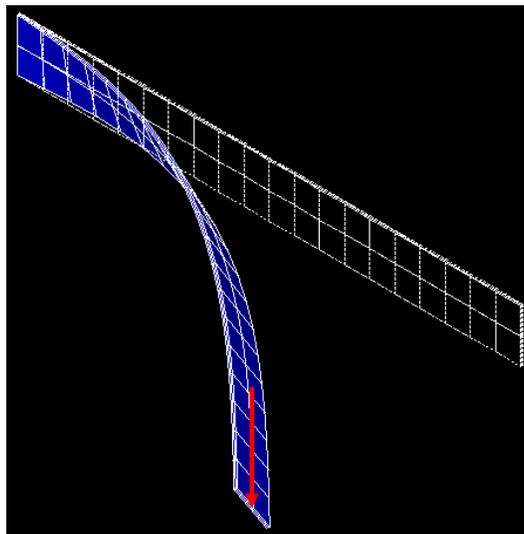


Figure 1: Lateral torsional buckling

A common mechanism that shows the usefulness of this phenomenon is a bistable hair clip, illustrated in Figure 2. This mechanism is made of two cantilever beams fixed at one end and pinned together with a rivet at the other. In order to pin the two beams together, they first must be buckled. Once pinned, they will flip between two stable buckled positions when an adequate load is applied to the pinned end.

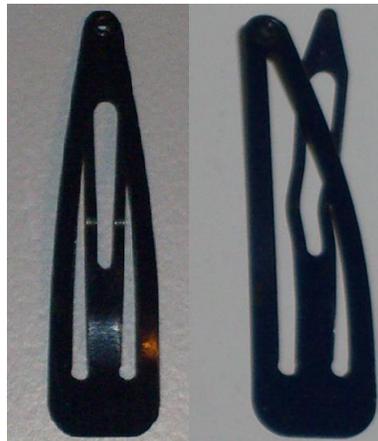


Figure 2: Bistable hair clip

Research on lateral torsional buckling has focused on small deflections [1, 2] leading into the undesirable [3] buckling initiation process and the study of buckling resistances [4]. However, large deflections of lateral torsional buckling may prove useful in mechanism design [5].

Pseudo-rigid-body models provide a simple method of analyzing systems that undergo large, nonlinear deflections. Pseudo-rigid-body models have been used with great success in predicting force-deflection characteristics of compliant mechanisms in two dimensions [6]. Currently, pseudo-rigid-body models have been defined for many two-dimensional cases where it is assumed that all deflections remain in a single plane. A pseudo-rigid-body model enables engineers to replace complex models of flexible members with a much simpler model of two-dimensional rigid links and springs.

II. THE CHAIN ALGORITHM

Beams undergoing complex loading conditions often do not lend themselves to a pseudo-rigid-body model with a single link. A chain algorithm is another technique used to solve large deflection problems. The chain algorithm requires discretization of the object being modeled into beam elements and each element is analyzed in succession [6]. In the past, 2-D and 3-D [7] chain algorithms have depended on small deflection formulas to describe the displacements for each small beam segment. This theory allows different parts of a beam, along its length, to undergo different local displacement magnitudes.

The chain calculation procedure is an alternative to finite element analysis for nonlinear beam-element applications. A chain algorithm employs similar elements and stiffness matrix theory as traditional finite elements, but a different technique is used to solve the resulting equations [6].

Each beam element is calculated in succession as a cantilever beam. Once an element's deflections are solved, all succeeding elements or those closer to the beam tip are rotated and lined up starting at the deflected element's tip. Figure 3 illustrates how a single element is analyzed.

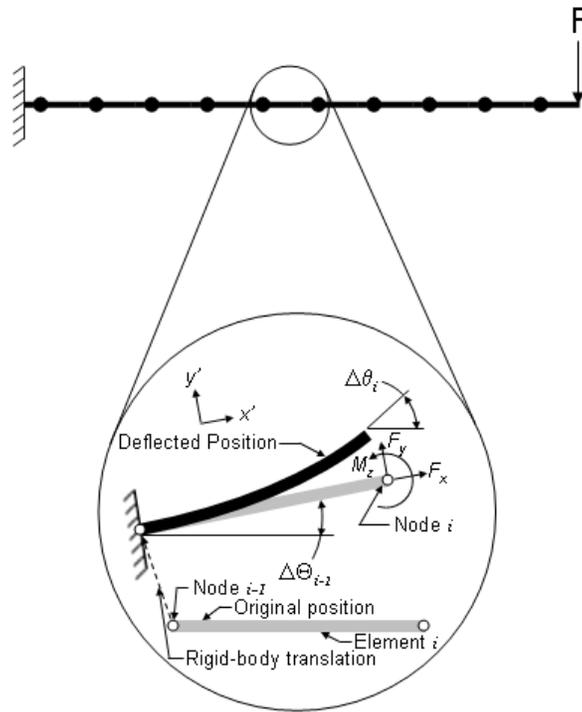


Figure 3: Analysis of a single chain algorithm element

Starting at the beam root, the deflection path of each element is calculated. At element i , the beam has undergone rigid-body translation because of the deflection of previous elements. It has also undergone rigid-body rotation $\Delta\Theta_{i-1}$ because of the elastic deflection caused by previous elements. The forces are broken down into the element's local coordinate system and any

moment is also added to the beam tip. Load direction remains constant during deformation, thus there are no “follower” forces [8]. The loading problem is then solved at the element level, causing a rotation $\Delta\theta_i$ at the beam tip node i .

Once an element has been analyzed, forces and moments on elements down the line need to be recalculated with the latest position and rotation information. Iterating through the succession of elements can be repeated until the desired accuracy is achieved. Load increments can be adjusted for better efficiency. Load increments also need to be small enough to allow the chain algorithm to converge instead of becoming unstable. In general, using a large number of load increments results in a smaller number of required iterations, and vice versa [6].

III. PSEUDO-RIGID-BODY MODEL CHAIN

The idea of a pseudo-rigid-body model chain algorithm for 2-D problems was introduced by Pauly and Midha [9, 10]. It relies on pseudo-rigid-body elements while implementing the chain algorithm in the same way as previously discussed. “Such a ‘pseudo-rigid-body model chain’ would possess dual advantages of expediency of modeling through the use of pseudo-rigid-body representations of compliant segments, and the inherent flexibility of the chain algorithm to geometry and load boundary conditions” [9].

Two limitations to using pseudo-rigid-body modeling techniques are noted:

1. “In its current form, suitable modifications to existing conventions in the pseudo-rigid-body modeling technique are required in order to apply the parametric relationships developed...to determine the deformation kinematics of compliant segments for varied end force loading situations” [9].
2. “Presently, generalized pseudo-rigid-body models of reasonable accuracy are unavailable for compliant members with combined end force and moment load boundary conditions” [9].

The method of superposition was introduced by Pauly and Midha as a way to remedy these limitations [9]. The tip force and tip moment load pseudo-rigid-body models would be calculated separately, and then their displacement results would be superposed on each other. This method provided reasonable approximations, but one limitation remained in that error was introduced into the chain approximation because of incompatibilities in superimposing the pseudo-rigid-body models on each other [9].

Pauly and Midha, in a second paper, proposed using a “rudimentary equivalent pseudo-rigid-body model to represent compliant segments with combined load boundary conditions in the pseudo-rigid-body model chain algorithm” [10]. This method finds an equivalent pseudo-rigid-body model by adding the displacement of the two pseudo-rigid-body models as

$$\Theta_i^e = \Theta_i^f + \Theta_i^m \quad (1)$$

where Θ_i^f represents the angular displacement in the force tip loaded pseudo-rigid-body model, Θ_i^m represents the angular displacement in the end-moment loaded pseudo-rigid-body model, and Θ_i^e represents the equivalent angular displacement. The subscript i denotes the equivalent model's location along the chain algorithm. The equivalent characteristic radius factor γ_i^e is set to equal the force model's characteristic radius factor γ_i^f with negligible error. The equivalent spring stiffness coefficient k^e , is found using equation 2:

$$\frac{1}{k^e} = \left(\frac{1}{k_1} + \frac{1}{k_2} \right) \quad (2)$$

where k_1 and k_2 are the spring constants for the two pseudo-rigid-body models being combined. The equivalent pseudo-rigid-body model remains incomplete because a suitable equivalent parametric angle coefficient has yet to be discovered.

IV. MODELING ORTHOGONAL DEFLECTIONS

This paper introduces a method of using pseudo-rigid body model elements in a 3-D chain algorithm that avoids the difficulty caused by the equivalent parametric angle coefficient by only superposing pseudo-rigid-body models on each other that act orthogonal to each other. Figure 4 illustrates three types of deflections with a rectangular beam that, when converted to a pseudo-rigid-body model, act orthogonally. The first is twist along the beam length. The second is bending in the slender direction. The third is bending in the stiff direction.

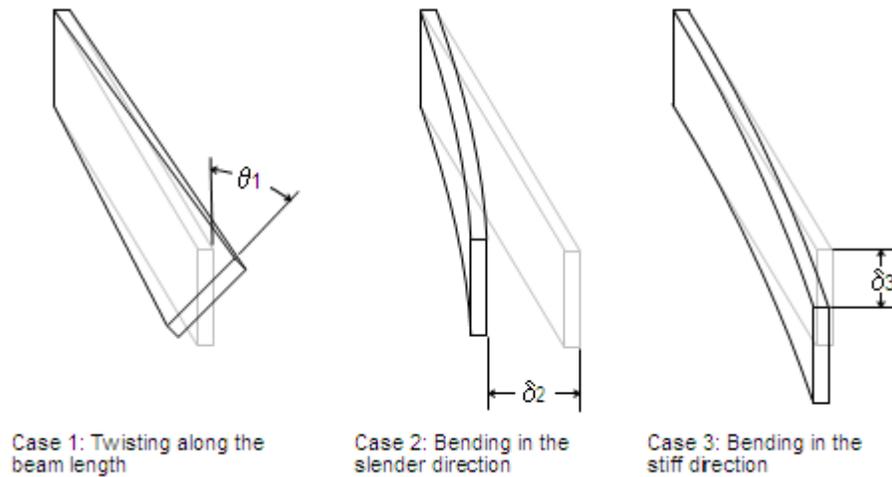


Figure 4: Three orthogonal deflection components

V. DEVELOPMENT OF A 3-D PSEUDO-RIGID-BODY-MODEL

The three beam deflections described in Figure 4 can be modeled with simple two-dimensional pseudo-rigid-body models. Case 2 is the beam bending in the slender direction and case 3 is the beam bending in the stiff direction. These are the same phenomena with different stiffnesses.

Now consider the classic two-dimensional cantilever beam, shown in Figure 5.

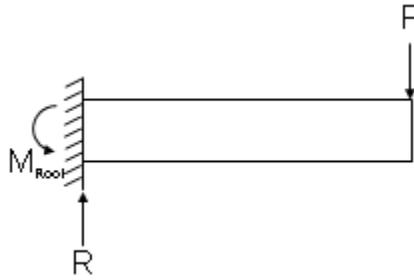


Figure 5: The classic cantilever beam

One end is fixed while the other end is free. The free end has a vertical force applied to it. The reaction caused by the applied force, F , is an opposing force and moment located at the fixed end of the beam. The reaction force is felt as a constant shear force (V) through the beam length (see Figure 6).

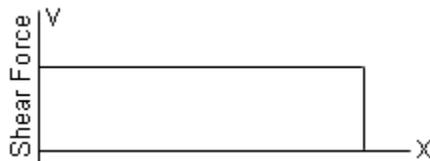


Figure 6: Shear force diagram

As is done in classical beam theory, the shear force is neglected because the beam deflection caused by shear is very small compared to the beam deflection caused by the bending moment in the beam [11]. The moment at the beam tip is zero and increases linearly along the beam length, reaching a maximum magnitude at the beam's fixed end (see Figure 7).

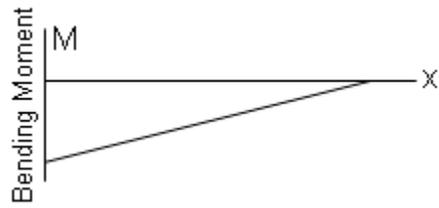


Figure 7: Moment load diagram

The beam is discretized into small elements. Consider an arbitrary element and its loads as illustrated in Figure 8.

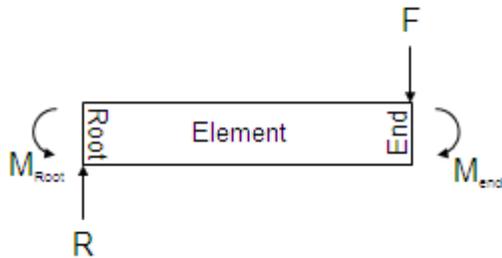
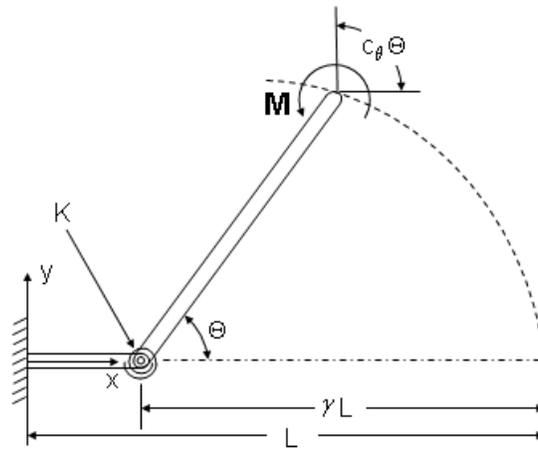


Figure 8: Small arbitrary element

At the element end is the applied load, F , and a moment, M_{end} , representing the applied force multiplied by the distance between the applied force and the element end. At the element root is

a reaction force, R , and a moment, M_{root} , representing the force applied at the beam tip multiplied by the distance from the beam tip to the element root.

The beam segment is now shrunk in length to an infinitesimally small length. The moments also converge together. A pseudo-rigid-body model for a cantilever beam with a moment at the free end is now applied to the infinitely short beam as shown in Figure 9.



Characteristic Radius Factor: $\gamma = 0.7346$

Parametric Angle Coefficient: $c_{\theta} = 1.5164$

Spring Constant: $K = \gamma K_{\theta} EI/L$

Stiffness Coefficient: $K_{\theta} = 2.0643$

Figure 9: Components of the end moment loaded pseudo-rigid-body model

A torsional spring representing the deflection of the beam segment is placed near the beam segment root. Its distance from the element tip is 0.7346 multiplied by the element length [6]. As

the linkage deflects from the applied moment, the beam tip angle is represented by the linkage angle Θ , multiplied by, c_θ , the parametric angle coefficient. The torsional spring constant for the fixed-free beam shown in Figure 9 is K . This pseudo-rigid-body model approximates a pure moment in a two-dimensional cantilever beam with less than 0.5% error over a range of 124.4 degrees from the undeflected position. [6]

The beam deflection described in Case 1 is twisted around its centroidal axis. The angular deflection for this element is [12]

$$\theta = \frac{TL}{KG} \quad (3)$$

where T is the torque, L is the length of the beam, K is the torsional spring constant and G is the shear modulus. The torsional spring constant K for this element is [6]

$$K = bh^3 \left[\frac{1}{3} - 0.21 \frac{h}{b} \left(1 - \frac{h^4}{12b^4} \right) \right], \text{ where } b \geq h \quad (4)$$

Here, b represents the height of the beam and h represents its width. This spring can be placed anywhere along the length of the beam segment. For convenience it will be placed at the same location as the two torsional springs previously discussed. The error associated with the torsional spring constant K is no greater than 4% [12].

These three 2-D pseudo-rigid-body models can now be superimposed as illustrated in Figure 10. Each element contains the three models with the three springs placed at exactly the same point. The three springs are positioned orthogonal to each other so that if a torque were applied in the plane of one spring, it would not influence the other two springs. The spring characterizing bending in the stiffer direction of the beam corresponds to the Y-direction. The spring characterizing bending in the slender direction corresponds to the Z-direction. The final spring describes axial torsion along the X-axis.

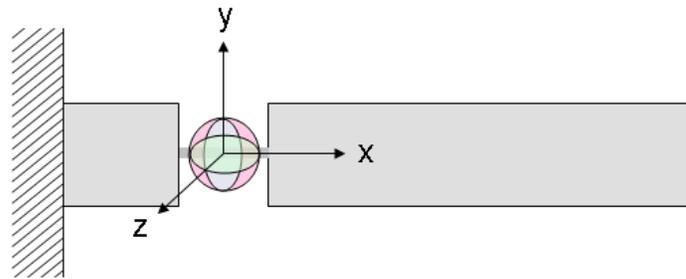


Figure 10: Three models superimposed on each other

Before combining the new 3-D pseudo-rigid-body model linkage with others into a chain, two differences between this element and 2-D chain elements should be pointed out. First, each node of the 3-D pseudo-rigid-body linkage will have an X, Y, and Z displacement component. Also, where a two-dimensional chain of elements would require a single axis of rotation with one spring, 3-D lateral torsional buckling requires three springs with three axes. The 3-D linkage is represented with vectors, and the spring axes are represented as a group by a local coordinate system. When the element rotates around one of the three spring axes characterized by a unit

vector, the other two spring vectors must rotate around the first vector in the same direction and with the same magnitude. This can be accomplished with the following formula [13], which rotates a vector around another vector or axis:

$$\vec{v}' = \vec{v} \cos \varphi + (1 - \cos \varphi)(\vec{v} \cdot \hat{w})\hat{w} - \vec{v} \times \hat{w} \sin \varphi \quad (5)$$

where the vector v is rotated counterclockwise around the axis w by an angle φ . The length of the vector is preserved.

The second difference between this element and 2-D chain elements is that the 3-D chain cantilever is indeterminate. In a 2-D chain, each segment is attached to a node with one axis of rotation. As the segment rotates along that axis, a circle is drawn with a radius equal to the length of the segment. In the 3-D chain cantilever, only two of the three degrees of freedom are required for the free segment to touch any point on a sphere with a radius equal to the length of the free segment. This indeterminacy is overcome by analyzing each of the element's orthogonal rotations in succession. First, the second and third degrees of freedom are held rigid and the element is analyzed and rotated around the first degree of freedom. Then the first and third degrees of freedom are held rigid while the second degree of freedom is analyzed and rotated. Finally, the first and second degrees of freedom are held rigid and the element is rotated around its third degree of freedom. Although the overall deflection is large, each element is small rotates only a small amount, so each orthogonal rotation can be added together like a vector with negligible error.

Now the 3-D chain cantilevers can be placed end to end to form a chain as illustrated in Figure 11.

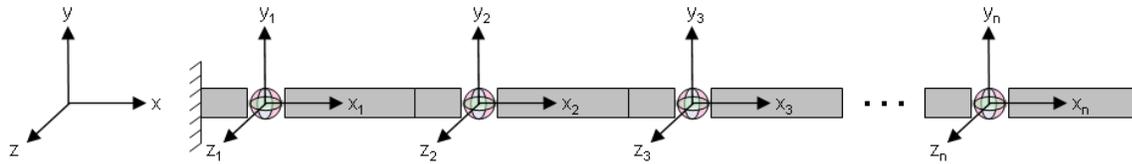


Figure 11: 3-D chain of elements

Every node will be placed in 3-D space with respect to the global X-Y-Z coordinate system.

Each node will have its own local coordinate system starting with $X_1-Y_1-Z_1$ and ending with $X_n-Y_n-Z_n$ where n represents the last node in the final 3-D chain cantilever.

As an example, the 3-D chain algorithm will be used to analyze a cantilever beam with a single tip load as illustrated in Figure 12.

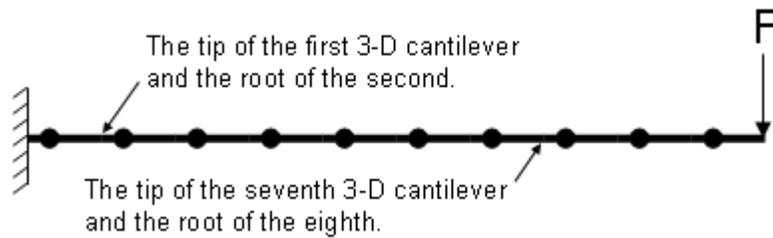


Figure 12: 3-D chain algorithm scenario

Before the moments can be found, the location of each 3-D element tip and the location of the acting force must be calculated. At the beginning, the initial orientation of each element will be defined and then each element will be aligned root to tip down the chain. The equations used to find the location of each beam tip are

$$\vec{P}_i = \begin{bmatrix} P_{x_i} \\ P_{y_i} \\ P_{z_i} \end{bmatrix} = \begin{bmatrix} P_{x_{i-1}} + ux_{x_{i-1}} S_L (1-0.7346) + ux_{x_i} S_L (0.7346) \\ P_{y_{i-1}} + ux_{y_{i-1}} S_L (1-0.7346) + ux_{y_i} S_L (0.7346) \\ P_{z_{i-1}} + ux_{z_{i-1}} S_L (1-0.7346) + ux_{z_i} S_L (0.7346) \end{bmatrix} \quad (6)$$

$$\vec{ux}_i = \begin{bmatrix} ux_{x_i} \\ ux_{y_i} \\ ux_{z_i} \end{bmatrix} \quad (7)$$

where \vec{P}_i represents the current location of each beam tip, and \vec{ux}_i is a unit vector and part of the local coordinate system for each node that points in the axial direction of each 3-D cantilevered beam segment. It should be noted that \vec{ux}_0 represents a unit vector in the global coordinate system and is positioned at

$$\vec{P}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

S_L in equation (6) is multiplied by each unit vector to give elements their true lengths. Because each element has a joint, two vectors are needed to describe the orientation of the element tip. The unit vector of the previous element multiplied by $S_L(1-0.7346)$ represents the orientation of the root end of the current element as shown in Figure 10. S_L multiplied by 0.7346 gives the orientation of the tip end. By adding up all the element vectors, the position of the overall beam tip force is also known. It is located on the tip of the 3-D element \vec{P}_i (see Figure 13).

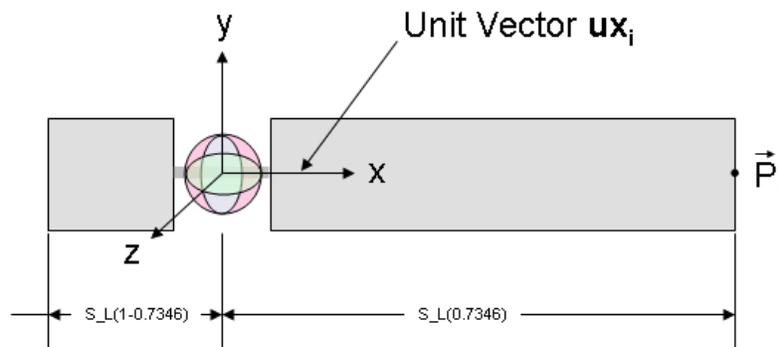


Figure 13: 3-D element coordinate system

The moment can now be calculated for node 1 using the moment equation:

$$\vec{M} = \vec{P}_1 \times \vec{F} \quad (9)$$

The moment \vec{M} is now broken down into three components with respect to the local coordinate system of node 1.

$$\begin{aligned}\vec{M}_{x_1} &= \vec{M} \cdot \vec{u}_x \\ \vec{M}_{y_1} &= \vec{M} \cdot \vec{u}_y \\ \vec{M}_{z_1} &= \vec{M} \cdot \vec{u}_z\end{aligned}\tag{10}$$

Next, the rotation of the spring whose axis lies on the local coordinate system unit vector \vec{u}_x is

$$\varphi_{x_1} = \frac{M_{x_1} L}{K_t G} - \varphi_{x_1 Last}\tag{11}$$

where φ_{x_1} is the rotation of the element around the element's x-axis. In this equation

K_t represents the torsional spring constant described in equation (4). The variable $\varphi_{x_1 Last}$

represents the deflection of that same spring for its previous iteration. If this was the first time this calculation was performed then $\varphi_{x_1 Last} = 0$.

The rotation of the X-axis spring of the local coordinate system for node 1 requires that the Y- and Z-axes for this local coordinate system also rotate around the X-axis the same amount and in the same direction. This is done with the vector rotation formula described in equation (5). In this case, the unit vector \vec{u}_x is the axis of rotation. The vectors \vec{u}_y and \vec{u}_z are rotated around \vec{u}_x by an angle φ_{x_1} . Now all unit vectors in local coordinate systems at each node with a subscript higher than that of the current node must rotate by the angle φ_{x_1} around the axis \vec{u}_x .

Next, $\theta_{1_{y_1}}$, the rotation of the spring associated with the local unit vector \vec{u}_{y_1} , is calculated.

$$\theta_{1_{y_1}} = \frac{M_{y_1}}{K_y} - \theta_{1_{y_1 Last}} \quad (12)$$

Similar to what was done earlier, the unit vectors \vec{u}_{x_1} and \vec{u}_{z_1} at that same node are then rotated around axis \vec{u}_{y_1} by an angle $\theta_{1_{y_1}}$. However, all unit vectors in local coordinate systems at each node with a subscript higher than that of the current node do not rotate by the angle $\theta_{1_{y_1}}$, although they do rotate around \vec{u}_{y_1} . If the end angle of the actual end-moment loaded beam were compared to the end angle of the end-moment loaded pseudo-rigid-body model, it would be evident that the difference between these angles grows as the beam displacement increases. This comparison is illustrated in Figure 14.

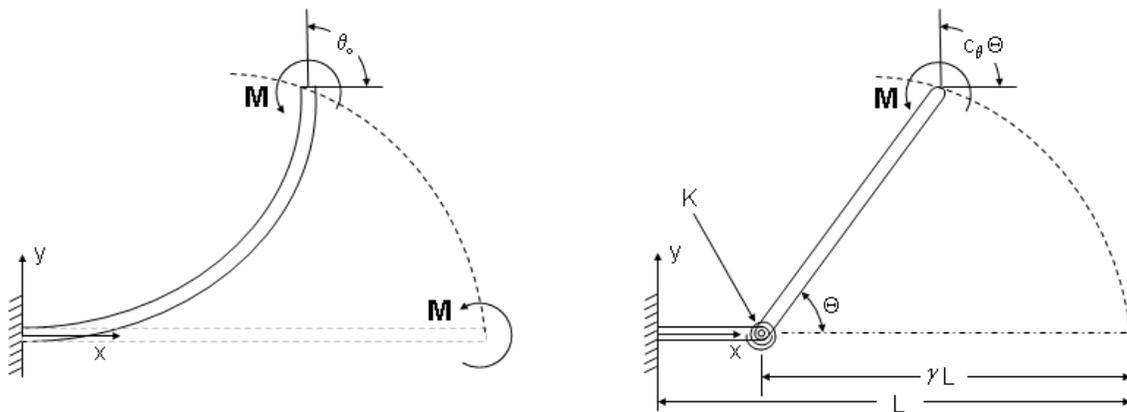


Figure 14: Actual beam tip angle compared to pseudo-rigid-body model tip angle

As noted in Figure 9, the correction factor or parametric angle coefficient is 1.5164. Before the unit vectors at nodes with a subscript higher than the current subscript can be rotated, this parametric angle coefficient must be multiplied by θ_{1,y_1} .

Finally, the last spring can be analyzed in the same way as the \vec{u}_{y_1} spring.

$$\theta_{2_{z_1}} = \frac{M_{z_1}}{K_z} - \theta_{2_{z_1}Last} \quad (13)$$

The angle $\theta_{2_{z_1}}$ is found, followed by the rotation of the \vec{u}_{x_1} and \vec{u}_{y_1} around the axis \vec{u}_{z_1} . The same parametric angle coefficient is then multiplied by the angle $\theta_{2_{z_1}}$ and all unit vectors for nodes with a subscript higher than the current subscript are rotated around \vec{u}_{z_1} by an angle $1.5164\theta_{2_{z_1}}$.

This process continues by moving from node 2 to the final node n. All nodes with a number lower than the current node number being analyzed are held rigid, and before any node can be analyzed, the positions of each node that has been modified must be found.

This iterative process is repeated over and over. After each run of analyzing the first node to the last node, the distance between a particular node's old position and its new position should converge to zero. When enough accuracy has been achieved the process can be terminated.

In the simple tip-loaded cantilever case used to describe the chain algorithm process, it should be noted that the moment on the last 3-D cantilever is zero because the distance from the beam tip to the applied force is zero. This illustrates how the 3-D chain algorithm approximates a continuous process by discretizing the moment as shown in Figure 15.

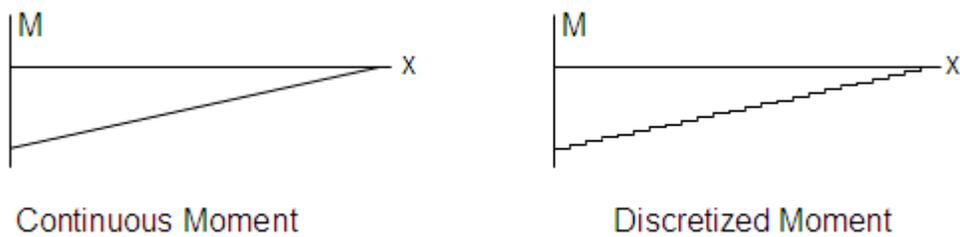


Figure 15: Discretization of moment load

This 3-D chain algorithm is useful in applications where applied forces and moments are known. It is robust and can converge to solutions with fairly large load increments. Multiple forces and moments can be placed at any node along the chain to simulate different or very complex displacements.

VI. THE CHAIN ALGORITHM COMPARED TO FINITE ELEMENT ANALYSIS

In this comparison, the 3-D chain algorithm and a commercial finite element analysis program capable of nonlinear deflection analysis (ANSYS) were used to analyze lateral torsional buckling of a beam with the same geometric and material properties. The test case beam was 100 inches long, 5 inches high, and 0.125 inches wide. This gave it an aspect ratio of 40. It was isotropic and had material properties approximating polypropylene, the Young's modulus was 200,000 psi and Poisson's ratio was 0.34. Both FEA and the 3-D chain algorithm used 200 elements. A small perturbation force was applied in the Z-direction to initiate lateral buckling of the beam. A Y-direction force was applied in two-hundred increments. After 200 increments the beam deflected 80% of its length in the Y-direction. The FEA model used equal displacement loads while the chain algorithm added equal force load increments. Both the ANSYS approach and the 3-D chain algorithm finished with essentially the same 3-D displacement.

In the XY plane, there was a 0.24% difference between FEA and the chain algorithm with respect to the Y-deflection found after the beam had displaced 40% of its length (see Figure 16).

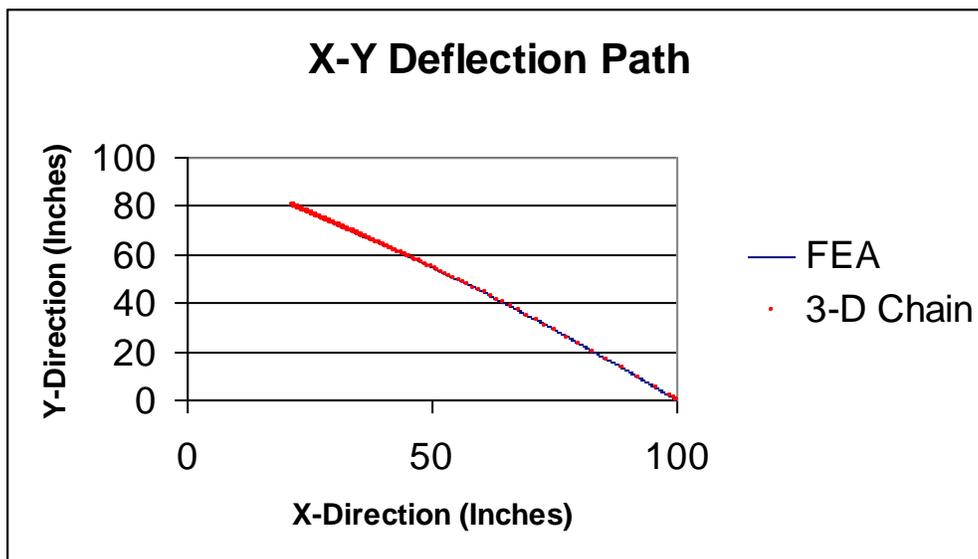


Figure 16: XY plane beam tip path

In the YZ plane there was a 0.37% difference with respect to the Y-deflection after deflecting 40% of the length (see Figure 17).

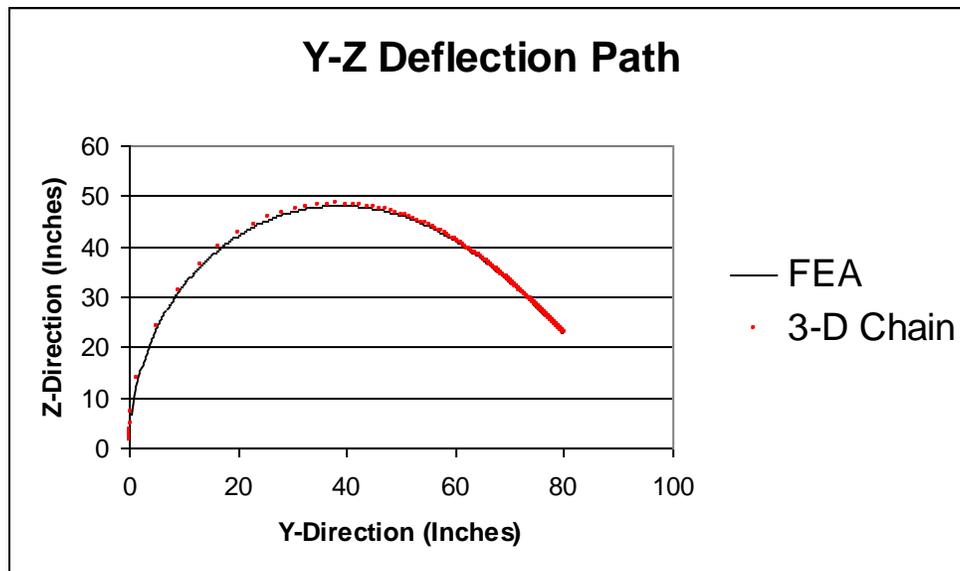


Figure 17: YZ plane beam tip path

The force was found to have a 0.0008% difference with respect to the Y-deflection after the beam had deflected 40% of its length in the Y-direction (see Figure 18).

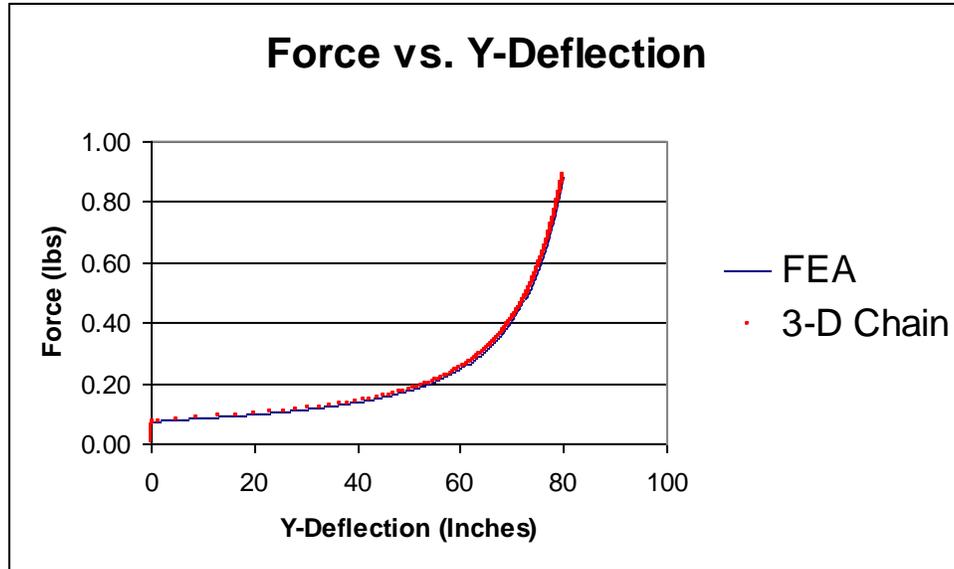


Figure 18: Relationship between force and Y-deflection

Also, it should be noted that FEA required about 75 minutes on a BYU supercomputer while the chain algorithm required only 18 minutes on a laptop to complete the same number of steps. This illustrates the accuracy and efficiency of the 3-D chain algorithm. The chain algorithm is not intended to replace FEA, but it is a powerful tool for certain problems where many iterations are required, such as in the early phases of design. This is particularly important for flexible segments that undergo large complex 3-D motions in compliant mechanisms.

VII. CONCLUSION

The 3-D chain algorithm with pseudo-rigid-body model elements has great potential in describing large complex deflections in beams because it models large 3-D beam displacements accurately and, in many applications, faster than the finite element method. In particular, the rigid links and springs that make up the elements of the chain describe the force deflection relationship and displacements at each node. The use of pseudo-rigid-body model elements greatly simplifies the 3-D chain algorithm, allowing it to be programmed in a few pages of code. Because the three pseudo-rigid-body models for each beam segment act orthogonally to each other, no interactions between models need to be addressed. Each model brings to the element excellent accuracy and, in the case of the end-moment loaded pseudo-rigid-body models, large deflection analysis potential.

The 3-D chain algorithm can include multiple forces and moments to be placed at any node along the beam. Curved beams and beams with varying cross-sections can also be analyzed as long as the product of inertia is zero. The 3-D chain algorithm is a general beam design tool that can describe beam deflections and force deflection relationships with good accuracy much faster than finite element methods for some applications, and thus is valuable in the early stages of design where many design iterations often need to be analyzed. The algorithm is showing promise in research in the design of compliant devices under lateral-torsional loads [5]. In the past, few applications of elastic beams undergoing lateral torsional buckling have been demonstrated, but design tools supporting their analysis, such as the 3-D chain algorithm, may lead to new applications not previously possible.

REFERENCES

1. Love, A.E.H. *The Mathematical Theory of Elasticity*; New York Dover Publications: New York, 1944; 419.
2. Villaggio, P. *Mathematical Models for Elastic Structures*; Cambridge University Press: Cambridge, UK, 1997; 219.
3. Petroski, H. *Design Paradigms*; Press Syndicate of the University of Cambridge: New York, 1994; 93.
4. Trahair, N. 1993. *Flexural-Torsional Buckling of Structures*; CRC Press: Florida, 1993; 174
5. Chase, R. 2006. *Large 3-D Deflection and Force Analysis of Lateral Torsional Buckled Beams*; Master's Thesis, Brigham Young University: Provo, Utah, 2006.
6. Howell, L.L. *Compliant Mechanisms*; John Wiley & Sons: New York, 2001; 261, 266, 166, 192, 191.
7. Nahvi, H. *Static and Dynamic Analysis of Compliant Mechanisms containing Highly Flexible Members*; Ph.D. dissertation, Purdue University: West Lafayette, Indiana; 1991.
8. Coulter, B.A.; Miller, R.E. Numerical Analysis of a Generalized Plane 'Elastica' with Non-linear Material Behaviour. *Internat. J. Numer. Methods Engrg.* **1988**, 26, 617–630.
9. Pauly, J.; Midha A. Pseudo-Rigid-Body Model Chain Algorithm, Part 1: Introduction and Concept Development. *Proceedings of 2006 ASME International Design Engineering Technical Conferences & Computers and Information In Engineering Conference*, Philadelphia, PA, September 10–13, 2006; DETC2006-99460, 1, 2, 8.

10. Pauly, J.; Midha, A. Pseudo-Rigid-Body Model Chain Algorithm, Part 2: Equivalent Representations for Combined Load Boundary Conditions. Proceedings of 2006 ASME International Design Engineering Technical Conferences & Computers and Information In Engineering Conference, Philadelphia, PA, September 10–13, 2006; DETC 2006-99463, 2.
11. Gere, J. M. *Mechanics of Materials*, 5th Ed.; Brooks/Cole: Pacific Grove, CA, 2001; 632.
12. Young, W.C.; Budynas, R.G. *Roark's Formulas for Stress and Strain*; McGraw-Hill, New York, 2002, 64, 383.
13. Murison, A., 1998. "How to Rotate a Vector." USNO Astronomical Applications Technical Note. 1998. 21 Oct. 2006 <<http://www.alphertatz.net/murison/papers/Notes/RotateVector/>>