

MeshPerturb

MATLAB codes for mesh perturbation and automated pre and post processing of post-bifurcation analyses via COMSOL

Sourabh K. Saha^{*} and Martin L. Culpepper[#]
Department of Mechanical Engineering
Massachusetts Institute of Technology

May 10, 2014

Abstract

Recently, there has been an interest in designing systems that employ buckling bifurcation as a desired mechanism to generate motion and/or form. As buckling has traditionally been considered a failure mode to be avoided, computational tools for predicting the state of the system after buckling are not well developed. For example, modules for post-bifurcation studies are not available in most of the commercial finite element software packages. Herein, we provide MATLAB codes that add the ability to perform post-bifurcation studies to the COMSOL package. This is achieved by implementing mesh perturbations as a scheme to introduce geometric imperfections to the system. Additionally, we provide codes to automate pre/post processing the studies. These codes enable performing (i) post-bifurcation analysis, (ii) studies on sensitivity to mesh imperfections, and (iii) unattended parametric studies that require re-meshing the geometry.

1. General purpose: Post-bifurcation analysis

In engineering applications, solving a bifurcation problem commonly arises during design and analysis of structures suspect to buckling failure. Solution to a bifurcation problem has two distinct steps: (i) predicting the onset of bifurcation and (ii) predicting the behavior of the system after the onset of bifurcation. These steps require two distinct sets of predictive tools. Herein, we focus on developing the predictive tools for solving the second step, i.e., for post-bifurcation analysis.

As buckling has traditionally been considered a failure mode, the first step of predicting the onset of bifurcation has received more attention. Thus, predictive tools for post-bifurcation analysis of structures are not as well developed as those for predicting the onset. This trend is also reflected in the development of commercially available finite element analysis (FEA) software packages. For example, dedicated stability analysis based modules are available for predicting the onset of bifurcation but modules for post-bifurcation analysis are absent from most FEA packages. In the past, this lack of predictive tools has not been a major issue due to the limited engineering interest in post-bifurcated systems. However, recently there has been an interest in designing and building systems that employ buckling bifurcation as a desired mechanism to generate motion and/or form. Post-bifurcation analysis is essential for an accurate prediction of form/motion in these systems. Our software codes add the ability to perform post-bifurcation studies to the commercially available COMSOL FEA package. This enables one to systematically perform post-bifurcation analyses that were previously not feasible within COMSOL.

^{*}sksaha@mit.edu, [#]culpepper@mit.edu.

This document is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2. Specific goal: Automated mesh perturbation

Providing an external perturbation to the system is often necessary for observing buckling bifurcation during numerical simulations. The bifurcation point is characterized by intersection of multiple feasible modes/paths that the system may follow after bifurcation. Out of these paths, one or more may be energetically favorable. In the absence of any imperfection, the system is unable to “see” any of these alternate paths. Therefore, the system may remain on an unfavorable path even after the bifurcation point.

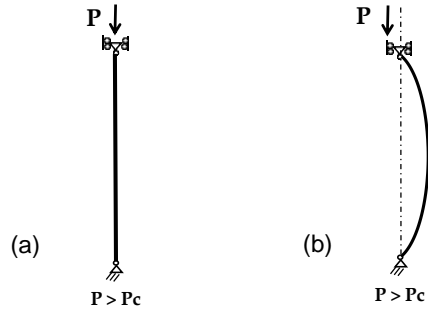


Figure 1: Imperfection is essential for bifurcation. (a) A “perfect” column does not buckle even at very high loads. (b) Real column buckles at high loads due to presence of imperfections.

For example, during buckling of a column there are two feasible modes at the bifurcation point (i) pure compression and (ii) bending. Bending is energetically favorable beyond the bifurcation point. However, a ‘perfect’ column with no imperfections would not buckle even beyond the bifurcation load (Fig. 1). In reality, this ‘non-buckling beyond bifurcation’ phenomenon is never observed due to the presence of imperfections in a real system. However, such non-buckling are routinely observed in an idealized finite element model. Therefore, an imperfection must be applied to the FEA model to ensure that the system follows the favorable post-bifurcation path. The imperfection may be in the form of (i) geometric/mesh imperfection and/or (ii) imperfection in the applied loads and boundary conditions. Herein, we provide this imperfection via perturbation of the mesh.

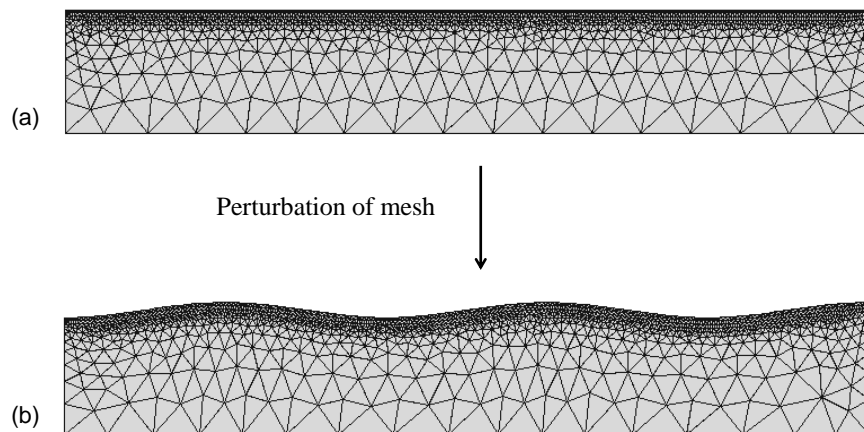


Figure 2: Mesh perturbation as imperfection that enables bifurcation. (a) Perfect mesh that does not bifurcate. (b) Perturbed mesh that bifurcates (perturbation exaggerated for clarity). A perturbed mesh can be obtained from a perfect mesh by ‘moving’ the mesh points around. Our codes enable one to systematically perform this perturbation process.

Perturbing the mesh is a reliable and sometimes the only technique to introduce imperfections to a system in such a way that the appropriate post-bifurcation behavior is observed. Perturbing a mesh means moving the mesh points around so that the resulting geometry is ‘somewhat but not substantially’ different from the original geometry (Fig. 2). Although mesh perturbation is an important technique for post-bifurcation analysis, it is not available via the existing modules in COMSOL. Therefore, we have developed these MATLAB codes to perform automated mesh perturbations. Additionally, we have developed the codes to automate the pre and post processing of the FEA models. These codes enable one to perform (i) post-bifurcation analysis, (ii) studies on sensitivity to mesh imperfections, and (iii) parametric studies that require re-meshing the geometry.

3. Technical description

3.1. Specific bifurcation problem: wrinkling

We have developed MATLAB codes to solve the specific case of wrinkling bifurcation that is observed during compression of bilayer materials. With minor modifications, these codes may also be used for other buckling bifurcation systems such as for buckling of spherical shells. Herein, we summarize the problem and solution scheme for bilayer wrinkling. Later in Sec. 3.6, we discuss how our codes may be modified to solve other bifurcation systems.

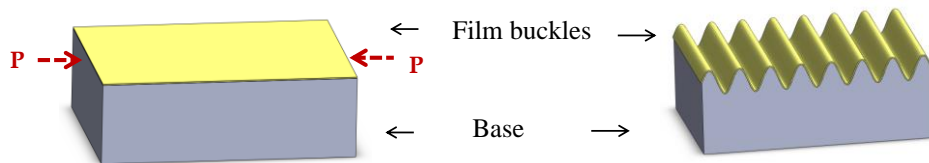


Figure 3: Schematic of the wrinkling process. Wrinkles are formed via periodic bending of the film/base bilayer upon compression.

Wrinkles are formed on a thin film as a result of buckling-based instabilities and the mechanism is similar to Euler buckling of beams under compressive loads (Fig. 3). Wrinkling of thin films can be used as an inexpensive fabrication technique for generating micro and nano scale periodic patterns [1]. The wrinkling system consists of a thin film of a stiff material that lies on top of a soft base. As the film is compressed, it undergoes buckling bifurcation at a critical compression. Beyond the critical compression, pure compression becomes unstable and wrinkles are formed via periodic bending of the film/base. In the absence of a base, the film buckles into the largest feasible period that is determined by the in-plane film dimensions. However, in the presence of a base, a period that is independent of in-plane dimensions is obtained. This is because of the competing strain energy dependence of the film and the base on the wrinkling wavelength. The film bending energy decreases with wavelength, whereas the base stretching energy increases with wavelength. Thus, the total strain energy is minimized at an intermediate period. The period and amplitude of the wrinkles depend on the applied compression, mechanical properties of the film and base, and the film thickness. Thus, periodic patterns of the desired period/amplitude may be obtained by controlling these parameters.

In the absence of analytical models, finite element techniques are used for predictive design of complex wrinkling patterns. As wrinkles are formed via a bifurcation process, the accuracy of these models is dependent on the initial geometric imperfection to the system. Without this initial imperfection, the

‘perfect’ system does not bifurcate and wrinkled patterns are not observed (Fig. 4); when excess imperfection is added, the system changes over into a different one. Thus, there exists a specific range of imperfection values over which ‘proper’ bifurcation occurs. Our codes ensure that mesh imperfections are within this range.

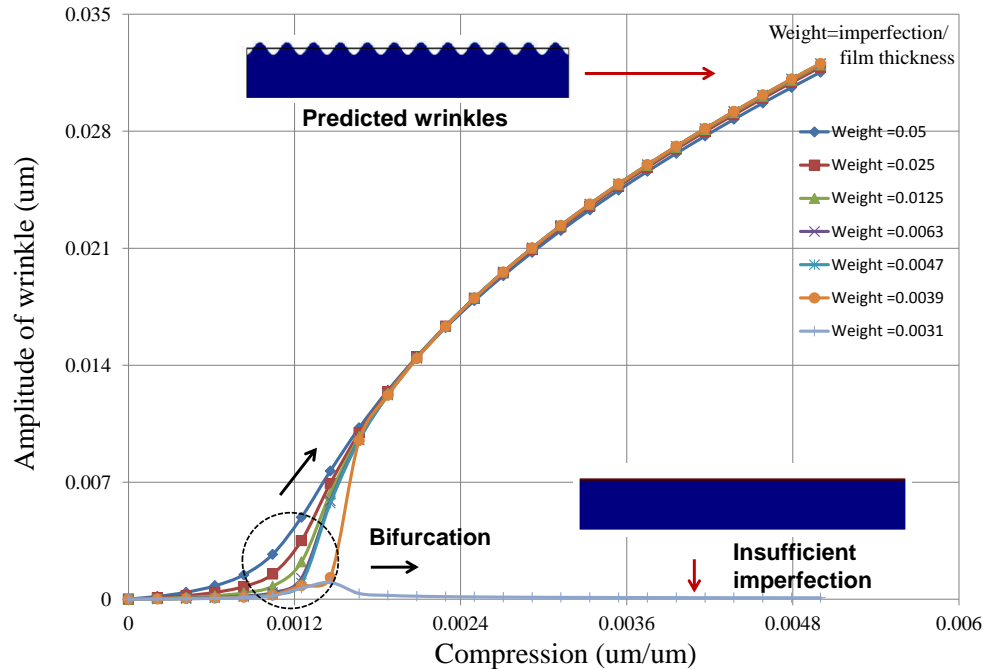


Figure 4: Effect of insufficient mesh perturbation on bifurcation behavior. The system does not bifurcate unless a sufficiently large perturbation is provided.

3.2. FEA modeling

We have developed 2-D and 3-D finite element models of wrinkling using the Structural Mechanics module of the COMSOL 4.2 software package [2]. The COMSOL software package was selected because of its ability to (i) perform coupled multi-physics simulations and (ii) link to MATLAB to pre/post process the models. The 2-D models were developed by implementing buckling of wide plates under the plane strain condition; whereas 3-D models were developed by implementing buckling of finite sized plates. In both cases, the top film was modeled as an elastic plate and the bottom layer as a Neo-Hookean foundation. Also, two different types of loading were applied: (i) pure compression of top and bottom layers and (ii) initial pre-stretch of the bottom layer followed by release of this pre-stretch. The pre-stretch based loading was implemented by applying an initial strain to the top layer that corresponds to a uniaxial/biaxial tension. This was followed by compression of both the layers.

Modeling was performed in two steps: (i) linear buckling analysis to predict the mode shapes, i.e., the period of the wrinkles and (ii) a nonlinear post-buckling analysis to predict the shape and amplitude of the wrinkles after buckling bifurcation. In this two-step process, first the displacements of the 1st mode shape obtained from linear buckling analysis were extracted; then the displacements were added as the initial imperfection to the mesh. The displacements were weighted by a multiple of the top layer thickness. The studies were solved in COMSOL. LiveLink for MATLAB was used to (i) generate the initial weighted imperfections, (ii) set-up the nonlinear studies and (iii) post-process the nonlinear analysis.

3.3. Mesh perturbation

3.3.1. Mathematical background

In our scheme, we generate the perturbed mesh from the original mesh by displacing the mesh points, i.e., by ‘moving around’ the nodes. This transformation preserves the connectivity, ordering, total number of nodes, and the type of elements in the mesh. If $X_{i,j}$ are the coordinates of the j^{th} node in the original mesh and $x_{i,j}$ are the coordinates of the same node in the perturbed mesh, the two are related by:

$$x_{i,j} = X_{i,j} + (wh)\bar{u}_{i,j} \quad (\text{Eq. 1})$$

$$\bar{u}_{i,j} = \frac{u_{i,j}}{\max |u_{N,j}|} \quad (\text{Eq. 2})$$

Here, ‘ w ’ is a non-dimensional weight factor, ‘ h ’ is thickness of the top layer and ‘ $\bar{u}_{i,j}$ ’ are non-dimensional perturbation shape functions. The perturbation shape functions are generated by non-dimensionalizing the displacement field, $u_{i,j}$, with the maximum out-of-plane displacement. The size of the perturbation is determined by the weight factor and the top layer thickness. The subscript ‘ i ’ runs from 1 to N for an N -dimensional system; i.e., $N=2$ for a 2-D system and $N=3$ for a 3-D system. The displacement functions $u_{i,j}$ may be chosen arbitrarily or according to a predetermined scheme. Typically, during post-bifurcation analysis, these functions are chosen to be either (i) the Eigen modes of the linear buckling analysis or (ii) the geometric imperfections that are observed in a ‘real’ system. Our codes enable applying both of these two types of mesh perturbations.

3.3.2. Implementation: Mesh as ‘.mphtxt’ file

In order to implement the perturbation scheme discussed in Sec. 3.3.1, we have developed codes that transform an existing COMSOL mesh into the perturbed mesh. This is achieved by generating a new file with the perturbed mesh data from an existing ‘.mphtxt’ text file that contains the original mesh data. An ‘.mphtxt’ mesh file contains all data that are required to fully define a mesh. Details of the format are available in the COMSOL Reference Guide [3]. The original mesh file can be obtained from a COMSOL study by exporting the undeformed mesh. The perturbed mesh is generated by replacing the mesh point coordinates of the original mesh with the transformed coordinates while keeping the rest of the mesh data intact. To do this we have developed codes that perform (i) file read/write operations to extract and replace mesh point coordinates of an ‘.mphtxt’ file and (ii) mathematical transformations to generate coordinates of the perturbed mesh for a given set of perturbation shape functions. A COMSOL study may then be solved by importing the perturbed mesh and defining the study on this new mesh.

3.4. Algorithm

In addition to the codes to perturb a mesh, to solve a post-bifurcation problem one needs to (i) extract the original mesh, (ii) generate the perturbation shape functions, and (iii) define the COMSOL study files. Although these steps may be performed manually, codes for automated processing are necessary if unattended parametric studies are required. Automation was made possible by (i) using MATLAB LiveLink to transfer data across the MATLAB and COMSOL desktop environments and (ii) using

COMSOL model seed files to define a template of the post-bifurcation study. Herein, we summarize the sequence of steps involved in defining and solving a post-buckling study.

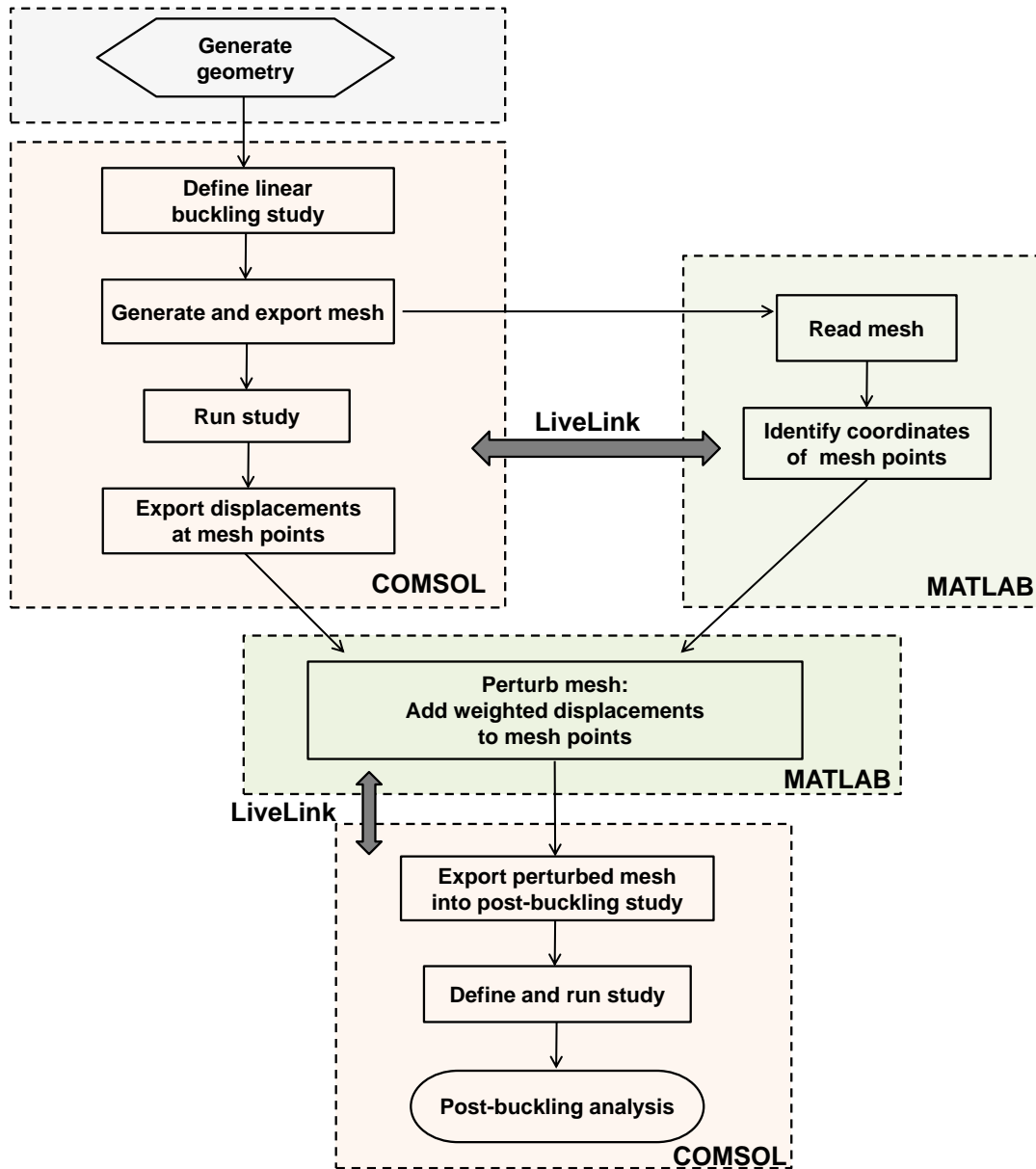


Figure 5: Algorithm for performing post-bifurcation analysis with imperfections defined by Eigen modes of the pre-buckling analysis.

3.4.1. With Eigen modes as mesh perturbation

When Eigen modes are used as mesh perturbations, a linear buckling analysis is performed before defining the post-buckling study. The mesh perturbation displacement field in Eq. 2 is obtained from Eigen modes of the linear buckling analysis. The perturbed mesh is obtained by performing the transformation of Eq. 1 on the original undeformed mesh of the linear buckling analysis. The post-buckling study is then defined on this perturbed mesh. As the amplitude of the modes is undetermined in a linear buckling analysis, only the shape of the modes is used as the perturbation displacement function. The size of the perturbation is determined by the user defined weight factor and is proportional to the top

layer thickness. A flowchart of the algorithm is shown in Fig. 5. Both the linear buckling and the post-buckling studies are solved in the COMSOL environment. However, the studies are defined and post-processed in MATLAB. Data is exchanged between the two environments via the MATLAB LiveLink module of COMSOL.

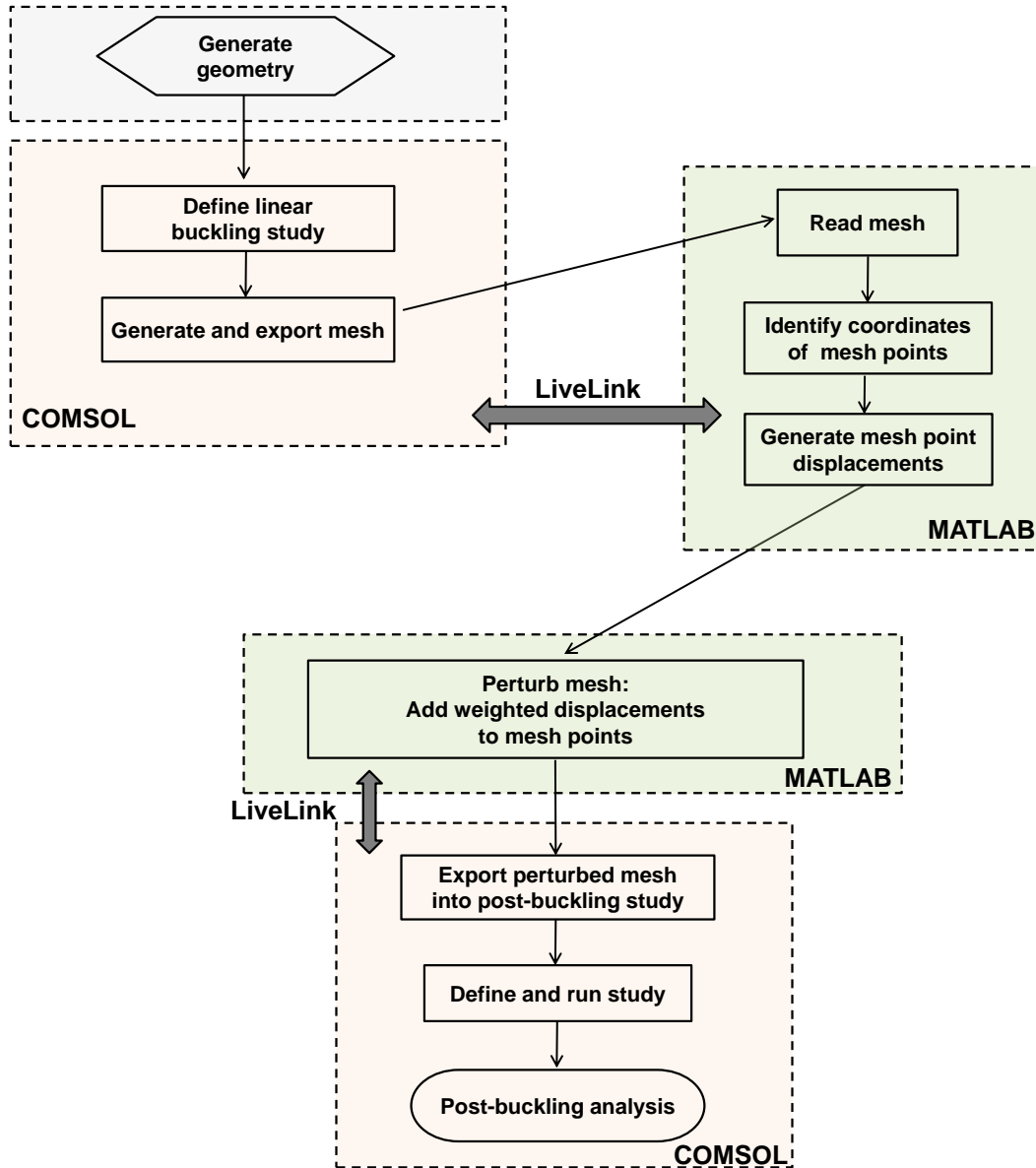


Figure 6: Algorithm for performing post-bifurcation analysis with externally provided imperfections.

3.4.2. Externally provided mesh perturbation

Running a linear buckling analysis is not necessary when an external mesh perturbation is applied. Instead, only the original undeformed mesh is extracted from the study and the mesh is then transformed with the external perturbation. The post-buckling study is then defined on this perturbed mesh. For convenience, the schemes and codes for Sec. 3.4.1 are reused with these modifications: (i) the linear buckling study is defined to extract the mesh but the study is not solved and (ii) external perturbation displacement field is generated and used to perturb the mesh. The steps are summarized in Fig. 6.

3.5. Automation via seed models and LiveLink

During a parametric study that involves varying the geometric parameters, it is necessary to regenerate a perturbed mesh for each combination of parameters. This requires repeatedly solving the linear buckling studies, generating the perturbation displacement field, and defining the post-buckling studies. Automation of this process can substantially reduce the time and effort required to run such parametric analyses. We have achieved this automation by using seed models to define the COMSOL models and LiveLink to exchange data between MATLAB and COMSOL.

3.5.1. Seed models

When a geometric parameter is varied, we use seed models to generate the new models. Seed models are the preliminary unsolved COMSOL models that contain all the necessary definitions required to solve the model for a specific set of parameters. Seed models are necessary because post-buckling studies are not defined on a geometry; instead, they are defined on an imported mesh. Therefore, changing a geometric parameter requires changing the mesh in this study. The initial linear buckling study in the COMSOL ‘.mph’ format is directly used as the seed model. The seed model for the post-buckling study is obtained by exporting the post-buckling COMSOL study as a ‘.m’ script. This script contains the sequence of MATLAB commands required to define the study [4]. The linear buckling seed model and the post-buckling MATLAB script can be modified to generate new models whenever the model parameters are changed. When Eigen modes are used as perturbations, the new linear buckling studies are solved to generate the perturbation displacement field. However, when external perturbations are used, the linear models are used only as a source of the initial undeformed mesh.

3.5.2. LiveLink

We have used the LiveLink module of COMSOL to ‘connect’ the COMSOL desktop environment to the MATLAB environment. Details on this module are available in the COMSOL documentation [5]. This connection enables one to access or modify data in a model from both of these environments. For example, one may define a study and/or modify the parameters via MATLAB scripts and then use COMSOL to solve the study. This connection has been used to automate parametric analysis by implementing these data transfer operations: (i) accessing and modifying parameters of the seed models to generate new models, (ii) extracting the undeformed mesh and perturbation displacement fields from the linear buckling models, (iii) defining a new post-buckling study on the perturbed mesh, and (iv) extracting results of solved models to perform post-processing via MATLAB. Automated data transfer during these steps enables performing parametric studies in a completely unattended mode.

3.6. Modifications for other systems

The codes provided here were written specifically for the wrinkling bifurcation system. However, with slight modifications these codes may also be used to solve other buckling systems. To solve other buckling systems one must: (i) replace ‘ h ’ in Eq. 1 with a length scale parameter that is physically relevant to the system, (ii) edit, delete, or add parameters that are required to properly define the new system, (iii) modify the seed files to represent the new system, and (iv) modify the codes to ensure that the parameter names and other variable names are consistent across the MATLAB codes and the seed models.

3.7. Morphing geometries via mesh perturbation

In addition to enabling post-buckling analysis, mesh perturbations may also be used as a technique for morphing geometries. A given geometry can be morphed into a different one by providing a moderate to large perturbation displacement to the undeformed mesh. For example, one may apply sinusoidal or other periodic perturbations to morph the initially flat top layer into a pre-patterned top layer. As this morphing is achieved via mesh perturbations, the same code and seed models may also be used to perform the post-buckling analysis on these morphed geometries.

4. References

- [1] S.K. Saha and M.L. Culpepper, [Predicting the Quality of One-Dimensional Periodic Micro and Nano Structures Fabricated Via Wrinkling](#), *ASME 2012 International Mechanical Engineering Congress and Exposition*, Houston TX, USA, Nov 2012.
- [2] *Structural Mechanics Module User's Guide*, COMSOL 4.2 installation documentation.
- [3] The COMSOL File Formats, *COMSOL Multiphysics Reference Guide*, pg. 481, COMSOL 4.2 installation documentation.
- [4] Building a COMSOL Model, *COMSOL Multiphysics User's Guide*, pg. 357, COMSOL 4.2 installation documentation.
- [5] *LiveLink for MATLAB User's Guide*, COMSOL 4.2 installation documentation.