Post-buckling Due to The Eigenstrain

Zhiyan Wei

1. Introduction

After spending a whole semester to learn the theory of elasticity and some basic concepts in large deformation, we feel natural to understand the deformation when an external force is applied on the object now. However, as I mentioned in my proposal for this project, even without external force, CPU chips deform when they are working. The reason is that the heat generated causes thermal expansion, and misfit in thermal expansion of different layers induces deformation and internal stresses. My classmate Matt's research topic for Abaqus project is related to this. To better understand the deformation without external forces, one may find it useful to understand eigenstrains and eigenstresses first.

The eigenstrain is a generic name given to such nonelastic strains as thermal expansions, plastic strains, misfit strains, photostrains and etc in a body which is free of external forces and surface constraints **[1]**. The eigenstress is a generic name given to such internal self equilibrated stresses caused by the eigenstrain. One perfect example of the eigenstrain is the light induced deformation of nematic liquid-crystal elastomers (LCEs). Nematic LCEs are rubbers whose rodlike constituent molecules are uniaxially oriented. A remarkable feature of such soft material is that they are able to deform as much as 400% when straddling the Nematic-Isotropic (NI) phase transition temperature **[2]**. The same magnitude of deformation as thermal response can also be achieved for the elastomers by other methods such as optical stimuli. It means that there is a mapping relation between these two kinds of eigenstrains—thermal strains and photo strains. This is the most fundamental reason why I use non-uniform temperature field inside the object to introduce an eigenstrain. It is a straightforward thermal strain, but one can regard it as any kind of eigenstrain by mapping them together. In the picture **[3]** below, for a certain range of incident lights, the deformation direction of the thin LCE layer keeps the same as the polarization direction of the incident light. If such a light-sensitive film is put on the water surface and illuminated by a linearly polarized light, it will swim away from the light. Mahadevan analyzed photoinduced small deformation of structural elements made of LCEs and found that beams of these materials can have two elastically neutral planes and plates have a typically saddle shape **[4]**.

My topic for Abaqus project is simply related to a common phenomenon in the nature—large long leaves can hardly keep in plane. They more or less buckle on the edges or the whole bodies curve. As we see, there are no external forces exerted on leaves, but they do deform. This is also an eigenstrain problem, in which the non-uniform growth rate plays the role. One submitted paper by Haiyi and Mahadevan both analytically and numerically analyzes this problem, which inspires me to model and test it in Abaqus. If Abaqus works, it might be a good choice to use it to complete the phase diagram. I choose pictures of seaweeds because in water, the buoyancy nearly counterbalances the gravity effect, so it might make us more comfortable to believe that the buckling is not caused by gravity, although I cannot rigorously prove it.

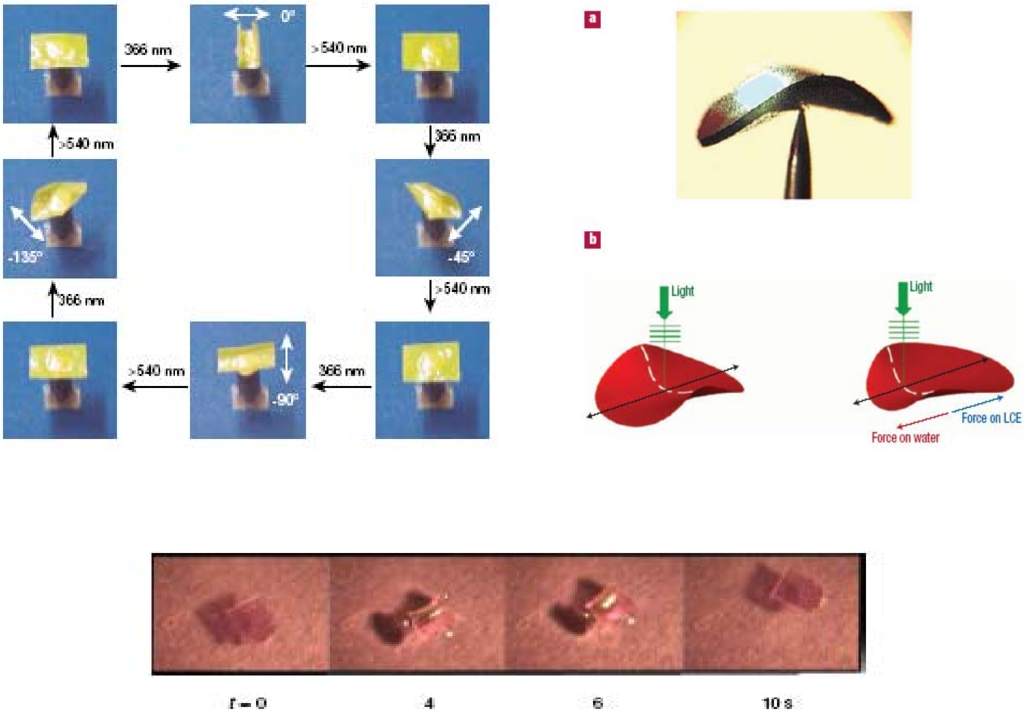


Fig. 1 photo eigenstrains

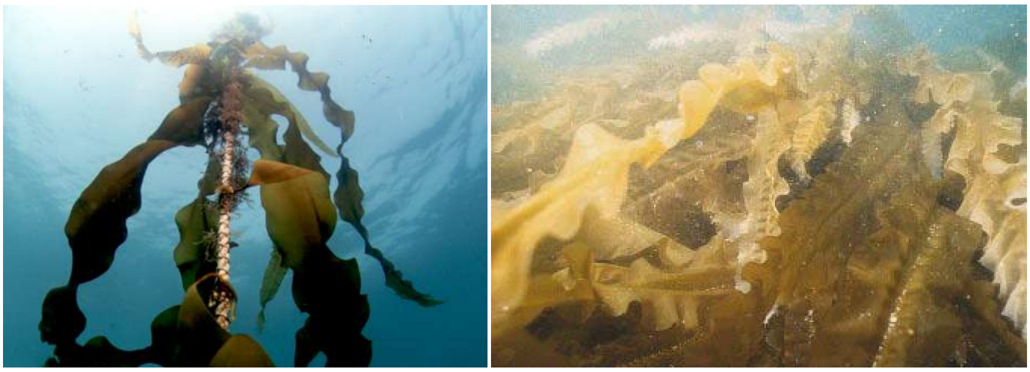


Fig. 2 buckling of long large seaweeds

2. FEA Model

FEA method is employed to analyze the post-buckling behavior of the eigenstrain problem. Post-buckling is a finite deformation. Abaqus should have no problem dealing with the finite deformation. (Note that in our problem, finite deformation really means small local stretching but large rotation.) The choice of Young's modulus **E** turns out making no differences in this problem. I think the reason is that what I specify is the temperature field (not the force) inside the sample, which can be converted to eigenstrain (a dimensionless parameter). The energy analysis manifests that as post buckling proceeds, stretching energy decreases while bending energy increases. Both of them contain the Young's modulus **E**, so after energy scaling, **E** does not appear in the argument. Maybe this is why **E** only influences the stress state, but has nothing to do with the strain state in the end. Poisson's ration is chosen to be 0.3. The coefficient of thermal expansion is 0.01.

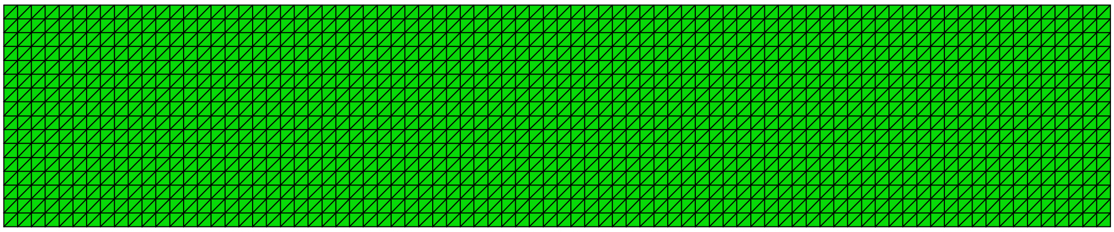
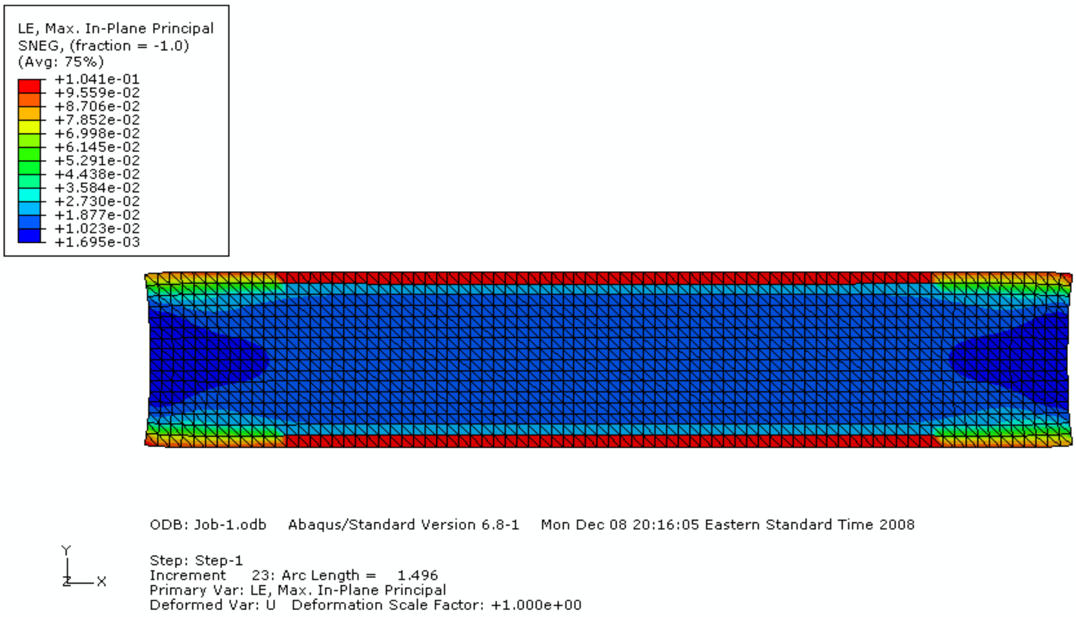


Fig. 3 FEA Model

- Part: 3D Deformable Shell
- Size:
  The size is varied in computation. The complete plan also includes testing different shapes such as ellipses with different major and minor axes, which can asymptotically approximate rectangular. If everything works perfectly, to model a real leaf is more interesting.
- Boundary Condition: fixed in the center
  The purpose is to eliminate the rigid body motion. A more experienced user of Abaqus suggested me not to use a single-point-BC because in large deformation, it is more likely to result in a divergence in computation. I am not sure if it is true or not. However, after I tried tens of times, every time the computation diverged, changing BC did not help at all. Once the computation was convergent, any BC that was sufficient to eliminate the rigid body motion was fine. Hence, my conclusion is that at least in this case, BC is not the reason for divergence.
- Temperature field:

y is the coordinate along the width. Experiment results show that the leaf grows much faster at the edge than in the middle. This form is only a roughly estimate. Abaqus allows two ways (as I find) to add this non-uniform temperature field. One is to use User's Subroutine UTEMP, and the other is to use Analytical Fields Function. It is worth to mention that Abaqus seems to have a bug in the second method. When I applied the temperature and directly submitted the job, the program did not yield a correct result. No matter how large or how small the temperature magnitude is, the deformation is always the same, which is very weird. However, if I saved the file into an .inp file after applying the temperature field and reloaded without doing anything else, the program yielded a correct result.

- Mesh Elements: linear triangular elements
  The choice of linear triangular elements is also from the experience gained after try. Linear triangular elements seem to have the highest tolerance. Once it diverges, other types of elements do not work either. On the contrary, when other types of elements do not work, it is possible for triangular elements to give a reliable result. My superficial understanding is that linear triangular elements have constant strains inside them. They lose some accuracy but gain more tolerance.
- We expect that if no out-plane perturbation is introduced. No matter how large the eigenstrain is, the deformation still remains in plane, because FEA only solves the equation $KU = F$ for displacement. To make ourselves convinced, see the result below. $\alpha = 5, \beta = 10$, so the eigenstrain is $\varepsilon^{*} = 0.05\left(y/200\right)^{10}$.

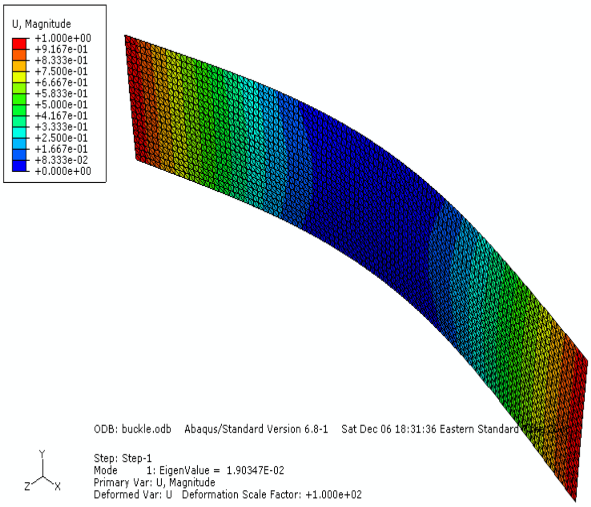


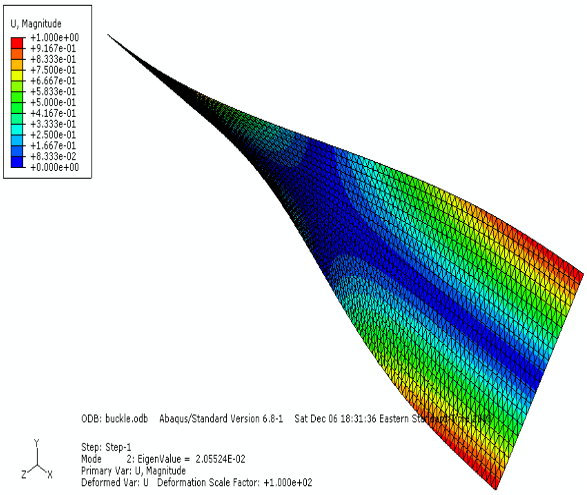
```
ODB: Job-1.odb    Abaqus/Standard Version 6.8-1    Mon Dec 08 20:16:05 Eastern Standard Time 2008

Step: Step-1
Increment     23: Arc Length =     1.496
Primary Var: LE, Max. In-Plane Principal
Deformed Var: U   Deformation Scale Factor: +1.000e+00
```

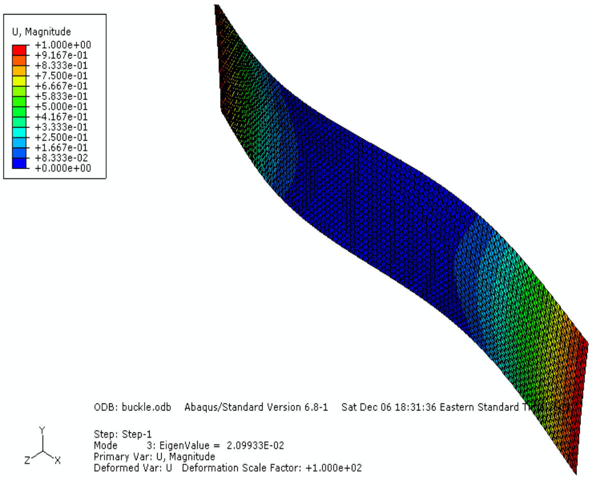
Fig.4 in plane deformation

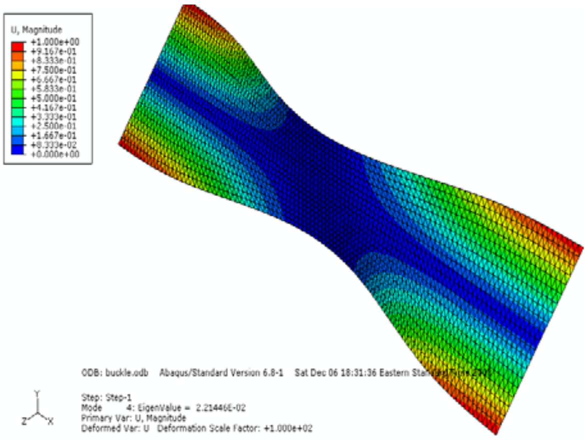
3. Buckling Mode Analysis

The linear perturbation method is employed to study the buckling modes. It is the pre-step the post-buckling analysis. Note that this is only a result got from the linear analysis. Post-buckling is a finite deformation process. The first 10 buckling modes are listed below:


U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb   Abaqus/Standard Version 6.8-1   Sat Dec 06 18:31:36 Eastern Standard
Y
Z
X
Step: Step-1
Mode   1: EigenValue = 1.90347E-02
Primary Var: U, Magnitude
Deformed Var: U   Deformation Scale Factor: +1.000e+02


U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb   Abaqus/Standard Version 6.8-1   Sat Dec 06 18:31:36 Eastern Standard Time 2008
Y
Z
X
Step: Step-1
Mode   2: EigenValue = 2.05524E-02
Primary Var: U, Magnitude
Deformed Var: U   Deformation Scale Factor: +1.000e+02


U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb   Abaqus/Standard Version 6.8-1   Sat Dec 06 18:31:36 Eastern Standard
Y
Z
X
Step: Step-1
Mode   3: EigenValue = 2.09933E-02
Primary Var: U, Magnitude
Deformed Var: U   Deformation Scale Factor: +1.000e+02


U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb   Abaqus/Standard Version 6.8-1   Sat Dec 06 18:31:36 Eastern Standard Time 2008
Y
Z
X
Step: Step-1
Mode   4: EigenValue = 2.21446E-02
Primary Var: U, Magnitude
Deformed Var: U   Deformation Scale Factor: +1.000e+02

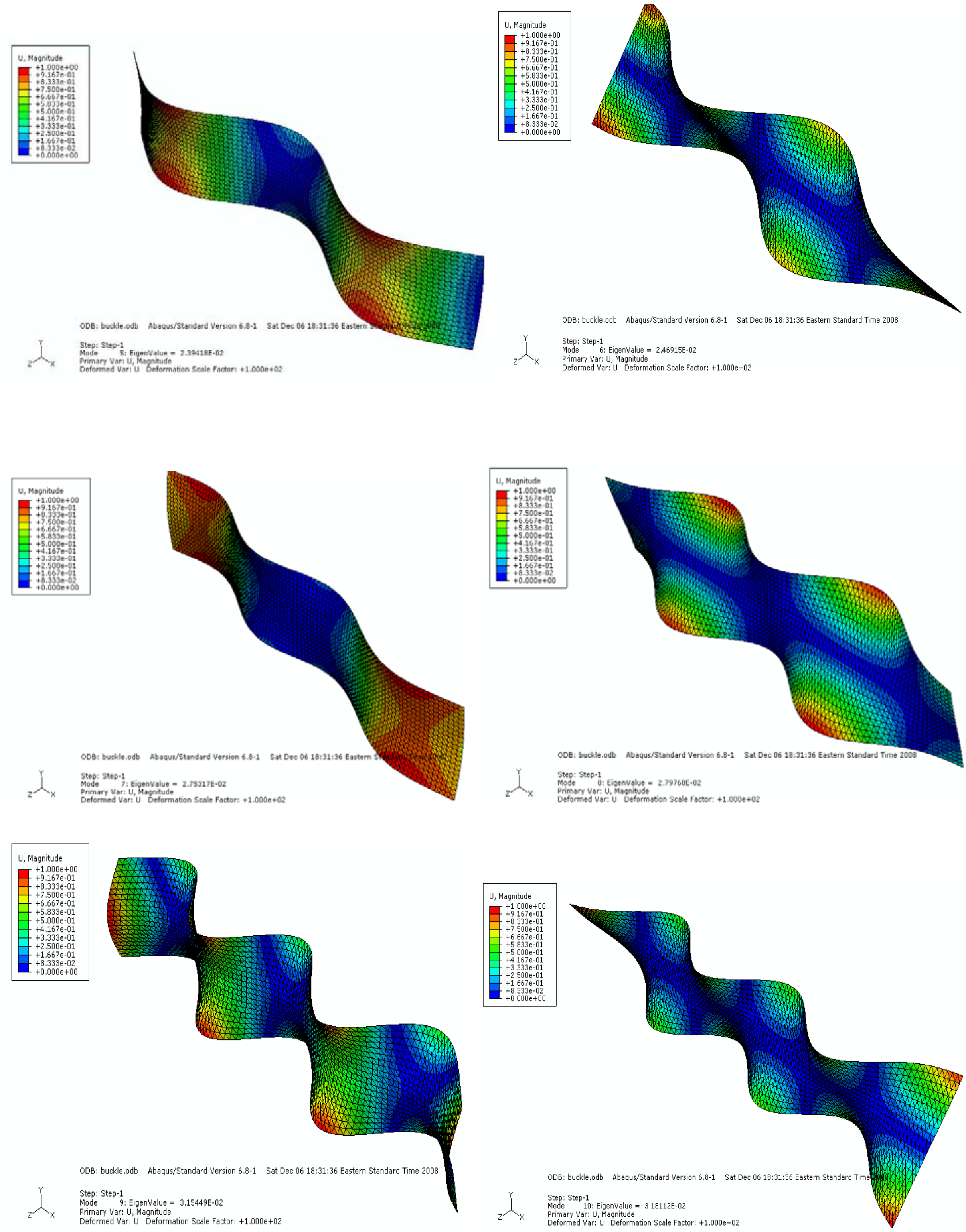

U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern
Step: Step-1
Mode 5: EigenValue = 2.39418E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X
U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern Standard Time 2008
Step: Step-1
Mode 6: EigenValue = 2.46915E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X
U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern
Step: Step-1
Mode 7: EigenValue = 2.75317E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X
U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern Standard Time 2008
Step: Step-1
Mode 8: EigenValue = 2.79760E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X
U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern Standard Time 2008
Step: Step-1
Mode 9: EigenValue = 3.15449E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X
U, Magnitude
+1.000e+00
+9.167e-01
+8.333e-01
+7.500e-01
+6.667e-01
+5.833e-01
+5.000e-01
+4.167e-01
+3.333e-01
+2.500e-01
+1.667e-01
+8.333e-02
+0.000e+00
ODB: buckle.odb Abaqus/Standard Version 6.8-1 Sat Dec 06 18:31:36 Eastern Standard Time
Step: Step-1
Mode 10: EigenValue = 3.18112E-02
Primary Var: U, Magnitude
Deformed Var: U Deformation Scale Factor: +1.000e+02
Y
Z
X

Fig.5 first 10 buckling modes

The eigenvalues of the first 10 buckling modes are closely spaced. As the eigenvalues of the buckling modes are corresponding to the critical forces at which buckling occurs, the closely spaced eigenvalues mean that the system is sensitive to the perturbation.

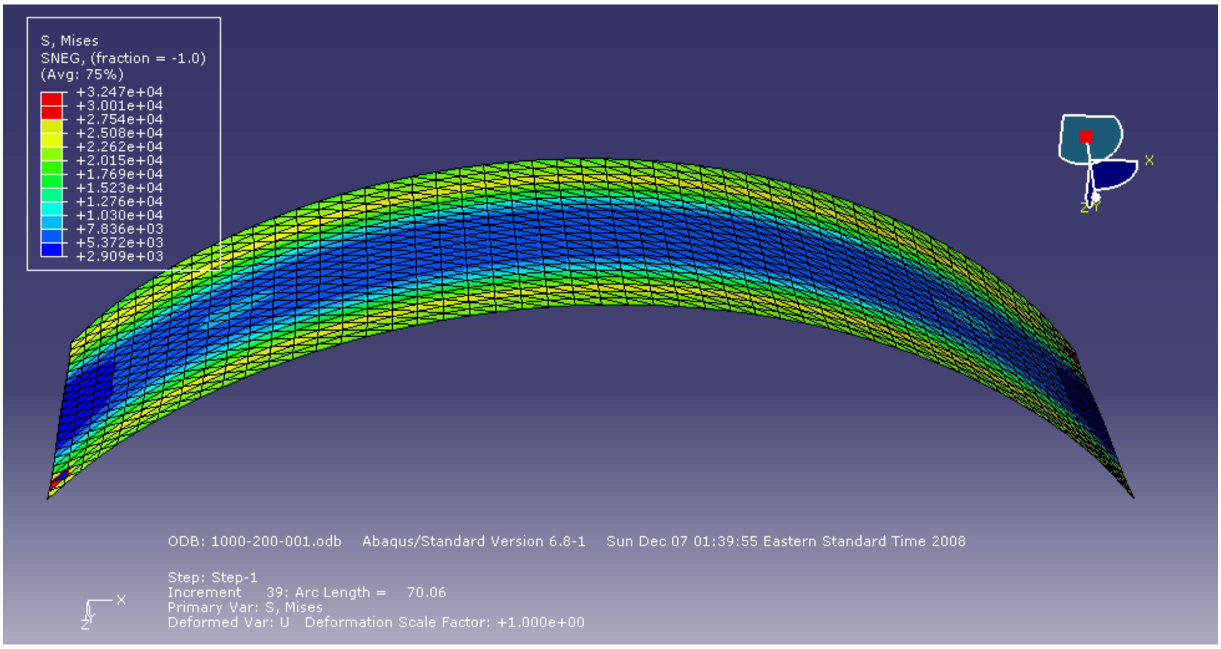
4. Post Buckling Analysis

For the geometrically nonlinear static problems that involve buckling or collapse behavior, Static Riks is a good method to study the buckling behavior. The first step is to introduce out of plane perturbation. The standard way to do this is to linearly add up displacements of different modes using different weights. This is another reason why we do the buckling mode analysis at first. Another method is to use Matlab to generate a Gaussian distribution perturbation by Randn function. The effects of these two methods are the same as far as I have tried.

The results turn out to be not very satisfactory in the end. For the eigenstrain $\varepsilon_{\max}^{*} \le 0.1$, the program is able to converge, but sometimes is sensitive to the value of initial arc length (a controlling parameter in Static Riks method). How to choose the value depends more on experience. And no matter how the perturbation is varied, the thin plate always buckles in the first mode. I have never seen higher mode buckling $\varepsilon_{\max}^{*} \le 0.1$ so far as I tried. For the eigenstrain $\varepsilon_{\max}^{*} > 0.1$, the program is not stable. The first mode buckling always appears at the beginning. However, as the increment steps increase, the thin plate gradually recovers to its original state and then begins to buckle on the opposite direction or exhibits higher mode buckling. Since the eigenvalues of this system are closely spaced, we expect transitions between different modes when small perturbation is added. But a natural question to ask is how the perturbation is introduced in the middle of computation? What I did is to only introduce an out plane perturbation at the initial state. Does the perturbation come from the computational error? The general tendency appears to be reasonable. But the problem is that after higher mode buckling emerges, the program is always terminated due to the same error which says all displacements are constricted to be zero. It means the program is divergent. Maybe the results generated when $\varepsilon_{\max}^{*} > 0.1$ are not reliable at all. And the last but not least problem I have found is that the thickness of the plate does not affect the results much. As Prof. Suo suggested during the presentation, shell elements may not be a good choice.

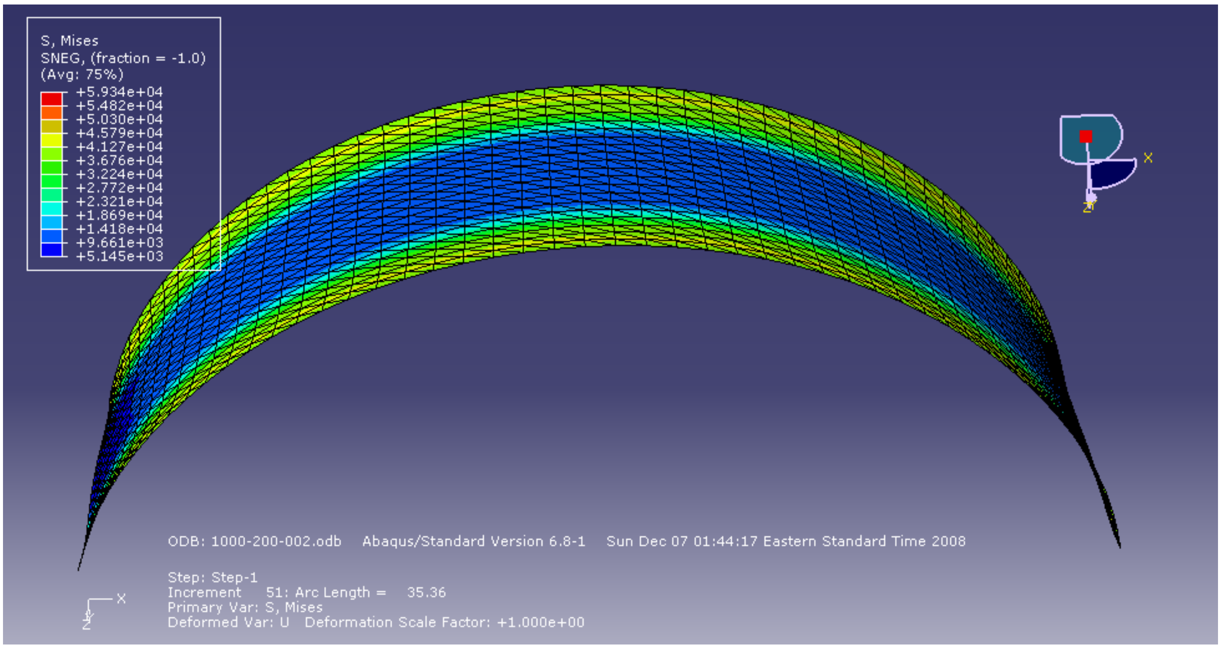
Considering $\varepsilon_{\max}^{*} = 0.1$ is already a large strain for leaf growth, some results corresponding to $\varepsilon_{\max}^{*} \le 0.1$ are temporarily employed to study the behavior of post buckling due to the eigenstain. A further understanding about how the Abaqus program works and its limitation for post buckling problem is required. See the pictures below:

- $\beta = 10, \quad \alpha = 1$



```
S, Mises
SNEG, (fraction = -1.0)
(Avg: 75%)
  +3.247e+04
  +3.001e+04
  +2.754e+04
  +2.508e+04
  +2.262e+04
  +2.015e+04
  +1.769e+04
  +1.523e+04
  +1.276e+04
  +1.030e+04
  +7.836e+03
  +5.372e+03
  +2.909e+03

ODB: 1000-200-001.odb    Abaqus/Standard Version 6.8-1    Sun Dec 07 01:39:55 Eastern Standard Time 2008

Step: Step-1
Increment     39: Arc Length =    70.06
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +1.000e+00
```

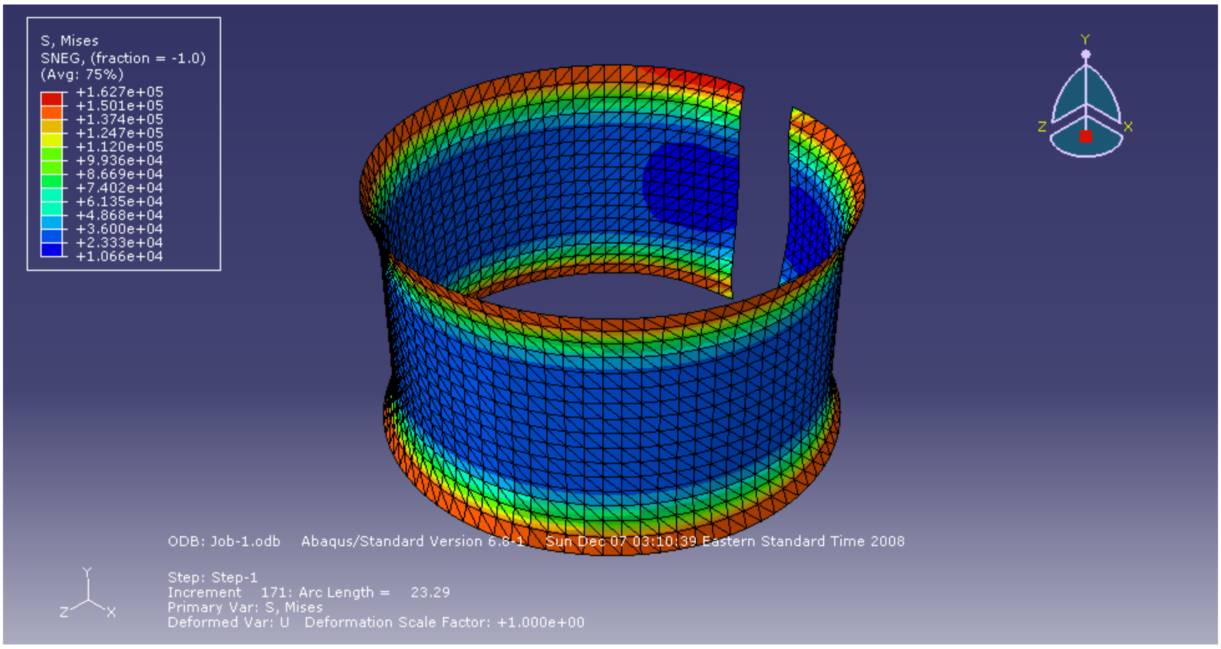
- $\beta = 10, \quad \alpha = 2$



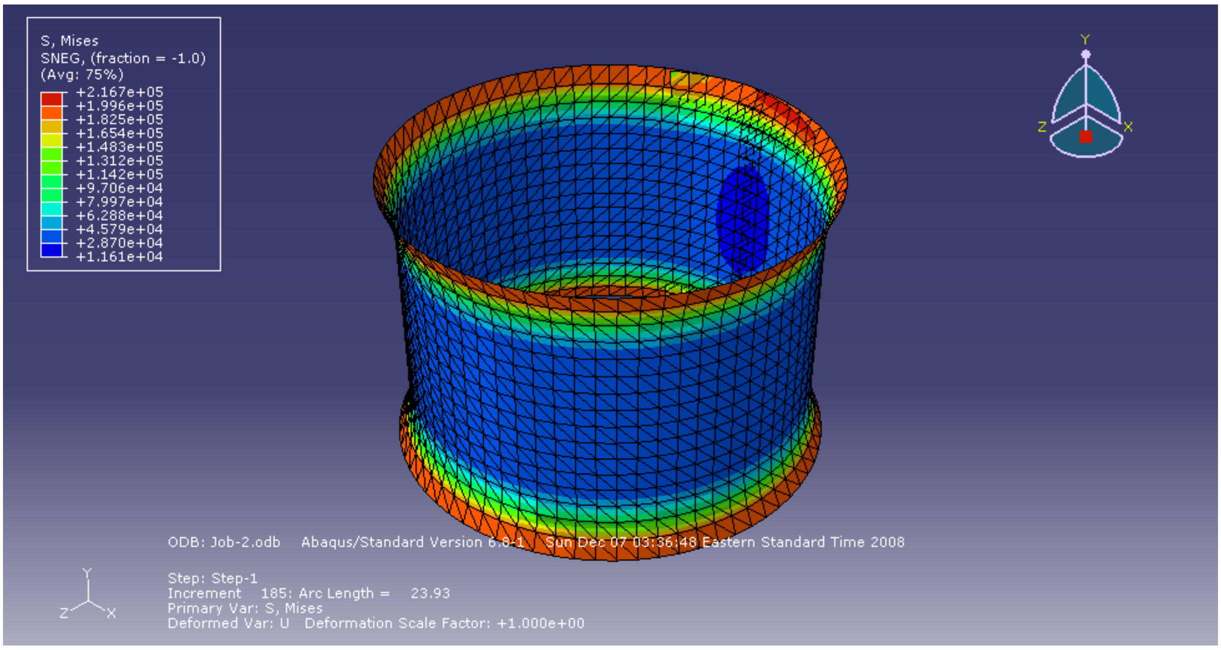
```
S, Mises
SNEG, (fraction = -1.0)
(Avg: 75%)
  +5.934e+04
  +5.482e+04
  +5.030e+04
  +4.579e+04
  +4.127e+04
  +3.676e+04
  +3.224e+04
  +2.772e+04
  +2.321e+04
  +1.869e+04
  +1.418e+04
  +9.661e+03
  +5.145e+03

ODB: 1000-200-002.odb    Abaqus/Standard Version 6.8-1    Sun Dec 07 01:44:17 Eastern Standard Time 2008

Step: Step-1
Increment     51: Arc Length =    35.36
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +1.000e+00
```

As the eigenstrain increases, the buckling deformation increases. Notice that at two free ends, there is a little bulge. It is because according to the form of eigenstrain, the edges tend to extend more than the middle, so at the free ends, the edges compress the middle, which results a bulge.

- $\beta = 10, \quad \alpha = 8$

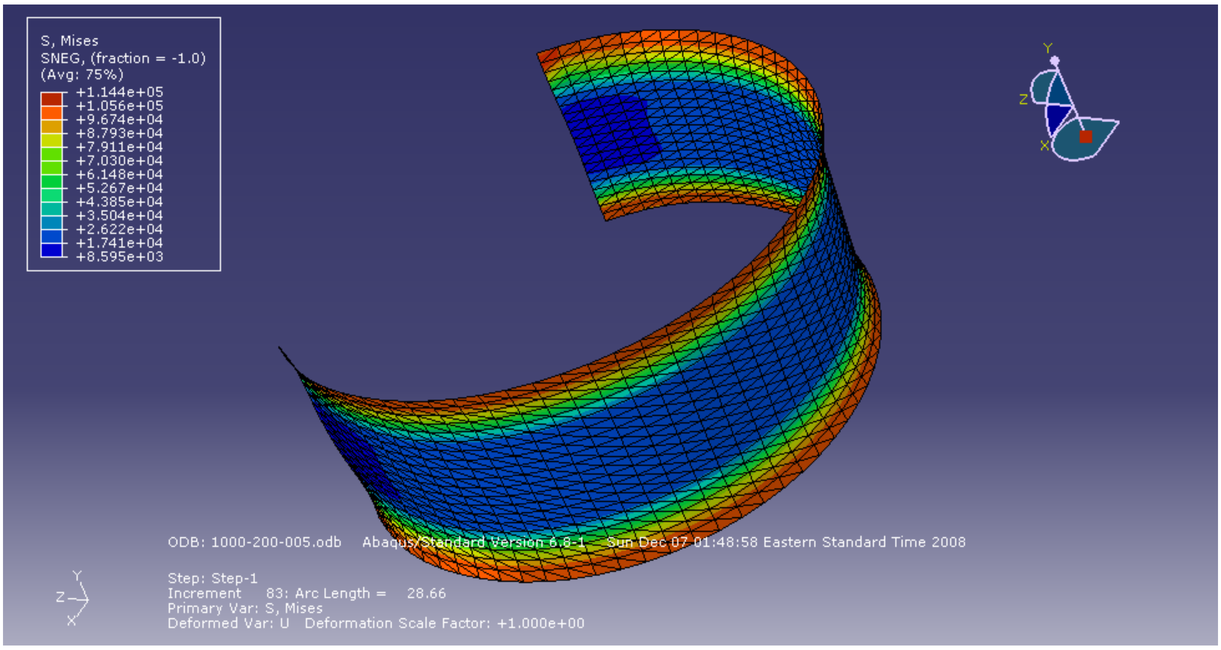


- $\beta = 10, \quad \alpha = 10$



As the eigenstrains increase more, the boundary effect becomes more obvious. And the plate can curve more than a circle. As to the boundary layer width, we can use the Gaussian curvature and transition between stretching energy and bending energy to do the scale analysis.

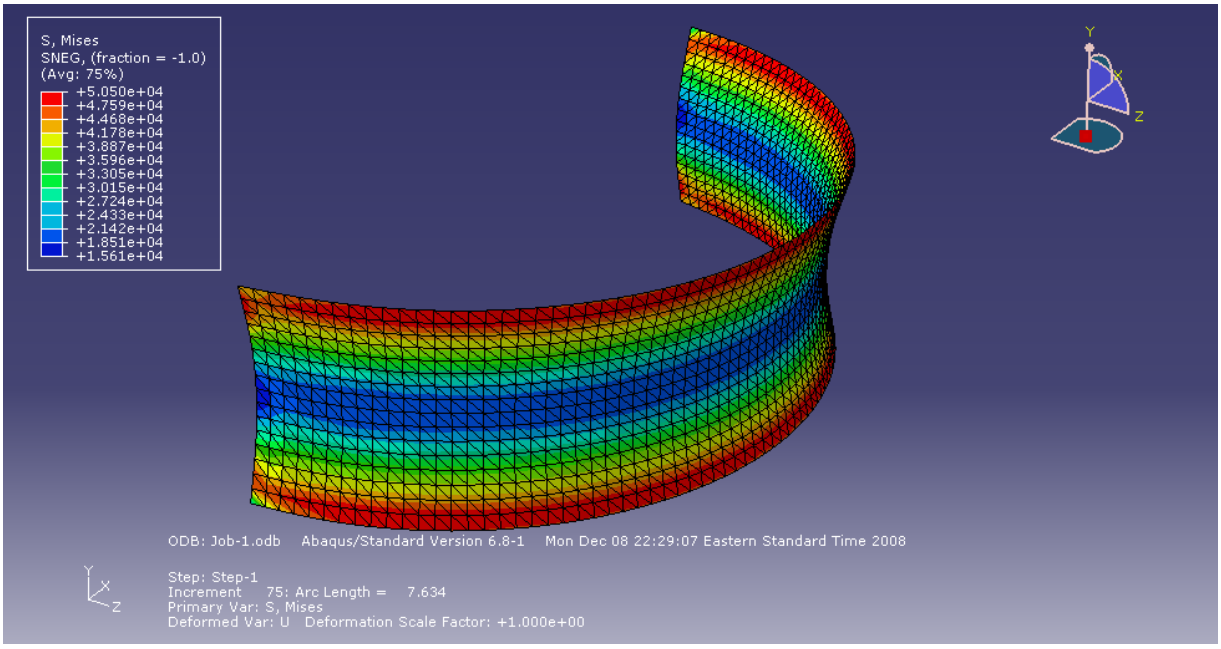
- $\beta = 10, \quad \alpha = 5$



```
S, Mises
SNEG, (fraction = -1.0)
(Avg: 75%)
  +1.144e+05
  +1.056e+05
  +9.674e+04
  +8.793e+04
  +7.911e+04
  +7.030e+04
  +6.148e+04
  +5.267e+04
  +4.385e+04
  +3.504e+04
  +2.622e+04
  +1.741e+04
  +8.595e+03
```

```
ODB: 1000-200-005.odb    Abaqus/Standard Version 6.8-1    Sun Dec 07 01:48:58 Eastern Standard Time 2008

Step: Step-1
Increment     83: Arc Length =    28.66
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +1.000e+00
```

- $\beta = 2, \quad \alpha = 5$



```
S, Mises
SNEG, (fraction = -1.0)
(Avg: 75%)
  +5.050e+04
  +4.759e+04
  +4.468e+04
  +4.178e+04
  +3.887e+04
  +3.596e+04
  +3.305e+04
  +3.015e+04
  +2.724e+04
  +2.433e+04
  +2.142e+04
  +1.851e+04
  +1.561e+04
```

```
ODB: Job-1.odb    Abaqus/Standard Version 6.8-1    Mon Dec 08 22:29:07 Eastern Standard Time 2008

Step: Step-1
Increment     75: Arc Length =    7.634
Primary Var: S, Mises
Deformed Var: U   Deformation Scale Factor: +1.000e+00
```

The parameter β controls the geometry across the width.

5. Conclusion

In this eigenstrain problem, at medium large deformation $\varepsilon^{*} \leq 0.1$, Abaqus is able to calculate the post buckling which contains nonlinear geometry, although the results are entirely the first mode. To add different perturbation at different steps may be a possible way to find transitions between different buckling modes. For large eigenstrain $\varepsilon^{*} > 0.1$, Static Riks method appears not capable to deal with the buckling due to eigenstrain. The author needs a further study in Abaqus.

References

[1] Mura, T., *Micromechanics of Solids with Defects*. 1982, The Hague: Martinus Nijhoff.

[2] H. Finkelmann, and H. Wermter, Abstr. Pap. Am. Chem. Soc. 219, 189 (2000); A. R. Tajbakhsh and E. M. Terentjev, cond-mat/0106138.

[3] Yu, Y.L., M. Nakano, and T. Ikeda, *Directed bending of a polymer film by light - Miniaturizing a simple photomechanical system could expand its range of applications.* Nature, 2003. **425**(6954): p. 145-145.

[4] Warner, M. and L. Mahadevan, *Photoinduced deformations of beams, plates, and films.* Physical Review Letters, 2004. **92**(13)