Progress Report

Our project is progressing well. We have been successful in creating and implementing axisymmetric elements (thanks to your help) in pyfem2. We are currently in the process of determining if our implementation of the elements is correct. We have begun constructing code for verification tests and implementing them into pyfem2.

In the project assignment details, it is outlined that a project must complete two of the three following items:

1. Computational implementation

2. Analytical formulation

3. Commercial or production code use

In keeping with these items, we've determined that we would like to focus on the computational implementation and the analytical formulation.

In addition to these portions of the project, we are exerting a great deal of effort toward the verification and error analysis portion of the project. We feel that this part of our project is most likely to produce re-usable results. We hope that the tests we create will help future developers in their development of axisymmetric elements in pyfem2.

The following is an outline of our final report. This summary captures what we aim to accomplish (or what has already been accomplished) by our work on this project.

Outline:

- Introduction

- Analytical formulation

    Petrov-Galerkin (Santos) formulation

    Galerkin (Fellipa) formulation

- Computational implementation

    CSDAX4F- Four node, full integration element

    CSDAX4R- Four node, reduced integration with hourglass control

    CSDAX4S- Four node, selectively reduced integration

- Verification problems

    Circular plate bending

    Thick walled pressure vessels

    Axially loaded bar (compression/tension test)

    Belleville washer

- Computational implementation of verification tests

    Testing framework with Python's *unittest*

    Test against verification problems under varying poisson's ratio and young's moduli

- Error analysis & element performance

    Detailed report of the performance of each element with respect to different potential operating conditions in the verification tests