









Protocol for mapping the variability in cell wall mechanical bending behavior in living leaf pavement cells

Wenlong Li ¹, Sedighe Keynia ¹, Samuel A. Belteton ^{2,3}, Faezeh Afshar-Hatam ¹,
Daniel B. Szymanski ^{2,3} and Joseph A. Turner ^{1,*†}

1 Mechanical and Materials Engineering, University of Nebraska-Lincoln, Lincoln, Nebraska, USA

2 Department of Botany and Plant Pathology, Purdue University, West Lafayette, Indiana, USA

3 Department of Biological Sciences, Purdue University, West Lafayette, Indiana, USA

*Author for communication: jaturner@unl.edu

†Senior author.

Additional Abaqus information



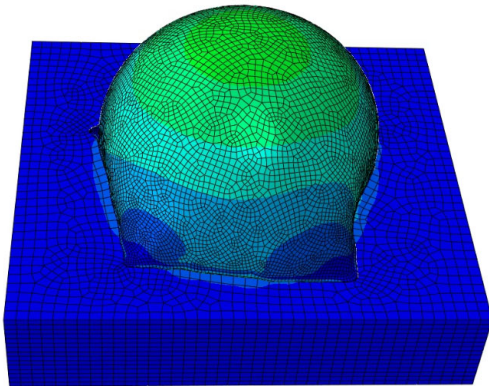
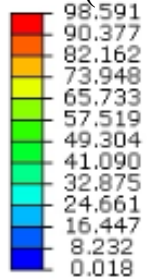
Dr. Wenlong Li created the Abaqus files shown here

- Abaqus 2019 version was used for the FE analysis.
- The Abaqus CAE and Python subscript files were created for each cell model in the manuscript which are titled by the figure number.
- For questions, please contact J. Turner (jaturner@unl.edu)

Fig. 1D, E (INP file)

Simulation results (Fig. 1 D)

Von Mises
Stress (MPa)



Turgor pressure = 0.6 MPa

Material properties (Neo Hooke) for Fig. 1D

1. Pavement cells: $E = 300 \text{ MPa}$, $\nu=0.47$

Relaxation time = 6.88s, $G_i/G_0 = 0.15$

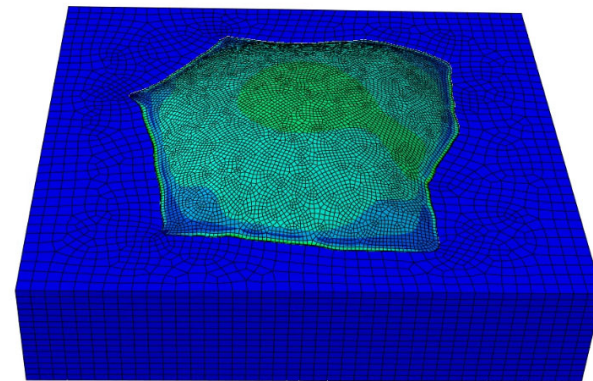
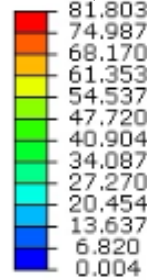
2. Surrounding materials:

$E = 100 \text{ MPa}$, $\nu=0.47$

Relaxation time = 6.88s, $G_i/G_0 = 0.15$

Simulation results (Fig. 1 E)

Von Mises
Stress (MPa)



Turgor pressure = 0.6 MPa

Material properties (Neo Hooke) for Fig. 1E

1. Pavement cells: $E = 30 \text{ MPa}$, $\nu=0.47$

Relaxation time = 6.88s, $G_i/G_0 = 0.15$

2. Surrounding materials:

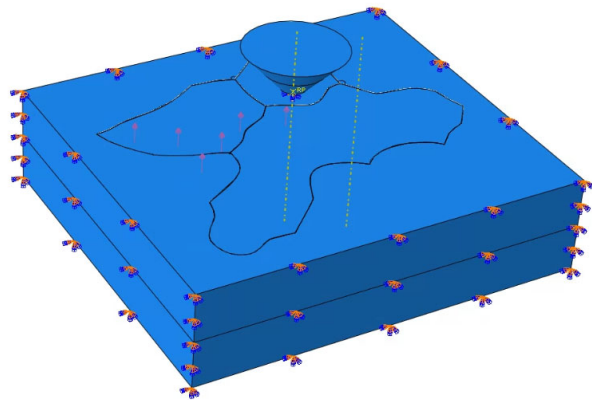
$E = 100 \text{ MPa}$, $\nu=0.47$

Relaxation time = 6.88s, $G_i/G_0 = 0.15$

Fig. 3D, E, F (INP file)

Only Fig. 3E is shown here, and other FE models have an analogous configuration.

Assembly and boundary condition.



Turgor pressure = 0.6 MPa

Material properties: (neo-Hookean)

1. All the pavement cells: $E = 360 \text{ MPa}$, $\nu = 0.47$

Relaxation time = 6.88 s, $G_i/G_0 = 0.25$

2. Middle lamellar and the surrounding materials:

$E = 100 \text{ MPa}$, $\nu = 0.47$

Relaxation time = 6.88 s, $G_i/G_0 = 0.25$

Simulation results

Von Mises
Stress (MPa)

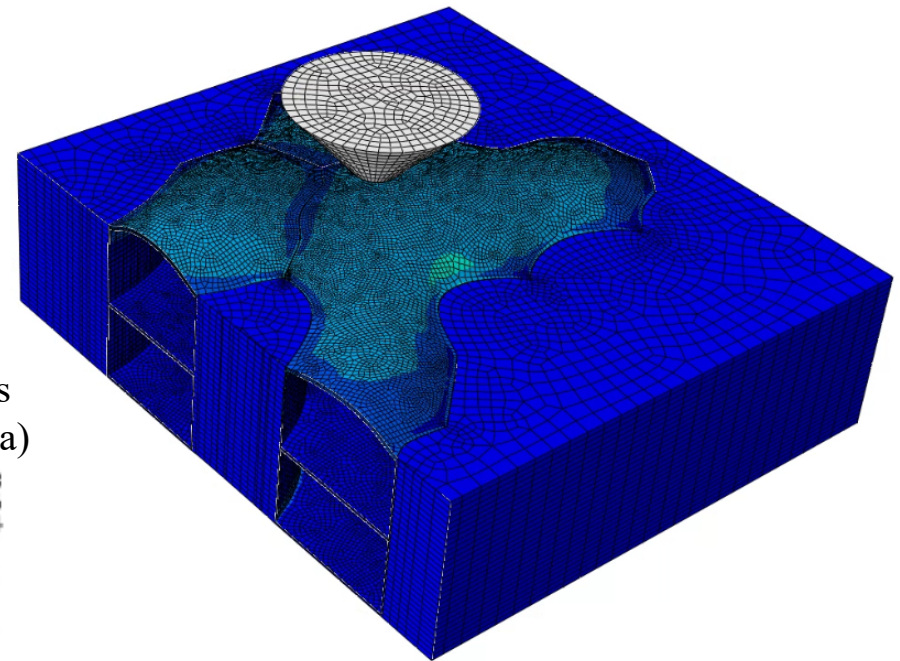
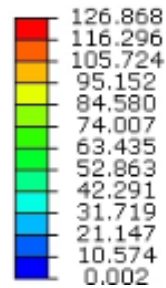
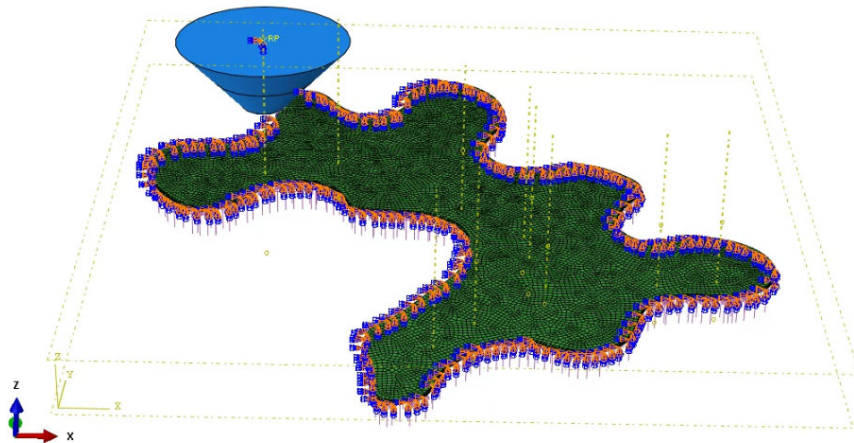


Fig. 5A, B, C (Abaqus CAE and Python files)

1. The CAE file contains one upper periclinal wall and one probe. The boundary and loading conditions are shown in the below image.



*Note: For shortening the calculation time, the uniform, isotropic material surrounding the cell will be removed. It has been demonstrated that this change will not influence the final iteration results.

2-1. The CAE file will be called by the iteration codes to automatically create INP files, and then are submitted to the Abaqus solver.

2-2. When the iterative calculation is finished, one has to manually re-locate the probe part in a new expected area, and update the node number in the center of the area for reading the relative expansion height. Addition, in the 'iteration code.py' file, update the variable slope150_Exp, slope1540_Exp and heightAfpref from the experiments.

2-3. After each FE calculation, the results will be written into the "parameters.txt" file. In the file, it shows E_o , turgor, height, and stiffness at each indentation depth.

*Note: E_o is not elastic modulus. To convert to elastic modulus, use $E = E_o * 2 * 2 * (1 + \text{Poisson's ratio})$.

3. Partial Python subscript codes (iteration codes)
See next page.

Fig. 5A,B,C (Python file)

3. Partial Python subscript codes (Iteration codes)

```
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from abaqus import *
from abaqusConstants import *
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
import numpy as np

# Load the module
mdb.openAuxMdb(pathName='Pavement_cell_3rd_remesh.cae')
mdb.copyAuxMdbModel(fromName='Model-10_remesh_p6_003base', toName='Model-10_remesh_p6_003base')
mdb.closeAuxMdb()

def runABAQUS(jobname,E_o,Turgor,E_i_ratio,maxh):
    poissonratio = 0.47
    G = 2.0*E_o
    bulk = 2.0*G*(1.0+poissonratio)/3.0/(1.0-2.0*poissonratio)
    D1 = 2.0/bulk
    # Assign these parameters to the module
    mdb.models['Model-10_remesh_p6_003base'].materials['ANTICLINAL-UNIFORM'].viscoelastic.setValues(
        domain=TIME, table=((E_i_ratio, 0.0, 6.88), ), time=PRONY)
    mdb.models['Model-10_remesh_p6_003base'].materials['ANTICLINAL-UNIFORM'].hyperelastic.setValues(
        table=((E_o, D1), ))
```

Call CAE file.

Fig. 5A, B, C (Python file)

3. Partial Python subscript codes (Iteration codes)

```
# -*- main -*-
# Please input experiment data
slope150_Exp = 23.4
slope1540_Exp = 40.4
heightAfpres = 1.077
# P4 and P6: 150 = 2; 1540 = 3.5; P3 150 = 3 and 1.5; 1540 = 4.5;
# P1 : 150 = 3, 1540 = 5.0; p5 : 150 = 2 , 1540 = 3;
error_slope150 = 2.0
error_slope1540 = 5.0
error_height = 0.2

i = 1
f = 0
j = 0
k = 0
number_limitation = 20

jobname = 'job-'+ str(i)
E_o = 101.5 #Note: E_o is the input parameter; the young's modulus = E_o*2*2*(1+0.47)
Turgor = 0.66
maxh = 1.3
E_pectin = 6.2
E_i_ratio = E_pectin/E_o

runABAQUS(jobname,E_o,Turgor,E_i_ratio,maxh)
slope150,slope1540,maxheight,slope240,slope360,slope970,slope1240 = extractForcDisp(jobname)
diffHeight = abs(maxheight-heightAfpres)
diffslope150 = abs(slope150_Exp-slope150)
diffslope1540 = abs(slope1540_Exp-slope1540)

f11 = open('parameters.txt', 'w')
f11.write(jobname)
f11.write('f')
f11.write(str(f))
```

← Input stiffness at 150nm and 1250 nm depth. Don't mind the variable name '1540'.

The other cells in Fig. 5 and Fig. 6 were used for the same iteration codes. But the FE structural model of each cell should be constructed in the CAE file.