Report. Python scripting to solve Biot's poroelastic equations with GetFEM

Anne-Cécile Lesage
MD Anderson Cancer Center
ajlesage@mdanderson.org

October 20, 2021

Abstract

This report describes python scripting to solve Biot's poroelastic equations with GetFEM

I. Biot theory equations

We propose to use the equations proposed by Paulsen et al. [cite] for consolidation in soft tissue. They are adapted from the poroelastic equations derived by Biot for soil consolidation [cite]. The governing equations can be written as

$$\nabla.G\nabla\mathbf{u} + \nabla\frac{G}{1-2\nu}(\nabla.\mathbf{u}) - \alpha\nabla p = 0 \tag{1}$$

$$\alpha\frac{\partial}{\partial t}(\nabla.\mathbf{u}) + \frac{1}{S}\frac{\partial p}{\partial t} - \nabla.k\nabla p = 0 \tag{2}$$

where
G shear modulus (Pa);
ν Poisson's ratio;
$\mathbf{u}$ displacement vector (m);
p pore fluid pressure (Pa);
α ratio of fluid volume extrcated to volume change of the tissue under compression;
k hydraulic conductivity (m^3s/kg);
$1/S$ amount of fluid which can be forced into the tissue under constant volume ($1/Pa$)
The equation assumes that the solid tissue behaves in a linearly elastic fashion and the pore fluid is incompressible. Equation 1 is equivalent to

$$\nabla.\overline{\sigma}_e - \alpha\nabla p = 0 \tag{3}$$

with $\overline{\sigma}_e = \lambda tr(\overline{\epsilon}) + 2\mu\overline{\epsilon}$ and $\overline{\epsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$.

II. Finite element discretization

i. Time discretization

After volume integration on test functions ϕ_i and using the ϕ_j test function for the Galerkin discretization, we obtain the following weak form of equations (1) and (2)

$$\sum_j \mathbf{u}_j < G\nabla\phi_j.\nabla\phi_i > + \sum_j \mathbf{u}_j. < \nabla\phi_j \frac{G}{1-2\nu}\nabla\phi_i >$$
$$+ \sum_j p_j < \nabla\phi_j\phi_i >= \oint G\mathbf{n}\phi_i ds + \oint \frac{G}{1-2\nu}\mathbf{n}(\nabla.\mathbf{u})\phi_i ds \quad (4)$$

$$\sum_j \frac{\partial \mathbf{u}_j}{\partial t} + \sum_j p_j(k\nabla\phi_j.\nabla\phi_i) = \oint k\mathbf{n}.\nabla p\phi_i ds \quad (5)$$

Equations (4) and (5) are integrated in time using the simple two-point weighting

$$\int_{t_n}^{t_{n+1}} f(t)dt = \Delta t[\theta f(t_{n+1} + (1-\theta)f(t_n)] \quad (6)$$

with a time discretization $\Delta t = t_{n+1} - t_n$ and $0 \leq \theta \leq 1$. It gives the follwing matrix equations

$$AU^{n+1} = BU^n + C^{n+\theta} \quad (7)$$

with

$$U_j^n = \begin{Bmatrix} u_{x_j}(t_n) \\ u_{y_j}(t_n) \\ u_{z_j}(t_n) \\ p_j(t_n \end{Bmatrix} \quad (8)$$

$$C_i^{n+\theta} = \begin{Bmatrix} \hat{x}\oint \sigma_s(t_{n+\theta}).\hat{n}\phi_i ds \\ \hat{y}\oint \sigma_s(t_{n+\theta}).\hat{n}\phi_i ds \\ \hat{z}\oint \sigma_s(t_{n+\theta}).\hat{n}\phi_i ds \\ \Delta t \oint k\nabla p(t_{n+\theta}).\phi_i ds \end{Bmatrix} \quad (9)$$

$A_{ij} =$
$$\begin{matrix} \theta G < \frac{2(1-\nu)}{1-2\nu}\delta_{xx} + \delta_{yy} + \delta_{zz} > & \theta G < \frac{2\nu}{1-2\nu}\delta_{yx} + \delta_{xy} > & \theta G < \frac{2\nu}{1-2\nu}\delta_{zx} + \delta_{xz} > & \theta < \delta x > \\ \theta G < \frac{2\nu}{1-2\nu}\delta_{xy} + \delta_{yx} > & \theta G < \frac{2(1-\nu)}{1-2\nu}\delta_{yy} + \delta_{xx} + \delta_{zz} > & \theta G < \frac{2\nu}{1-2\nu}\delta_{zy} + \delta_{yz} > & \theta < \delta y > \\ \theta G < \frac{2\nu}{1-2\nu}\delta_{xz} + \delta_{yx} > & \theta G < \frac{2\nu}{1-2\nu}\delta_{yz} + \delta_{zy} > & \theta G < \frac{2(1-\nu)}{1-2\nu}\delta_{zz} + \delta_{xx} + \delta_{yy} > & \theta < \delta z > \\ < \delta_x > & < \delta_y > & < \delta_z > & \theta\Delta tk < \delta_{xx} + \delta_{yy} + \delta_{zz} > \end{matrix}$$

with $\delta_k = (\partial\phi_j/\partial k)\phi_i$, $\delta_{kl} = (\partial\phi_j/\partial k)(\partial\phi_i/\partial l)$, where k and l take on the values x, y, or z. B is almost identical to A with the expection of changing θ for $\tilde{\theta} = \theta - 1$.

III. Numerical tests. Python script with GetFEM

i. Consolidation 2D test 1

Let us examine the particular case of a column of soil supporting a load $p_0 = -\sigma_z$ and confined laterally in a rigid stealth so that no lateral expansion can occur. It is assumed also that no water can escape laterally or through the bottom while it is free to escape at the upper surface by applying the load through a very porous slab.

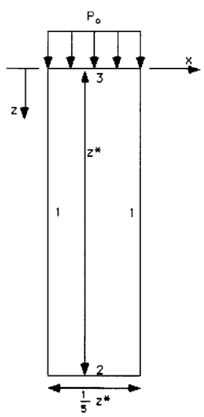

Figure 1: *Consolidation test.*

The problem is illustrated in Figure 1.
Material properties $G = 1.0 \times 10^7$ Pa, $\nu = 0.3$,
$k = 1.0 \times = 1.0 \times 10^{-13} m^3 s/kg$;
Running properties $dt = 1.0 \times 10^3 s$, $\theta = 1$,
steps = 1.0×10^3;
Boundary 1 $u_n = 0$, $\sigma_t = 0$, $\frac{\partial p}{\partial n} = 0$;
Boundary 2 $\sigma_t = 0$, $u_n = 0$, $\frac{\partial p}{\partial n} = 0$;
Boundary 3 $\sigma_t = 0$, $\sigma_n = p_0$, $p = 0$;
Initial conditions at $t = 0$, $\mathbf{u} = 0$ and $p = p_0$
where σ_t and σ_n are the shear and normal stresses, respectively and u_n is the normal displacement at the boundary face.

The analytic solution to this problem for a saturated media is given by Biot [Biot, 1941]. Lets rewrite the analytic solution derivation.

Take the z axis positive downward; the only component of displacement in thisase will be w. Both w and the pressure p will depend only on the coordinate z and the time t. The differential eqs. (1) and (2) become

$$\frac{1}{a}\frac{\partial^2 w}{\partial z^2} - \alpha \frac{\partial p}{\partial z} = 0, \tag{10}$$

$$k\frac{\partial^2 p}{\partial z^2} = \alpha \frac{\partial^2 w}{\partial z \partial t} + \frac{1}{S}\frac{\partial p}{\partial t} \tag{11}$$

with $a = \frac{1-2\nu}{2G(1-\nu)}$ called the final compressibility.

The stress σ_z throughout the loaded column is a constant. Thus we have from the stress equations derived by Biot in [Biot, 1941].

$$\sigma_z = 2G\left(\frac{\partial w}{\partial z} + \frac{\nu div(\mathbf{u})}{1-2\nu}\right) - \alpha p \tag{12}$$

$$p_0 = -\sigma_z = -\frac{1}{a}\frac{\partial w}{\partial z} + \alpha p$$

Equation 12 implies that

$$-\frac{1}{a}\frac{\partial w}{\partial t \partial z} = \alpha \frac{\partial p}{\partial t} \tag{13}$$

Equation 13 carried into eq.11 gives

$$\frac{\partial^2 p}{\partial z^2} = \frac{1}{c}\frac{\partial p}{\partial t} \tag{14}$$

with

$$\frac{1}{c} = \alpha^2 \frac{a}{k} + \frac{1}{Sk} \tag{15}$$

Constant c is called the consolidation constant. Equation 14 shows the important result that the water pressure satisfies the well-known equation of heat conduction. This equation along with the boundary and the initial conditions leads to a complete solitipn of the problem of consolidation. Taking the height of the soil to be h and z=0 at the top we have the boundary conditions

$$p = 0 \qquad \textbf{for } z = 0 \tag{16}$$

$$\frac{\partial p}{\partial z} = 0 \qquad \textbf{for } z = h \tag{17}$$

The first condition expresses that the pressure of the water under the load is zero because the permeability of the slab through whcih the load is applied is assumed to be large with respect to that of the soil. The second condition expresses that no water escapes through the bottom. The initial condition is that the change of water content θ is zero when the load is applied beacuse the water must escape with a finite velocity. Hence we have

$$\theta = \alpha \frac{\partial w}{\partial z} + \frac{p}{S} \textbf{ for } t = 0 \tag{18}$$

Carrying this in eq.12 we derive the initial value of the water pressure

$$p = p_0 / (\frac{1}{\alpha a S} + \alpha) = p_0 \; for \; t = 0 \tag{19}$$

with values given by [**?**] for brain tissue. The solution of the differential equation 14 with boundary conditions (eq. 16 and 17) and initial condition (eq. 19) may me witten in the form of a series

$$p = \frac{4}{\pi} p_0 \sum_{i\ odd} \{ \frac{1}{i} exp[-(\frac{i\pi}{2h})^2 ct] sin \frac{i\pi z}{2h} \} \tag{20}$$

ii. Python script. Test 1

Import library GetFem. First line is to give the path to Ubuntu 20

```
sys.path.append('/usr/local/lib/python3.8/site-packages/getfem')

import numpy as np

import getfem as gf
```

Create parameter

```
# solve classical consoludation column for poroelastic material
# see Biot 1941 and Paulsen 1999
#
# Physical parameters
#
epsilon = 0.01     # Thickness of the plate (
nu    = 0.3        # Poisson ratio
k     = 1.0E-13    # hydraulic conductivity
G     = 1.0E-7     # shear modulus
alpha = 1.0        # ratio of fluid volume extracted to volume change of the tissue under compression
#
# Numerical parameters
#
dt    = 1.0E3             # time step
theta = 1.0
nt    = 1000              # number of time step
elements_degree = 1       # Degree of the finite element methods
export_mesh = True        # Draw the mesh after mesh generation or not
```

Import Mesh format GiD. Create mesh does not work. GMSH format is too heavy and complex.
Export mesh to VTK format
Print mesh parameters. Elements Number. Point numbers.

```
print('Read Mesh GiD')
#gf.util('trace level', 2)   # No trace for mesh generation
#mesh = gf.Mesh('generate', mo, h, 2)

m=gf.Mesh('import','gid','mesh2d_h0_2.msh')
```

```
# print mesh vtk format
if (export_mesh):
    m.export_to_vtk('mesh.vtk');
    print('\nYou can view the mesh for instance with');
    print('mayavi2 -d mesh.vtk -f ExtractEdges -m Surface \n');
```

```
print('elements number=%d, points number=%d' % \
      (m.nbcvs(),m.nbpts()))
```

Define three boundary regions. Adapted from script example demo_thermo_elastic*py

```
#
# Boundary selection
#
fb1 = m.outer_faces_with_direction([ 1., 0.], 0.01) # Right boundary
fb2 = m.outer_faces_with_direction([-1., 0.], 0.01) # Left boundary
fb3 = m.outer_faces_with_direction([0.,  1.], 0.01) # Top boundary
fb4 = m.outer_faces_with_direction([0., -1.], 0.01) # Bottom boundary

RIGHT_BOUND=1; LEFT_BOUND=2; TOP_BOUND=3; BOTTOM_BOUND=4;

# Define mesh region
m.set_region( RIGHT_BOUND, fb1)
m.set_region(LATERAL_BOUND, fb2)
m.set_region(    TOP_BOUND, fb3)
m.set_region(BOTTOM_BOUND, fb4)

mesh.region_merge(LATERAL_BOUND, RIGHT_BOUND)
```

Define FEM

```
#
# Definition of finite elements methods and integration method
#
print('Define fem')
mfu = gf.MeshFem(m, 2)  # Finite element for the elastic displacement
mfu.set_classical_fem(elements_degree)
mfp = gf.MeshFem(m, 1)  # Finite element for pressure field
mfp.set_classical_fem(elements_degree)
mim = gf.MeshIm(m, elements_degree*2)   # Integration meth

#
# Model definition
#

md=gf.Model('real');
md.add_fem_variable('u', mfu)       # Displacement of the structure
md.add_fem_variable('press', mfp)   # Pressure
```

Define boundary condition on top boundary

```
# Homegeneous Dirichlet condition for pressure on top boundary
md.add_Dirichlet_condition_with_multipliers(mim, 'press', elements_degree-1, TOP_BOUND)

# Condition on sigma_n on top boundary
md.add_initialized_data('P0', [P0])
md.add_source_term_brick(mim, 'u', 'P0', TOP_BOUND)
```

Define matrix A

```
md.add_initialized_data('G', [G])
md.add_initialized_data('Gnu', [Gnu])
md.add_initialized_data('dt', [dt])
md.add_initialized_data('k', [k])

md.add_linear_term(mim, "G*Grad_u.Grad_Test_u+Gnu*Div_u.Grad_Test_u+Grad_press.Test_u");

md.add_linear_term(mim, "Grad_u*Test_press+k*dt*Grad_press.Grad_Test_press");
```

References

[Biot, 1941] Biot, M.A. (1941). General theory of three dimensional consolidation *J. Appl. Phys.*, vol. 12, pp. 155-164, 1941.