14 – finite element method – density growth - theory

balance equations for open systems - mass

\[ \dot{D_t} \rho_0 = \text{Div}(\mathbf{R}) + \mathcal{R}_0 \]

- mass flux \( \mathbf{R} \)
  - cell movement (migration)
- mass source \( \mathcal{R}_0 \)
  - cell growth (proliferation)
  - cell division (hyperplasia)
  - cell enlargement (hypertrophy)

biological equilibrium


growing bone as open system

where are we???

balance equations for open systems - momentum

- volume specific version
  \[ \dot{D_t} (\rho_0 \mathbf{v}) = \text{Div}(\mathbf{P} + \mathbf{v} \otimes \mathbf{R}) + [\mathbf{b}_0 + \mathbf{v} \mathcal{R}_0 - \nabla \cdot \mathbf{v} \cdot \mathbf{R}] \]
- subtraction of weighted balance of mass
  \[ \mathbf{v} \dot{D_t} \rho_0 = \text{Div}(\mathbf{v} \otimes \mathbf{R}) + \mathbf{v} \mathcal{R}_0 - \nabla \cdot \mathbf{v} \cdot \mathbf{R} \]
- mass specific version
  \[ \rho_0 \dot{D_t} \mathbf{v} = \text{Div}(\mathbf{P}) + \mathbf{b}_0 \]

mechanical equilibrium

growing bone as open system
constitutive equations for open systems

- free energy
  \[ \psi_0 = \left( \frac{\rho_0}{\rho_0^*} \right)^n \psi_0^{\text{neo}}(F) \]
- stress
  \[ P = \left( \frac{\rho_0}{\rho_0^*} \right)^n P^{\text{neo}}(F) \]
- mass flux
  \[ R = R_0 \nabla_X \rho_0 \]
- mass source
  \[ \mathcal{R}_0 = \left( \frac{\rho_0}{\rho_0^*} \right)^{-m} \psi_0(F) - \psi_0^* \]

coupling of growth and deformation

Recipe for finite element modeling

\[ \begin{align*}
D_t \rho_0 &= \text{Div}(R) + \mathcal{R}_0 \\
\rho_0 D_t \mathbf{v} &= \text{Div}(P) + b_0
\end{align*} \]

from continuous problem...

- temporal discretization
- spatial discretization
- staggered/simultaneous
- linearization

... to linearized discrete initial boundary value problem
key transformation - from strong form to weak form (1D)

- strong / differential form
  \[ \sum f = f^{\text{int}} + f^{\text{ext}} = 0 \quad f^{\text{int}} = P'(\varphi) \]
- strong form / residual format
  \[ R(\varphi) = P'(\varphi) + f^{\text{ext}} = 0 \]
- weak / integral form - nonsymmetric \( \forall \delta \varphi \)
  \[ G(\delta \varphi; \varphi) = \int \delta \varphi \cdot [P'(\varphi) + f^{\text{ext}}] \, dx = 0 \]
- integration by parts
  \[ \int \delta \varphi \cdot P' \, dx = \int \left[ \delta \varphi \cdot P \right]_x^l \, dx - \int \delta \varphi' \cdot P \, dx \]
- integral theorem & neumann bc's
  \[ \int \left[ \delta \varphi \cdot P \right]_x^l \, dx = \delta \varphi \cdot P^{l=0}_r \]
- weak form / integral form - symmetric \( \forall \delta \varphi \)
  \[ \int \delta \varphi' \cdot P \, dx - \delta \varphi \cdot P^{l=0}_r = \int \delta \varphi \cdot f^{\text{ext}} = 0 \]

finite elements – integration point based

residual equation...

- strong / differential form
  \[ R^\varphi = \rho_0 D_t \varphi - \text{Div}(P) - b_0 = 0 \quad \text{in } B_0 \]
- dirichlet / essential boundary conditions (displacements)
  \( \varphi - \varphi = 0 \) on \( \partial B_0^\varphi \) with \( \partial B_0^\varphi \cup \partial B_0^{T\varphi} = \partial B_0 \)
- neumann / natural boundary conditions (tractions)
  \( P \cdot N - T^\varphi = 0 \) on \( \partial B_0^{T\varphi} \) and \( \partial B_0^\varphi \cap \partial B_0^{T\varphi} = \emptyset \)

... and boundary conditions

finite elements – integration point based

from equilibrium equation...

- start with nonlinear mechanical equilibrium equation
  \[ \rho_0 D_t \varphi \approx 0 \quad \text{quasi-static} \]
  \[ \rho_0 D_t \varphi = \text{Div}(P) + b_0 \approx 0 \quad \text{no gravity} \]
- cast it into its residual format
  \[ R^\varphi(\varphi) = 0 \quad \text{in } B_0 \]
- with residual
  \[ R^\varphi = \rho_0 D_t \varphi - \text{Div}(P) - b_0 \]

... to residual format

finite elements – integration point based

from strong form...

- strong / differential form
  \[ R^\varphi = \rho_0 D_t \varphi - \text{Div}(P) - b_0 = 0 \quad \text{in } B_0 \]
- multiplication with test function & integration
  \[ G^\varphi (\delta \varphi; \varphi) = \int_{B_0} \delta \varphi \cdot R^\varphi \, dV = 0 \quad \forall \delta \varphi \quad \text{in } H_0^1(B_0) \]
- weak form / nonsymmetric \( \forall \delta \varphi \neq 0 \) second derivative
  \[ G^\varphi = \int_{B_0} \delta \varphi \cdot \rho_0 D_t \varphi \, dV - \int_{B_0} \delta \varphi \cdot \text{Div}(P) \, dV - \int_{B_0} \delta \varphi \cdot b_0 \, dV \]

... to nonsymmetric weak form

finite elements – integration point based
from non-symmetric weak form...

- integration by parts
  \[ \int_{B_0} \delta \varphi \text{Div}(P) dV = \int_{B_0} \text{Div}(\delta \varphi \cdot P) dV - \int_{B_0} \nabla \delta \varphi : P dV \]
- gauss theorem & boundary conditions
  \[ \int_{B_0} \text{Div}(\delta \varphi \cdot P) dV = \int_{\partial B_0} \delta \varphi \cdot P \cdot N dA - \int_{B_0} \delta \varphi \cdot \nabla \varphi : T \varphi dA \]
- weak form / symmetric  
  \[ G = \int_{B_0} \delta \varphi \rho_0 \, dV + \int_{B_0} \nabla \delta \varphi : \dot{P} dV - \int_{\partial B_0} \delta \varphi \cdot \dot{T} \varphi \cdot T \varphi dA - \int_{B_0} \delta \varphi \cdot b_0 dV \]

... to symmetric weak form

finite elements – integration point based

from discrete weak form...

- discrete weak form
  \[ G = \delta \varphi_J \cdot R_j^c(\varphi_{n+1}) = 0 \quad \forall \delta \varphi_J \]
- discrete residual format
  \[ R_j^c(\varphi_{n+1}) = 0 \quad \forall \, J = 1, \ldots, n_{np} \]
- discrete residual
  \[ R_j^c = A^{rel}_{rel} \int_{B_0} N_j^i N_\varphi D_I \varphi_{n+1} dV + \int_{B_0} \nabla N_j^i \cdot P_{n+1} dV \]
  \[ - \int_{\partial B_0} N_j^i \nabla \varphi_{n+1} dA - \int_{B_0} N_j^i b_{0n+1} dV \]

... to discrete residual

finite elements - integration point based

integration point based solution of balance of mass

loop over all time steps
  
  global newton iteration
    
    loop over all elements
      
      loop over all quadrature points
        
        local newton iteration $p_{n+1}$
          
          determine element residual & tangent
          
          determine global residual and tangent

  
  determine $\varphi_{n+1}$
  
  determine state of biological equilibrium

staggered solution of density and displacements

finite elements - integration point based

finite elements - integration point based

spatial discretization

- discretization
  \[ B_0 = \bigcup_{e=1}^{n_{el}} B_e^c \]
- interpolation of test functions
  \[ \delta \varphi^h_{B_0} = \sum_{i=1}^{n_{en}} N_i^j \delta \varphi_j \in H^1_0(B_0) \]
  \[ \nabla \delta \varphi^h_{B_0} = \sum_{i=1}^{n_{en}} \delta \varphi_j \otimes \nabla N_i^j \]
- interpolation of trial functions
  \[ \varphi^h_{B_0} = \sum_{i=1}^{n_{en}} N_i^l \varphi_l \in H_0(B_0) \]
  \[ \nabla \varphi^h_{B_0} = \sum_{i=1}^{n_{en}} \varphi_l \otimes \nabla N_i^l \]

... to discrete weak form
%% loop over all load steps %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for is = (nsteps+1):(nsteps+inpstep);
    iter = 0;  residuum = 1;
    %%% global newton-raphson iteration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while residuum > tol
        iter=iter+1;
        R = zeros(ndof,1); K = sparse(ndof,ndof);
        %%% loop over all elements %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        for ie = 1:nel
            [Ke,Re,Ie] = element1(e_mat(ie,:),e_spa(ie,:),i_var(ie,:),mat);
            [K, R, I] = assm_sys(edof(ie,:),K,Ke,R,Re,I,Ie);
        end
        %%% loop over all elements %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        u_inc(:,2)=dt*u_pre(:,2);  R = R - time*F_pre;   dofold = dof;
        [dof,F] = solve_nr(K,R,dof,iter,u_inc);
        residuum= res_norm((dof-dofold),u_inc);
    end
    %%% global newton-raphson iteration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    time = time + dt;   i_var = I;   plot_int(e_spa,i_var,nel,is);
end

% discrete residual
R^φ = \sum_{i=1}^{nE} \int_{B_i} \nabla N_i^j \cdot P \, n \, dV

% residual of mechanical equilibrium/balance of momentum
check in matlab!

righthand side vector for global system of equations

% linearized residual
K^φ_\varphi \frac{∂R^φ_j}{∂φ_L} = \sum_{i=1}^{nE} \int_{B_i} N_i^j \rho D(\varphi) (D(\varphi))^t N_i^j \, dV
+ \sum_{i=1}^{nE} \int_{B_i} \nabla N_i^j \cdot D_\varphi (\varphi) N_i^j \, dV

4th order tensor - derivatives of 2nd order tensors wrt 2nd order tensor

linearization of residual wrt nodal dofs

iteration matrix for global system of equations
finite elements - integration point based

quads_2d.m

% element stiffness matrix Ke, residual Re, internal variables Ie
function [Ke,Re,Ie]=element1(e_mat,e_spa,i_var,mat)

% integration points
indy=[1;3;5;7];  ey_mat=e_mat(indy);
indx=[2;4;6;8];  ex_mat=e_mat(indy);
% shape functions and derivatives in isoparametric space
N(:,3)=(1+eta).*N(:,3);N(:,4)=(1-eta).*N(:,4);
N(:,1)=(1-eta).*N(:,1);N(:,2)=(1+eta).*N(:,2);
% assemble local element contributions to global tangent & residual
% assemble element stiffness matrix Ke, residual Re, internal variables Ie

finite elements - integration point based

finite elements - integration point based
@ integration point level

- constitutive equations - given $F = \nabla \varphi$ calculate $P$

- update density for current stress state from $\rho_0$ and $D_t \rho_0 = \left[ \frac{\rho_0}{\rho_0} \right]^{-m} \psi_0(F) - \dot{\psi}_0$ calculate $\rho_{0n+1}$

- calculate first piola kirchhoff stress of solid material $P^\text{neo}(F) = \mu_0 F + [\lambda_0 \ln(\det(F))] - \mu_0 ] F^{-t}$

- calculate first piola kirchhoff stress of porous material $P(F') = \left[ \frac{\rho_0}{\rho_0} \right]^{n} P^\text{neo}$

stress for righthand side vector

finite elements - integration point based

@ integration point level

- constitutive equations - given $F$ calculate $D_F P$

\[
D_F P = \partial_F P - \partial_{p_0} P \left[ \partial_{p_0} R_0 \right]^{-1} \partial_F R_0
\]

with

\[
\partial_F P = \left[ \frac{\rho_0}{\rho_0} \right]^{n} \mu I \otimes I + \lambda F^{-t} \otimes F^{-t} - [\lambda \ln J - \mu] F^{-t} \otimes F^{-1}
\]

\[
\partial_{p_0} P = \left[ \frac{\rho_0}{\rho_0} \right]^{n} P
\]

\[
\partial_{p_0} R_0 = \left[ \frac{\rho_0}{\rho_0} \right]^{-m} \frac{1}{\rho_0} [n - m] \psi_0
\]

\[
\partial_F R_0 = \left[ \frac{\rho_0}{\rho_0} \right]^{-m} P
\]

tangent for iteration matrix

finite elements - integration point based

@ integration point level

- stress calculation @ integration point level

- check in matlab!

\[
P(F) = \left[ \frac{\rho_0}{\rho_0} \right]^{n} \mu_0 F + [\lambda_0 \ln(\det(F))] - \mu_0 ] F^{-t}
\]

finite elements - integration point based

@ integration point level

- tangent operator / constitutive moduli

check in matlab!

\[
A = D_F P = \partial_F P - \partial_{p_0} P \left[ \partial_{p_0} R_0 \right]^{-1} \partial_F R_0
\]

tangent for iteration matrix

finite elements - integration point based

@ integration point level

- linearization of stress wrt deformation gradient
finite elements - integration point based

recipe for temporal discretization

**implicit euler backward**
- evolution of density
  \[ D_t \rho_0 = \frac{1}{\Delta t} [\rho_{0,n+1} - \rho_{0,n}] \]
  finite difference approximation
- discrete residual
  \[ R^\rho_{n+1} = \frac{1}{\Delta t} [\rho_{0,n+1} - \rho_{0,n}] - [\rho_{0,n+1} \rho_0^{-m}]^m \psi_0(F) - \psi_0^* \]
  \[ R^{\rho,k}_{n+1} = R^{\rho,k}_{n+1} + dR^\rho = 0 \]
- local newton iteration
  \[ \rho_{0,n+1} \leftarrow \rho_{0,n+1} + \frac{d\rho_0}{d\rho_0} R^{\rho,k}_{n+1} \]
  iterative update

unconditionally stable - larger time steps

**explicit euler forward**
- evolution of density
  \[ D_t \rho_0 = \frac{1}{\Delta t} [\rho_{0,n+1} - \rho_{0,n}] \]
  finite difference approximation
- direct update of growth multiplier
  \[ \rho_{0,n+1} = \rho_{0,n} + \left[ \frac{\rho_{0,n} \rho_0^m}{\rho_0^*} \right] \psi_0(F) - \psi_0^* \]
  \[ \Delta t \]

conditionally stable - limited time step size

@ integration point level
- discrete residual of density update
  \[ R^\rho_{n+1} = \frac{1}{\Delta t} [\rho_{0,n+1} - \rho_{0,n}] - [\rho_{0,n+1} \rho_0^{-m}]^m \psi_0(F) - \psi_0^* \]
  \[ \Delta t \]
  check in matlab!

- residual of biological equilibrium / balance of mass
  \[ \text{local newton iteration} \]
**finite elements - integration point based**

*form follows function - bioinspired design*

**integration point based solution of balance of mass**

- **global newton iteration**
  - **loop over all quadrature points**
    - **local newton iteration**
      - **P0**
      - **n+1**
  - **determine element residual & tangent**
  - **determine global residual and tangent**
  - **determine \( \xi_{n+1} \)**
  - **determine state of biological equilibrium**

**staggered solution of density and displacements**

**finite elements - integration point based**

**ex_frame.m**

```matlab
function [q0, edof, bc, F_ext, mat, nel, node, ndof, nip] = ex_frame
    % input data for frame example
    emod = 10000; nue = 0.3; rho0 = 1.0; psi0 = 1.0;
    expm = 3.0; expn = 2.0; dt = 1.0;
    mat = [emod, nue, rho0, psi0, expm, expn, dt];
    
    [q0, edof] = mesh_sqp(xbox, ybox, nx, ny);
    nel, sizen] = size(edof); [ndof, sizen] = size(q0);
    node = ndof/2; nip = 4;
    
    % dirichlet boundary conditions
    bc(1,1) = 2*(ny+1)*q0; bc(1,2) = 0;
    bc(2,1) = 2*(ny+1)*q0; bc(2,2) = 0;
    bc(3,1) = 2*(ny+1)*(nx/2+1); bc(3,2) = 0;
    bc(4,1) = 2*(ny+1)*(nx/2+1); bc(4,2) = -ybox(2)/50;
    
    % neumann boundary conditions
    F_ext = zeros(ndof,1);
    % input data for frame example
```

**example - topology optimization**

- **load case I**
- **optimal material distribution**
- **load case II**

**frame example**

- **load case I**
- **optimal material distribution**
- **load case II**

find the lightest structure to support a given set of loads
form follows function - bioinspired design

ex_frame.m

find the lightest structure to support a given set of loads

e
x
a
m
p
l
-

t
p
o
l
o
g
y
o
p
t
i
m
i
z
a
t
i
o
n
f
o
r
f
o
l
l
w
s
f
u
n
c
t
i
o
n
b
i
c
c
l
f
r
a
m
e
s
1
8
1
7
-
2
0
0
5

example - topology optimization

bicycle frames 1817-2005

example - topology optimization

form follows function

functional adaptation of proxima femur

\[ D_t \rho_0 = \mathcal{R}_0 \]

\[ \mathcal{R}_0 = \left[ \frac{\rho_1}{\rho_0} \right]^{-m} \psi_0 - \psi^* \]

the density develops such that the tissue can just support the given mechanical load

design of bicycle frame

example - topology optimization

growing bone
Femoral neck deformity

The femoral neck normally forms an angle of 120-135 degrees with the shaft of the bone. This acts as the lever in easing the action of the muscles around the hip joint. An increase or decrease in this angle beyond the normal limits causes improper action of muscles, and interferes with walking. An increase in the angle beyond 135 degrees is called coxa valga or outward curvature of the hip joint. A decrease in the angle below 120 degrees is called coxa vara or inward curvature of the hip joint.

Example - Femoral neck deformity

Simulation vs. X-ray scans

Coxa vara
Coxa norma
Coxa valga

Excellent agreement of simulation and X-ray pattern

Pauwels [1973], Balle [2004], Kuhl & Balle [2005]

Example - Femoral neck deformity