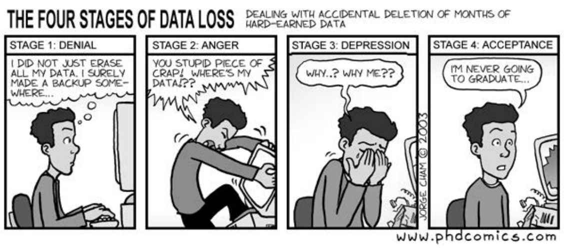
11 - finite element method - volume growth - implementation

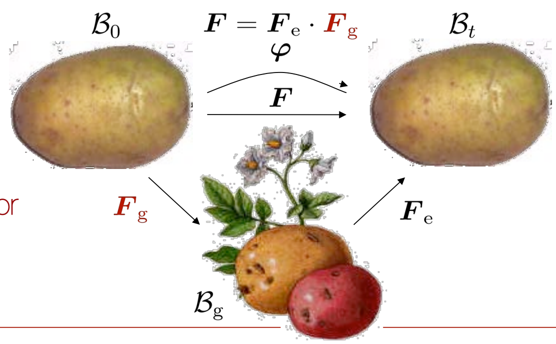


day	date		topic
tue	jan	08	motivation - everything grows!
thu	jan	10	basics maths - notation and tensors
tue	jan	15	basic kinematics - large deformation and growth
thu	jan	17	kinematics - growing hearts
tue	jan	22	guest lecture - growing surfaces
thu	jan	24	kinematics - growing leaflets
tue	jan	29	basic balance equations - closed and open systems
thu	jan	31	basic constitutive equations - growing muscle
tue	feb	05	basic constitutive equations - growing tumors
thu	feb	07	volume growth - finite elements for growth - theory
tue	feb	12	volume growth - finite elements for growth - matlab
thu	feb	14	volume growth - growing hearts
tue	feb	19	basic constitutive equations - growing bones
thu	feb	21	density growth - finite elements for growth
tue	feb	26	density growth - growing bones
thu	feb	28	everything grows! - midterm summary
tue	mar	05	midterm
thu	mar	07	remodeling - remodeling arteries and tendons
tue	mar	12	class project - discussion, presentation, evaluation
thu	mar	14	class project - discussion, presentation, evaluation
thu	mar	14	written part of final projects due

where are we???

the potato equations - kinematics

growth tensor

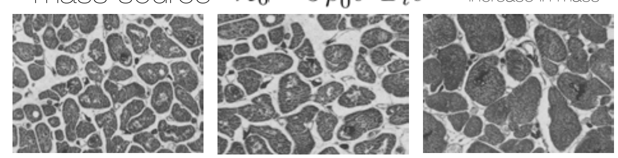


multiplicative decomposition

lee [1969], simo [1992], rodriguez, hoger & mc culloch [1994], epstein & maugin [2000], humphrey [2002], ambrosi & mollica [2002], himpel, kuhl, menzel & steinmann [2005]

volume growth at constant density

- free energy $\psi_0 = \psi_0^{\mathrm{neo}}(\boldsymbol{F}_\mathrm{e})$
- stress $\boldsymbol{P}_\mathrm{e} = \boldsymbol{P}_\mathrm{e}^{\mathrm{neo}}(\boldsymbol{F}_\mathrm{e})$
- growth tensor $\boldsymbol{F}_\mathrm{g} = \vartheta\, \boldsymbol{I}$ $\quad \mathrm{D}_t\vartheta = k_\vartheta(\vartheta)\ \mathrm{tr}(\boldsymbol{C}_\mathrm{e} \cdot \boldsymbol{S}_\mathrm{e})$
  growth function pressure
- mass source $\mathcal{R}_0 = 3\, \rho_0^* \vartheta^2 \mathrm{D}_t \vartheta$
  increase in mass

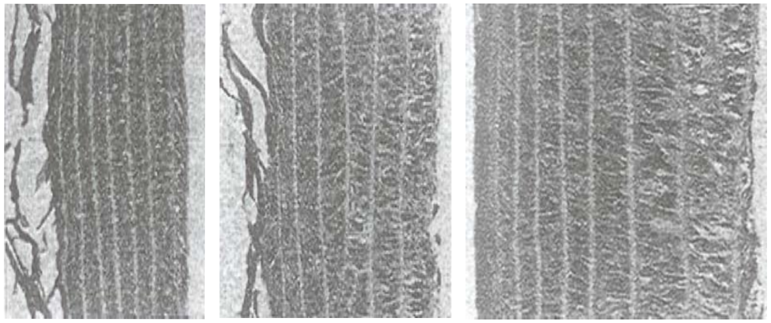


kinematic coupling of growth and deformation

rodriguez, hoger & mc culloch [1994], epstein & maugin [2000], humphrey [2002]

volume growth of the aortic wall

normosensitive — hypersensitive — severeley hypersensitive



wall thickening - thickening of musculoelastic fascicles

matsumoto & hayashi [1996], humphrey [2002]

example - growth of aortic wall

compensatory wall thickening during atherosclerosis

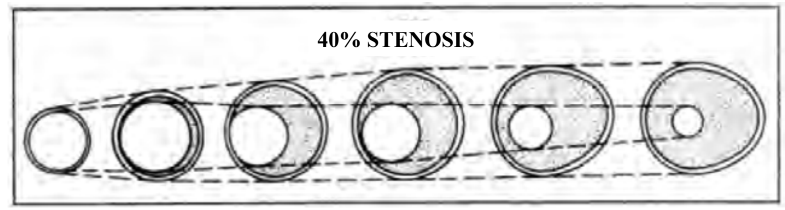
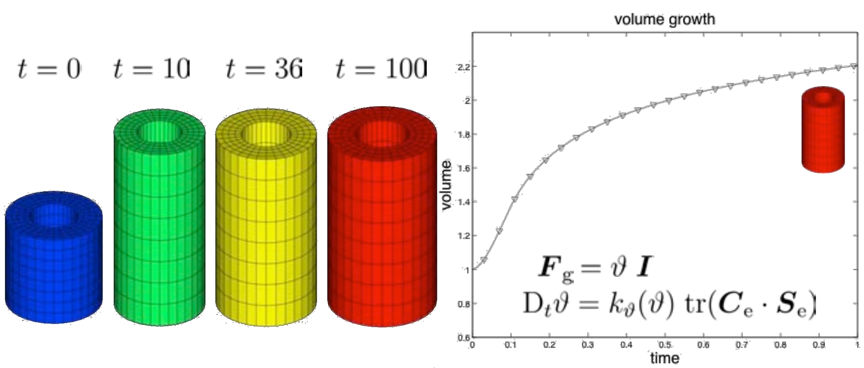


Figure 5. Diagrammic representation of a possible sequence of changes in atherosclerotic arteries leading eventually to lumen narrowing and consistent with the findings of this study. The artery enlarges initially (left to right in diagram) in association with the plaque accumulation to maintain an adequate, if not normal, lumen area. Early stages of lesion development may be associated with overcompensation. at more than 40% stenosis, however, the plaque area continues to increase to involve the entire circumference of the vessel, and the artery no longer enlarges at a rate suffcient to prevent the narrowing of the lumen.

glagov, weisenberg, zarins, stankunavicius, kolettis [1987]

example - growth of aortic wall

volume growth in cylindrical tube



stress-induced volume growth

himpel, kuhl, menzel & steinmann [2005]

example - growth of aortic wall

recipe for finite element modeling

from continuous problem…

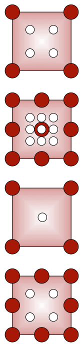
$$\rho_0 \, \mathrm{D}_t \boldsymbol{v} \quad = \mathrm{Div}(\boldsymbol{P}) + \boldsymbol{b}_0$$

- temporal discretization — implicit euler backward
- spatial discretization — finite element method
- staggered/simultaneous — newton raphson iteration
- linearization — gateaux derivative

… to linearized discrete initial boundary value problem

finite element method

integration point based solution of growth equation



loop over all time steps	nlin_fem
global newton iteration	nlin_fem
loop over all elements	brick_3d
loop over all quadrature points	cnst_vol
local newton iteration ϑ_{n+1}	upd_vol
determine element residual & tangent	cnst_vol
determine global residual and tangent	brick_3d
determine φ_{n+1}	nlin_fem
determine state of biological equilibrium	nlin_fem

growth multiplier ϑ as internal variable

nlin_fem.m

```
%%% 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);
      e_spa = extr_dof(edof,dof);
%%%%% 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
%%% loop over all load steps %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


@ the element level

- determine global residual

check in matlab!

$$\mathbf{R}_J^{\varphi} = \mathbf{A}_{e=1}^{n_{\mathrm{el}}} \int_{\mathcal{B}_0^e} \nabla N_{\varphi}^j \cdot \boldsymbol{P}_{n+1} \mathrm{d}V$$

- residual of mechanical equilibrium/balance of momentum

righthand side vector for global system of equations

discrete residual

check in matlab!

$$\mathbf{R}_J^{\varphi} = \mathbf{A}_{e=1}^{n_{\mathrm{el}}} \int_{\mathcal{B}_0^e} \nabla N_{\varphi}^j \cdot \boldsymbol{P}_{n+1} \mathrm{d}V$$

quads_2d.m/brick_3d.m

```
for i=1:nod
  en=(i-1)*2;
  Re(en+ 1) = Re(en+ 1) +(P(1,1)*dNx(1,i)' ...
                  + P(1,2)*dNx(2,i)') * detJ * wp(ip);
  Re(en+ 2) = Re(en+ 2) +(P(2,1)*dNx(1,i)' ...
                  + P(2,2)*dNx(2,i)') * detJ * wp(ip);
end
```

righthand side vector for global system of equations

@ the element level

- stiffness matrix / iteration matrix

check in matlab!

$$\mathbf{K}^{\varphi\varphi}_{JL} = \frac{\partial \mathbf{R}^{\varphi}_{J}}{\partial \boldsymbol{\varphi}_{L}} = \mathbf{A}_{e=1}^{n_{\mathrm{el}}} \int_{\mathcal{B}_0^e} \nabla N^j_{\varphi} \cdot \mathrm{D}_F \boldsymbol{P} \cdot \nabla N^l_{\varphi} \, \mathrm{d}V$$

- linearization of residual wrt nodal dofs

iteration matrix for global system of equations

linearized residual

check in matlab!

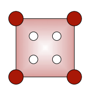
$$\mathbf{K}^{\varphi\varphi}_{JL} = \frac{\partial \mathbf{R}^{\varphi}_{J}}{\partial \boldsymbol{\varphi}_{L}} = \mathbf{A}_{e=1}^{n_{\mathrm{el}}} \int_{\mathcal{B}_0^e} \nabla N^j_{\varphi} \cdot \mathrm{D}_F \boldsymbol{P} \cdot \nabla N^l_{\varphi} \, \mathrm{d}V$$

quads_2d.m/brick_3d.m

```
for i=1:nod; for j=1:nod
  eni=(i-1)*2; enj=(j-1)*2;
  Ke(eni+1,enj+1)=Ke(eni+1,enj+1)+(dNx(1,i)*A(1,1,1,1)*dNx(1,j) ...
                +dNx(1,i)*A(1,1,1,2)*dNx(2,j) ...
                +dNx(2,i)*A(1,2,1,1)*dNx(1,j) ...
                +dNx(2,i)*A(1,2,1,2)*dNx(2,j))*detJ*wp(ip);
end; end
```

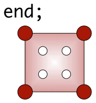
iteration matrix for global system of equations

quads_2d.m



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Ke,Re,Ie]=element1(e_mat,e_spa,i_var,mat)
%%% element stiffness matrix Ke, residual Re, internal variables Ie %%%%
Ie = i_var;
Re = zeros(8,1);
Ke = zeros(8,8);
nod=4;              delta = eye(2);
indx=[1;3;5;7];  ex_mat=e_mat(indx);
indy=[2;4;6;8];  ey_mat=e_mat(indy);
%%% integration points %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
g1=0.577350269189626; w1=1;
gp(:,1)=[-g1; g1;-g1; g1];   w(:,1)=[ w1; w1; w1; w1];
gp(:,2)=[-g1;-g1; g1; g1];   w(:,2)=[ w1; w1; w1; w1];
wp=w(:,1).*w(:,2);      xsi=gp(:,1);      eta=gp(:,2);
%%% shape functions and derivatives in isoparametric space %%%%%%%%%%%%%
N(:,1)=(1-xsi).*(1-eta)/4;   N(:,2)=(1+xsi).*(1-eta)/4;
N(:,3)=(1+xsi).*(1+eta)/4;   N(:,4)=(1-xsi).*(1+eta)/4;
dNr(1:2:8  ,1)=-(1-eta)/4;   dNr(1:2:8  ,2)= (1-eta)/4;
dNr(1:2:8  ,3)= (1+eta)/4;   dNr(1:2:8  ,4)=-(1+eta)/4;
dNr(2:2:8+1,1)=-(1-xsi)/4;   dNr(2:2:8+1,2)=-(1+xsi)/4;
dNr(2:2:8+1,3)= (1+xsi)/4;   dNr(2:2:8+1,4)= (1-xsi)/4;
JT=dNr*[ex_mat;ey_mat]';
%%% element stiffness matrix Ke, residual Re, internal variables Ie %%%%
```

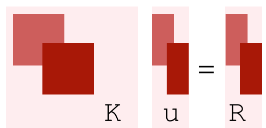

quads_2d.m



```
%%% loop over all integration points %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ip=1:4
indx=[2*ip-1; 2*ip];  detJ=det(JT(indx,:));
if detJ<10*eps;  disp('Jacobi determinant less than zero!');  end;
JTinv=inv(JT(indx,:));  dNx=JTinv*dNr(indx,:);
F=zeros(2,2);
for j=1:4
   jndx=[2*j-1; 2*j];
   F=F+e_spa(jndx)'*dNx(:,j)';
end
var = i_var(ip);
[A,P,var]=cnst_law(F,var,mat);
Ie(ip) = var;
for i=1:nod
   en=(i-1)*2;
   Re(en+ 1) =  Re(en+ 1) +(P(1,1)*dNx(1,i)' ...
                          + P(1,2)*dNx(2,i)') * detJ * wp(ip);
   Re(en+ 2) =  Re(en+ 2) +(P(2,1)*dNx(1,i)' ...
                          + P(2,2)*dNx(2,i)') * detJ * wp(ip);
end
%%% loop over all integration points %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% element stiffness matrix Ke, residual Re, internal variables Ie %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

assm_sys.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [K,R,I]=assm_sys(edof,K,Ke,R,Re,I,Ie)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% assemble local element contributions to global tangent & residual %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% input:  edof   = [ elem X1 Y1 X2 Y2 ]    ... incidence matrix
%%%         Ke     = [ nedof x nedof ]       ... element tangent Ke
%%%         Re     = [ fx_1 fy_1 fx_2 fy_2 ] ... element residual Re
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% output: K      = [ ndof x ndof ]         ... global tangent K
%%%         R      = [ ndof x 1 ]            ... global residual R
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[nie,n]=size(edof);
I(edof(:,1),:)=Ie(:);
t=edof(:,2:n);
for i = 1:nie
   K(t(i,:),t(i,:)) = K(t(i,:),t(i,:))+Ke;
   R(t(i,:))         =R(t(i,:))       +Re;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



@ the integration point level

- constitutive equations - given $\boldsymbol{F}$ calculate $\boldsymbol{P}$

check in matlab!

$$\boldsymbol{P}(\boldsymbol{F}^{\mathrm{e}}) = \mu\,\boldsymbol{F}^{\mathrm{e}} + [\,\lambda\,\ln(\det(\boldsymbol{F}^{\mathrm{e}})) - \mu\,]\boldsymbol{F}^{\mathrm{e}\,-\,\mathrm{t}}$$

- stress calculation @ integration point level

stress for righthand side vector

@ the integration point level

check in matlab!

$$\boldsymbol{P}(\boldsymbol{F}^{\mathrm{e}}) = \mu\,\boldsymbol{F}^{\mathrm{e}} + [\,\lambda\,\ln(\det(\boldsymbol{F}^{\mathrm{e}})) - \mu\,]\,\boldsymbol{F}^{\mathrm{e}\,-\,\mathrm{t}}$$

const_vol.m

```
Fe     = F / theta;
Fe_inv = inv(Fe);
Je     = det(Fe);
delta  = eye(ndim);

P = xmu * Fe + (xlm * log(Je) - xmu) * Fe_inv';
```

stress for righthand side vector

@ the integration point level

- tangent operator / constitutive moduli

check in matlab!

$$\mathsf{A} = \frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}\boldsymbol{F}} = \left.\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^{\mathrm{g}}} + \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}^{\mathrm{g}}} : \frac{\partial \boldsymbol{F}^{\mathrm{g}}}{\partial \vartheta} \otimes \left.\frac{\partial \vartheta}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}}$$

- linearization of stress wrt deformation gradient

tangents for iteration matrix

@ the integration point level

check in matlab!

$$\mathbf{A} = \frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}\boldsymbol{F}} = \left.\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^{\mathrm{g}}} + \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}^{\mathrm{g}}} : \frac{\partial \boldsymbol{F}^{\mathrm{g}}}{\partial \vartheta} \otimes \left.\frac{\vartheta}{\boldsymbol{F}}\right|_{\boldsymbol{F}}$$

const_vol.m

```
for i=1:ndim; for j=1:ndim; for k=1:ndim; for l=1:ndim
 A(i,j,k,l) = xlm * Fe_inv(j,i)*Fe_inv(l,k) ...
         - (xlm * log(Je) - xmu) * Fe_inv(l,i)*Fe_inv(j,k) ...
         + xmu * delta(i,k)* delta(j,l) ...
         + ten1(i,j)* ten2(k,l);
end, end, end, end
A = A / theta;
```

tangent for iteration matrix

cnst_vol.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function[A,P,var]=cnst_vol(F,var,mat,ndim)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
determine tangent, stress and internal variable %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
emod = mat(1);   nue = mat(2);  kt = mat(3);  kc = mat(4);
mt   = mat(5);   mc  = mat(6);  tt = mat(7);  tc = mat(8); dt=mat(9);
xmu = emod / 2 / (1+nue);       xlm= emod * nue / (1+nue) /(1-2*nue);
%%% update internal variable%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[var,ten1,ten2]=updt_vol(F,var,mat,ndim);
theta =var(1)+1; Fe=F/theta;Fe_inv=inv(Fe);Je=det(Fe);delta=eye(ndim);
%%% first piola kirchhoff stress %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P = xmu * Fe + (xlm * log(Je) - xmu) * Fe_inv';
%%% tangent %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:ndim; for j=1:ndim; for k=1:ndim; for l=1:ndim
    A(i,j,k,l) =  xlm * Fe_inv(j,i)*Fe_inv(l,k) ...
               - (xlm * log(Je) - xmu) * Fe_inv(l,i)*Fe_inv(j,k) ...
               +                 xmu  *  delta(i,k)* delta(j,l) ...
               +                         ten1(i,j)*  ten2(k,l);
end, end, end, end;
A = A / theta; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
```


@ the integration point level

- discrete update of growth multiplier

check in matlab!

$$\mathsf{R}^{\vartheta}_{\mathrm{n}+1} = \vartheta_{\mathrm{n}+1} - \vartheta_{\mathrm{n}} - k\,\mathrm{tr}(\boldsymbol{M}^{\mathrm{e}})\,\Delta t$$

- residual of biological equilibrium

local newton iteration

@ the integration point level

check in matlab!

$$\mathsf{R}^{\vartheta}_{\mathrm{n}+1} = \vartheta_{\mathrm{n}+1} - \vartheta_{\mathrm{n}} - k\,\mathrm{tr}(\boldsymbol{M}^{\mathrm{e}})\,\Delta t$$

updt_vol.m

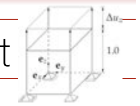
```
while abs(res) > tol
   res  = k * tr_Me * dt - the_k1 + the_k0;
   dres =(dk_dthe * tr_Me + k * dtrM_dthe)*dt -1;
end
```

local newton iteration

updt_vol.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% local newton-raphson iteration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while abs(res) > tol
  iter=iter+1;
  Fe = F/the_k1; Fe_inv = inv(Fe);   Ce = Fe'*Fe; Ce_inv = inv(Ce);
  Je = det(Fe);  delta  = eye(ndim);
  Se = xmu * delta + (xlm * log(Je) - xmu) * Ce_inv;
  Me = Ce*Se;      tr_Me = trace(Me);
  CeLeCe = ndim * ndim * xlm - 2 * ndim * (xlm * log(Je) - xmu);
  dtrM_dthe = - 1/the_k1 * ( 2*tr_Me + CeLeCe );
  if tr_Me > 0
     k        = kt*((tt-the_k1)/(tt-1))^mt;
     dk_dthe = k  /(the_k1-tt)        *mt;
  else
     k        = kc*((the_k1-tc)/(1-tc))^mc;
     dk_dthe = k  /(the_k1-tc)        *mc;
  end
res   = k * tr_Me * dt - the_k1 + the_k0;
dres =(dk_dthe * tr_Me + k * dtrM_dthe)*dt -1;
the_k1 =  the_k1 -res/dres;
if(iter>20); disp(['*** NO LOCAL CONVERGENCE ***']); return; end;
%%% local newton-raphson iteration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


probing the material @the integration point



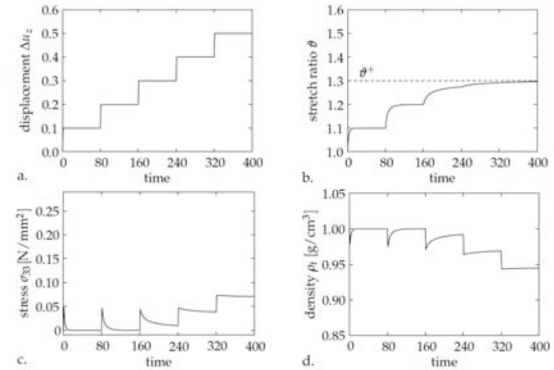
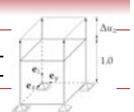


Figure 4.2: Isotropic simple tension test on a growing cube. (a) An incrementally increasing stretch is applied. (b) The stretch ratio converges time-dependently to the biological equilibrium. (c) The stresses vanish in the biological equilibrium state as long as $\vartheta < \vartheta^+$. (d) The density in the biological equilibrium state does not change as long as $\vartheta < \vartheta^+$.

himpel, kuhl, menzel & steinmann [2005]

probing the material @the integration point



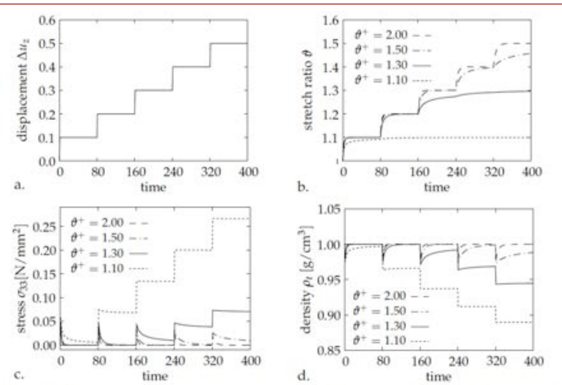
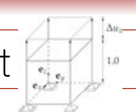


Figure 4.3: Variation of the limiting stretch ratio ϑ^+ in the simple tension test. The stretch ratio increases until the limiting value is reached. If the limiting value of the stretch ratio is reached the material behavior is purely elastic.

himpel, kuhl, menzel & steinmann [2005]

probing the material @the integration point



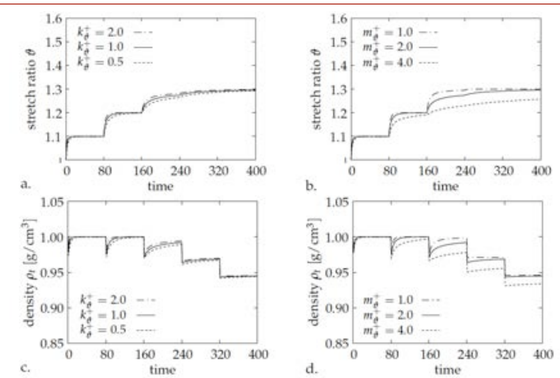
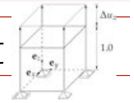


Figure 4.4: Variation of the material parameters k_ϑ^+ and m_ϑ^+ in the simple tension test. They influence the relaxation time, but not the final state at biological equilibrium.

himpel, kuhl, menzel & steinmann [2005]

probing the material @the integration point



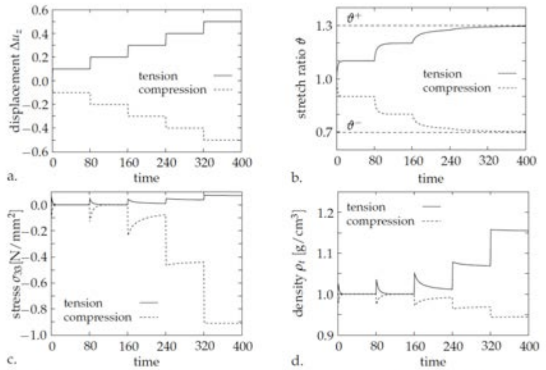


Figure 4.5: The material distinguishes between tension and compression. In case of tension the material grows, and in case of compression the material decreases.

himpel, kuhl, menzel & steinmann [2005]

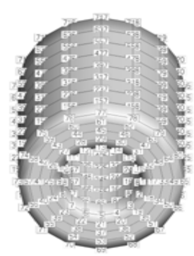
ex_tube1.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [q0,edof,emat,bc,F_ext,mat,ndim,nel,node,ndof,nip,nlod] = ex_tube1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% material parameters for volume growth %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
emod = 3.0; nue = 0.3;
kt   = 0.5; kc   = 0.25; mt = 4.0; mc = 5.0; tt = 1.5; tc = 0.5;dt=1.0;
mat=[emod,nue,kt,kc,mt,mc,tt,tc,dt];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
l    =    2.0;      % length
ra   =    1.0;      % inner radius
ri   =    0.5;      % outer radius
nez =       8;      % number of elements in longitudinal direction
ner =       4;      % number of elements in radial direction
nep =      16;      % number of elements in circumferential direction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tol   = 1e-8;
ndim = 3;
nip   = 8;
nel = nez * ner * nep;
node= (nez+1)*(ner+1)*nep;
ndof = ndim*node;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```


ex_tube1.m

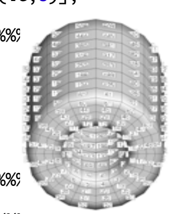
```
%%% coordinates %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
q0 = zeros(ndim*node,1);
nn = 0;

delta_z =       l / nez;
delta_r = (ra-ri) / ner;
delta_t =    2*pi / nep;
for iz = 0:nez
  z = iz * delta_z;
  for ir = 0:ner
    r = ri + ir * delta_r;
    for ip = 0:(nep-1)
      p = ip * delta_t;
      nn = nn +ndim;
      q0(nn-2,1) = r*cos(p);
      q0(nn-1,1) = r*sin(p);
      q0(nn  ,1) = z;
    end
  end
end
%%% coordinates %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



ex_tube1.m

```
%%% connectivity %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ie = 1:nel
   edof(ie,:)=[ie, ndim*enod(ie,1)-2  ndim*enod(ie,1)-1  ndim*enod(ie,1) ...
                   ndim*enod(ie,2)-2  ndim*enod(ie,2)-1  ndim*enod(ie,2) ...
                   ndim*enod(ie,3)-2  ndim*enod(ie,3)-1  ndim*enod(ie,3) ...
                   ndim*enod(ie,4)-2  ndim*enod(ie,4)-1  ndim*enod(ie,4) ...
                   ndim*enod(ie,5)-2  ndim*enod(ie,5)-1  ndim*enod(ie,5) ...
                   ndim*enod(ie,6)-2  ndim*enod(ie,6)-1  ndim*enod(ie,6) ...
                   ndim*enod(ie,7)-2  ndim*enod(ie,7)-1  ndim*enod(ie,7) ...
                   ndim*enod(ie,8)-2  ndim*enod(ie,8)-1  ndim*enod(ie,8)];
end
%%% boundary conditions %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
du = l/2; nb = 0;
for ib = 1:(nep*(ner+1))
  if(abs(q0(ndim*(node-nep*(ner+1))+ndim*ib-2)-0.0)<tol)
    nb = nb+1;     bc(nb,:) = [ndim*ib-2  0];
  end  if(abs(q0(ndim*(node-nep*(ner+1))+ndim*ib-1)-0.0)<tol)
end
%%% loading %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load  = 0.0;F_ext = zeros(ndof,1);  nlod  = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



ex_tube1.m

Copyright © 2005 Tech Science Press
Eng Sci. 2005;8:119-134
Comp Meth

Computational modelling of isotropic multiplicative growth

G. Himpel, E. Kuhl, A. Menzel, P. Steinmann

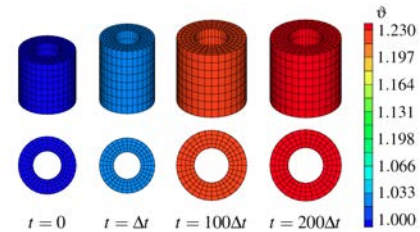


Figure 10 : Deformation of the tube and evolution of the stretch ratio for an axial stretch $u = 1.0$.

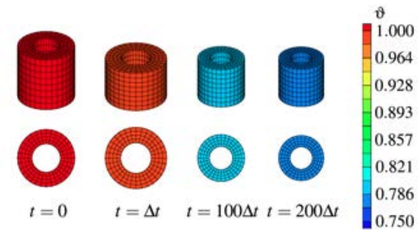
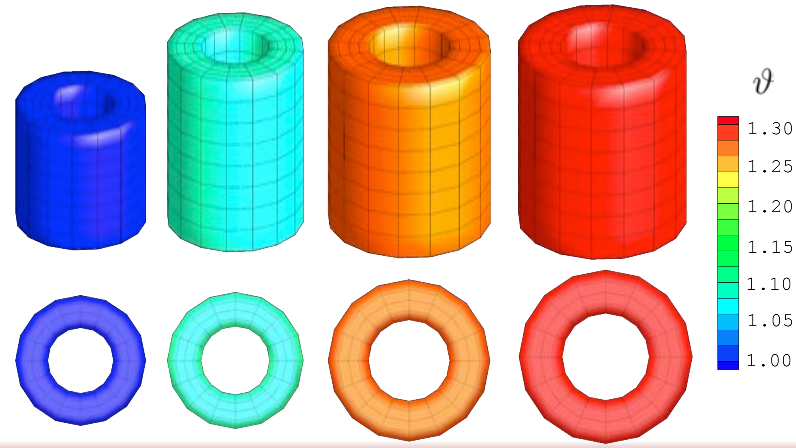


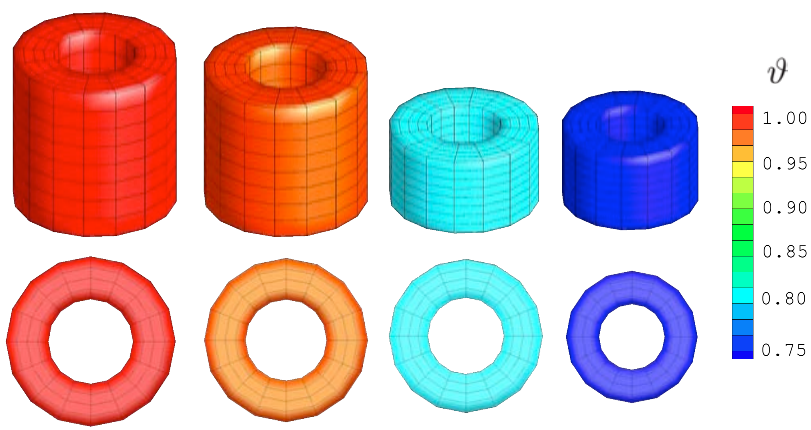
Figure 12 : Deformation of the tube and evolution of the stretch ratio for an axial compression $u = -1.0$.

himpel, kuhl, menzel & steinmann [2005]

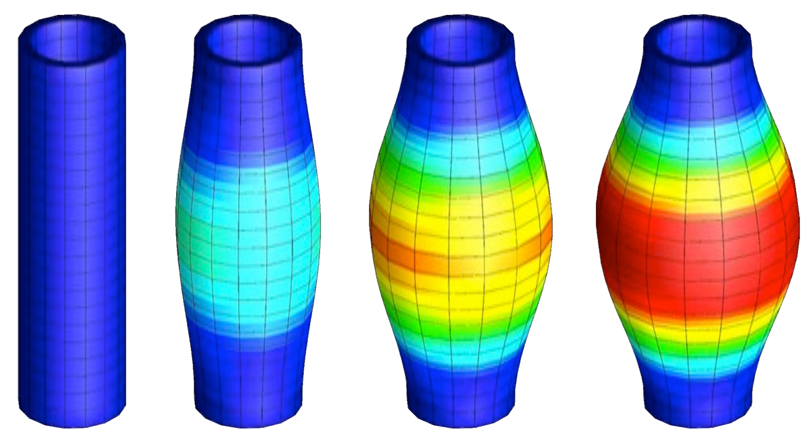
ex_tube1.m



ex_tube2.m



ex_tube3.m



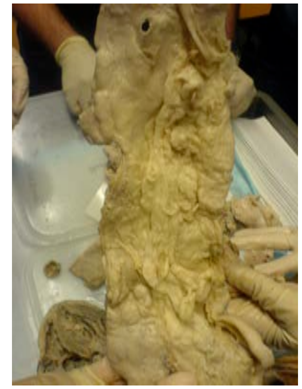
atherosclerosis

atherosclerosis is a condition in which an artery wall thickens as the result of a build-up of fatty materials. the atheromatous plaques, although compensated for by artery enlargement, eventually lead to plaque rupture and clots inside the arterial lumen. the clots leave behind stenosis, a narrowing of the artery, and insufficient blood supply to the tissues and organ it feeds. if the artery enlargement is excessive, a net aneurysm results. these complications of advanced atherosclerosis are chronic, slowly progressive and cumulative.

WIKIPEDIA
The Free Encyclopedia

example - atherosclerosis

atherosclerosis

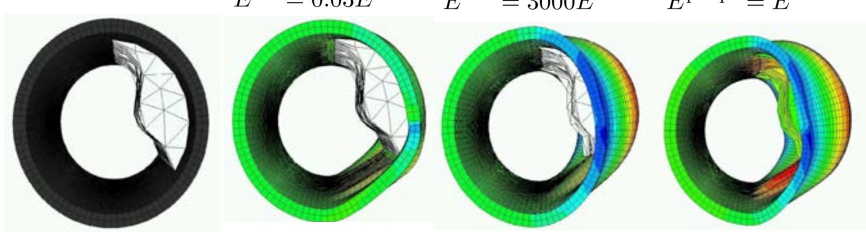




[greek] arteria = artery / sclerosis = hardening

example - atherosclerosis

qualitative simulation of atherosclerosis

initial conditions | adiposis $E^{\text{fat}} = 0.03 E^{\text{tissue}}$ | calcification $E^{\text{lime}} = 3000 E^{\text{tissue}}$ | growing plaque $E^{\text{plaque}} = E^{\text{tissue}}$



early stage
soft plaque pressed into wall

later stage
stiff plaque induces high stresses

overall thickening - thickening of individual fascicles

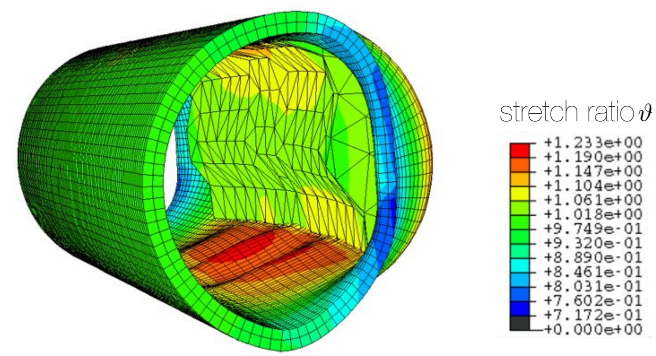
holzapfel [2001], holzapfel & ogden [2003], kuhl, maas, himpel & menzel [2007]

example - atherosclerosis

qualitative simulation of atherosclerosis

restenotic conditions

stretch ratio ϑ



re-nerrowing of x-section in response to high stress

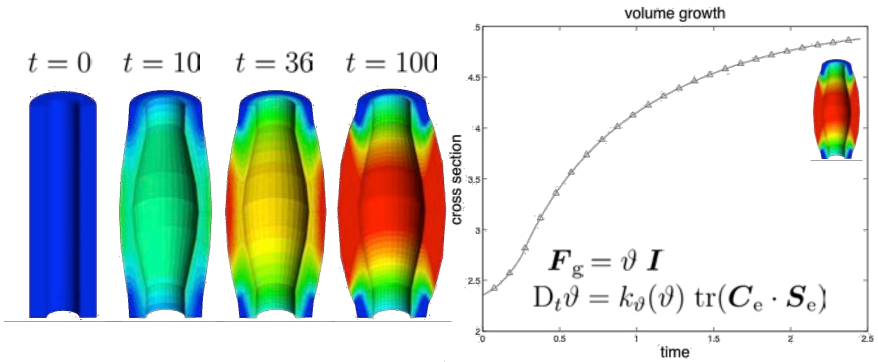
kuhl, maas, himpel & menzel [2007]

example - atherosclerosis

in-stent restenosis

restenosis is the reoccurrence of stenosis, the narrowing of a blood vessel, leading to restricted blood flow. restenosis usually pertains to a blood vessel that has become narrowed, received treatment, and subsequently became renarrowed. in some cases, surgical procedures to widen blood vessels can cause further narrowing. during balloon angioplasty, the balloon 'smashes' the plaques against the arterial wall to widen the size of the lumen. however, this damages the wall which responds by using physiological mechanisms to repair the damage and the wall thickens.

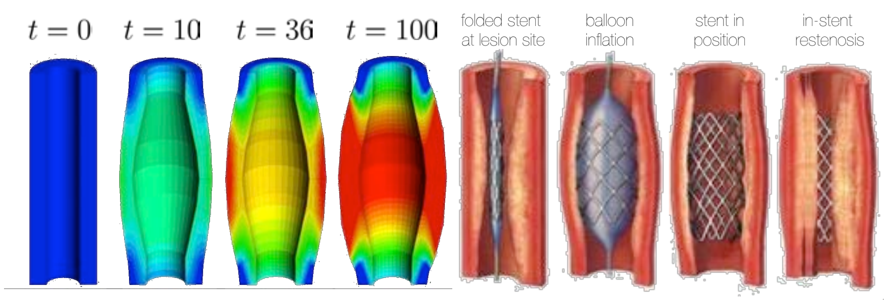
qualitative simulation of stent implantation



stress-induced volume growth

kuhl, maas, himpel & menzel [2007]

qualitative simulation of stent implantation



stress-induced volume growth

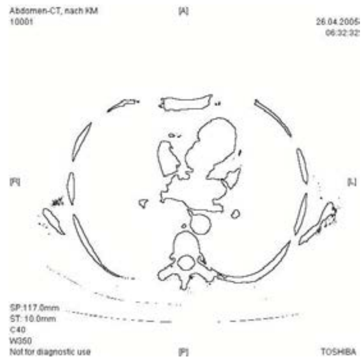
kuhl, maas, himpel & menzel [2007]

generation of patient specific model

computer tomography - typical cross section

kuhl, maas, himpel & menzel [2007]

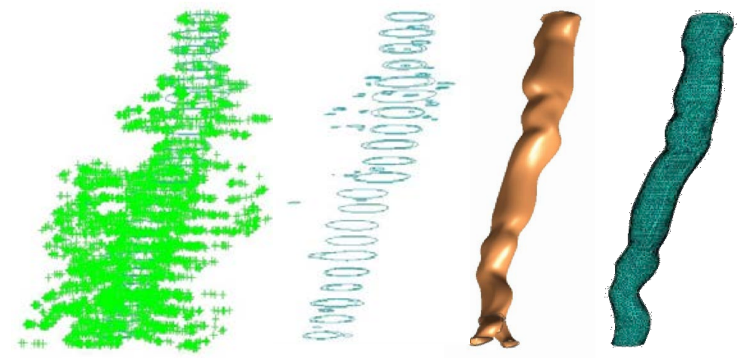
generation of patient specific model



outline of ct image - typical cross section

kuhl, maas, himpel & menzel [2007]

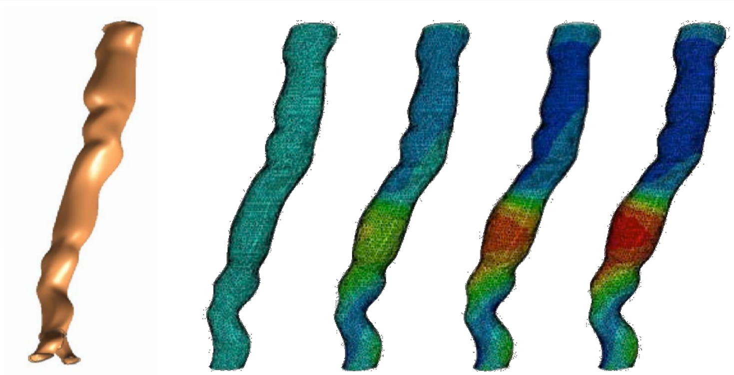
generation of patient specific model



from computer tomography to finite element model

kuhl, maas, himpel & menzel [2007]

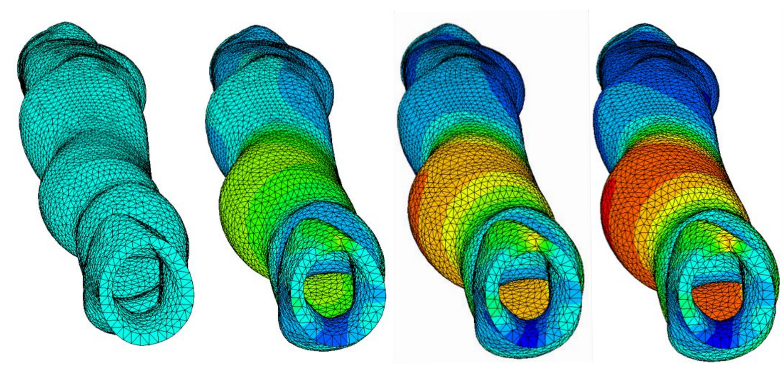
virtual stent implantation - patient specific model



tissue growth - response to virtual stent implantation

kuhl, maas, himpel & menzel [2007]

virtual stent implantation - patient specific model



tissue growth - response to virtual stent implantation

kuhl, maas, himpel & menzel [2007]