AnisoMPM: Animating Anisotropic Damage Mechanics: Supplemental Document

Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang,
Ziyin Qu, Jiecong Lu, Meggie Cheng, Chenfanfu Jiang

Contents

1 Pseudocode **2**
1.1 General Routines . . . 2
1.2 AnisoMPM Explicit Damage . . . 3
1.3 AnisoMPM Implicit Damage . . . 4
1.4 Inextensibility Solve . . . 5

2 Anisotropic Damage **6**
2.1 Explicit Damage Derivation . . . 6
2.2 Implicit Damage Derivation . . . 6

3 Anisotropic Elasticity **7**
3.1 QR-based General Anisotropic Hyperelasticity . . . 7
3.2 Stress . . . 7
3.3 Stress Differential . . . 8
3.4 Stress Derivative . . . 8
3.4.1 Derivative of R . . . 9
3.4.2 Derivative of Q . . . 9
3.4.3 MATLAB code . . . 9
3.5 Bounded Stress . . . 10
3.5.1 2D . . . 10
3.5.2 3D . . . 11
3.6 Energy Degradation . . . 13

4 Inextensibility **13**
4.1 Derivation of Constraint Equation . . . 13
4.2 Derivation of Weak Form . . . 14
4.2.1 Momentum Equation . . . 14
4.2.2 Constraint . . . 15
4.3 KKT Relaxation . . . 15

1 Pseudocode

Here we present algorithms for all of our new methods, putting them in context within the MPM data flow to highlight them as MPM augmentations that can be added to any existing pipeline.

1.1 General Routines

Algorithm 1 Initialize

$\{\mathbf{a_i}\}$ = set of structural directors (one for transverse isotropy, two for orthotropy)
$\{\alpha_i\}$ = set of corresponding weights $\in [-1, \infty)$ (we only choose $\alpha_i \in \{-1, 0\}$)
Set $l_0 = 0.5 * \Delta x$
Set $\zeta = 1$
Empirically tune σ_c (good starting value is to use a stress corresponding to some uniform stretch)
Empirically tune η //controls crack speed like M_c

Algorithm 2 Construct Structural Tensor $(\{\mathbf{a_i}\}, \{\alpha_\mathbf{i}\}, \mathbf{F})$

$\mathbf{F} = \mathbf{RS}$ //Grab rotation, $\mathbf{R}$, from $\mathbf{F}$
if transverse isotropic **then**
    return $\mathbf{A} = \mathbf{I} + \alpha_\mathbf{1}(\mathbf{Ra_1} \otimes \mathbf{Ra_1})$
else if orthotropic **then**
    return $\mathbf{A} = \mathbf{I} + \alpha_\mathbf{1}(\mathbf{Ra_1} \otimes \mathbf{Ra_1}) + \alpha_\mathbf{2}(\mathbf{Ra_2} \otimes \mathbf{Ra_2})$

1.2 AnisoMPM Explicit Damage

Algorithm 3 AnisoMPM Explicit Damage

initializeParameters()
for each substep **do**
    Begin MPM like usual until just before computeGridVelocity
    computeLaplacians()
    updateDamage()
    Continue MPM like usual using the new updated damage values in computeGridVelocity

Algorithm 4 Compute Laplacians (Explicit Only)

//Transfer damage to grid
Compute interpolation weights, w_{ip}^n //We must use cubic B-spline
for each grid node, i, **do**
    $d_i^n = \frac{\sum_p w_{ip}^n d_p^n}{\sum_p w_{ip}^n}$
    $\Delta N_i^n = N_i''(x)N_i(y)N_i(z) + N_i(x)N_i''(y)N_i(z) + N_i(x)N_i(y)N_i''(z)$ //Requires cubic N
//Compute and store Laplacians
for each particle, p, **do**
    $\Delta d_p^n = \sum_i d_i^n \Delta N_i^n$

Algorithm 5 Update Damage (Explicit Only)

for each particle, p, **do**
    //Compute geometric resistance: D_c^n
    $D_c^n = d_p^n - l_0^2 \Delta d_p^n$
    //Compute driving force: $(1 - d_p^n)\tilde{D}_p^n$
    Get *un-degraded* Cauchy stress, $\boldsymbol{\sigma}_p^n$, from $\boldsymbol{F}^n$ and constitutive model
    Get eigenvalues, $\{\sigma_a\}_{a=1,2,3}$, and eigenvectors, $\{\mathbf{n}_a\}_{a=1,2,3}$, of $\boldsymbol{\sigma}_p$
    $\boldsymbol{\sigma}_p^{+n} = \sum_{a=1}^3 \langle\sigma_a\rangle \mathbf{n}_a \otimes \mathbf{n}_a$ //With $\langle x\rangle = \frac{x+|x|}{2}$
    $\boldsymbol{A}_{\boldsymbol{p}}^n$ = constructStructuralTensor($\{\boldsymbol{a}_i\}_p, \{\alpha_i\}_p, \boldsymbol{F}_p^n$) //pass structure directors and F
    $\Phi(\boldsymbol{\sigma}_p^{+n}) = \frac{1}{\boldsymbol{\sigma}_{crit}^2}(\boldsymbol{A}_{\boldsymbol{p}}^n \boldsymbol{\sigma}_{\boldsymbol{p}}^{+n}) : (\boldsymbol{\sigma}_{\boldsymbol{p}}^{+n} \boldsymbol{A}_{\boldsymbol{p}}^n)$ //See tech doc for derivation
    $\tilde{D}_p^n = \max(\zeta\langle\Phi(\boldsymbol{\sigma}_p^{+n}) - 1\rangle, \tilde{D}_p^{H,n-1})$
    $\tilde{D}_p^{H,n} = \tilde{D}_p^n$ //update $\tilde{D}$ history to keep the max
    if $(1 - d_p^n)\tilde{D}_p^n > D_c^n$ **then**
        $d_p^{n+1} = \min(1, d_p^n + \frac{\Delta t}{\eta}\langle(1 - d_p^n)\tilde{D}_p^n - D_c^n\rangle)$
    else
        $d_p^{n+1} = d_p^n$

1.3 AnisoMPM Implicit Damage

Algorithm 6 AnisoMPM Implicit Damage

initializeParameters()
for each substep **do**
    Begin MPM like usual until just before computeGridVelocity
    damageP2G()
    update$\tilde{D}$()
    damageSolve()
    damageG2P()
    Continue MPM like usual using the new updated damage values in computeGridVelocity

Algorithm 7 Damage P2G (Implicit Only)

//Transfer damage to grid
Compute interpolation weights, w_{ip}^n //We must use cubic B-spline
for each grid node, i, **do**
    $d_i^n = \frac{\sum_p w_{ip}^n d_p^n}{\sum_p w_{ip}^n}$

Algorithm 8 Update $\tilde{D}$ (Implicit Only)

for each particle, p, **do**
    //Compute driving force: $(1 - d_p^n)\tilde{D}_p^n$
    Get *un-degraded* Cauchy stress, $\boldsymbol{\sigma}_p^n$, from $\boldsymbol{F}^n$ and constitutive model
    Get eigenvalues, $\{\sigma_a\}_{a=1,2,3}$, and eigenvectors, $\{\mathbf{n}_a\}_{a=1,2,3}$, of $\boldsymbol{\sigma}_p$
    $\boldsymbol{\sigma}_p^{+n} = \sum_{a=1}^3 \langle\sigma_a\rangle \mathbf{n}_a \otimes \mathbf{n}_a$ //With $\langle x\rangle = \frac{x+|x|}{2}$
    $\boldsymbol{A}_{\boldsymbol{p}}^n$ = constructStructuralTensor($\{\boldsymbol{a}_i\}_p, \{\alpha_i\}_p, \boldsymbol{F}_p^n$) //pass structure directors and F
    $\Phi(\boldsymbol{\sigma}_p^{+n}) = \frac{1}{\boldsymbol{\sigma}_{crit}^2}(\boldsymbol{A}_{\boldsymbol{p}}^n \boldsymbol{\sigma}_{\boldsymbol{p}}^{+n}) : (\boldsymbol{\sigma}_{\boldsymbol{p}}^{+n} \boldsymbol{A}_{\boldsymbol{p}}^n)$ //See tech doc for derivation
    $\tilde{D}_p^n = \max(\zeta\langle\Phi(\boldsymbol{\sigma}_p^{+n}) - 1\rangle, \tilde{D}_p^{H,n-1})$
    $\tilde{D}_p^{H,n} = \tilde{D}_p^n$ //update $\tilde{D}$ history to keep the max

Algorithm 9 Damage Solve (Implicit Only)

//Goal is to contruct and solve this system for $\boldsymbol{d}$: $(\mathcal{A} + \mathcal{B})\boldsymbol{d} = \boldsymbol{c}$
$\boldsymbol{c} = [c_i] = \sum_p V_p^n \left(\frac{\Delta t}{\eta}\tilde{D}_p^n + d_i^n\right) w_{ip}^n$ //Build rhs
$\mathcal{B} = [\mathcal{B}_{ij}] = \sum_p V_p^n \left(\frac{\Delta t}{\eta} l_0^2\right) \left(\nabla\Theta_i(\boldsymbol{x}_p^n)\right)^T \left(\nabla\Theta_j(\boldsymbol{x}_p^n)\right)$ //Build MPM discrete Laplace operator
$\mathcal{A} = [\mathcal{A}_{ii}] = \sum_p V_p^n \left(1 + \frac{\Delta t}{\eta}(\tilde{D}_p^n + 1)\right) w_{ip}^n$ //Build diagonal matrix
Solve the system with PCG (we use Jacobi preconditioner)

Algorithm 10 Damage G2P (Implicit Only)

//Transfer damage from the grid
Compute interpolation weights, w_{ip}^n //We must use cubic B-spline
for each particle, p, **do**
    $d_p^{n+1} = \min(1, \sum_i d_i^{n+1} w_{ip}^n)$

1.4 Inextensibility Solve

Algorithm 11 Enforce Inextensibility

procedure COMPUTE COEFFICIENT
  //We store $i\alpha$ at $(i * d + \alpha)$-th position in an array
  d = simulation dimension, N_n = number of grid nodes, N_c = number of cell centers
  Initialize N_i as quadratic basis over grid nodes, Γ_z as linear basis over grid cells
  Set $\boldsymbol{M}^{-1} = Zero(d * N_n, d * N_n), \boldsymbol{B} = Zero(N_c, d * N_n), \boldsymbol{b} = Zero(d * N_n, 1)$
  Compute $(\boldsymbol{M}^{-1})_{i\alpha i\alpha} = \Delta t / m_i$
  Compute $b_{i\alpha} = \frac{m_i}{\Delta t} v_{i\alpha}^n - \sum_p V_p^n \sigma_{p\alpha\beta}^n N_{i,\beta}(\boldsymbol{x}_p)$ //Loop over particles and sum up
  Compute $K_{c,i\alpha} = \sum_p V_p^n \Gamma_z(\boldsymbol{x}_p) a_\alpha (\boldsymbol{a} \cdot \nabla N_j(\boldsymbol{x}_p))$//Loop over particles and sum up
procedure SOLVE LINEAR SYSTEM
  Compute coefficient matrix: $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{M}^{-1}\boldsymbol{B}^T$, assemble RHS: $\boldsymbol{r} = \boldsymbol{B}\boldsymbol{M}^{-1}\boldsymbol{b}$
  Solve linear system $\boldsymbol{L}\boldsymbol{\lambda} = \boldsymbol{r}$ with conjugate gradient method
  Compute $\boldsymbol{v} = \{v_{i\alpha}\}_{i=1\cdots n_{node}, \alpha=1\cdots d} = \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{B}^T\boldsymbol{\lambda})$
  Update grid velocity using $v_{i\alpha}$

2 Anisotropic Damage

2.1 Explicit Damage Derivation

Starting from the continuous damage evolution equation in [6], we can use explicit integration to get a local damage evolution equation as follows:

$$\begin{aligned}
\dot{d} &= \frac{1}{\eta}\left\langle (1-d)\tilde{D} - (d - l_0^2\Delta d)\right\rangle \\
\frac{d^{n+1}-d^n}{\Delta t} &= \frac{1}{\eta}\left\langle (1-d^n)\tilde{D}^n - (d^n - l_0^2\Delta d^n)\right\rangle \\
d^{n+1} &= d^n + \frac{\Delta t}{\eta}\left\langle (1-d^n)\tilde{D}^n - (d^n - l_0^2\Delta d^n)\right\rangle
\end{aligned}$$

Thus, our explicit damage update scheme becomes:

$$d_p^{n+1} = \min\left(1, d_p^n + \frac{\Delta t}{\eta}\langle (1-d_p^n)\tilde{D}_p^n - (d_p^n - l_0^2\Delta d_p^n)\rangle\right)$$

where we take the min with 1 to enforce $d \in [0,1]$. $\tilde{D}_p^n$ is the maximum value in the history of particle p's driving force, $\tilde{D}_p^H$, such that $\tilde{D}_p^n = \max(\tilde{D}_p^H, \zeta\langle\Phi(\boldsymbol{\sigma}_p^{+n}) - 1\rangle)$. Note that, for explicit damage, we directly update damage over particles, while the grid is only used for computing the damage Laplacians like so:

$$\Delta d_p^n = \sum_i d_i^n \Delta N_i^n \quad \text{with} \quad d_i^n = \frac{\sum_p w_{ip}^n d_p^n}{\sum_p w_{ip}^n},$$

and $\Delta N_i^n = N_i''(x)N_i(y)N_i(z) + N_i(x)N_i''(y)N_i(z) + N_i(x)N_i(y)N_i''(z)$ where $N_{\boldsymbol{i}}(\boldsymbol{x}_p)$ is the interpolation function used for transfer. Note that this computation requires the second derivative, mandating that we use at least cubic $N_{\boldsymbol{i}}(\boldsymbol{x}_p)$.

2.2 Implicit Damage Derivation

Here we begin with the same local damage evolution equation and discretize it using implicit integration as follows:

$$\begin{aligned}
\eta\dot{d} &= (1-d)\tilde{D} - (d - l_0^2\Delta d) \text{ //Local damage evolution rule} \\
\frac{d^{n+1}-d^n}{\Delta t} &= \frac{1}{\eta}\left\langle (1-d^{n+1})\tilde{D} - (d^{n+1} - l_0^2\Delta d^{n+1})\right\rangle \\
d^{n+1} - d^n &= \frac{\Delta t}{\eta}\left\langle \tilde{D} - d^{n+1}\tilde{D} - d^{n+1} + l_0^2\Delta d^{n+1})\right\rangle \\
d^{n+1} - d^n &= \frac{\Delta t}{\eta}\left(\tilde{D} - d^{n+1}\tilde{D} - d^{n+1} + l_0^2\Delta d^{n+1})\right) \text{ //Can remove Macauley brackets if we enforce history of } \tilde{D} \\
d^{n+1} - d^n &= \frac{\Delta t}{\eta}\tilde{D} - \frac{\Delta t}{\eta}d^{n+1}\tilde{D} - \frac{\Delta t}{\eta}d^{n+1} + \frac{\Delta t}{\eta}l_0^2\Delta d^{n+1} \\
d^{n+1} + \frac{\Delta t}{\eta}d^{n+1}\tilde{D} + \frac{\Delta t}{\eta}d^{n+1} - \frac{\Delta t}{\eta}l_0^2\Delta d^{n+1} &= \frac{\Delta t}{\eta}\tilde{D} + d^n \\
d^{n+1}(1 + \frac{\Delta t}{\eta}\tilde{D} + \frac{\Delta t}{\eta}) - \frac{\Delta t}{\eta}l_0^2\Delta d^{n+1} &= \frac{\Delta t}{\eta}\tilde{D} + d^n \\
(1 + \frac{\Delta t}{\eta}(\tilde{D}^n + 1))d^{n+1} - (\frac{\Delta t}{\eta}l_0^2)\Delta d^{n+1} &= \frac{\Delta t}{\eta}\tilde{D}^n + d^n
\end{aligned}$$

At this point we may write an Eulerian weak form of this PDE as follows:

$$\int_{\Omega^t} \omega((\alpha - \beta\Delta)d^{n+1} - \gamma)d\boldsymbol{x} = 0,$$

where $\alpha(\boldsymbol{x},t)$, $\beta(\boldsymbol{x},t)$, and $\gamma(\boldsymbol{x},t)$ are corresponding coefficients, and $\omega(\boldsymbol{x},t)$ is the arbitrary test function. From this point we use MLS-MPM [1] to get a positive semi definite linear system to solve for unknown grid damage values, d_i^{n+1}. For this system we follow the derivation presented by Wolper et al. 2019 in section 3 of their supplemental document [9].

3 Anisotropic Elasticity

3.1 QR-based General Anisotropic Hyperelasticity

Drawing inspirations from [2], we can design arbitrary anisotropic hyperelastic constitutive models through QR decomposition of $\boldsymbol{F}$, i.e. $\boldsymbol{F}=\boldsymbol{Q}\boldsymbol{R}$ where $\boldsymbol{Q}$ is a rotation, and $\boldsymbol{R}$ is upper-triangular.

$$\boldsymbol{R}=\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \end{pmatrix}$$

Similar to standard SVD type elasticity, in 2D we require a sign convention that r_{11} is non-negative to make sure that $\boldsymbol{Q}$ is a rotation matrix. Since

$$\boldsymbol{F}=[r_{11}\boldsymbol{Q}_1 \quad r_{12}\boldsymbol{Q}_1+r_{22}\boldsymbol{Q}_2]$$

we can enforce this by flipping the sign of the entire $\boldsymbol{Q}$ and $\boldsymbol{R}$ if after QR decomposition we detect $r_{11}<0$. Similarly, since in 3D we have

$$\boldsymbol{F}=[r_{11}\boldsymbol{Q}_1 \quad r_{12}\boldsymbol{Q}_1+r_{22}\boldsymbol{Q}_2 \quad r_{13}\boldsymbol{Q}_1+r_{23}\boldsymbol{Q}_2+r_{33}\boldsymbol{Q}_3]$$

We can enforce a sign convention that r_{11} and r_{22} are non-negative by saying: if after QR, we detect $r_{11}<0$, then we flip the sign of r_{11}, $\boldsymbol{Q}_1$, r_{12}, r_{13}, r_{33}, $\boldsymbol{Q}_3$. Then if we further detect $r_{22}<0$, we flip the sign of r_{22}, $\boldsymbol{Q}_2$, r_{23}, r_{33}, $\boldsymbol{Q}_3$.

We define our elastic energy as:

$$\hat{\Psi}=\frac{\mu}{2}(\sum_{ij} r_{ij}^2-d)-\mu(J-1)+\frac{\lambda}{2}(J-1)^2+\frac{k_x}{2}(r_{11}-1)^2+\frac{k_y}{2}(\sqrt{r_{12}^2+r_{22}^2}-1)^2.$$

The first three terms, inspired by stable Neo-Hookean [8], consist of the isotropic part of our energy. We add the last two terms to penalize the x and y direction fiber stretching. For 2D, we add only the k_x term, and the energy formula becomes

$$\hat{\Psi}=\frac{\mu}{2}(\sum_{ij} r_{ij}^2-d)-\mu(J-1)+\frac{\lambda}{2}(J-1)^2+\frac{k_x}{2}(r_{11}-1)^2.$$

3.2 Stress

In [3] it was proven that given $\Psi(\boldsymbol{F})=\hat{\Psi}(\boldsymbol{R})$, the stress $\boldsymbol{P}$ can be computed through

$$\boldsymbol{Q}^T\boldsymbol{P}\boldsymbol{R}^T=\boldsymbol{Q}^T\frac{\partial\Psi}{\partial\boldsymbol{F}}\boldsymbol{R}^T=\boldsymbol{A}=\begin{pmatrix} \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{11} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{12} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{13} \\ \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{12} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{22} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{23} \\ \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{13} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{23} & \left(\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T\right)_{33} \end{pmatrix}$$

Note that, in our method, $J=\prod r_{ii}$; thus, for 3D we have

$$\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}=\begin{bmatrix} K_1 & \mu r_{12}+k_y r_{12}(1-\frac{1}{\sqrt{r_{12}^2+r_{22}^2}}) & \mu r_{13} \\ 0 & K_2 & \mu r_{23} \\ 0 & 0 & K_3 \end{bmatrix}$$

where $K_1 = \mu(r_{11} - r_{22}r_{33}) + \lambda(J-1)r_{22}r_{33} + k_x(r_{11}-1)$, $K_2 = \mu(r_{22} - r_{11}r_{33}) + \lambda(J-1)r_{33}r_{11} + k_y r_{22}(1 - \frac{1}{\sqrt{r_{12}^2 + r_{22}^2}})$ and $K_3 = \mu(r_{33} - r_{11}r_{22}) + \lambda(J-1)r_{11}r_{22}$. For 2D we have

$$\frac{\partial \hat{\Psi}}{\partial \boldsymbol{R}} = \begin{bmatrix} \mu(r_{11} - r_{22}) + \lambda(J-1)r_{22} + k_x(r_{11}-1) & \mu r_{12} \\ 0 & \mu(r_{22} - r_{11}) + \lambda(J-1)r_{11} \end{bmatrix}$$

We can then compute $\boldsymbol{P}$ using the above formula. Later, we will show that our QR-based energy always has a bounded first Piola-Kirchhoff stress under arbitrary circumstances.

3.3 Stress Differential

Note that [3] has shown how to compute $\delta \boldsymbol{Q}$ and $\delta \boldsymbol{R}$. Here we further derive $\delta \boldsymbol{P}$, which is essential for implicit QR energy. Starting from $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{A}\boldsymbol{R}^{-T}$, we have

$$\delta \boldsymbol{P} = \delta \boldsymbol{Q}\boldsymbol{A}\boldsymbol{R}^{-T} + \boldsymbol{Q}\delta \boldsymbol{A}\boldsymbol{R}^{-T} + \boldsymbol{Q}\boldsymbol{A}\delta(\boldsymbol{R}^{-T}).$$

The first term is

$$\begin{aligned} \delta \boldsymbol{Q}\boldsymbol{A}\boldsymbol{R}^{-T} &= \delta \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{P}\boldsymbol{R}^T\boldsymbol{R}^{-T} \\ &= -\boldsymbol{Q}\delta \boldsymbol{Q}^T\boldsymbol{P} \end{aligned}$$

where we have used $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$ to get the last equality.

For the second term, note that for a given energy, $\hat{\Psi}$, we explicitly know the entries in $\boldsymbol{A}$ as a function of entries of $\boldsymbol{R}$. Thus we can explicitly build $\delta \boldsymbol{A}$. That allows us to compute the second term.

The third term is

$$\begin{aligned} \boldsymbol{Q}\boldsymbol{A}\delta(\boldsymbol{R}^{-T}) &= \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{P}\boldsymbol{R}^T\delta(\boldsymbol{R}^{-T}) \\ &= \boldsymbol{P}\boldsymbol{R}^T\delta(\boldsymbol{R}^{-T}) \\ &= -\boldsymbol{P}\delta \boldsymbol{R}^T\boldsymbol{R}^{-T} \end{aligned}$$

where we have used $\boldsymbol{R}^T\boldsymbol{R}^{-T} = \boldsymbol{I}$ to get the last equality.

Combining the above we get $\delta \boldsymbol{P}$.

3.4 Stress Derivative

We further show how to derive the stress derivative $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}$. Again, starting from $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{A}\boldsymbol{R}^{-T}$, we have

$$\begin{aligned} P_{ij} &= Q_{ik}A_{kl}(R^{-T})_{lj} \\ \frac{\partial P_{ij}}{\partial F_{ab}} &= \frac{\partial Q_{ik}}{\partial F_{ab}}A_{kl}(R^{-T})_{lj} + Q_{ik}\frac{\partial A_{kl}}{\partial F_{ab}}(R^{-T})_{lj} + Q_{ik}A_{kl}\frac{\partial (R^{-T})_{lj}}{\partial F_{ab}} \\ &= \frac{\partial Q_{ik}}{\partial F_{ab}}A_{kl}(R^{-T})_{lj} + Q_{ik}\frac{\partial A_{kl}}{\partial r_{cd}}\frac{\partial r_{cd}}{\partial F_{ab}}(R^{-T})_{lj} + Q_{ik}A_{kl}\frac{\partial (R^{-T})_{lj}}{\partial F_{ab}} \end{aligned}$$

The first term can be computed if we know how to compute $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{F}}$, the second term can be constructed if we know how to compute $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}}$, and the third term can be computed by noticing

$$\boldsymbol{R}^{-T} = \boldsymbol{Q}^T\frac{1}{J}J\boldsymbol{F}^{-T},$$

whose derivative can be computed if we can compute $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{F}}$ and the derivative of the cofactor matrix $J\boldsymbol{F}^{-T}$.

3.4.1 Derivative of R

If we can compute $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{F}}$, then we can construct $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}}$ through:

$$\begin{aligned} r_{ij} &= Q_{ik}^T F_{kj} \\ \frac{\partial r_{ij}}{\partial F_{ab}} &= \frac{\partial Q_{ki}}{\partial F_{ab}} F_{kj} + Q_{ik}^T \delta_{ka} \delta_{jb} \end{aligned}$$

3.4.2 Derivative of Q

All that is left is computing $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{F}}$. Through differentiating both sides of $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$ w.r.t. F_{ij} we obtain:

$$\frac{\partial \boldsymbol{Q}^T}{\partial F_{ij}} \boldsymbol{Q} + \boldsymbol{Q}^T \frac{\partial \boldsymbol{Q}}{\partial F_{ij}} = 0,$$

which means $\boldsymbol{Q}^T \frac{\partial \boldsymbol{Q}}{\partial F_{ij}}$ is skew symmetric and can be written as:

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

By differentiating both sides of $\boldsymbol{F} = \boldsymbol{Q}\boldsymbol{R}$ w.r.t. F_{ij}, and left multiplying $\boldsymbol{Q}^T$ on both sides we obtain:

$$\boldsymbol{Q}^T \frac{\partial \boldsymbol{F}}{\partial F_{ij}} = \boldsymbol{Q}^T \frac{\partial \boldsymbol{Q}}{\partial F_{ij}} \boldsymbol{R} + \frac{\partial \boldsymbol{R}}{\partial F_{ij}}$$

which is essentially:

$$\begin{bmatrix} (Q^T \frac{\partial F}{\partial F_{ij}})_{11} & (Q^T \frac{\partial F}{\partial F_{ij}})_{12} & (Q^T \frac{\partial F}{\partial F_{ij}})_{13} \\ (Q^T \frac{\partial F}{\partial F_{ij}})_{21} & (Q^T \frac{\partial F}{\partial F_{ij}})_{22} & (Q^T \frac{\partial F}{\partial F_{ij}})_{23} \\ (Q^T \frac{\partial F}{\partial F_{ij}})_{31} & (Q^T \frac{\partial F}{\partial F_{ij}})_{32} & (Q^T \frac{\partial F}{\partial F_{ij}})_{33} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} + \begin{bmatrix} (\frac{\partial R}{\partial F_{ij}})_{11} & (\frac{\partial R}{\partial F_{ij}})_{12} & (\frac{\partial R}{\partial F_{ij}})_{13} \\ 0 & (\frac{\partial R}{\partial F_{ij}})_{22} & (\frac{\partial R}{\partial F_{ij}})_{23} \\ 0 & 0 & (\frac{\partial R}{\partial F_{ij}})_{33} \end{bmatrix},$$

and then we know:

$$\begin{aligned} Q_{i2}\delta_{j1} &= \omega_3 r_{11} \\ Q_{i3}\delta_{j1} &= -\omega_2 r_{11} \\ Q_{i3}\delta_{j2} &= -\omega_2 r_{12} + \omega_1 r_{22} \end{aligned}$$

which can be used to construct $\frac{\partial \boldsymbol{Q}}{\partial F_{ij}}$ by computing:

$$\frac{\partial \boldsymbol{Q}}{\partial F_{ij}} = \boldsymbol{Q}(\boldsymbol{Q}^T \frac{\partial \boldsymbol{Q}}{\partial F_{ij}}) = \begin{bmatrix} Q_{12}\omega_3 - Q_{13}\omega_2 & -Q_{11}\omega_3 + Q_{13}\omega_1 & Q_{11}\omega_2 - Q_{12}\omega_1 \\ Q_{22}\omega_3 - Q_{23}\omega_2 & -Q_{21}\omega_3 + Q_{23}\omega_1 & Q_{21}\omega_2 - Q_{22}\omega_1 \\ Q_{32}\omega_3 - Q_{33}\omega_2 & -Q_{31}\omega_3 + Q_{33}\omega_1 & Q_{31}\omega_2 - Q_{32}\omega_1 \end{bmatrix}.$$

3.4.3 MATLAB code

compute_dQ_div_dF.m

```
F = randn(3,3);
[Q, R] = qr(F);

% for faster computation
one_div_R11 = 1.0 / R(1,1);
one_div_R22 = 1.0 / R(2,2);
```

```
% 1,1
w3 = Q(1,2) * one_div_R11;
w2 = -Q(1,3) * one_div_R11;
w1 = w2 * R(1,2) * one_div_R22;
dQ_div_dF11 = Q_mult_Omega(Q, w1, w2, w3);

% check with finite diff
perturb = 1e-7;
[Q_p, R_p] = qr(F + [perturb, 0, 0; 0, 0, 0; 0, 0, 0]);
err = (Q_p - Q) / perturb - dQ_div_dF11
```

Q_mult_Omega.m

```
function result = Q_mult_Omega(Q, w1, w2, w3)
    result = [Q(1,2) * w3 - Q(1,3) * w2, ...
        -Q(1,1) * w3 + Q(1,3) * w1, ...
        Q(1,1) * w2 - Q(1,2) * w1;
        Q(2,2) * w3 - Q(2,3) * w2, ...
        -Q(2,1) * w3 + Q(2,3) * w1, ...
        Q(2,1) * w2 - Q(2,2) * w1;
        Q(3,2) * w3 - Q(3,3) * w2, ...
        -Q(3,1) * w3 + Q(3,3) * w1, ...
        Q(3,1) * w2 - Q(3,2) * w1];
end
```

3.5 Bounded Stress

We now address the design choices of our energy density function and, furthermore, show that our energy has bounded stress everywhere.

3.5.1 2D

For arbitrary energy $\hat{\Psi} = \hat{\Psi}(\boldsymbol{R})$ in 2D, we now derive the stress using the method mentioned above to see what necessary conditions should be met.

$$\boldsymbol{Q}^T\boldsymbol{P}\boldsymbol{R}^T = \boldsymbol{A}$$

$$\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}} = \begin{pmatrix} \frac{\partial\hat{\Psi}}{\partial r_{11}} & \frac{\partial\hat{\Psi}}{\partial r_{12}} \\ & \frac{\partial\hat{\Psi}}{\partial r_{22}} \end{pmatrix}$$

$$\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}\boldsymbol{R}^T = \begin{pmatrix} \frac{\partial\hat{\Psi}}{\partial r_{11}}r_{11} + \frac{\partial\hat{\Psi}}{\partial r_{12}}r_{12} & \frac{\partial\hat{\Psi}}{\partial r_{12}}r_{22} \\ * & \frac{\partial\hat{\Psi}}{\partial r_{22}}r_{22} \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} \frac{\partial\hat{\Psi}}{\partial r_{11}}r_{11} + \frac{\partial\hat{\Psi}}{\partial r_{12}}r_{12} & \frac{\partial\hat{\Psi}}{\partial r_{12}}r_{22} \\ \frac{\partial\hat{\Psi}}{\partial r_{12}}r_{22} & \frac{\partial\hat{\Psi}}{\partial r_{22}}r_{22} \end{pmatrix}$$

$$\boldsymbol{R}^{-T} = \frac{1}{r_{11}r_{22}}\begin{pmatrix} r_{22} & 0 \\ -r_{12} & r_{11} \end{pmatrix}$$

$$\boldsymbol{Q}^T\boldsymbol{P} = \boldsymbol{A}\boldsymbol{R}^{-T} = \begin{pmatrix} \frac{\partial\hat{\Psi}}{\partial r_{11}} & \frac{\partial\hat{\Psi}}{\partial r_{12}} \\ \frac{\partial\hat{\Psi}}{\partial r_{12}}\frac{r_{22}}{r_{11}} - \frac{\partial\hat{\Psi}}{\partial r_{22}}\frac{r_{12}}{r_{11}} & \frac{\partial\hat{\Psi}}{\partial r_{22}} \end{pmatrix}$$

Ultimately for $\boldsymbol{P}$ to be well defined, we need:

$$\frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} - \frac{\partial \hat{\Psi}}{\partial r_{22}} r_{12}$$

to either have r_{11} as a factor, or be equal to zero.

Using the 2D version of $\frac{\partial \hat{\Psi}}{\partial \boldsymbol{R}}$ derived in 3.2, we can easily verify that for our energy this term equals to $r_{11} r_{12}(\mu - \lambda(J-1))$, which is a multiple of r_{11}; hence, the computed stress is always bounded.

3.5.2 3D

$$\boldsymbol{Q}^T \boldsymbol{P} \boldsymbol{R}^T = \boldsymbol{A}$$

$$\frac{\partial \hat{\Psi}}{\partial \boldsymbol{R}} = \begin{pmatrix} \frac{\partial \hat{\Psi}}{\partial r_{11}} & \frac{\partial \hat{\Psi}}{\partial r_{12}} & \frac{\partial \hat{\Psi}}{\partial r_{13}} \\ & \frac{\partial \hat{\Psi}}{\partial r_{22}} & \frac{\partial \hat{\Psi}}{\partial r_{23}} \\ & & \frac{\partial \hat{\Psi}}{\partial r_{33}} \end{pmatrix}$$

$$\frac{\partial \hat{\Psi}}{\partial \boldsymbol{R}} \boldsymbol{R}^T = \begin{pmatrix} \frac{\partial \hat{\Psi}}{\partial r_{11}} r_{11} + \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{12} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{13} & \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} \\ * & \frac{\partial \hat{\Psi}}{\partial r_{22}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{33} \\ * & * & \frac{\partial \hat{\Psi}}{\partial r_{33}} r_{33} \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} \frac{\partial \hat{\Psi}}{\partial r_{11}} r_{11} + \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{12} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{13} & \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} \\ \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{22}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{33} \\ \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} & \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{33} & \frac{\partial \hat{\Psi}}{\partial r_{33}} r_{33} \end{pmatrix}$$

$$\boldsymbol{R}^{-T} = \frac{1}{r_{11} r_{22} r_{33}} \begin{pmatrix} r_{22} r_{33} & & \\ -r_{12} r_{33} & r_{11} r_{33} & \\ r_{12} r_{23} - r_{13} r_{22} & -r_{11} r_{23} & r_{11} r_{22} \end{pmatrix}$$

$$\boldsymbol{Q}^T \boldsymbol{P} = \boldsymbol{A} \boldsymbol{R}^{-T} = \begin{pmatrix} \frac{\partial \hat{\Psi}}{\partial r_{11}} r_{11} + \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{12} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{13} & \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} \\ \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{22}} r_{22} + \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{23} & \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{33} \\ \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} & \frac{\partial \hat{\Psi}}{\partial r_{23}} r_{33} & \frac{\partial \hat{\Psi}}{\partial r_{33}} r_{33} \end{pmatrix} \frac{1}{r_{11} r_{22} r_{33}}$$
$$\begin{pmatrix} r_{22} r_{33} & & \\ -r_{12} r_{33} & r_{11} r_{33} & \\ r_{12} r_{23} - r_{13} r_{22} & -r_{11} r_{23} & r_{11} r_{22} \end{pmatrix}$$

The $(1,1)$ term is $\frac{1}{r_{11} r_{22} r_{33}}$ times

$$\frac{\partial \hat{\Psi}}{\partial r_{11}} r_{11} r_{22} r_{33} + \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{12} r_{22} r_{33} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{13} r_{22} r_{33} - \frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} r_{12} r_{33} - \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} r_{12} r_{33} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} r_{12} r_{23} - \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} r_{13} r_{22}$$

giving no divisions.

The $(1,2)$ term is $\frac{1}{r_{11} r_{22} r_{33}}$ times

$$\frac{\partial \hat{\Psi}}{\partial r_{12}} r_{22} r_{11} r_{33} + \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{23} r_{11} r_{33} - \frac{\partial \hat{\Psi}}{\partial r_{13}} r_{33} r_{11} r_{23}$$

giving no divisions.

The $(1,3)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{13}}r_{33}r_{11}r_{22}$$

giving no divisions.

The $(2,1)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{12}}r_{22}r_{22}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{13}}r_{23}r_{22}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{22}}r_{22}r_{12}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{23}r_{12}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{12}r_{23}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{13}r_{22}$$

The $(2,2)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{22}}r_{22}r_{11}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{23}r_{11}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{11}r_{23}$$

giving no divisions.

The $(2,3)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{11}r_{22}$$

giving no divisions.

The $(3,1)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{13}}r_{33}r_{22}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{12}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{12}r_{23}-\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{13}r_{22}$$

The $(3,2)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{11}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{11}r_{23}$$

The $(3,3)$ term is $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{11}r_{22}$$

giving no divisons.

In summary, the appearance of divisions are all in the lower left non-diagonal terms: $\frac{1}{r_{11}r_{22}r_{33}}$ times

$$\begin{aligned}
&\frac{\partial\hat{\Psi}}{\partial r_{12}}r_{22}r_{22}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{13}}r_{23}r_{22}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{22}}r_{22}r_{12}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{23}r_{12}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{12}r_{23}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{13}r_{22}\\
&\frac{\partial\hat{\Psi}}{\partial r_{13}}r_{33}r_{22}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{12}r_{33}+\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{12}r_{23}-\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{13}r_{22}\\
&\frac{\partial\hat{\Psi}}{\partial r_{23}}r_{33}r_{11}r_{33}-\frac{\partial\hat{\Psi}}{\partial r_{33}}r_{33}r_{11}r_{23}
\end{aligned}$$

which are

$$\begin{aligned}
&\frac{\partial\hat{\Psi}}{\partial r_{12}}\frac{r_{22}}{r_{11}}+\frac{\partial\hat{\Psi}}{\partial r_{13}}\frac{r_{23}}{r_{11}}-\frac{\partial\hat{\Psi}}{\partial r_{22}}\frac{r_{12}}{r_{11}}-\frac{\partial\hat{\Psi}}{\partial r_{23}}\frac{r_{13}}{r_{11}}\\
&\frac{\partial\hat{\Psi}}{\partial r_{13}}\frac{r_{33}}{r_{11}}-\frac{\partial\hat{\Psi}}{\partial r_{23}}\frac{r_{12}r_{33}}{r_{11}r_{22}}+\frac{\partial\hat{\Psi}}{\partial r_{33}}\frac{r_{12}r_{23}}{r_{11}r_{22}}-\frac{\partial\hat{\Psi}}{\partial r_{33}}\frac{r_{13}}{r_{11}}\\
&\frac{\partial\hat{\Psi}}{\partial r_{23}}\frac{r_{33}}{r_{22}}-\frac{\partial\hat{\Psi}}{\partial r_{33}}\frac{r_{23}}{r_{22}}
\end{aligned}$$

Using $\frac{\partial\hat{\Psi}}{\partial\boldsymbol{R}}$ derived in 3.2, we can verify that all three of these expressions give no division; thus, our elastic energy can be used to compute stress under an arbitrary setting.

3.6 Energy Degradation

To allow damage to degrade our elastic potential, we use a typical degradation function [7, 5], $g(d)$, such that $g \in [0, 1]$ and choose to selectively degrade specific terms like so:

- We always degrade the *shearing term* (our two μ terms) by multiplying it by g.
- We conditionally degrade the *volumetric term* (the λ term) when $J > 1$ (volume increase, associated with tension) by multiplying it by g.
- We **never** degrade the k_x and k_y terms to allow for the fiber elasticity to remain intact.

We write our QR-elasticity in the following form, splitting it into a tension term, $\Psi^+(\boldsymbol{R})$, a compression term, $\Psi^-(\boldsymbol{R})$, and a fiber term, $\Psi^f(\boldsymbol{R})$, as follows:

$$\Psi(\boldsymbol{R}) = \Psi^+(\boldsymbol{R}) + \Psi^-(\boldsymbol{R}) + \Psi^f(\boldsymbol{R}).$$

This incorporates our aforementioned terms in the following way:

$$\Psi^+(\boldsymbol{R}) = \begin{cases} \Psi^\mu(\boldsymbol{R}) + \Psi^\lambda(\boldsymbol{R}) & J \geq 1 \\ \Psi^\mu(\boldsymbol{R}) & J < 1 \end{cases}, \quad \Psi^-(\boldsymbol{R}) = \begin{cases} 0 & J \geq 1 \\ \Psi^\lambda(\boldsymbol{R}) & J < 1 \end{cases}.$$

Thus, we can simply degrade elasticity like so, successfully only degrading the tension portion of our elasticity:

$$\hat{\Psi}(\boldsymbol{R}) = g(d)\Psi^+(\boldsymbol{R}) + \Psi^-(\boldsymbol{R}) + \Psi^f(\boldsymbol{R}).$$

4 Inextensibility

4.1 Derivation of Constraint Equation

Our inextensibility constraint equation is:

$$(\boldsymbol{a}\boldsymbol{a}^T) : \boldsymbol{d} = 0. \tag{1}$$

In the following context, we will use notations and concepts from standard MPM and continuum mechanics. We list some of them here for reference. Assuming we have a deformation map $\varphi(\boldsymbol{X}, t)$, the deformation gradient is defined as $\boldsymbol{F} = \frac{\partial \varphi}{\partial \boldsymbol{X}}$. $J = |\boldsymbol{F}|$ denotes the determinant of $\boldsymbol{F}$. $\mathbf{C}$ denotes Cauchy-Green tensor defined as $\mathbf{C} = \boldsymbol{F}^T\boldsymbol{F}$ and $\boldsymbol{E}$ denotes the Green-Lagrangian strain tensor defined as $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}) = \frac{1}{2}(\mathbf{C} - 1)$.

Let $\boldsymbol{a}^0 = \boldsymbol{a}(\boldsymbol{x}, 0) = (a^0_\alpha)_{\alpha=1\cdots d}$ denote the initial fiber direction. The structural tensor is defined $\boldsymbol{M} = \boldsymbol{a}^0 \otimes \boldsymbol{a}^0$. In mechanics literature like [10], the constraint relation for inextensibility is $\boldsymbol{a}^0 \cdot \boldsymbol{E}\boldsymbol{a}^0 = 0$. We can rewrite the above condition as: $\boldsymbol{a}^0 \cdot \mathbf{C}\boldsymbol{a}^0 = (\boldsymbol{a}^0)^T\boldsymbol{a}^0 = 1$ ($\boldsymbol{a}^0$ is a unit vector). Notice that $\boldsymbol{a}^0 \cdot \mathbf{C}\boldsymbol{a}^0 = \mathbf{C} : \boldsymbol{M} = tr(\mathbf{C}\boldsymbol{M})$ ($\boldsymbol{a}^0 \cdot \mathbf{C}\boldsymbol{a}^0 = a^0_i(\mathbf{C}\boldsymbol{a}^0)_i = a^0_i C_{ij} a^0_j, tr(\mathbf{C}\boldsymbol{M}) = C_{ij}M_{ji} = C_{ij}a^0_j a^0_i$). Thus, the inextensibility constraint becomes:

$$tr(\mathbf{C}\boldsymbol{M}) = 1. \tag{2}$$

Note that $tr(\mathbf{C}\boldsymbol{M})$ equals 1 in the initial configuration. In order to keep the fiber inextensible, the derivative should equal 0 in all configurations. Thus, $\mathbf{C}$ satisfies $\frac{D(tr(\mathbf{C}\boldsymbol{M}))}{Dt} = 0$. We can derive that

$$0 = \frac{D(tr(\mathbf{C}\boldsymbol{M}))}{Dt} = tr(\dot{\mathbf{C}}\boldsymbol{M}) = tr(\boldsymbol{F}^T(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})\boldsymbol{F}\boldsymbol{M}). \tag{3}$$

Here the upper dot represents derivative with respect to time, and ∇ represents the spatial gradient. The second equal sign holds because $\boldsymbol{M}$ is a constant tensor. To derive the third equation, we can see from MPM literature that $\dot{\boldsymbol{F}} = \nabla \boldsymbol{v}\boldsymbol{F}$ Hence we have $\dot{\mathbf{C}} = (\dot{\boldsymbol{F}^T}\boldsymbol{F}) = \dot{\boldsymbol{F}}^T\boldsymbol{F} + \boldsymbol{F}^T\dot{\boldsymbol{F}} = (\nabla \boldsymbol{v}\boldsymbol{F})^T\boldsymbol{F} + \boldsymbol{F}^T\nabla \boldsymbol{v}\boldsymbol{F} = \boldsymbol{F}^T(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})\boldsymbol{F}$.

Our formulation then becomes $tr(\boldsymbol{F}^T(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})\boldsymbol{F}^n\boldsymbol{M}) = 0$. Hence, we have:

$$0 = \boldsymbol{F}_{ij}^T(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})_{jk}\boldsymbol{F}_{kl}a_l^0 a_i^0 = \boldsymbol{F}_{ij}^T a_i^0(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})_{jk}\boldsymbol{F}_{kl}a_l^0 = (\boldsymbol{F}\boldsymbol{a}^0)_j(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})_{jk}(\boldsymbol{F}\boldsymbol{a}^0)_k \tag{4}$$

where summation over i, j, k, l is implied. Notice that $\boldsymbol{a} = \boldsymbol{F}\boldsymbol{a}^0$(up to a scalar scaling) and $\boldsymbol{d} = \frac{1}{2}(\nabla \boldsymbol{v}^T + \nabla \boldsymbol{v})$. The above derivation gives $(\boldsymbol{a}\boldsymbol{a}^T) : \boldsymbol{d} = tr(\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{d}) = tr(\boldsymbol{a}^T\boldsymbol{d}\boldsymbol{a}) = a_j d_{jk} a_k = 0$. Thus, our formulation in Eq. (1) is proven.

4.2 Derivation of Weak Form

4.2.1 Momentum Equation

$$\rho(\boldsymbol{x}, t)\frac{D\boldsymbol{v}}{Dt} = \nabla^{\boldsymbol{x}} \cdot \boldsymbol{\sigma}_{\text{full}} + \rho(\boldsymbol{x}, t)\boldsymbol{g} \tag{5}$$

Through introducing test function, $\boldsymbol{q}(\boldsymbol{x}, t)$, and ignoring gravity for simplicity, the weak form of Eq. (5) is:

$$\int_{\Omega^t} \boldsymbol{q}(\boldsymbol{x}, t)\rho(\boldsymbol{x}, t)\frac{D\boldsymbol{v}}{Dt}d\boldsymbol{x} = \int_{\Omega^t} \boldsymbol{q}(\boldsymbol{x}, t)\nabla^{\boldsymbol{x}} \cdot (\boldsymbol{\sigma} + \lambda \boldsymbol{a}\boldsymbol{a}^T)d\boldsymbol{x}. \tag{6}$$

After applying integration by parts and divergence theorem, this becomes:

$$\int_{\Omega^t} q_\alpha(\boldsymbol{x}, t)\rho(\boldsymbol{x}, t)\frac{Dv_\alpha(\boldsymbol{x}, t)}{Dt}d\boldsymbol{x} = \int_{\partial\Omega^t} q_\alpha t_\alpha ds(\boldsymbol{x}) - \left(\int_{\Omega^t} q_{\alpha,\beta}\sigma_{\alpha\beta}d\boldsymbol{x} + \int_{\Omega^t} q_{\alpha,\beta}\lambda(\boldsymbol{x}, t)a_\alpha(\boldsymbol{x}, t)a_\beta(\boldsymbol{x}, t)d\boldsymbol{x}\right).$$

We'll assume a zero-traction boundary condition, so let's remove the boundary integral term and to arrive at:

$$\int_{\Omega^t} q_\alpha(\boldsymbol{x}, t)\rho(\boldsymbol{x}, t)\frac{Dv_\alpha(\boldsymbol{x}, t)}{Dt}d\boldsymbol{x} = -\left(\int_{\Omega^t} q_{\alpha,k}\sigma_{\alpha\beta}d\boldsymbol{x} + \int_{\Omega^t} q_{\alpha,k}\lambda(\boldsymbol{x}, t)a_\alpha(\boldsymbol{x}, t)a_\beta(\boldsymbol{x}, t)d\boldsymbol{x}\right). \tag{7}$$

Now, we follow the standard MPM weak form derivation and only focus on the final integration since the others are proven in [4]. By expanding velocity and q at nodes, and λ at cell centers as:

$$q_\alpha(\boldsymbol{x}) = q_{i\alpha}N_i(\boldsymbol{x}), \quad v_\alpha(\boldsymbol{x}) = v_{i\alpha}N_i(\boldsymbol{x}), \quad \lambda(\boldsymbol{x}) = \lambda_z\Gamma_z(\boldsymbol{x}),$$

we get:

$$\frac{1}{\Delta t}\int_{\Omega^t} q_{i\alpha}^n N_i(\boldsymbol{x})\rho(\boldsymbol{x}, t^n)(v_{j\alpha}^{n+1} - v_{j\alpha}^n)N_j(\boldsymbol{x}) = -\int_{\Omega^{t^n}} q_{i\alpha}^n N_{i,\beta}(\boldsymbol{x})\sigma_{\alpha\beta}(\boldsymbol{x}, t^n)d\boldsymbol{x} - \int_{\Omega^{t^n}} N_{i,\beta}a_\alpha a_\beta\lambda_z\Gamma_z(\boldsymbol{x})d\boldsymbol{x}. \tag{8}$$

Thus, following the same mass lumping strategy from [4], we have:

$$\frac{(mv)_{i\alpha}^{n+1} - (mv)_{i\alpha}^n}{\Delta t} = -\int_{\Omega^{t^n}} N_{i,\beta}(\boldsymbol{x})\sigma_{\alpha\beta}(\boldsymbol{x}, t^n)d\boldsymbol{x} - \int_{\Omega^{t^n}} N_{i,\beta}a_\alpha a_\beta\lambda_z\Gamma_z(\boldsymbol{x})d\boldsymbol{x}. \tag{9}$$

Rearranging this equation, we get an equation of unknown λ and $\boldsymbol{v}$:

$$\frac{m_i}{\Delta t}v_{i\alpha}^{n+1} + \left(\int_{\Omega^{t^n}} a_\alpha(\boldsymbol{a} \cdot \nabla N_i)\Gamma_z(\boldsymbol{x})d\boldsymbol{x}\right)\lambda_z = \frac{m_i}{\Delta t}v_{i\alpha}^n - \int_{\Omega^{t^n}} N_{i,\beta}(\boldsymbol{x})\sigma_{\alpha\beta}(\boldsymbol{x}, t^n)d\boldsymbol{x}. \tag{10}$$

For simplicity, let dim be the dimension of simulation, N_n be the number of grid nodes, and N_c be the number of grid cells. Eq.(9) gives $dim \times N_n$ equations of $v_{i\alpha}^{n+1}$ and λ_z. We further use $B_{(z,i\alpha)}$ to denote the coefficient of λ_z while using $b_{i\alpha}$ to denote the right hand side. This item relates the z-th cell and the α-th dimension of i-th node, i.e.

$$\frac{m_i}{\Delta t}v_{i\alpha}^{n+1} + B_{(z,i\alpha)}\lambda_z = b_{i\alpha} \tag{11}$$

4.2.2 Constraint

Recall Eq. (1). By introducing the test function, $h(\boldsymbol{x}, t)$, the weak form of Eq. (1) is:

$$\int_{\Omega^t} h(\boldsymbol{x}, t)(\boldsymbol{a}\boldsymbol{a}^T) : \boldsymbol{d} d\boldsymbol{x} = 0.$$

This is:

$$\int_{\Omega^t} h(\boldsymbol{x}, t)(\boldsymbol{a}\boldsymbol{a}^T) : \frac{1}{2}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}^T\right) d\boldsymbol{x} = 0$$
$$\frac{1}{2}\int_{\Omega^t} h a_\alpha a_\beta \left(\frac{\partial v_\alpha}{\partial x_\beta} + \frac{\partial v_\beta}{\partial x_\alpha}\right) d\boldsymbol{x} = 0$$

Through expanding velocity at nodes and h at cell centers like so:

$$v_\alpha(\boldsymbol{x}) = v_{i\alpha} N_i(\boldsymbol{x}), \quad h(\boldsymbol{x}) = h_z \Gamma_z(\boldsymbol{x}),$$

we get:

$$\frac{1}{2}\int_{\Omega^t} h_z \Gamma_z(\boldsymbol{x}) a_\alpha a_\beta \left(v_{i\alpha}\frac{\partial N_i(\boldsymbol{x})}{\partial x_\beta} + v_{k\beta}\frac{\partial N_k(\boldsymbol{x})}{\partial x_\alpha}\right) d\boldsymbol{x} = 0$$
$$\frac{1}{2}\int_{\Omega^t} h_z \Gamma_z(\boldsymbol{x}) a_\alpha a_\beta v_{i\alpha}\frac{\partial N_i(\boldsymbol{x})}{\partial x_\beta} + h_z \Gamma_z(\boldsymbol{x}) a_\alpha a_\beta v_{k\beta}\frac{\partial N_k(\boldsymbol{x})}{\partial x_\alpha} d\boldsymbol{x} = 0.$$

Let $h_z = \delta_{zc}$ (in turn picking an equation for each cell center c):

$$\frac{1}{2}\int_{\Omega^t} \Gamma_c(\boldsymbol{x}) a_\alpha a_\beta v_{i\alpha}\frac{\partial N_i(\boldsymbol{x})}{\partial x_\beta} + \Gamma_c(\boldsymbol{x}) a_\alpha a_\beta v_{k\beta}\frac{\partial N_k(\boldsymbol{x})}{\partial x_\alpha} d\boldsymbol{x} = 0.$$

Note that the two terms are equal to each other (to see this, we could switch k to i and swap α and β in the second term, since they are all dummy indices); therefore, we get:

$$\int_{\Omega^t} \Gamma_c(\boldsymbol{x}) a_\alpha a_\beta v_{i\alpha}\frac{\partial N_i(\boldsymbol{x})}{\partial x_\beta} d\boldsymbol{x} = 0,$$

i.e.,

$$\int_{\Omega^t} \Gamma_c(\boldsymbol{x}) a_\alpha (\boldsymbol{a} \cdot \nabla N_i) d\boldsymbol{x} v_{i\alpha} = 0 \tag{12}$$

In a stacked view, this defines a linear constraint on $\boldsymbol{v}$: $\boldsymbol{B}\boldsymbol{v} = 0$. One constraint per cell c.

Comparing Eq.(12) with Eq.(9) we can easily see that the coefficient of $v_{i\alpha}$ is exactly $B_{(c,i\alpha)}$. This gives N_c equations of $v_{i\alpha}$ and λ_c(with zero coefficient), i.e.

$$B_{(c,i\alpha)} v_{i\alpha} = 0. \tag{13}$$

4.3 KKT Relaxation

Directly solving the KKT system can be difficult both because it introduces more degrees of freedom (the Lagrange multipliers) as well as non-symmetric positive definite (SPD) nature of the system. Even if the Schur-complement is applied to solve for Lagrange multipliers and original unknowns in sequence to make the sub-systems SPD, the computational overhead is still large, especially when the upper-left part is not diagonal (when implicit time integration is applied). Likewise, the system can be super stiff and ill-conditioned. Thus, we investigate relaxing the KKT system by augmenting the lower right block with a diagonal matrix $\frac{1}{\eta}\boldsymbol{I}$.

Suppose we are solving:

$$\min_{\boldsymbol{x}} \boldsymbol{E}(\boldsymbol{x}) \quad s.t. \quad \mathbf{c}(\boldsymbol{x}) = 0 \tag{14}$$

with nonlinear energy, $\boldsymbol{E}(\boldsymbol{x})$, and linear constraints, $\mathbf{c}(\boldsymbol{x})$ and $(\nabla^2 \mathbf{c}(\boldsymbol{x}) = 0)$, the KKT system at iteration i is:

$$\begin{pmatrix} \nabla^2 \boldsymbol{E}(\boldsymbol{x}^i) & -\nabla^T \mathbf{c}(\boldsymbol{x}^i) \\ -\nabla \mathbf{c}(\boldsymbol{x}^i) & O \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla \boldsymbol{E}(\boldsymbol{x}^i) \\ \mathbf{c}(\boldsymbol{x}^i) \end{pmatrix}, \tag{15}$$

then the relaxed KKT system would become:

$$\begin{pmatrix} \nabla^2 \boldsymbol{E}(\boldsymbol{x}^i) & -\nabla^T \mathbf{c}(\boldsymbol{x}^i) \\ -\nabla \mathbf{c}(\boldsymbol{x}^i) & \frac{1}{\eta} \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla \boldsymbol{E}(\boldsymbol{x}^i) \\ \mathbf{c}(\boldsymbol{x}^i) \end{pmatrix}. \tag{16}$$

When the optimization converges at $\boldsymbol{x}^*$, the above system will be solved to give $\Delta \boldsymbol{x} = 0$, which means we have:

$$-\nabla^T \mathbf{c}(\boldsymbol{x}^*)\lambda^* = -\nabla \boldsymbol{E}(\boldsymbol{x}^*) \tag{17}$$

$$\frac{1}{\eta}\lambda^* = \mathbf{c}(\boldsymbol{x}^*). \tag{18}$$

By combining (18) and (17) to eliminate λ^* we obtain:

$$\nabla \boldsymbol{E}(\boldsymbol{x}^*) - \nabla^T \mathbf{c}(\boldsymbol{x}^*)\eta \mathbf{c}(\boldsymbol{x}^*) = 0, \tag{19}$$

which means the solved $\boldsymbol{x}^*$ from our relaxed KKT is also a local optimum of the penalty-based soft constraint version of our original problem:

$$\min_{\boldsymbol{x}} \boldsymbol{E}(\boldsymbol{x}) + \frac{-\eta}{2}||\mathbf{c}(\boldsymbol{x})||^2. \tag{20}$$

When $\eta \to -\infty$, the constraint in Eq. (20) will be strictly satisfied after convergence, which is essentially equivalent to not adding relaxation on our KKT formulation.

In this way, we can further demonstrate that a relaxation on our KKT formulation is essentially equivalent to solving anisotropic elasticity by properly defining $\mathbf{c}(\boldsymbol{x})$.

References

[1] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):150, 2018.

[2] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans Graph*, 36(4), 2017.

[3] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact supplementary technical document. 2017.

[4] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Course*, pages 24:1–24:52, 2016.

[5] C. Miehe, M. Hofacker, and F. Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2765–2778, 2010.

[6] C. Miehe, L.-M. Schänzel, and H. Ulmer. Phase field modeling of fracture in multi-physics problems. part i. balance of crack surface and failure criteria for brittle crack propagation in thermo-elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 294:449 – 485, 2015.

[7] C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field fe implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311, 2010.

[8] B. Smith, F. de Goes, and T. Kim. Stable neo-hookean flesh simulation. *ACM Trans. Graph.*, 37(2):12, 2018.

[9] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao, and C. Jiang. CD-MPM: Continuum damage material point methods for dynamic fracture animation. *ACM Trans. Graph.*, 38(4), July 2019.

[10] P. Wriggers, J Schröder, and F. Auricchio. Finite element formulations for large strain anisotropic material with inextensible fibers. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):25, 2016.