Today, we will address the last (for now...) challenge associated with quasistatic robust simulation of nonlinear materials \(\Rightarrow\) the positive-definiteness of the stiffness matrix \(K\) (or lack thereof).

\(\text{Remember:}\) We are using Newton's method to find the minimum-energy configuration for the function \(E(x)\). We do so by iterating

\[
x_{(0)} \leftarrow \text{initial guess} \\
\text{for } k = 0, 1, \ldots,
\]

\[
\text{Solve } \frac{dfe}{dx} \bigg|_{x_{(k)}} \cdot \delta x = f_{el} (x_{(k)}) \tag{1}
\]

\[
\text{Update } x_{(k+1)} = x_{(k)} + \delta x
\]

where eq (1) can also be seen as

\[
K(x_{(k)}) \delta x = f_{el} (x_{(k)}) \tag{2}
\]

or even

\[
\frac{\delta^2 E}{dx^2} \bigg|_{x_{(k)}} \cdot \delta x = -\frac{dE}{dx} \bigg|_{x_{(k)}} \tag{3}
\]
Our problem is that the matrix \( K(x) \) is not guaranteed to be positive definite for any \( x \).

**Quiz:**

→ What would be an example of a configuration \( x \) where \( K(x) \) would be expected to be indefinite?

→ Would you expect \( K(x_{(k)}) \) to be positive-definite for a "late" iteration, where \( x_{(k)} \) is almost converged and why? (or why not?)

The general philosophy behind our remedy:

→ Replace the stiffness matrix \( K(x_{(k)}) \) with a modified version \( \hat{K}(x_{(k)}) \) which is designed to be positive-definite (CG can be used!)

→ The method becomes modified Newton, iterating

\[
\hat{K}(x_{(k)}) \delta x = \text{feq}(x_{(k)})
\]
But why is this a reasonable remedy?

⇒ It depends on the nature of the modification \((k)\) but we can say the following:

⇒ For the specific method we will see, practice shows that modified Newton is very robust, especially when combined with inversion-robust fixes (truncating singular values to small positive).

⇒ Once the modified iteration is close enough to convergence (enough for \(KC(z)\) to be s.p.d.) we can show that

\[
p \left( I - \left[ K(x_N) \right]^{-1} K(x_N) \right) < 1
\]

\[\text{spectral radius}\]

then the modified Newton process is guaranteed to converge to the actual minimum.

(i.e. if we are almost converged, a small modification does not compromise the ability to converge all the way)

(In-class discussion: 2D nonzero-restlengthspring
\[E(\underline{z}) = \frac{k}{2} (\|\underline{x} - \underline{x}_0\| - l_0)^2\])
Outline of the modification

→ Split the stiffness matrix to the contribution of individual elements

\[ K(z) = \sum_{e} K^e(z) \]

e: element

Each \( K^e \) corresponds to the contribution of just the \( e \)-th element to the energy, i.e.

\[ E(z) = \sum_{e} E^e(z) \]

then \[ K^e : = - \frac{\partial E^e(z)}{\partial z} \]

if \((i, j, k, l)\) are the vertex indices, then only the blocks of \( K^e \) corresponding to row/column blocks \( i / j / k / l \) will be nonzero

\[ K_e = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]

Each "\( x \)" is a 3×3 matrix, so, effectively \( K_e \) is a 12×12 matrix padded by zeroes to become full-size.
The prescription for the remedy, then, becomes:

"Project" each elemental stiffness matrix to its positive definite part

$$\mathbf{K}^e = \mathbf{Q} \Lambda \mathbf{Q}^T$$

Only 12 of the \( \Lambda \)'s can be nonzero

(in fact, just \( \frac{9}{2} \) of them maximum... why?)

If we define

$$\hat{(\Lambda)}_{ii} = \begin{cases} (\Lambda)_{ii}, & \text{if } \Lambda_{ii} \geq 0 \\ 0, & \text{if } \Lambda_{ii} < 0 \end{cases}$$

and

$$\hat{K}^e := \mathbf{Q} \hat{(\Lambda)} \mathbf{Q}^T$$

Construct the "global" modified stiffness by assembly:

$$\hat{\mathbf{K}} := \sum_{e} \hat{K}^e$$

(some textbooks use an "assembly" operator \( \mathbf{A} \) to write

$$\hat{\mathbf{K}} := \mathbf{A} \hat{\mathbf{K}}^e$$

where \( \hat{K}^e \) are 12x12 matrices)
If we were prepared to pay the cost of a 12x12 eigenanalysis (9x9, with some effort...) this concludes our fix! (in some cases, we would have no other option).

There is, however, a shortcut to computing exactly the same result, with much reduced cost in the case of isotropic materials (those where $\psi(\Sigma)$)

**Implementation**

Remember the matrix-free computation of the differential $\delta f[x; \delta x]$

- **add Force Differential**
  
  foreach element $e=(i,j,k,l)$
  
  $D_e, F_e \leftarrow$ as before
  
  $\delta D_e, \delta F_e \leftarrow$ as before
  
  $\delta P^e = \delta P[F_e \delta F_e]$

- $\delta H_e \leftarrow \text{Vol}_e \cdot \delta P^e (D_e^{-1})$

- $\delta H_e \rightarrow \delta f_{ijkl}$ as before

**Definiteness fix**

Replace $\delta P[F_i \delta F]$ with modified $\hat{\delta P}[F_i \delta F]$ (next page)
And, here is the prescription for the modified "stress differential"

(Proofs: Teran et al '05 "Robust Quasistatics and Flesh Simulation", SCA '05
Stomakhin et al '12 "Energetically Consistent Invertible Elasticity", SCA '12)

Definitions

\[
\begin{align*}
\text{vec} \left[ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \right] &= \\
\text{mat} \left[ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{pmatrix} \right] &= \\
\begin{pmatrix} a_{11} \\ a_{22} \\ a_{33} \\ a_{12} \\ a_{21} \\ a_{13} \\ a_{31} \\ a_{23} \\ a_{32} \\ a_1 & a_4 & a_6 \\ a_5 & a_2 & a_8 \\ a_7 & a_9 & a_3 \end{pmatrix}
\end{align*}
\]

We can show that \( \Sigma P = U \text{mat} \left[ T \cdot \text{vec} (U^T SFV) \right] V^T \)

\( (F=U \Sigma V^T) \)
Where \( T = \begin{bmatrix} A & B_{12} \\ (3x3) & (2x2) \end{bmatrix} \)

\[ A_{ij}^2 = \frac{\partial^2 \psi}{\partial \sigma_i \partial \sigma_j} \]

\[ B_{ij} = \frac{1}{2} \left( \frac{\partial \psi / \partial \sigma_i - \partial \psi / \partial \sigma_j}{\sigma_i - \sigma_j} \right) \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \]

\[ + \frac{1}{2} \left( \frac{\partial \psi / \partial \sigma_i + \partial \psi / \partial \sigma_j}{\sigma_i + \sigma_j} \right) \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \]