Robustness - Implicit integration w/ nonlinearity

We previously identified several challenges associated with the simulation of nonlinear materials; today, we will start addressing those!

We will initially focus on Quasistatic simulation, as it will help demonstrate almost all of the proposed remedies We will extend to dynamic simulation later:

For quasistatics:

→ Velocities are presumed always zero
→ No damping forces
→ Any motion ⇒ Due to motion of boundary conditions — NOT inertia.
→ Newton update equation

Solve: $$\left[-\left.\frac{\partial f_{el}}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}_{(k)}} \right] \cdot \delta \underline{x} = f_{el}\left(\underline{x}_{(k)}\right)$$

Update: $$\underline{x}_{(k+1)} \longleftarrow \underline{x}_{(k)} + \delta \underline{x}$$

This is equivalent to solving a minimization problem (for the elastic potential energy)

$$\min_{\underline{x}} E(\underline{x})$$

(where $\underline{x}$ is those degrees of freedom that are allowed to move by the simulation).

Finding the minimum, $\underline{x}^*$, is done by requiring that the gradient is equal to zero:

$$\left.\frac{\partial E}{\partial \underline{x}}\right|_{\underline{x}^*} = 0 \quad \Longleftrightarrow \quad -f_{el}(\underline{x}^*) = 0$$

(or, just $f_{el}(\underline{x}^*) = 0$)

This is a nonlinear equation itself, so we use Newton's method (approximate with Taylor expansion) to get an approximation ($\underline{x}_{(k+1)}$) of $\underline{x}^*$, starting from a guess $\underline{x}_{(k)}$

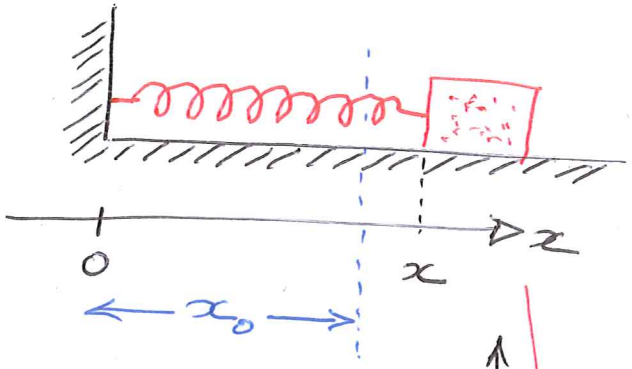
$$0 = f_{el}(\underline{x}_{(k+1)}) \approx f_{el}(\underline{x}_{(k)}) + \left.\frac{\partial f_{el}}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}_{(k)}} \cdot \delta \underline{x}$$

$(\delta x := \underline{x}_{(k+1)} - \underline{x}_{(k)})$

$$\Rightarrow \left[-\left.\frac{\partial f_{el}}{\partial \underline{x}}\right|_{x_{(k)}}\right] \cdot \delta \underline{x} = f(\underline{x}_{(k)}) \quad \underline{\text{or}} \quad \left(\left.\frac{\partial^2 E}{\partial \underline{x}^2}\right|_{\underline{x}_{(k)}}\right) \cdot \delta \underline{x} = -\left.\frac{\partial E}{\partial \underline{x}}\right|_{\underline{x}_{(k)}}$$

(1)

Let's try to intuitively understand this last formula. For this, consider the over-simplified case of x just being a single number, e.g.



← Imagine this being a "nonlinear spring" with the energy profile:

Local quadratic fit at $x_{(k)}$

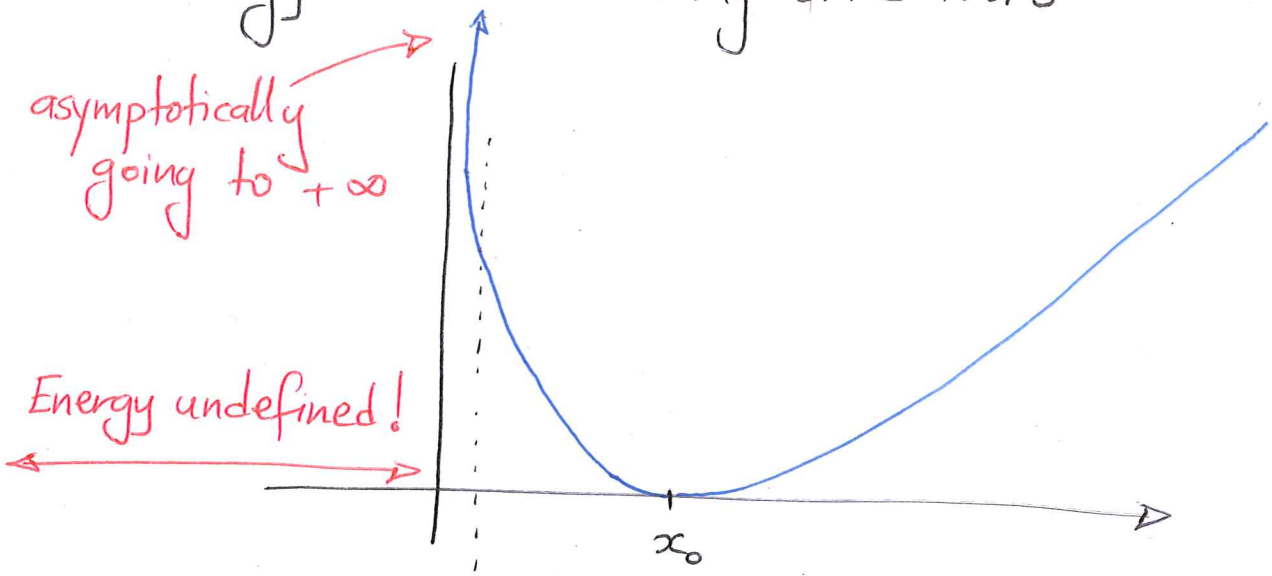
The minimum of this local quadratic approximation is taken as $x_{(k+1)}$!

This process is iterated to convergence. The multi-dimensional version of this (jumping to the minimum of a local, multi-D, quadratic fit) is exactly what equation (1) describes!

Now, we are ready to model (in simple 1D) one of the aforementioned challenges (forces becoming undefined). Consider a "Neo"-hookean spring, with

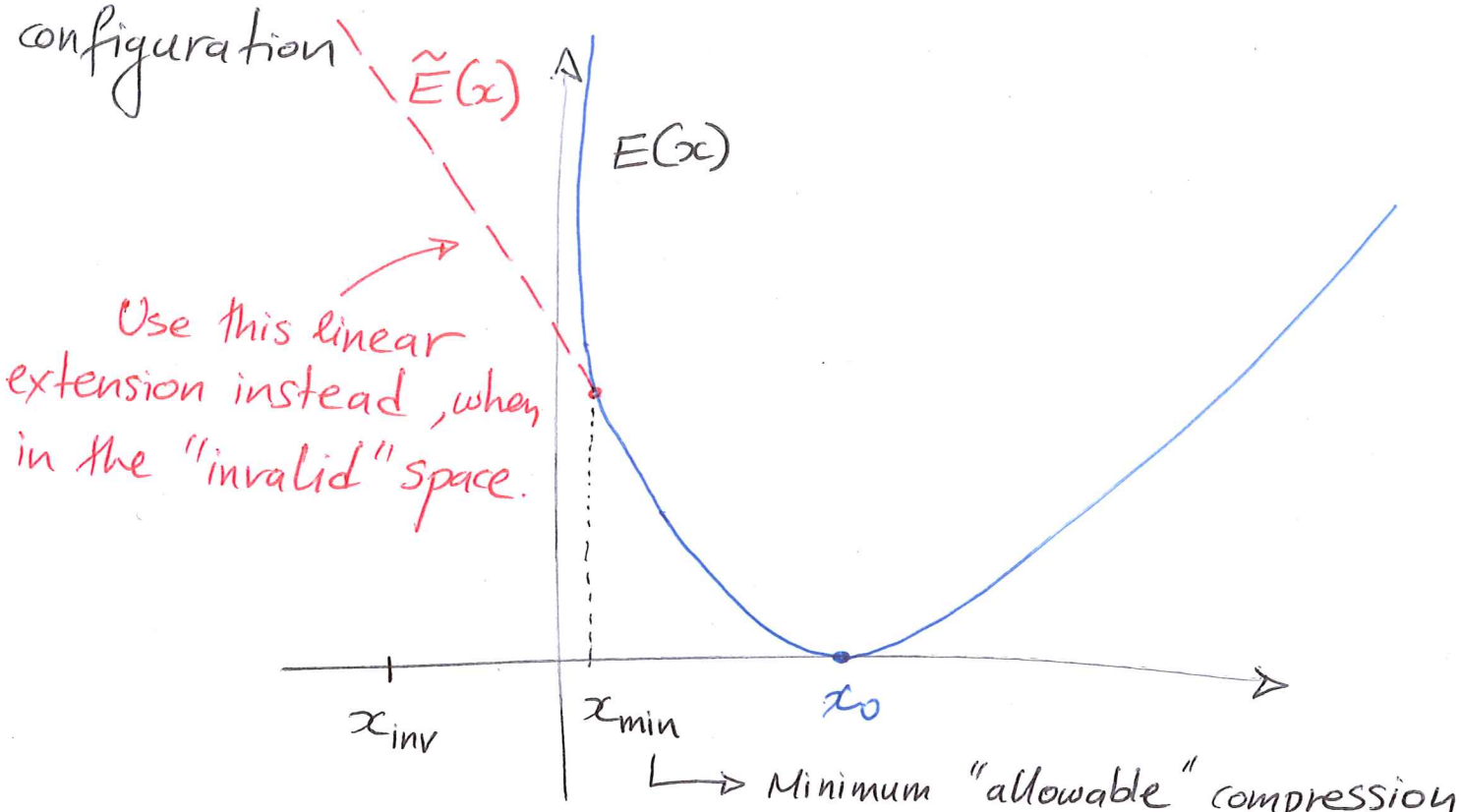
$$J := \frac{x}{x_0} \qquad E(x) := \frac{\mu}{2}(J^2 - 1) - \mu \log(J) + \frac{\lambda}{2}\log^2(J)$$

This energy looks something like this



When $x < x_0$, even if forces are still defined, they are certainly "uncomfortably" large for simulation. In the real world, the extreme energy penalty would prevent the spring from "flipping over", or even severely compressing $\to 0$ (there would be too high of an energy penalty to pay!). Of course if <u>in practice</u> x ever becomes too small – or negative – the forces will become undefined, and the simulation will fail!

Irving et.al (2004) "Invertible Finite Elements" proposed the following remedy: When getting "too close" to an invalid configuration, replace the energy curve with the linear extension at a limit/threshold configuration

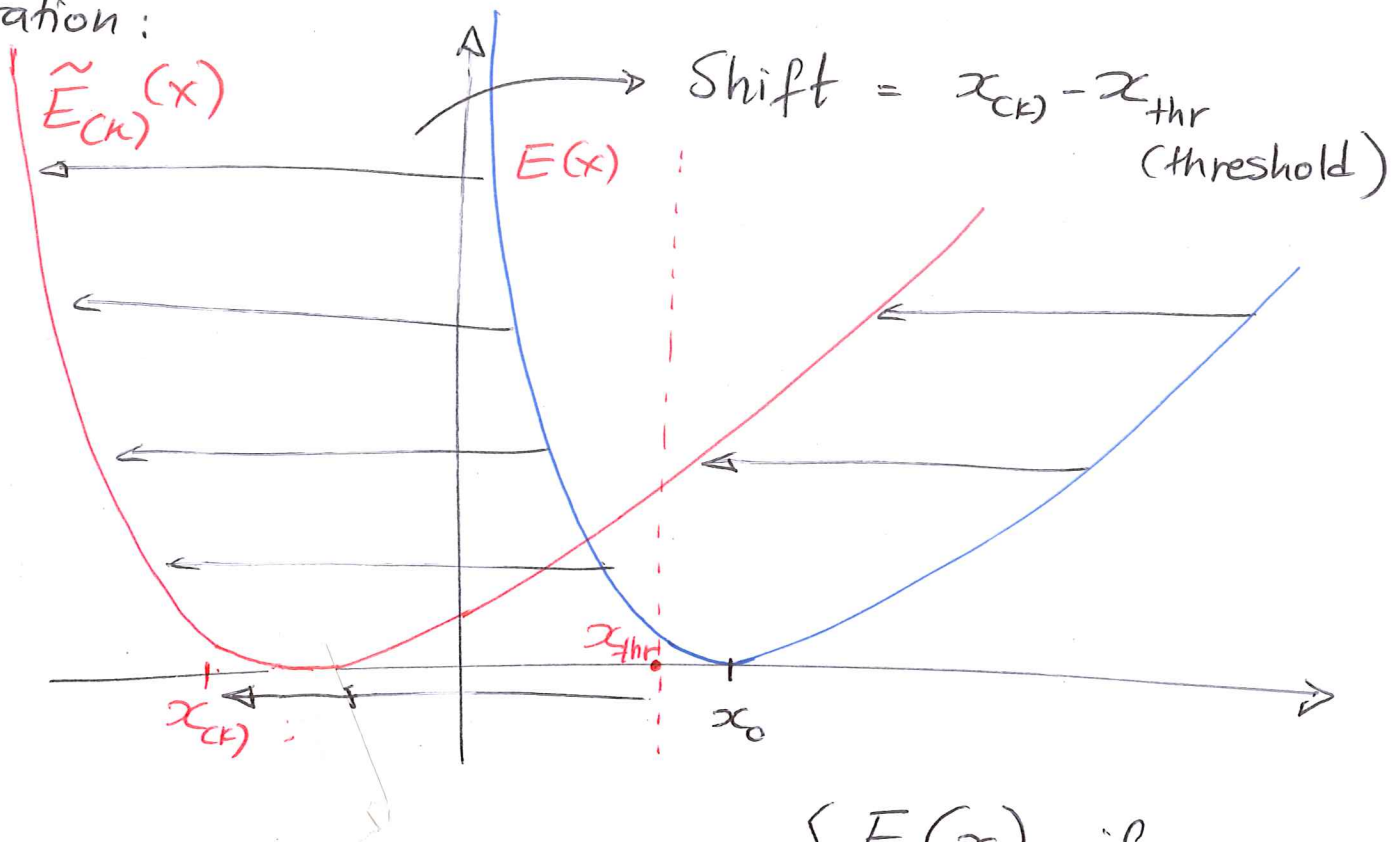


Thus, if we use $\tilde{E}$ instead, we can compute "valid" forces, even at invalid configurations. e.g.

$$\tilde{f}(x_{inv}) = \left.\frac{d\tilde{E}}{dx}\right|_{x_{inv}} = \left.\frac{dE}{dx}\right|_{x_{min}} = f(x_{min}) \quad !$$

Irving et al used this with an explicit time integration (not Backward Euler) so this was practically all that was needed. However, this does not work with eq (1), since $\left.\frac{d^2\tilde{E}}{dx^2}\right|_{x_{inv}} = 0$!

The more effective (and general) remedy is, instead, to momentarily "shift" the entire energy curve to the left, for the purposes of just one Newton iteration:



So, we define $\tilde{E}_{(k)}(x) = \begin{cases} E(x) & \text{if } x_{(k)} \geq x_{thr} \\ E(x - x_{(k)} + x_{thr}) & \text{if } x_{(k)} < x_{thr} \end{cases}$

then, equation (1) becomes:

$$\left(\left. \frac{\partial^2 \tilde{E}}{\partial x^2} \right|_{x_{(k)}} \right) \delta x = - \left. \frac{\partial \tilde{E}}{\partial x} \right|_{x_{(k)}} \Longrightarrow \begin{cases} \left. \frac{\partial^2 E}{\partial x^2} \right|_{x_{thr}} \cdot \delta x = \underbrace{- \left. \frac{\partial E}{\partial x} \right|_{x_{thr}}}_{= f(x_{thr})} & \text{if } x < x_{thr} \\ \text{Eq (1)}, \text{ otherwise.} \end{cases} \quad (2)$$

How does this extend to 3D elasticity?

Remember: the Singular Value Decomposition

$$F = U \Sigma V^T$$

- V^T → Rotation Matrix
- Σ → Diagonal, $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3|$
- U → Rotation Matrix, $U^T U = I$ $\det(U) = 1$

Note (see Irving '04): Most SVD packages produce U, V matrices that satisfy $U^T U = V^T V = I$, but $\det(U) = \pm 1$ $\det(V) = \pm 1$ (and $\Sigma_{ii} \geq 0$)

We can convert such result to the convention above, as:

→ if $\det(U) < 0$, $\det(V) < 0$ ⇒

Replace $U \leftarrow -U$, $V \leftarrow -V$ no other change

→ if only $\det(U) < 0$

Replace $\sigma_3 \leftarrow -\sigma_3$ (negate smallest σ_i)

$\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$

Replace 3rd column of U

$u_3 \leftarrow -u_3$

→ if only $\det(V) < 0$

As before, negate σ_3 & 3rd column of V

We can easily verify that:

(1) All the nonlinear material models we have discussed have an energy density function that is purely a function of Σ (not of U or V) - this is a property of all isotropic materials (see femdefo.org notes).

Corotated $\psi(F) = \mu \|F-R\|_F^2 + \frac{\lambda}{2} tr^2(S-I) = \mu \|\Sigma - I\|_F^2 + \frac{\lambda}{2} tr^2(\Sigma - I)$

StVK $\psi(\Sigma) = \mu \|E_\Sigma\|_F^2 + \frac{\lambda}{2} tr^2(E_\Sigma - I)$

$$E_\Sigma = \Sigma^2 - I$$

Neohookean $\psi = \frac{\mu}{2}(I_1 - d) - \mu \log J + \frac{\lambda}{2} \log^2(J)$

$$J = \det(C)$$

so, it would make sense to "shift" just the Σ factor of the SVD of F!

(2) The "closest" matrix $\tilde{F}$ to F (in terms of $\|\tilde{F}-F\|_F$) whose singular values do not exceed σ_{thr} is

$$\tilde{F} = U \tilde{\Sigma} V^T \quad \text{where} \quad F = U \Sigma V^T$$

$$\text{and} \quad (\tilde{\Sigma})_{ii} = \begin{cases} \Sigma_{ii} & \text{if } \Sigma_{ii} \geq \sigma_{thr} \\ \sigma_{thr} & \text{if } \Sigma_{ii} < \sigma_{thr}. \end{cases}$$

We use this to define a "shifted" energy $\tilde{\Psi}_{(k)}(F)$ for each element in the k-th step of the newton iteration, and define the energy in eq (2) as

$$\tilde{E}_{(k)}(x) = \sum_{\text{element:e}} \tilde{E}^{e}_{(k)}(x)$$

in such a way that

$$\left.\frac{\partial^2 \tilde{E}^{e}}{\partial (F^e)^2}\right|_{F^e_{(k)}} := \left.\frac{\partial^2 E}{\partial F^2}\right|_{F = \tilde{F}^{e}_{(k)}}$$

← with truncated singular values

(↑ we haven't get used this expression)

and

$$\left.\frac{\partial \tilde{E}^{e}}{\partial F^e}\right|_{F^e_{(k)}} := \left.\frac{\partial E}{\partial F}\right|_{F = \tilde{F}^{e}_{(k)}}$$

← This is $P(F)$! (evaluated at $\tilde{F}_{(k)}$)

Force Differentials

In the Backward Euler system, we created a matrix (to be solved by CG) that required the (matrix-free) computation of products like

$$K\left(\underline{x}_{(k)}^{n+1}\right) \cdot \underline{w}$$

Let us introduce the following symbol:

For any function $f(x)$: $\mathbb{R}$ or $\mathbb{R}^n \longrightarrow \mathbb{R}$ or $\mathbb{R}^m$

Let $$\delta f[\underline{x}^*; \underline{h}] := \left.\frac{\partial f}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}^*} \cdot h$$

Often times we use the symbol "δx" instead, here

we call this the "differential of f at x^*, in the direction of h" (or, for an increment of h).

Then: $$K\left(\underline{x}_{(k)}^{n+1}\right) \cdot \underline{w} = -\left.\frac{\partial f_{el}}{\partial \underline{x}}\right|_{\underline{x}_{(k)}^{n+1}} \cdot \underline{w}$$

$$= -\delta f_{el}\left[\underline{x}_{(k)}^{n+1}; w\right]$$

We will dive into the algebra next lecture but, a procedure for computing $\delta f_{el}[x_i; \delta x]$:

Function addForce (in: x, out: f)

Foreach element $e = (i,j,k,l)$

$$D_s = \left[x_i - x_l \,\middle|\, x_j - x_l \,\middle|\, x_k - x_l \right]$$

$$F = D_s D_m^{-1}$$

$$\boxed{P = P(F)}$$ ← Material Specification

$$H = -Vol_e \cdot P \cdot D_m^{-T}$$

$$f_i \mathrel{+}= h_1, \quad f_j \mathrel{+}= h_2, \quad f_k \mathrel{+}= h_3$$

$$f_l \mathrel{-}= (f_i + f_j + f_k)$$

Function addForceDifferential (in: $x, \delta x$; out: δf)

Foreach element $e = (i,j,k,l)$

$$\delta D_s = \left[\delta x_i - \delta x_l \,\middle|\, \delta x_j - \delta x_l \,\middle|\, \delta x_k - \delta x_l \right]$$

$$\delta F = \delta D_s \cdot D_m^{-1}$$

$$\boxed{\delta P = \delta P[F; \delta D_s]}$$

$$\delta H = -Vol_e \, \delta P \cdot D_m^{-T}$$

$$\delta f_{ijk} \mathrel{+}= \delta h_{1|2|3}$$

$$\delta f_l \mathrel{-}= (\delta h_1 + \delta h_2 + \delta h_3)$$