Solution algorithm for nonlinear problems

We typically look for an unknown internal 'force' to balance external loads:

$$F^{int} = F^{ext}$$

$$F^{int} = N(d) \quad : \quad \mathbb{R}^{n_{eq}} \to \mathbb{R}^{n_{eq}}$$

Even for static problems, we introduce a time-like parameter to parameterize the external load. : $N(d(t)) \equiv F^{int}(t) = F^{ext}(t)$

$$R(t) := F^{ext}(t) - F^{int}(t)$$

at the 'time' sub-interval, given d_n $\left(0 = F^{ext}(t_n) - N(d_n) \right)$
determine d_{n+1} such that

$$0 = R_{n+1} = \underset{\substack{\parallel \\ F^{ext}(t_{n+1})}}{F^{ext}_{n+1}} - F^{int}(d_{n+1})$$

or, equivalently,

$\cancel{K \Delta d = R_{n+1}}$

determine $\Delta d = d_{n+1} - d_n$. $\rightarrow$ $d_{n+1} = d_n + \Delta d$

By algorithm $\qquad$ $n \leftarrow n+1$

Consistent tangent $K(d) = \frac{\partial N}{\partial d}$ or

$$K = [K_{PQ}] \quad K_{PQ} = \frac{\partial N_P}{\partial d_Q} \quad 1 \le P, Q \le n_{eq}$$

$$F^{ext}_{n+1} = N(d_{n+1})$$

$$= N(d_n) + \frac{\partial N}{\partial d}(d_n)\, \Delta d + \underbrace{\dots\dots}_{\text{higher-order terms}}$$

$$K(d_n)\, \Delta d \cong F^{ext}_{n+1} - N(d_n)$$

approximation, meaning d_{n+1} will be obtained <u>iteratively</u>.

<u>The Newton-Raphson method.</u>

Perform iteration in each load step to drive the residual to a sufficiently small value.

Stopping criterion $\| R \| \le \varepsilon$ ....

tolerance.

norm of a vector

iteration index i

$d^{(i)}_{n+1} = d_n$ is a reasonable guess for starting the algorithm.

$$i = 0$$

$$d^{(i)} = d_n$$

$$K(d^{(i)})\Delta d^{(i)} = R^{(i)} := F^{ext}_{n+1} - F^{int}(d^{(i)})$$

$$d^{(i+1)} = d^{(i)} + \Delta d^{(i)}$$

No — $\| R^{(i+1)} \| \overset{?}{\leq} \varepsilon \| R^{(0)} \|$

Yes

$$d_{n+1} = d^{(i+1)} \qquad n \leftarrow n+1$$

Remark 1: typically, we also monitor the absolute error and the number of iteration in the stopping condition.

$\| \cdot \|$ is a l_2-norm. but you may use other options.

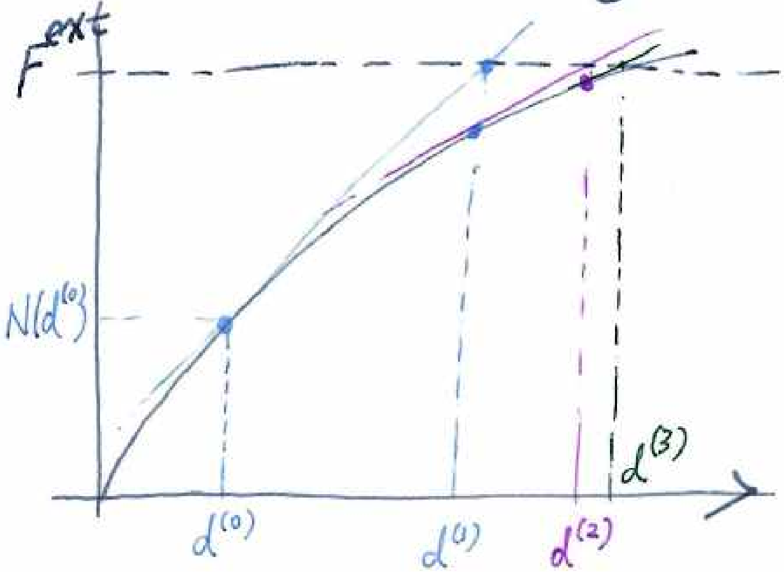
Remark 2: We say the algorithm is N.-R. only when K is <u>consistent</u>.

Remark 3: Most computation cost is due to the ① formation & ② factorization / preconditioning of the K matrix.

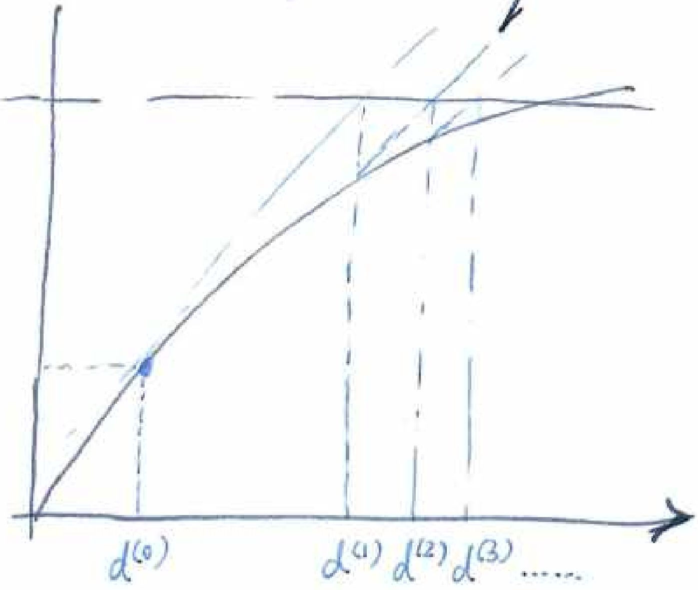
$\hookrightarrow$ avoid updating K and reuse the existing info

$\Rightarrow$ Modified Newton method.

Remark 4: Graphically, in a scalar prob. ($n_{eq}=1$)



consistent N.-R.



Modified N.-R.

Order of convergence: We measure the speed of a sequence, say $\{ d^{(i)} \}$, approaching to its limit. d^* by the <u>convergence order</u>. If

$$\lim_{i \to \infty} \frac{|d^{(i)} - d^*|}{|d^{(i-1)} - d^*|^k} = C \neq 0$$

We say the order of convergence is k.

Note: if $k=1$. $|C|$ has to be strictly less than 1.

Ex: assume $e^{(i)} := |d^{(i)} - d^*|$ satisfies $|e^{(i)}| / |e^{(i-1)}|^k = C$ for all i, $e^{(0)} = 0.9$, $C = 0.5$.

Case 1: $k=1$

Iterate i	Error $\lvert e^{(i)} \rvert = C \lvert e^{(i-1)} \rvert$
1	4.5×10^{-1}
2	1.3×10^{-1}
3	6.3×10^{-2}
4	3.1×10^{-2}
5	1.6×10^{-2}
6	7.8×10^{-3}
7	3.9×10^{-3}

Case 2. $k = 2$

Iterate i	Error $\lvert e^{(i)} \rvert = C \lvert e^{(i-1)} \rvert^2$
1	4.1×10^{-1}
2	8.2×10^{-2}
3	3.4×10^{-3}
4	5.7×10^{-6}
5	1.6×10^{-11}
6	1.3×10^{-22}
7	8.2×10^{-45}

\# numbers of leading zeros doubles in each succeeding iteration

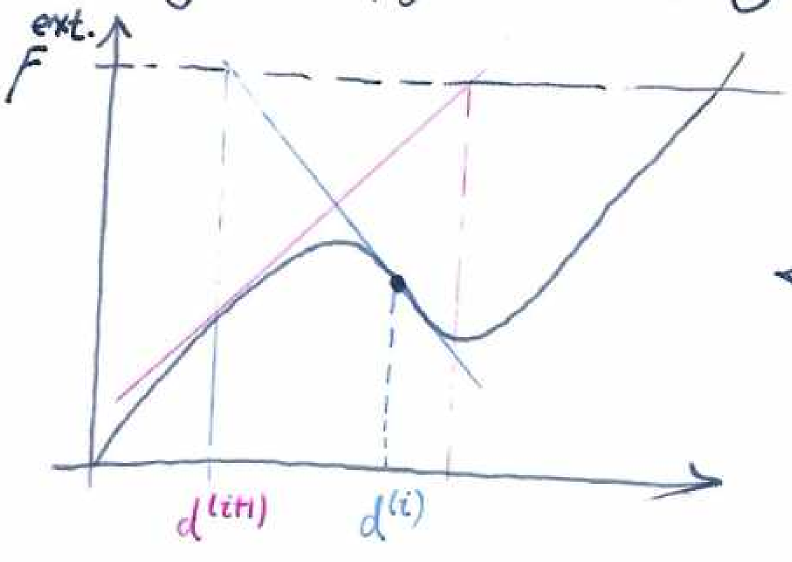
Consistent N.-R. : $k = 2$
modified N.-R. : $k = 1$
} there is a tradeoff in computation cost.

It's hard to give a rigid criterion. One needs to test the two for a specific problem.

Remark 5: In the solver for nonlinear problems, we may tolerate errors, sometimes we purposely ignore certain terms.

It is rather <u>critical</u> to make sure R is assembled <u>correctly</u>.

Remark 6: When we are far away from the true solution, consistent tangent can perform rather poorly.



$\Leftarrow$ one may converge or diverge in this situation.

$$\underbrace{K(d^{(i)})}_{<0} \Delta d = \underbrace{F^{ext} - N(d^{(i)})}_{>0}$$

need better methods or applied external loads slowly.

Soft & Stiff behavior of the solver.

$$N(d) = Kd$$ ← scalar linear problem

$$\tilde{K}\,\Delta d = F^{ext} - K d^{(i)}$$

Some approximation to the true stiffness K.

$$\Rightarrow \quad \underline{\tilde{K} = \alpha K. \qquad \alpha > 0}$$

- $\alpha < 1$: $\tilde{K}$ is a <u>soft</u> approximation
- $\alpha > 1$: $\tilde{K}$ is a <u>stiff</u> approximation

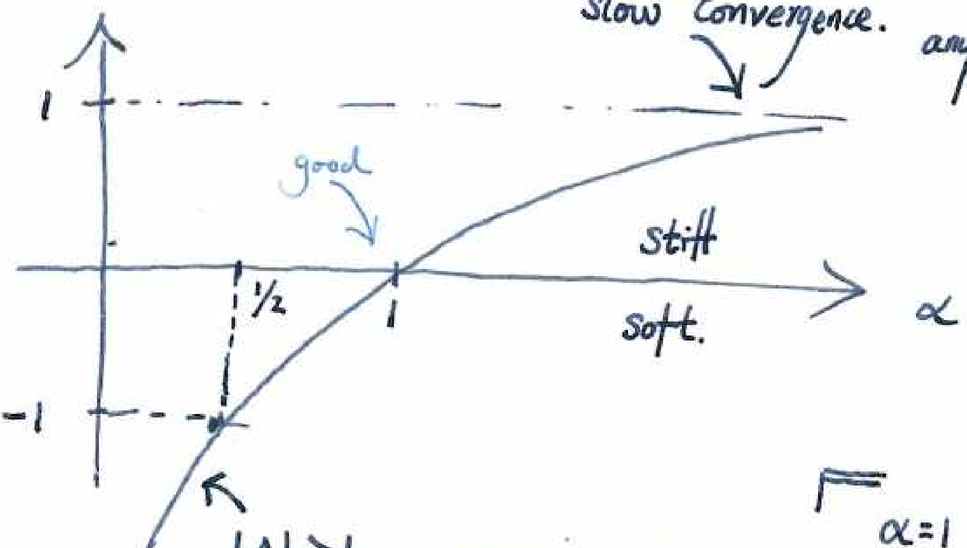
$$\alpha K \left(d^{(i+1)} - d^{(i)} \right) = F^{ext} - K d^{(i)}$$

$$\Rightarrow \quad \alpha K \left(e^{(i+1)} - e^{(i)} \right) = -K e^{(i)} \qquad e^{(i)} = d^{(i)} - d$$

$$\Rightarrow \quad e^{(i+1)} = A e^{(i)} \quad \text{with} \quad A := \frac{\alpha - 1}{\alpha} = 1 - \frac{1}{\alpha}$$

amplification factor

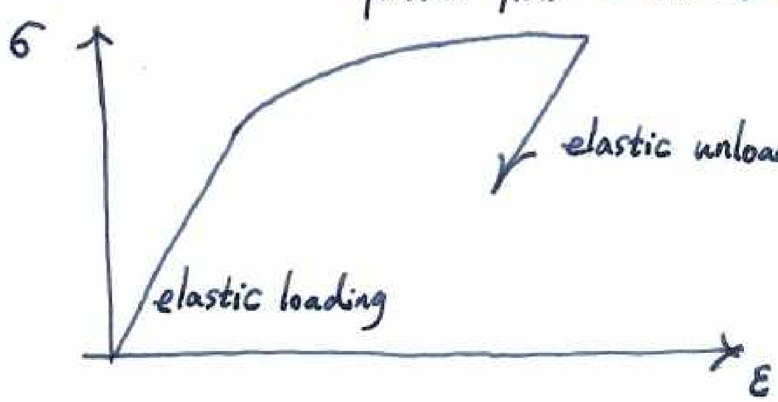
slow convergence.



$|A| > 1$: error grows. divergence

$|A| < 1$ is necessary for convergence.

$\alpha = 1 \quad A = 0$: linear problem converge in 1 iteration



plastic flow: consistent tangent is relatively soft.

elastic unloading: consistent tangent becomes stiffer.

retaining the tangent from plastic stage in unloading will cause divergence.

Similar phenomena observed in contact

<u>Multi-degree-of-freedom problem</u>

$$\hat{K}\,\Delta d^{(i)} = F^{ext} - K d^{(i)}$$

both K & $\hat{K}$ are positive definite.

eigenvalue problem. $(K - \lambda \hat{K})\psi = 0$ gives $\{\lambda, \psi\}_1^{n_{eq}}$

$$\Delta d_\ell^{(i+1)} = F_\ell^{ext} - \lambda_\ell d_\ell^{(i)} \quad \text{for each } \ell = 1, \cdots, n_{eq}.$$

modal subscript

no sum

$$e_\ell^{(i+1)} = (1 - \lambda_\ell)\, e_\ell^{(i)}$$

modal amplification factor. ($\lambda = 1/\alpha$)

therefore, a good approximated matrix $\hat{K}$ is one that has $\lambda_\ell \approx 1$ for all ℓ's.

Soft modal approximation may lead to divergence.