

NonLinear Solution Procedures

Summary

This document contains introduction to nonlinear solution procedures.

1 Solution procedures for nonlinear systems

We illustrate this on problem of nonlinear mechanics. Our starting point is general form of equilibrium equations expressing the balance between internal \mathbf{f}^{int} and external \mathbf{f}^{ext}

$$\mathbf{f}^{int}(\mathbf{r}) = \mathbf{f}^{ext}$$

Suppose we are looking for an equilibrium at the end of load increment $\Delta\mathbf{f}^{ext}$

$$\mathbf{f}^{int}(\mathbf{r} + \Delta\mathbf{r}) = \mathbf{f}^{ext} + \Delta\mathbf{f}^{ext} \quad (1)$$

By the linearization of the nodal force vector around known equilibrium state we can obtain

$$\mathbf{f}^{int}(\mathbf{r}) + \frac{\partial \mathbf{f}^{int}}{\partial \mathbf{r}} \Delta\mathbf{r} + O(\|\Delta\mathbf{r}\|^2) \quad (2)$$

The derivative of internal force vector with respect to nodal displacements is called jacobian matrix and in solid mechanics as tangent stiffness matrix. For the case of material non-linearity

$$\mathbf{f}^{e,int}(\mathbf{r}^e) = \int_{\Omega^e} \mathbf{B}^T \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{r}^e)) d\Omega \quad (3)$$

$$\frac{\partial \mathbf{f}^{e,int}}{\partial \mathbf{d}^e} = \int_{\Omega^e} \mathbf{B}^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{r}^e} d\Omega = \int_{\Omega^e} \mathbf{B}^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \mathbf{B} \mathbf{r}^e d\Omega = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{r}^e d\Omega \quad (4)$$

1.1 Newton-Raphson method

The method is based on splitting of the loading process into series of subsequent incremental loading steps in which the incremental loading vector $\Delta\mathbf{f}$ is applied. We are looking for the equilibrium at the end of loading step 1 using the iterative procedure outlined in 4. The algorithm is graphically outlined in Fig. 1.1 for a system with one unknown and summarized in Table 2.

Based on update strategy for stiffness matrix, one can obtain different variants of the method. When the stiffness matrix \mathbf{K}^i is updated in each iteration, the full Newton-Raphson method is obtained. When stiffness matrix only every n-th iteration, one speaks about modified Newton-Raphson method. Finally, when the stiffness matrix is updated only at the beginning of the loading step, one obtains so called initial stiffness method. For the full Newton-Raphson method a quadratic convergence is obtained.

One can implement two blends of Newton-Raphson algorithm, where the loading can be driven by load control or by displacement control, where the prescribed increments of displacements are applied to selected DOFs.

1.2 Arc-length method

We start by assuming the parametrized loading, in which the total external load vector is expressed as

$$\mathbf{f}^{ext}(\lambda) = +\lambda \mathbf{f}_p^{ext}$$

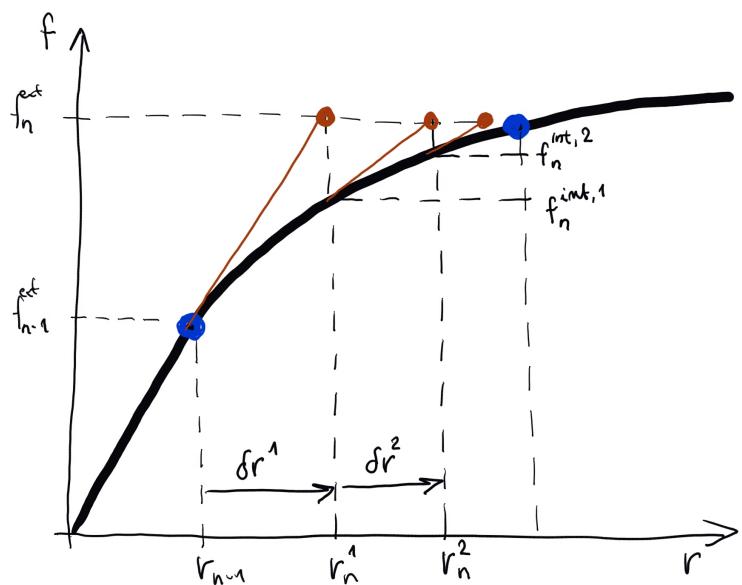


Figure 1: Illustration of Newton-Raphson method

Given
 \mathbf{f}_{n-1}^{ext}
 $\mathbf{f}_n^{ext} = \mathbf{f}_{n-1}^{ext} + \delta \mathbf{f}_n^{ext}$
 $\mathbf{r}_n^0 = \mathbf{r}_{n-1}$

Looking for \mathbf{r}_n , such that $\mathbf{f}^{int}(\mathbf{r}_n) = \mathbf{f}_n^{ext}$
Solve for $i = 1, 2, \dots$
 $\mathbf{K}^i \delta \mathbf{r}^i = \mathbf{f}_n^{ext} - \mathbf{f}_n^{int}(\mathbf{r}_n^{i-1})$
 $\mathbf{r}_n^i = \mathbf{r}_n^{i-1} + \delta \mathbf{r}^i$
Until $\|\mathbf{f}_n^{ext} - \mathbf{f}_n^{int}(\mathbf{r}_n^{i-1})\| \leq \varepsilon$

Table 1: Newton-Raphson method

where \mathbf{f}_p is proportional, reference load vector, and λ is load scaling parameter. The arc-length method is based on idea of controlling the length passed along the loading path. For the differential length of loading path we can write

$$\Delta l = \sqrt{\Delta \mathbf{r}^T \Delta \mathbf{r} + (c^2 \Delta \lambda^2 \mathbf{f}_p^T \mathbf{f}_p)} \quad (5)$$

where c is coefficient of generalized metrics used to define Δl (taking into account different units of displacement and load). For selected increment of loading path length Δl , we are looking for the equilibrium, where the unknowns are nodal displacements \mathbf{r} and the load scaling parameter λ . We have the equilibrium equation and additional scalar equation 5:

$$\mathbf{f}^{int}(\mathbf{r}_n) = \mathbf{f}^{ext}(\lambda_n \mathbf{f}_p) \quad (6)$$

$$\Delta l_n^2 = \Delta \mathbf{r}_n^T \Delta \mathbf{r}_n + c^2 \Delta \lambda^2 \mathbf{f}_p^T \mathbf{f}_p \quad (7)$$

At the end of n-th loading step and i-th iteration the displacement vector can be written as $\mathbf{r}_n^i = \mathbf{r}_n^{i-1} + \delta \mathbf{r}$ and similarly the load scaling parameter as $\lambda_n^i = \lambda_n^{i-1} + \delta \lambda$. Substituting this into equilibrium equation 6 we get

$$\mathbf{f}^{int}(\mathbf{r}_n^{i-1} + \delta \mathbf{r}) = \mathbf{f}^{ext}((\lambda_n^{i-1} + \delta \lambda) \mathbf{f}_p)$$

By linearization of \mathbf{F}^{int} around known state \mathbf{r}_n^{i-1} we get

$$\mathbf{f}_n^{int}(\mathbf{r}_n^{i-1}) + \mathbf{K}_n^{i-1} \delta \mathbf{r} = \mathbf{f}^{ext,i-1} + \delta \lambda \mathbf{f}_p$$

and finally for unknown $\delta \mathbf{r}$

$$\delta \mathbf{r} = \underbrace{(\mathbf{K}_n^{i-1})^{-1} (\mathbf{f}^{ext,i-1} - \mathbf{f}^{int}(\mathbf{r}_n^{i-1}))}_{\delta \mathbf{r}_r} + \underbrace{\delta \lambda (\mathbf{K}_n^{i-1})^{-1} \mathbf{f}_p}_{\delta \mathbf{r}_\lambda} \quad (8)$$

Note that the vectors $\delta \mathbf{r}_r$ and $\delta \mathbf{r}_\lambda$ can be computed and the only unknown remaining is the incremental change of loading parameter $\delta \lambda$, which could be determined from 7

$$\Delta l_n^2 = (\Delta \mathbf{r}_n^{i-1} + \delta \mathbf{r}_r + \delta \lambda \delta \mathbf{r}_\lambda)^T (\Delta \mathbf{r}_n^{i-1} + \delta \mathbf{r}_r + \delta \lambda \delta \mathbf{r}_\lambda) + c^2 (\Delta \lambda_n^{i-1} + \delta \lambda)^2 \mathbf{f}_p^T \mathbf{f}_p \quad (9)$$

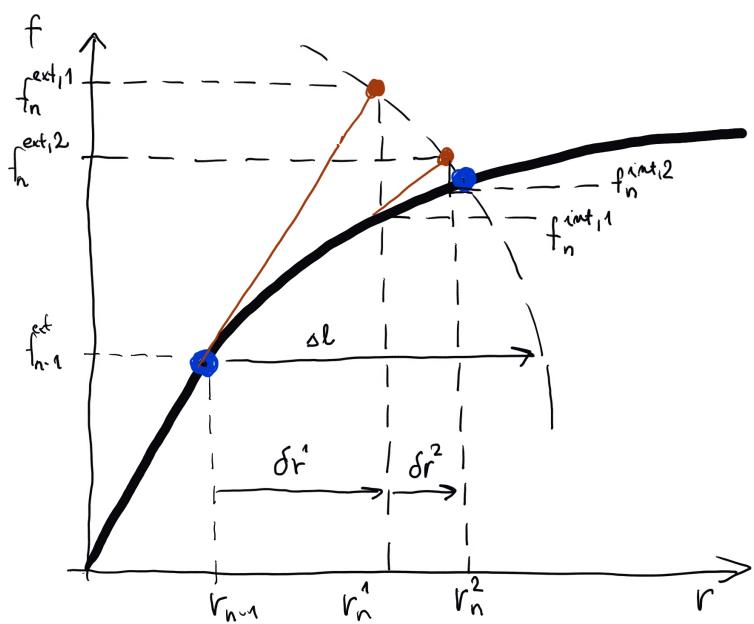


Figure 2: Illustration of Acr-length method

This finally yields a quadratic equation for unknown increment of loading parameter $\delta\lambda$. The algorithm is summarized in Table ??.

Given
$\mathbf{f}_{n-1}^{ext}, \mathbf{f}_p$
$\mathbf{r}_n^0 = \mathbf{r}_{n-1}$
Evaluate
$\delta\mathbf{r}_\lambda = (\mathbf{K}_n)^{-1} \mathbf{f}_p$
$\Delta\lambda^0 = \pm\Delta l / \sqrt{\delta\mathbf{r}_\lambda^T \delta\mathbf{r}_\lambda + c^2 \mathbf{f}_p^T \mathbf{f}_p}$
$\Delta\mathbf{r}_n^0 = (\mathbf{K}_n)^{-1}(\Delta\lambda \mathbf{f}_p) = (\mathbf{K}_n)^{-1}((\lambda_n + \Delta\lambda^0) \mathbf{f}_p - \mathbf{f}_n^{int,0})$
Repeat for $i = 1, 2, \dots$
$\delta\mathbf{r}_\lambda = (\mathbf{K}_n^{i-1})^{-1} \mathbf{f}_p$
$\delta\mathbf{r}_r = (\mathbf{K}_n^{i-1})^{-1}(\lambda_n^{i-1} \mathbf{f}_p - \mathbf{f}^{int}(\mathbf{r}_n^{i-1}))$
Solve quadratic equation 9 for $\delta\lambda$
$\delta\mathbf{r}^i = \delta\mathbf{r}_r + \delta\lambda \delta\mathbf{r}_\lambda$
$\Delta\mathbf{r}_n^i = \Delta\mathbf{r}^{i-1} + \delta\mathbf{r}^i, \mathbf{r}_n^i = \mathbf{r}_n^{i-1} + \delta\mathbf{r}^i$
$\lambda^i = \lambda^{i-1} + \delta\lambda, \Delta\Lambda_n^i = \Delta\Lambda_n^{i-1} + \delta\lambda$
Until convergence reached

Table 2: Newton-Raphson method

2 Non-stationary linear transport model

The weak form of diffusion-type differential equation leads to

$$\mathbf{K}\mathbf{r} + \mathbf{C} \frac{d\mathbf{r}}{dt} = \mathbf{F}, \quad (10)$$

where the matrix \mathbf{K} is a general non-symmetric conductivity matrix, \mathbf{C} is a general capacity matrix and the vector \mathbf{F} contains contributions from external and internal sources. The vector of unknowns, \mathbf{r} , can hold nodal values of temperature, humidity, or concentration fields, for example.

Time discretization is based on a generalized trapezoidal rule. Let us assume that the solution is known at time t and the time increment is Δt . The parameter $\alpha \in \langle 0, 1 \rangle$ defines a type of integration scheme; $\alpha = 0$ results in an explicit (forward) method, $\alpha = 0.5$ refers to the Crank-Nicolson method, and $\alpha = 1$ means an

implicit (backward) method. The approximation of solution vector and its time derivative yield

$$\tau = t + \alpha\Delta t = (t + \Delta t) - (1 - \alpha)\Delta t, \quad (11)$$

$$\mathbf{r}_\tau = (1 - \alpha)\mathbf{r}_t + \alpha\mathbf{r}_{t+\Delta t}, \quad (12)$$

$$\frac{d\mathbf{r}}{dt} = \frac{1}{\Delta t}(\mathbf{r}_{t+\Delta t} - \mathbf{r}_t). \quad (13)$$

$$\mathbf{F}_\tau = (1 - \alpha)\mathbf{F}_t + \alpha\mathbf{F}_{t+\Delta t}, \quad (14)$$

Let us assume that Eq. (10) should be satisfied at time τ . Inserting Eqs. (12)-(14) into Eq. (10) leads to

$$\left[\alpha\mathbf{K} + \frac{1}{\Delta t}\mathbf{C} \right] \mathbf{r}_{t+\Delta t} = \left[(\alpha - 1)\mathbf{K} + \frac{1}{\Delta t}\mathbf{C} \right] \mathbf{r}_t + (1 - \alpha)\mathbf{F}_t + \alpha\mathbf{F}_{t+\Delta t} \quad (15)$$

where the conductivity matrix \mathbf{K} contains also a contribution from convection, since it depends on $\mathbf{r}_{t+\Delta t}$

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \lambda \mathbf{B} d\Omega + \underbrace{\int_{\Gamma_{\bar{c}}} \mathbf{N}^T h \mathbf{N} d\Gamma}_{\text{Convection}} \quad (16)$$

The vectors \mathbf{F}_t or $\mathbf{F}_{t+\Delta t}$ contain all known contributions

$$\mathbf{F}_t = -\underbrace{\int_{\Gamma_{\bar{q}}} \mathbf{N}^T \bar{q}_t d\Gamma}_{\text{Given flow}} + \underbrace{\int_{\Gamma_{\bar{c}}} \mathbf{N}^T h T_{\infty,t} d\Gamma}_{\text{Convection}} + \underbrace{\int_{\Omega} \mathbf{N}^T \bar{Q}_t d\Omega}_{\text{Source}} \quad (17)$$

3 Non-stationary nonlinear transport model

In a nonlinear model, Eq. (10) is modified to

$$\mathbf{K}(\mathbf{r})\mathbf{r} + \mathbf{C}(\mathbf{r}) \frac{d\mathbf{r}}{dt} = \mathbf{F}(\mathbf{r}), \quad (18)$$

Time discretization is the same as in Eqs. (11)-(13) but the assumption in Eq. (17) is not true anymore. Let us assume that Eq. (18) should be satisfied at time $\tau \in \langle t, t + \Delta t \rangle$. By substituting of Eqs. (12)-(13) into Eq. (18) leads to the following equation

$$[(1 - \alpha)\mathbf{r}_t + \alpha\mathbf{r}_{t+\Delta t}] \mathbf{K}_\tau(\mathbf{r}_\tau) + \left[\frac{\mathbf{r}_{t+\Delta t} - \mathbf{r}_t}{\Delta t} \right] \mathbf{C}_\tau(\mathbf{r}_\tau) = \mathbf{F}_\tau(\mathbf{r}_\tau). \quad (19)$$

Eq. (19) is non-linear and the Newton method is used to obtain the solution. First, the Eq. (19) is transformed into a residual form with the residuum vector \mathbf{R}_τ , which should converge to the zero vector

$$\mathbf{R}_\tau = [(1 - \alpha)\mathbf{r}_t + \alpha\mathbf{r}_{t+\Delta t}] \mathbf{K}_\tau(\mathbf{r}_\tau) + \left[\frac{\mathbf{r}_{t+\Delta t} - \mathbf{r}_t}{\Delta t} \right] \mathbf{C}_\tau(\mathbf{r}_\tau) - \mathbf{F}_\tau(\mathbf{r}_\tau) \rightarrow \mathbf{0}. \quad (20)$$

A new residual vector at the next iteration, \mathbf{R}_τ^{i+1} , can be determined from the previous residual vector, \mathbf{R}_τ^i , and its derivative simply by linearization. Since the aim is getting an increment of solution vector, $\Delta \mathbf{r}_\tau^i$, the new residual vector \mathbf{R}_τ^{i+1} is set to zero

$$\mathbf{R}_\tau^{i+1} \approx \mathbf{R}_\tau^i + \frac{\partial \mathbf{R}_\tau^i}{\partial \mathbf{r}_t} \Delta \mathbf{r}_\tau^i = \mathbf{0}, \quad (21)$$

$$\Delta \mathbf{r}_\tau^i = - \left[\frac{\partial \mathbf{R}_\tau^i}{\partial \mathbf{r}_t} \right]^{-1} \mathbf{R}_\tau^i. \quad (22)$$

Deriving Eq. (20) and inserting to Eq. (22) leads to

$$\tilde{\mathbf{K}}_\tau^i = \frac{\partial \mathbf{R}_\tau^i}{\partial \mathbf{r}_t} = -\alpha \mathbf{K}_\tau^i(\mathbf{r}) - \frac{1}{\Delta t} \mathbf{C}_\tau^i(\mathbf{r}), \quad (23)$$

$$\Delta \mathbf{r}_\tau^i = - \left[\tilde{\mathbf{K}}_\tau^i \right]^{-1} \mathbf{R}_\tau^i, \quad (24)$$

which gives the resulting increment of the solution vector $\Delta \mathbf{r}_\tau^i$

$$\Delta \mathbf{r}_\tau^i = - \left[\tilde{\mathbf{K}}_\tau^i \right]^{-1} \left\{ [(1-\alpha)\mathbf{r}_t + \alpha \mathbf{r}_{t+\Delta t}] \mathbf{K}_\tau^i(\mathbf{r}_\tau) + \left[\frac{\mathbf{r}_{t+\Delta t} - \mathbf{r}_t}{\Delta t} \right] \mathbf{C}_\tau^i(\mathbf{r}_\tau) - \mathbf{F}_\tau(\mathbf{r}_\tau) \right\}, \quad (25)$$

and the new total solution vector at time $t + \Delta t$ is obtained in each iteration

$$\mathbf{r}_{t+\Delta t}^{i+1} = \mathbf{r}_{t+\Delta t}^i + \Delta \mathbf{r}_\tau^i. \quad (26)$$

There are two options how to initialize the solution vector at time $t + \Delta t$. While the first case applies linearization with a known derivative, the second case simply starts from the previous solution vector. The second method in Eq. (28) is implemented in OOFEM.

$$\mathbf{r}_{t+\Delta t}^0 = \mathbf{r}_t + \Delta t \frac{\partial \mathbf{r}_t}{\partial t}, \quad (27)$$

$$\mathbf{r}_{t+\Delta t}^0 = \mathbf{r}_t. \quad (28)$$

Note that the matrices $\mathbf{K}(\mathbf{r}_\tau)$, $\mathbf{C}(\mathbf{r}_\tau)$ and the vector $\mathbf{F}(\mathbf{r}_\tau)$ depend on the solution vector \mathbf{r}_τ . For this reason, the matrices are updated in each iteration step (Newton method) or only after several steps (modified Newton method). The residuum \mathbf{R}_τ^i and the vector $\mathbf{F}_\tau(\mathbf{r}_\tau)$ are updated in each iteration, using the most recent capacity and conductivity matrices.

3.1 Heat flux from radiation

Heat flow from a body surrounded by a medium at a temperature T_∞ is governed by the Stefan-Boltzmann Law

$$q(T, T_\infty) = \varepsilon\sigma(T^4 - T_\infty^4) \quad (29)$$

where $\varepsilon \in \langle 0, 1 \rangle$ represents emissivity between the surface and the boundary at temperature T_∞ . $\sigma = 5.67 \cdot 10^{-8} \text{ W/m}^{-2}\text{K}^{-4}$ stands for a Stefan-Boltzmann constant. Transport elements in OOFEM implement Eq. (29) and require non-linear solver.

Alternatively (not implemented), a linearization using Taylor expansion around T_∞ and neglecting higher-order terms results to

$$q(T, T_\infty) \approx q(T = T_\infty) + \frac{\partial q(T, T_\infty)}{\partial T_\infty}(T_\infty - T) = 4\varepsilon\sigma T_\infty^3(T - T_\infty) \quad (30)$$

leading to so-called radiation heat transfer coefficient $\alpha_{rad} = 4\varepsilon\sigma T_\infty^3$. The latter resembles similar coefficient as in convective heat transfer. Other methods for Eq. (29) could be based on Oseen or Newton-Kantorovich linearization. Also, radiative heat transfer coefficient α_{rad} can be expressed as [?, pp.28]

$$q(T, T_\infty) = \varepsilon\sigma \frac{T^4 - T_\infty^4}{T - T_\infty}(T - T_\infty) = \underbrace{\varepsilon\sigma(T^2 + T_\infty^2)(T + T_\infty)}_{\alpha_{rad}}(T - T_\infty) \quad (31)$$