

2 Numerical (iterative) solution of nonlinear equations

The formulation of a nonlinear problem in the FEM leads to a system of nonlinear equations whose unknown values represent the numerical values of the sought functions at the nodes of the computational model. The total number of equations is equal to the total number of degrees of freedom of the model. To solve the system of nonlinear equations, FEM programs use algorithms in the form of programmed solvers, based on basic mathematical methods of solving such problems. Most often they use an iterative approach based on the Newton-Raphson method, which we now remind with a special focus on the equations of structural problems.

Consider a bar of elastic material with Young's modulus E_0 and a cross-sectional area S_0 fixed and loaded according to Fig. 2.1. The bar has only one degree of freedom - the vertical

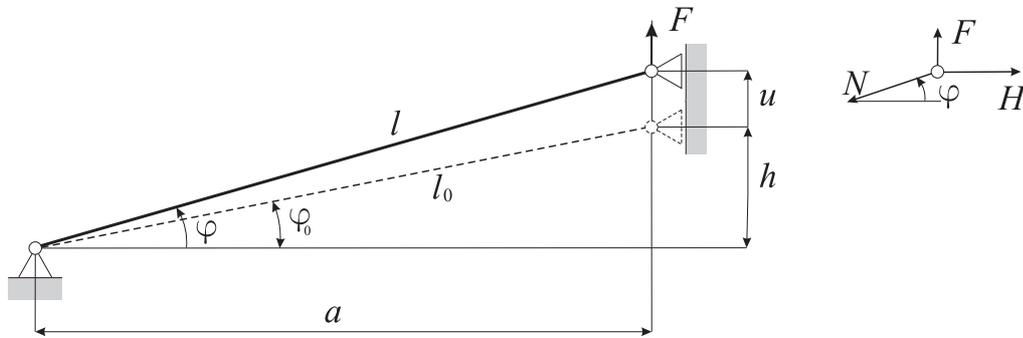


Fig. 2.1 Simple nonlinear problem

displacement u , whose dependence on the load force F can be approximately expressed with simple nonlinear equation [1]

$$\frac{E_0 S_0}{2l_0^3} (u^3 + 3hu^2 + 2h^2u) = F \quad (2.1)$$

With numerical values $E_0 = 200\,000$ MPa, $S_0 = 800$ mm², $l_0 = 2000$ mm, $h = 50$ mm, the equation (2.1) changes to

$$0,01(u^3 + 150u^2 + 5000u) = F \quad (2.2)$$

If we want to find the displacement u for a certain force value F iteratively, it is advantageous to write equation (2.2) in a form that makes it easy to control the iterative process

$$R(u) = 0,01(u^3 + 150u^2 + 5000u) - F = 0 \quad (2.3)$$

where $R(u)$ is called the *non-equilibrium force*, since the equilibrium equation of the bar (2.2) is fulfilled only when this force equals zero. If e.g. for the calculation u we choose force $F = 4000$ N, the non-equilibrium force will be

$$R(u) = 0,01(u^3 + 150u^2 + 5000u) - 4000 \quad (2.4)$$

A graphical representation of this function in Fig. 2.2 shows that the displacement of the bar at the force $F = 4000$ N is around the value of $u^* = 35$ mm.

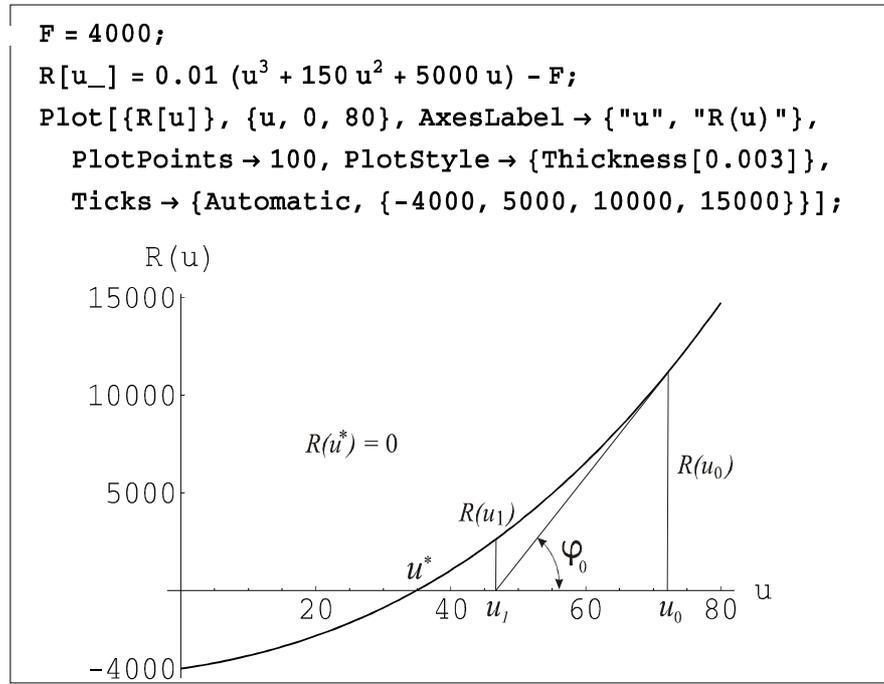


Fig. 2.2

The principle of the Newton-Raphson method can be easily explained on an equation with one unknown. Let us choose the initial value u_0 of the displacement (Fig. 2.2) and express the non-equilibrium force $R(u_0)$. At this point make a tangent to the curve (this can be called linearization of the function around this point), which marks the displacement u_1 on the u -axis as the first approximation to the solution. This displacement is easy to determine because, as you can see from the picture, it applies

$$\operatorname{tg}(\varphi_0) = (dR/du)_{u_0} = \frac{R(u_0)}{u_0 - u_1} \quad (2.5)$$

and from that

$$u_1 = u_0 - \frac{R(u_0)}{(dR/du)_{u_0}} = u_0 - \frac{R(u_0)}{(K_T)_{u_0}} \quad (2.6)$$

An analogous repetition of this procedure with value u_1 , u_2 and so on, is the basis for the necessary iterative algorithm. The derivative

$$K_T = \frac{dR}{du} \quad (2.7)$$

is so-called *tangential stiffness* which, as can be seen, is not constant in this process.

In the Newton-Rapson method, the root (solution) of a one-dimensional equation is thus searched as follows:

1. The gradient of the function $R(u)$, the so-called tangential stiffness, is calculated

$$K_T(u) = \frac{dR}{du} \quad (2.8)$$

2. The initial displacement u_0 is selected

3. In the cycle from $i = 0$ to the selected value of the maximum allowable number of iterations i_{max}

- a) the following displacement value is calculated

$$u_{i+1} = u_i - \frac{R(u_i)}{K_T(u_i)} \quad (2.9)$$

b) the absolute value of the non-equilibrium force $|R_{i+1}| = |R(u_{i+1})|$ is compared with the prescribed tolerance R_{tol} and the decision-making steps are taken

- if $|R_{i+1}| \geq R_{tol}$, go to 3, if not, finish the solution
- if $i > i_{max}$, finish the solution

Example 2.1

For the bar fastened and loaded according to Fig. 2.3 calculate by Newton-Raphson method the vertical displacement of the end point of the bar u_A with the permissible solution error $R_{tol} = 1$ N. Given: $F = 4000$ N, $l_0 = 2000$ mm, $h = 50$ mm, $E_0 = 200\,000$ MPa, $S_0 = 800$ mm².

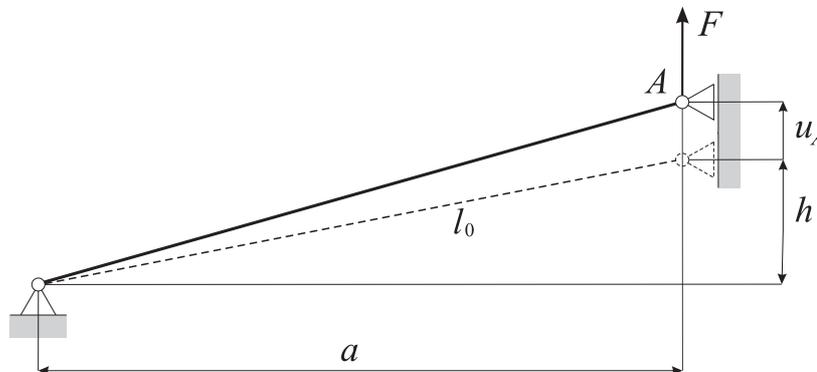


Fig. 2.3

The relationship between the displacement of the bar end point and the force F in the form of a non-equilibrium force is given by Equation (2.4)

$$R(u) = 0,01(u^3 + 150u^2 + 5000u) - 4000$$

with tangential stiffness $K_T(u) = \frac{dR}{du} = 0,01(3u^2 + 300u + 5000)$

The first step of the iterative solution: ($i = 0$):

The initial displacement (selected) $u_0 = 0$

The first displacement value is $u_1 = u_0 - \frac{R(0)}{K_T(0)} = 0 - \frac{-4000}{50} = 80$ mm

The absolute value of the non-equilibrium force at this offset value is (2.4) $|R_1| = 14720$ N

Second step ($i = 1$):

The corrected displacement value is $u_2 = u_1 - \frac{R(80)}{K_T(80)} = 80 - \frac{14720}{482} = 49,46$ mm

The non-equilibrium force is $|R_2| = 3352,5$ N

The third step ($i = 2$)

The corrected displacement value is $u_3 = u_2 - \frac{R(49,46)}{K_T(49,46)} = 49,46 - \frac{3352,53}{271,77} = 37,12$ mm

The non-equilibrium force dropped to $|R_3| = 435,28$ N

To calculate the next values we used the program shown in Fig. 2.4, which also shows the calculated values of the example in the next steps. A graphical illustration of the iterative solution process is shown in Fig. 2.5.

```

NR[F_, uzac_, imax_, tol_] :=
Module[{},
  r[u_] = 0.01 * (u^3 + 150 * u^2 + 5000 * u) - F;
  i = 0;
  u0 = uzac;
  Print[" u0 = ", u0, ", r0 = ", r[u0]];
  u1 = u0;
  While[i < imax && tol < Abs[r[u1]],
    u0 = u1;
    u1 = u0 - r[u0] / r'[u0];
    i = i + 1;
    Print[" i = ", i, ", u(i) = ",
      PaddedForm[u1, {6, 4}], ", r(i) = ", PaddedForm[r[u1], {9, 4}]];
  ];]

NR[4000, 0, 7, 1];

u0 = 0, r0 = -4000

i = 1, u(i) = 80.0000, r(i) = 14720.0000
i = 2, u(i) = 49.4606, r(i) = 3352.5311
i = 3, u(i) = 37.1248, r(i) = 435.2826
i = 4, u(i) = 34.9776, r(i) = 11.9515
i = 5, u(i) = 34.9152, r(i) = 0.0099

```

Fig. 2.4

From the result of the iteration it can be seen that a sufficiently accurate solution of equation (2.2) is the displacement with the value $u_A = 34.92$ mm.

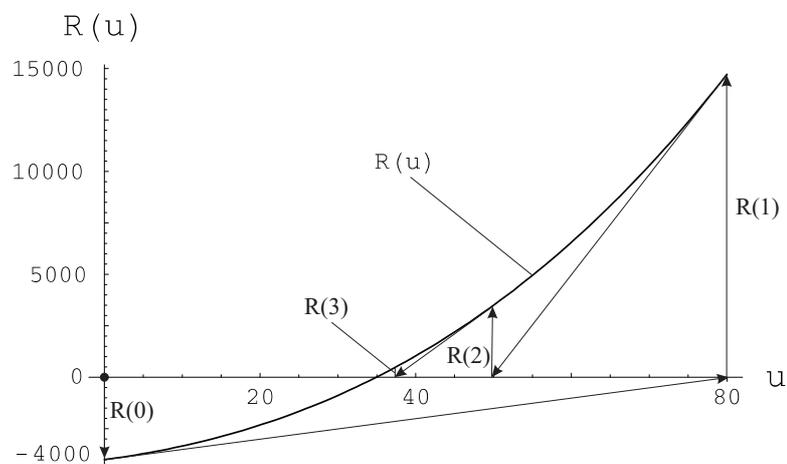


Fig. 2.5

For the general use of the method it is useful to recall its mathematical principle: the function $r(u)$ can be approximated (linearized) around the point u_i by decomposition into a truncated Taylor series

$$r(u) \approx r(u_i) + \frac{df(u_i)}{dx}(u - u_i) \quad (2.10)$$

The solution of the equation can then be obtained by successive iterative solution of the equation

$$\begin{aligned} r(u_i) + \frac{df(u_i)}{dx} \Delta u &= 0 \quad i = 0, 1, 2, \dots \\ u_{i+1} &= u_i + \Delta u \end{aligned} \quad (2.11)$$

to the extent of accuracy specified by the selected coverage condition (the convergence is not guaranteed for any type of function f).

The Newton-Raphson method can also be generalized for a system of nonlinear equations with several unknowns. In such a modification it is normally used in FEM since the application of this method to a nonlinear problem leads to a system of such equations. The generalized displacements of the nodal points arranged in the vector $\mathbf{u} = [u_1, u_2, u_3, \dots, u_n]^T$ are unknown. In matrix notation, the system of equations of body equilibrium has a shape

$$\mathbf{r}(\mathbf{u}) = \mathbf{q}(\mathbf{u}) - \mathbf{f} = \mathbf{0} \quad (2.12)$$

where \mathbf{f} is the known external nodal force vector of the body and $\mathbf{q}(\mathbf{u})$ is the internal nodal force vector of the FEM model calculated from finite element stresses containing nonlinear functions of unknowns \mathbf{u} ; equations (2.12) actually express that the body is in equilibrium when at each node the internal nodal forces are equal to the external nodal forces (in the nodes without external forces, the internal nodal forces themselves are in equilibrium). Using FEM we create a system (2.12) and then based on the selected initial values iteratively adjust the displacements so that the non-equilibrium forces in the vector drop to sufficiently small values according to a certain prescribed convergence condition.

The Newton-Raphson method for the system of nonlinear equations $\mathbf{r}(\mathbf{u}) = \mathbf{0}$, analogous to the one-dimensional case, consists in linearizing the components of the vector function

$$\mathbf{r}(\mathbf{u}) = [f_1(\mathbf{u}), f_2(\mathbf{u}), \dots, f_n(\mathbf{u})]^T$$

using their first differential and replacing the nonlinear system of equations with a linear system to correct the arguments $\Delta \mathbf{u}$. By solving this linear system we get a new iteration $\mathbf{u} + \Delta \mathbf{u}$ of approximate solution. By repeating this procedure we find the approximate value of the solution \mathbf{u} with the required tolerance of either for \mathbf{u} or \mathbf{r} .

By linearizing the system $\mathbf{r}(\mathbf{u}) = \mathbf{0}$ around the initial approximation \mathbf{u}_0 we get

$$\mathbf{r}_L(\mathbf{u}) = \mathbf{r}(\mathbf{u}_0) + \frac{\partial \mathbf{r}(\mathbf{u}_0)}{\partial \mathbf{u}}(\mathbf{u} - \mathbf{u}_0)$$

We are looking for the first approximation \mathbf{u}_1 so that $\mathbf{r}_L(\mathbf{u}_1) = \mathbf{0}$. Then it applies

$$\frac{\partial \mathbf{r}}{\partial \mathbf{u}}(\mathbf{u}_0)(\mathbf{u}_1 - \mathbf{u}_0) = -\mathbf{r}(\mathbf{u}_0) \quad (2.13)$$

After selecting the initial displacement values \mathbf{u}_0 , (2.13) represents a linear system of equations with an unknown vector $\Delta \mathbf{u}_0 = \mathbf{u}_1 - \mathbf{u}_0$. If the matrix of this system is regular then its solution is

$$\Delta \mathbf{u}_0 = -[\mathbf{K}_T(\mathbf{u}_0)]^{-1} \mathbf{r}(\mathbf{u}_0)$$

with the so-called tangential body stiffness matrix (Jacobi matrix)

$$\mathbf{K}_T = \frac{\partial \mathbf{r}}{\partial \mathbf{u}} = \begin{bmatrix} \partial R_1 / \partial u_1 & \partial R_1 / \partial u_2 & \dots & \partial R_1 / \partial u_n \\ \partial R_2 / \partial u_1 & \partial R_2 / \partial u_2 & \dots & \partial R_2 / \partial u_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial R_n / \partial u_1 & \partial R_n / \partial u_2 & \dots & \partial R_n / \partial u_n \end{bmatrix} \quad (2.14)$$

The desired approximation is then $\mathbf{u}_1 = \mathbf{u}_0 + \Delta \mathbf{u}_0$. By repeating this procedure we get the relation for the sequence of approximations

$$\mathbf{u}_{i+1} = \mathbf{u}_i - [\mathbf{K}_T(\mathbf{u}_i)]^{-1} \mathbf{r}(\mathbf{u}_i) \quad i = 0, 1, 2, \dots, i_{max} \quad (2.15)$$

If this sequence converges to the solution of the system of equations (2.12), the calculation is terminated by the condition for $\|\mathbf{u}_{i+1} - \mathbf{u}_i\|$ or $\|\mathbf{r}_{i+1} - \mathbf{r}_i\|$.

In cases where the external nodal forces in (2.12) are not dependent on displacements, the tangential stiffness matrix can be expressed from the internal nodal forces and the following applies:

$$\mathbf{K}_T = \frac{\partial \mathbf{r}}{\partial \mathbf{u}} = \frac{\partial \mathbf{q}}{\partial \mathbf{u}} \quad (2.16)$$

The procedure, as shown in (2.15), requires inversion of the tangential body stiffness matrix at each iteration step, which can be time consuming for large problems. On the other hand, in this case, the quadratic convergence of the iterative process is guaranteed and we are talking about the *full Newton-Raphson method*. In FEM programs it is usually possible to choose the so-called *modified Newton-Raphson method*, where the inversion of the tangential stiffness matrix

is performed only once - at the beginning of the iteration process and does not change in the next iteration steps. Thus, the same matrix is used in all iteration steps. The iterative process will be accelerated, but the number of iterative steps will increase as the rate of convergence decreases.

Equation (2.15) can be adjusted to

$$\Delta \mathbf{u}_{i+1} = \mathbf{u}_{i+1} - \mathbf{u}_i = - [\mathbf{K}_T(\mathbf{u}_i)]^{-1} (\mathbf{q}_i - \mathbf{f}) \quad (2.17)$$

and further multiplied from the left by $\mathbf{K}_T(\mathbf{u}_i)$ to the form in which it usually occurs in program manuals

$$\mathbf{K}_T(\mathbf{u}_i) \Delta \mathbf{u}_{i+1} = \mathbf{f} - \mathbf{q}_i \quad (2.18)$$

Suppose that the iteration procedure begins on an unloaded body. Then the node displacements are zero ($\mathbf{u}_0 = \mathbf{0} \rightarrow \mathbf{K}_T = \mathbf{K}_{lin}$) and since the stresses are zero ($\mathbf{q} = \mathbf{0}$) the initial iteration step is performed with a linear system of equations

$$\mathbf{K}_{lin} \mathbf{u}_1 = \mathbf{f}$$

From the calculated linear solution \mathbf{u}_1 the deformations and stresses in the elements are determined, from them \mathbf{q}_1 and the vector of non-equilibrium nodal forces of the first iteration step $\Delta \mathbf{r}_1 = \mathbf{f} - \mathbf{q}_1$ are determined. The tangential stiffness matrix is also created for the next iteration step. According to (2.17), the displacement corrections $\Delta \mathbf{u}_{i+1}$ are then calculated and with the new displacements $\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta \mathbf{u}_{i+1}$ the iteration process continued until the convergence conditions are met. In the case of a linear problem, convergence is achieved in the first iteration step.

Example 2.2

The application of the FEM to solve a simple geometrically non-linear truss system with a single free nodal point with three degrees of freedom led to a system of three nodal equations of equilibrium

$$R_1(u_1, u_2, u_3) = (0,01u_1^2 + 50)u_1 + 1,5u_1u_2^2 - 1,5u_1u_3^2 - 600 = 0$$

$$R_2(u_1, u_2, u_3) = 1,5u_1^2u_2 + (0,01u_2^2 + 50)u_2 + 1,5u_2u_3^2 - 800 = 0$$

$$R_3(u_1, u_2, u_3) = -1,5u_1^2u_3 + 1,5u_2^2u_3 + (0,01u_3^2 + 50)u_3 - 500 = 0$$

in matrix notation

$$\mathbf{r}(\mathbf{u}) = \mathbf{q}(\mathbf{u}) - \mathbf{f} = \mathbf{0}$$

where the vector \mathbf{f} contains the orthogonal components of the external force acting at that node.

Determine the nodal displacement components $\mathbf{u} = [u_1 \ u_2 \ u_3]^T$ by the Newton-Raphson method.

The Mathematica solution is shown in Fig. 2.6, where a set of equations is given in part a), the program for solving the problem is given in part b), the program start command with the necessary input parameters is in part c) and the last part d) contains an iterative sequence of solutions with the resulting non-equilibrium force vector satisfying the convergence condition.

```

R1[{u1_, u2_, u3_}] = (0.01 u1^2 + 50) u1 + 1.5 u2^2 u1 - 1.5 u3^2 u1 - 600;
R2[{u1_, u2_, u3_}] = 1.5 u1^2 u2 + (0.01 u2^2 + 50) u2 + 1.5 u3^2 u2 - 800;
R3[{u1_, u2_, u3_}] = -1.5 u1^2 u3 + 1.5 u2^2 u3 + (0.01 u3^2 + 50) u3 - 500;
R̄[{u1_, u2_, u3_}] = {R1[{u1, u2, u3}], R2[{u1, u2, u3}], R3[{u1, u2, u3}]}];

NewtonRaphson[tol_, max_] :=
Module[{norma = 1, i = 0},
  u0 = {0., 0., 0.};
  Print["u0 = ", u0];
  K_T[{u1_, u2_, u3_}] = Transpose[{∂u1 R̄[{u1, u2, u3}], ∂u2 R̄[{u1, u2, u3}], ∂u3 R̄[{u1, u2, u3}]}];
  While[And[i < max, norma > tol],
    u1 = u0 - (Inverse[K_T[u0]]) . R̄[u0];
    Print["u"_{i+1}, " = ", u1];
    norma = √((u1 - u0) . (u1 - u0));
    If[norma < tol, Print["u"_{i+1}, " = ", MatrixForm[R̄[u1]]]];
    u0 = u1;
    i = i + 1;];

NewtonRaphson[10^-3, 30];

u0 = {0., 0., 0.}
u1 = {12., 16., 10.}
u2 = {8.09783, 11.0354, 6.77938}
u3 = {5.70581, 8.60238, 4.85192}
u4 = {4.40778, 8.32121, 3.8068}
u5 = {3.8515, 8.92314, 3.32305}
u6 = {3.73247, 9.16435, 3.21805}
u7 = {3.72762, 9.17659, 3.21387}
u8 = {3.72761, 9.17661, 3.21386}
r8 = { -1.52482 × 10^-9
      -1.14767 × 10^-9
      -1.4295 × 10^-9 }

```

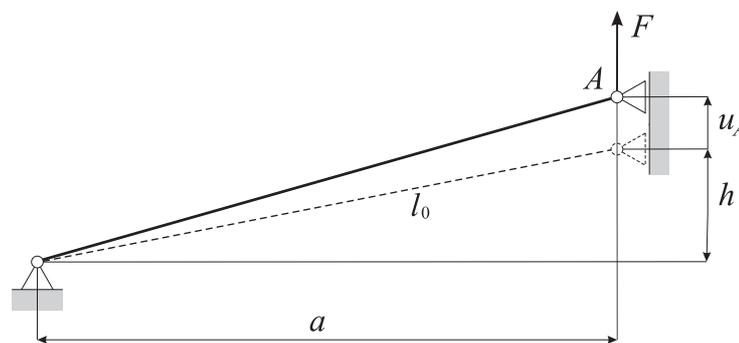
Fig. 2.6

From the results it can be seen that iteration begins with zero displacement components ($\mathbf{u}_0 = \mathbf{0}$), the vector \mathbf{u}_1 contains a linear solution and finally the vector \mathbf{u}_8 gives the values of the rectangular displacement components in the node (in mm) that satisfy the convergence condition. The vector \mathbf{r}_8 contains the values of the node non-equilibrium force components at such displacement values, i. j. values that meet the convergence condition.

In principle, the FEM programs use the Newton-Raphson method as well also ANSYS program, which will be used in this work for solving simple exemplary nonlinear problems.

Example 3.1

For the bar fixed and loaded as shown in the figure, use ANSYS to calculate the vertical displacement of the bar end point u_A when given: $F = 4000 \text{ N}$, $l_0 = 2000 \text{ mm}$, $h = 50 \text{ mm}$, $E_0 = 200000 \text{ MPa}$, $S_0 = 800 \text{ mm}^2$. Compare the result with an analytical-iterative solution (example 2.1).



Program ANSYS performed the calculation with the following commands:

1. Utility Menu, File, Change Jobname... /FILNAM = EXAMPLE3.1;
2. Main Menu, Preprocessor
3. Preprocessor, Element Type, Add/Edit/Delete, Add... Link 2D Spar 1, OK, Close;
4. Preprocessor, Real Constants, Add/Edit/Delete, Add... (Typ 1 Link1), OK, Area = 800, OK, Close;
5. Preprocessor, Material Props, Material Models... (Material Model Number 1), Structural, Linear, Elastic, Isotropic, EX = 200000, PRXY = 0.3, OK, Material, Exit;
6. Preprocessor, Modeling, Create, Nodes, In Active CS... X = 0, Y = 0, Apply,
X = sqrt(2000**2-50**2), Y = 50, OK;
7. Preprocessor, Modeling, Create, Elements, Auto Numbered, Thru Nodes, P: Node 1, P: Node 2, OK;
8. Main Menu, Solution, Define Loads, Apply, Structural, Displacement, On Nodes, P: Node 1, OK,
Mark UX, UY, Apply, P: Node 2, OK, Mark UX, OK;
9. Solution, Define Loads, Apply, Structural, Force/Moment, On Nodes↑ P: Node 2, OK, FY, VALUE = 4000, OK;

10. Solution, Analysis Type, Sol'n Controls... Basic, Large Displacement Static, OK;

11. Solution, Solve, Current LS

12. Main Menu, Solution, Finish

13. Main Menu, General Postproc, Results Summary

SET	TIME/FREQ	LOAD STEP	SUBSTEP	CUMULATIVE
1	1.0000	1	1	4

14. Main Menu, General Postproc, List Results, Nodal Solution... DOF Solution, Displacement vector sum, OK

PRINT U NODAL SOLUTION PER NODE

THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN GLOBAL COORDINATES

NODE	UX	UY	UZ	USUM
1	0.0000	0.0000	0.0000	0.0000
2	0.0000	34.996	0.0000	34.996

20. Ansys Toolbar, Quit, Save Geom+Loads, OK;

The result $u_A = 34.996$ mm is in good agreement with the result in Example 2.1.

References

[1] Benča, Š.: Solution of nonlinear strength problems using FEM. Slovak Technical University in Bratislava. Nakladateľstvo STU, Bratislava, 2009. (in Slovak)