Chapter 3
Finite Elements

Structures are constantly in motion, constantly changing their shape in accordance with the equilibrium conditions, and these tiny corrections are governed by influence functions. As the market woman knows, balance means equal work, $\delta W_e^{left} = \delta W_e^{right}$, means the work of the weights on both sides of the scale is the same. Install a shear hinge in a frame and spread the hinge! You will find that the work of the shear force and the load are the same—the market woman could have told you beforehand.

The structure in the architect's presentation may look pretty, but it does not "live". This requires finite elements. They turn the sketch into a living whole, which can react, deform, and find its balance.

Statics is kinematics and finite elements means life, means motion.

Sure, the FE-structure still more resembles a puppet on a string than a real structure, but even this restricted model has enough life in it, to give an engineer a sufficient idea of the equilibrium position.

Once the model is constructed, the engineer displaces the nodes one by one, $u_i = 1$ (all else are fixed, $u_j = 0$), remembers which forces are necessary for this maneuver, these are the *shape forces* p_i, and combines these loads p_i so, that the FE-load $p_h = \sum_i u_i p_i$ is **shake-equivalent** to the original load p

$$\delta W_e(p, \varphi_i) = \delta W_e(p_h, \varphi_i) \qquad \text{for all } \varphi_i \,. \tag{3.1}$$

This is the system $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$. It is the logic of the market woman.

What we have presented here in anticipation will certainly become clearer in the following. At the beginning, however, we want to introduce finite elements quite classically via the principle of minimum potential energy.

F. Hartmann and P. Jahn, *Statics and Influence Functions*,
Springer Series in Solid and Structural Mechanics 13,
https://doi.org/10.1007/978-3-030-55889-5_3

3.1 The Minimum

According to the *principle of minimum potential energy* the deflection of the rope in Fig. 3.1

$$-H\,w''(x) = p(x) \qquad w(0) = w(l) \qquad H = \text{ horizontal force in the rope}\,, \tag{3.2}$$

is the function which minimizes the potential energy of the rope

$$\Pi(w) = \frac{1}{2}\int_0^l \frac{V^2}{H}\,dx - \int_0^l p(x)\,w(x)\,dx \qquad (V = H\,w') \tag{3.3}$$

among all functions, which satisfy the boundary conditions, $w(0) = w(l) = 0$. The set of all these functions we call $\mathcal{V}$ (as in Variety).

Since the set $\mathcal{V}$ is too large to find $w(x)$ by a manual search, we restrict the search to a *finite dimensional* subspace $\mathcal{V}_h \subset \mathcal{V}$ and declare the function w_h, which minimizes the potential energy in this subspace, the best approximation. This is the idea of the **Ritz method**.

We start by subdividing the rope into small elements. An element is a piece of rope on which two linear functions are defined, the so-called element displacements, and by combining the element displacements of neighboring elements, we construct hat functions which represent ***unit displacements of the nodes***, see Fig. 3.1.

The unit displacements of the four internal nodes form the FE-solution

$$w_h(x) = w_1\,\varphi_1(x) + w_2\,\varphi_2(x) + w_3\,\varphi_3(x) + w_4\,\varphi_4(x)\,, \tag{3.4}$$

and we choose the nodal displacements w_i so that the FE-solution minimizes the potential energy

$$\Pi(w_h) = \frac{1}{2}\int_0^l H\,(w_h')^2\,dx - \int_0^l p\,w_h\,dx \tag{3.5}$$

on $\mathcal{V}_h \subset \mathcal{V}$, the span of the four hat functions $\varphi_i(x)$.

The function (3.4) wins the competition, if

$$\frac{\partial \Pi}{\partial w_i} = \frac{\partial}{\partial w_i}\{\frac{1}{2}\,\boldsymbol{w}^T\,\boldsymbol{K}\,\boldsymbol{w} - \boldsymbol{f}^T\,\boldsymbol{w}\} = 0 \qquad i = 1, 2, 3, 4\,, \tag{3.6}$$

if the vector $\boldsymbol{w}$ of the nodal displacements solves the system $\boldsymbol{K}\,\boldsymbol{w} = \boldsymbol{f}$, or

$$\frac{H}{l_e}\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.7}$$

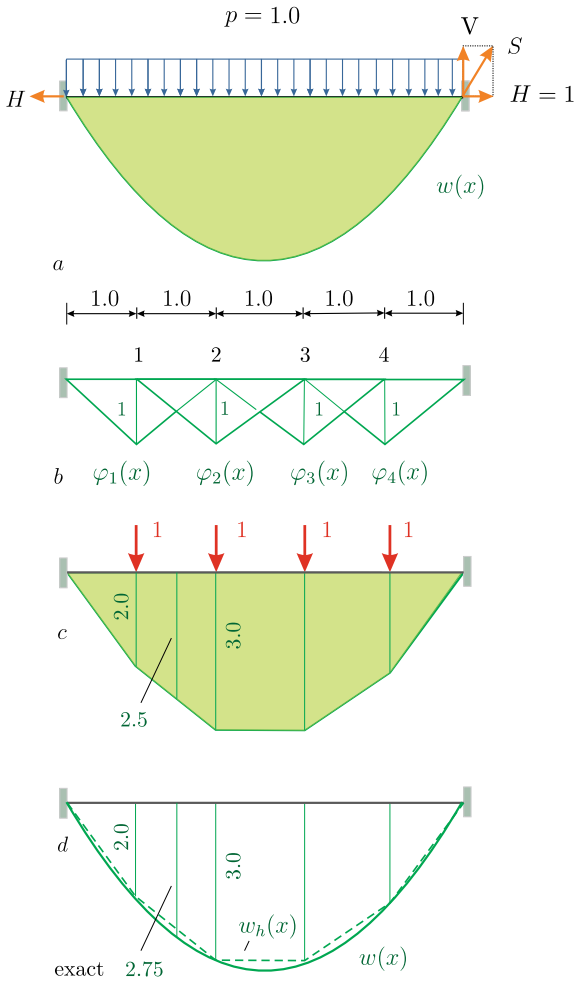

Fig. 3.1 FE-analysis of a rope, **a** system and load, **b** hat functions, **c** FE-solution $w_h(x)$, **d** comparison between $w(x)$ and $w_h(x)$

The elements k_{ij} of the stiffness matrix $\boldsymbol{K}$ are the **strain energy products** between the shape functions

$$k_{ij} = a(\varphi_i, \varphi_j) = \int_0^l H\, \varphi_i'(x)\, \varphi_j'(x)\, dx = \int_0^l \frac{V_i\, V_j}{H}\, dx\,, \tag{3.8}$$

and the **equivalent nodal forces** on the right side are the integrals

$$f_i = \int_0^l p(x)\, \varphi_i(x)\, dx\,. \tag{3.9}$$

If we let $H = 1$ and $l_e = 1$, the system (3.7) has the solution

$$w_1 = w_4 = 2 \qquad w_2 = w_3 = 3\,, \tag{3.10}$$

and so the function

$$w_h(x) = 2 \cdot \varphi_1(x) + 3 \cdot \varphi_2(x) + 3 \cdot \varphi_3(x) + 2 \cdot \varphi_4(x) \tag{3.11}$$

is on $\mathcal{V}_h$ the best approximation.

3.2 Why the Nodal Values of the Rope Are Exact

When we compare the FE-solution with the exact solution

$$w(x) = \frac{1}{2} \cdot (5\,x - x^2)\,, \tag{3.12}$$

we note that the FE-solution agrees with the exact solution at the nodes, $w_i = w(x_i)$. The reason is the following: The FE-program computes the deflection of the rope with the influence function

$$w_h(x) = \int_0^l G_h(y, x)\; p(y)\, dy\,, \tag{3.13}$$

only that it substitutes for the exact kernel $G(y, x)$ an approximate kernel, $G_h(y, x)$, the **best fit** it finds in the trial space $\mathcal{V}_h$. With regard to the nodes the best fit are the original Green's functions since they lie in $\mathcal{V}_h$. This is why the nodal values are exact.

If the Green's function $G(y, x)$ of a point x lies in $\mathcal{V}_h$, the FE-solution is exact at this point, $w(x) = w_h(x)$.

The influence function for the deflection $w(x_1)$ at the first node is the triangle $G(y, x_1)$ in Fig. 3.2b. Since the four shape functions can model this triangle exactly, $G_h(y, x_1) = G(y, x_1)$, the FE-solution agrees with the exact solution at the node x_1

$$w_h(x_1) = \int_0^l G(y, x_1)\; p(y)\, dy = w(x_1)\,. \tag{3.14}$$

This is true in any load case!

We do a test with $p(x) = \sin(\pi x/5)$. The equivalent nodal forces $f_i = (p, \varphi_i)$ are

$$\boldsymbol{f} = \{0.569,\ 0.920,\ 0.920,\ 0.569\}^T\,, \tag{3.15}$$

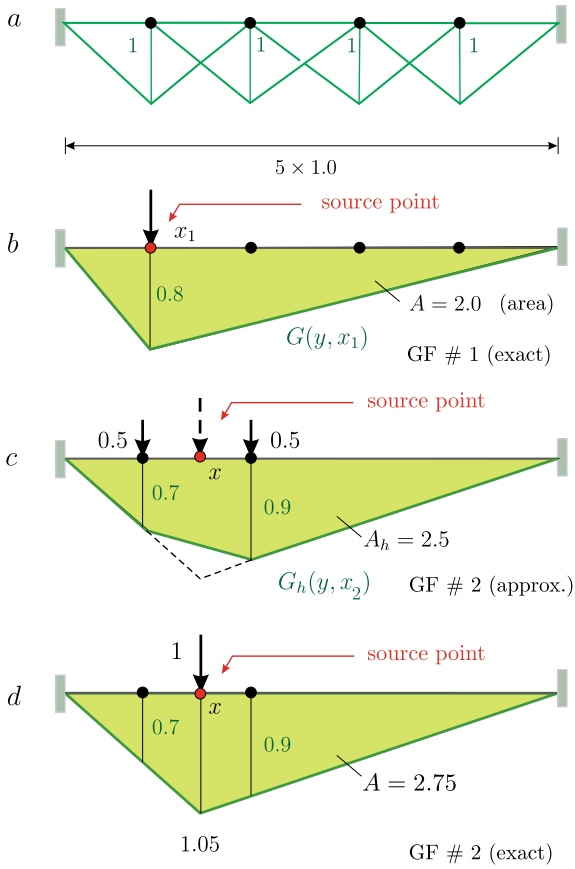

Fig. 3.2 FE-model of a rope, **a** shape functions, **b** influence function (GF) for $w(x_1)$, and **c** for the deflection $w(x)$ at an intermediate point, **d** the exact influence function for $w(x)$

and the solution $\boldsymbol{w} = \{1.489, 2.409, 2.409, 1.489\}^T$ of the system $\boldsymbol{K}\,\boldsymbol{w} = \boldsymbol{f}$ are the nodal values of the solution $w(x) = 25/\pi^2 \cdot \sin(\pi x/5)$.

But if a source point, say the point $x = 1.5$, is not a node, see Fig. 3.2c, the Green's function does not lie in $\mathcal{V}_h$. In this situation the program substitutes for the unit point load at x two half-sized point loads $P = 0.5$ at the neighboring nodes and it calculates with this approximation $G_h(y, x)$

$$w_h(x) = \int_0^l G_h(y, x)\, p(y)\, dy = 2.5 \neq 2.75 = w(x)\,, \tag{3.16}$$

but the result is then of course only an approximate value.

"But doesn't an FE-program calculate the nodal values by solving the system $\boldsymbol{K}\,\boldsymbol{w} = \boldsymbol{f}$ and the values in between by interpolating between the nodes?" Correct, but they are as large ***as if*** they had been calculated with the approximate influence functions. This is the key point.

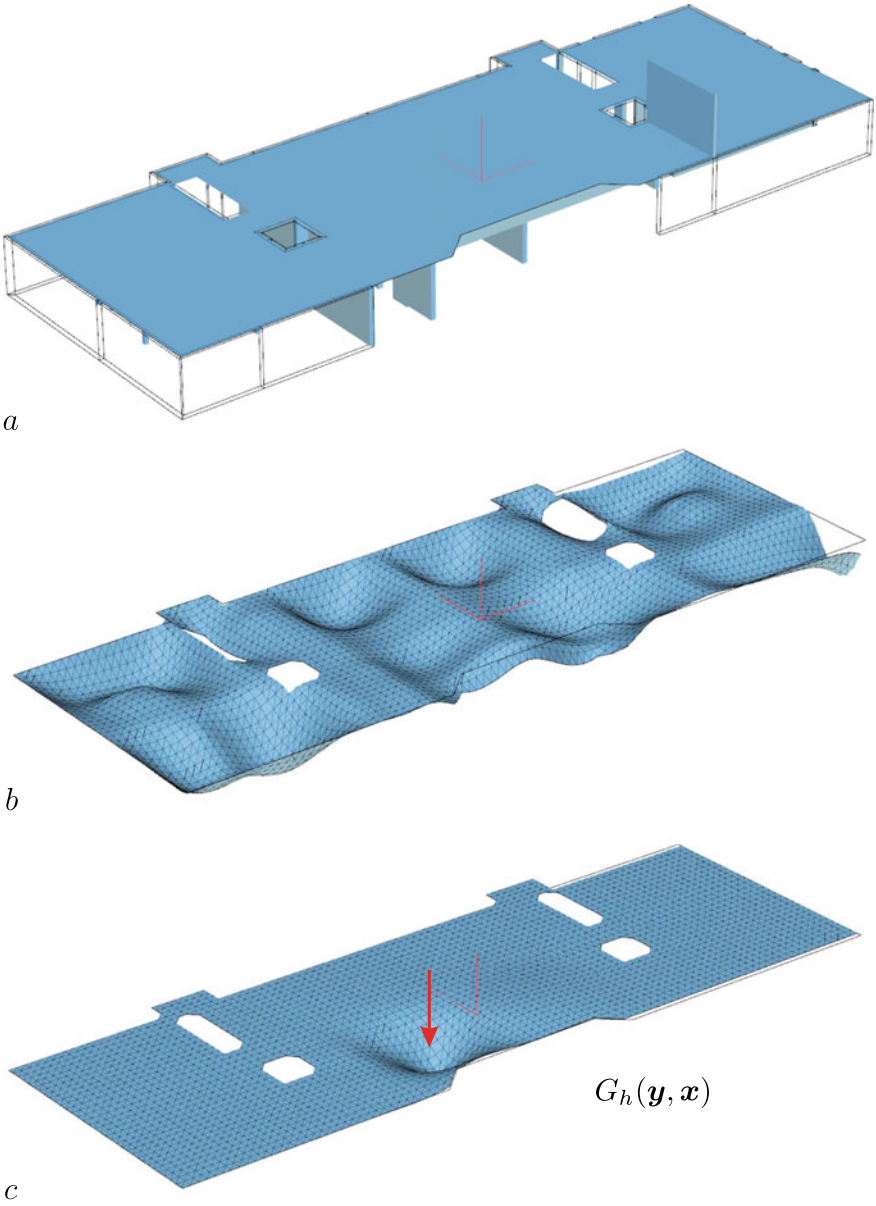

Fig. 3.3 Plate, **a** system, **b** deflection under gravity load, **c** influence function for the deflection w at a node $\boldsymbol{x}$

In 2-D and 3-D this is true as well. The nodal values of the plate in Fig. 3.3 are the result of the superposition of the nodal Green's functions with the load

$$w_h(\boldsymbol{x}_i) = \int_\Omega G_h(\boldsymbol{y}, \boldsymbol{x}_i)\, p(\boldsymbol{y})\, d\Omega_{\boldsymbol{y}}\,. \tag{3.17}$$

Of course, the nodal values w_i in the output are the solution of the system $\boldsymbol{K}\boldsymbol{w} = \boldsymbol{f}$, but they are just as large *as if* the FE-program had integrated over the whole plate, the results are the same[1]

$$w_h(\boldsymbol{x}_i) = w_i = \sum_j k_{ij}^{(-1)}\, f_j = \int_\Omega G_h(\boldsymbol{y}, \boldsymbol{x}_i)\, p(\boldsymbol{y})\, d\Omega_{\boldsymbol{y}}\,. \tag{3.18}$$

This is the secret, little-known law behind finite elements.

The quality of the influence functions determines the quality of the results.

3.3 Adding the Local Solution

When we calculate the deflection of the rope with an FE-program, we will not see a polygon on the screen, but a well curved parabola, i.e. the exact curve. How does the FE-program do this? It proceeds exactly as we said:

- It splits the rope into small elements.
- It reduces the load to the nodes and calculates the f_i.
- It solves the system $\boldsymbol{K}\boldsymbol{w} = \boldsymbol{f}$.

If it stopped now, we would see a rope polygon on the screen.

But there is one more step to follow. The program calculates in each element the so-called ***local solution*** w_{loc}, this is the deflection if the element is fixed at both ends, and it adds these shapes to the FE-polygon. This is how the curve in Fig. 3.4 was generated.

It is the same technique as in the **slope deflection method**, which first reduces all loads to the nodes, finds the equilibrium position, and then adds the local solutions.

The finite elements do it the same way, since the system of equations actually is

$$\boldsymbol{K}\boldsymbol{w} = \boldsymbol{f} + \boldsymbol{d}\,. \tag{3.19}$$

The f_i are the single forces, which act directly at the nodes, and the d_i are the equivalent nodal forces of the distributed load in between the nodes. In the FE-literature $f_i \equiv f_i + d_i$ usually stands for both parts.

[1] Skeptical readers may integrate the right side of (3.18) by parts.

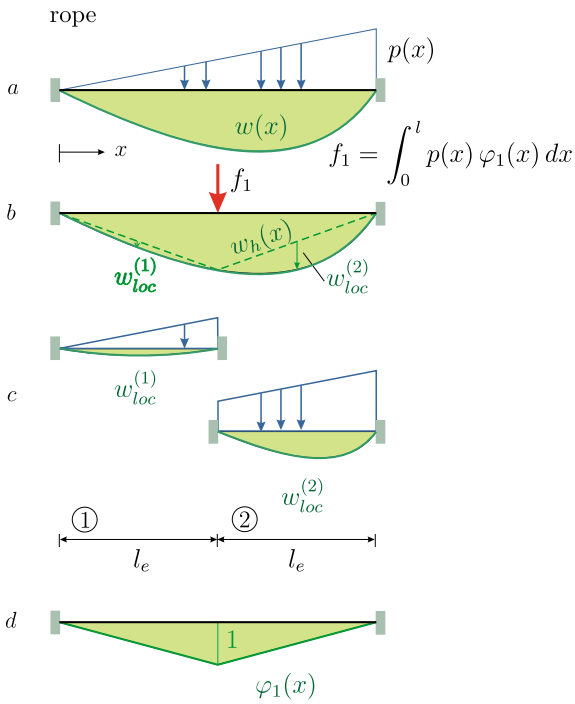

Fig. 3.4 Rope analysis with two elements, **a** load and deflection, **b** FE-solution + local solutions, **c** local solutions, **d** unit deflection of the central node

This way finite elements succeed, despite their **limited kinematics**, i.e. the use of

- piecewise linear functions for the longitudinal displacement
- cubic polynomials for the deflection

in producing the exact deformations, see Fig. 3.5.

Remark 3.1 To be precise this holds only true if EA and EI are constant, because only then are the element shape functions φ_i^e homogeneous solutions of the differential equations. In all other cases the nodal values are not exact.

The equivalence ***finite elements = slope deflection method*** also means: It makes no sense to further divide the individual elements of a frame into short elements. There is no gain in accuracy.

Remark 3.2 The finite element technique is often explained with 1-D problems. We do that too. For the finite elements to remain finite elements, we must agree that all these demonstrations refer to the time ***before*** the local solution is added to the FE-solution.

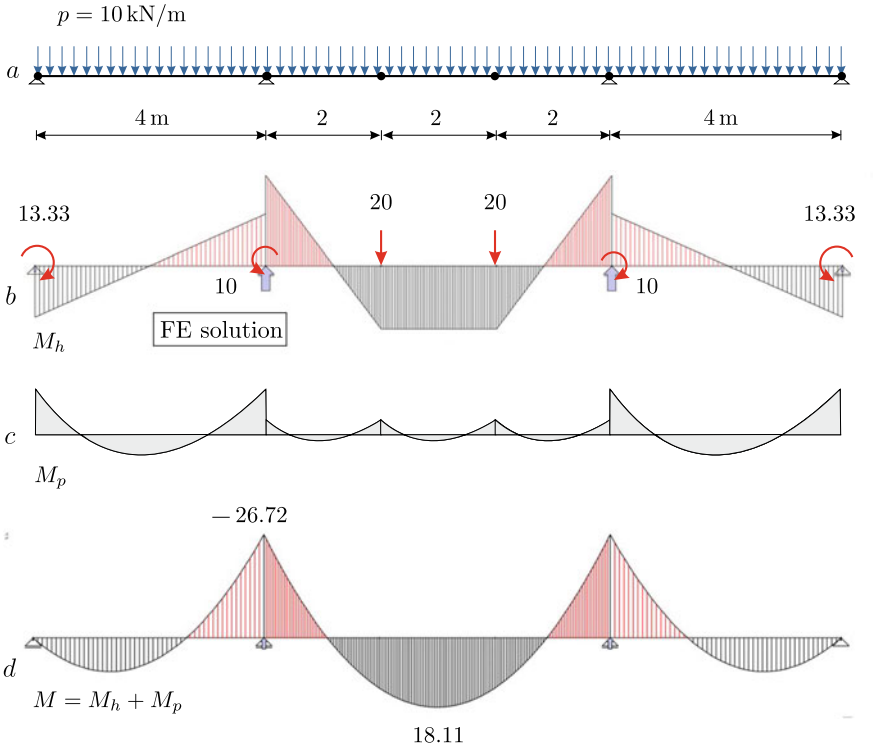

Fig. 3.5 Slope deflection method and finite elements are the same, **a** system, **b** first, the distributed load is reduced to the nodes and equilibrium at the nodes is established, and **c** the local solutions of the single clamped beams **d** are added

3.4 Projection

We arrived at $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$ via the principle of minimum potential energy, however the projection of the exact solution onto the trial space $\mathcal{V}_h$, the ***Galerkin method***, leads to the same equation.

The projection of a vector $\boldsymbol{x} = \{x_1, x_2, x_3\}^T$ onto the $x_1 - x_2$ plane is its **shadow** $\boldsymbol{x}' = c_1\boldsymbol{e}_1 + c_2\boldsymbol{e}_2$, see Fig. 3.6, which means the error is orthogonal to $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$

$$(\boldsymbol{x} - \boldsymbol{x}')^T\boldsymbol{e}_i = 0 \qquad i = 1, 2 \qquad \Rightarrow \qquad c_1 = x_1\,, c_2 = x_2\,, \tag{3.20}$$

and so the shadow is the vector in the plane with the shortest distance

$$|\boldsymbol{e}| = |\boldsymbol{x} - \boldsymbol{x}'| = \text{minimum} \tag{3.21}$$

to the tip of $\boldsymbol{x}$.

This idea is adopted by the *Galerkin method* for which the best approximation is the function w_h in $\mathcal{V}_h$, whose error $w - w_h$ is orthogonal to all φ_i,

$$a(w - w_h, \varphi_i) = 0 \qquad i = 1, 2, \ldots, n\,. \tag{3.22}$$

Because of Green's first identity

$$\mathscr{G}(w, \varphi_i) = \int_0^l p\,\varphi_i\,dx - a(w, \varphi_i) = 0 \quad \Rightarrow \quad f_i = \int_0^l p\,\varphi_i\,dx = a(w, \varphi_i) \tag{3.23}$$

this is equivalent to

$$a(w_h, \varphi_i) = f_i \qquad i = 1, 2, \ldots, n \tag{3.24}$$

or $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$. This implies, see (9.50), that w_h has the shortest distance in the energy metric from the exact solution w

$$a(w - w_h, w - w_h) = \text{minimum}\,. \tag{3.25}$$

Looking at Fig. 3.6 we realize that an additional projection will not lead any further. The shadow of the shadow $\boldsymbol{x}'$ is $\boldsymbol{x}'$. ***A projection stops after the first step***. This is the reason why it is not possible to improve the FE-solution on the same mesh, since the difference $p - p_h$ is orthogonal to the shape functions

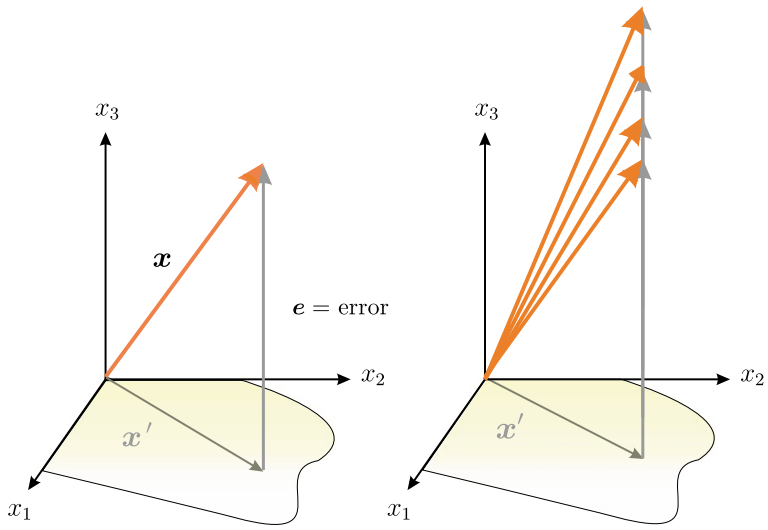

Fig. 3.6 The error vector $\boldsymbol{e}$ is orthogonal to the projection plane and all vectors directly above the vector $\boldsymbol{x}$ have the same shadow

$$f_i = \int_0^l (p - p_h)\, \varphi_i \, dx = 0\,. \tag{3.26}$$

Equation (3.22) is the ***Galerkin-orthogonality***. Because of $\delta W_i = \delta W_e$ it can also be "turned inside out", can be expressed as orthogonality in the virtual external work

$$\text{(inside)} \qquad a(w - w_h, \varphi_i) = \int_0^l (p - p_h)\, \varphi_i \, dx = 0 \qquad \text{(outside)}\,. \tag{3.27}$$

3.5 Equivalent Nodal Forces

The f_i on the right side of the system $\boldsymbol{K u} = \boldsymbol{f}$ are not forces in the true sense, but they are equivalent nodal forces, [Nm], are **energy quanta**[2]

$$f_i = \int_0^l p(x)\, \varphi_i(x)\, dx = [\text{N/m}] \cdot [\text{m}] \cdot [\text{m}] = [\text{Nm}]\,, \tag{3.28}$$

like the entries k_{ij}, which are **energy packets**, for example in a bar,

$$k_{ij} = \int_0^l E A\, \varphi_i' \, \varphi_j' \, dx = [\text{N}] \cdot [\,] \cdot [\,] \cdot [\text{m}] = [\text{Nm}]\,, \tag{3.29}$$

or a beam

$$k_{ij} = \int_0^l E I\, \varphi_i'' \, \varphi_j'' \, dx = [\text{Nm}^2] \cdot [1/\text{m}] \cdot [1/\text{m}] \cdot [\text{m}] = [\text{Nm}]\,. \tag{3.30}$$

Each derivative means a multiplication with $[\text{m}]^{-1}$.

$$\varphi_i \ [\text{m}] \qquad \varphi_i' = \frac{d\,\varphi_i}{dx} = [\ \ ] \qquad \varphi_i'' = \frac{d\,\varphi_i'}{dx} = \frac{1}{[\text{m}]}\,. \tag{3.31}$$

The nodal displacements u_i are (internally) *dimensionless* weights

$$u_h = \textstyle\sum_i u_i\, \varphi_i(x) = [\ \ ] \cdot [\text{m}] = [\text{m}] \tag{3.32}$$

In the output they have of course the dimension of a length, as the engineer wants to see it.

[2] For an alternative interpretation in beam analysis, see Sect. 9.21, page 449.

Calculi

The equivalent nodal forces f_i serve bookkeeping purposes—like the calculi on an *abacus*. The program moves from node to node, displaces the node by one meter in horizontal and vertical direction and makes a note on how much work is done by the load. These notes are the f_i.

A nodal force f_i of 10 Nm means that the loads in the neighborhood of the node contribute a work of 10 Nm on acting through $\boldsymbol{\varphi}_i$.

All what an FE-program does in the end is, it distributes **substitute loads** across the structure in such a way that these loads contribute the same amount of work as the real load, $f_i^h = f_i$.

The system $\boldsymbol{K}\,\boldsymbol{u} = \boldsymbol{f}$ is exactly this match $\boldsymbol{f}_h = \boldsymbol{f}$, since the left side $\boldsymbol{K}\,\boldsymbol{u}$ is the vector $\boldsymbol{f}_h$ of the equivalent nodal forces of the FE-solution.

In the notation of the next section the single f_i^h is the work done by the substitute FE-load

$$\boldsymbol{p}_h = \sum_j u_j\, \boldsymbol{p}_j \tag{3.33}$$

on acting through the nodal displacement $\boldsymbol{\varphi}_i$

$$f_i^h = \sum_j u_j\, \delta W_e(\boldsymbol{p}_j, \boldsymbol{\varphi}_i)\,. \tag{3.34}$$

3.6 Fixed End Forces

Imagine a beam element which is clamped at both ends, and which carries a distributed load p. In such a one-element beam the element shape functions $\varphi_i^e(x)$ are the nodal shape functions, and so the equivalent nodal forces are

$$f_i^e = \int_0^l p(x)\,\varphi_i^e(x)\,dx \qquad \textit{actio}\,. \tag{3.35}$$

Since the support reactions have the opposite direction

$$f_i^e \times (-1) \qquad \textit{reactio}\,, \tag{3.36}$$

we conclude that the element shape functions of a bar

$$\begin{aligned} \varphi_1^e(x) &= \frac{1-x}{l_e} \qquad & \varphi_1^e(0) &= 1\,, \qquad & \varphi_1^e(l_e) &= 0\,, \\ \varphi_2^e(x) &= \frac{x}{l_e} & \varphi_2^e(0) &= 0\,, & \varphi_2^e(l_e) &= 1 \end{aligned} \tag{3.37}$$

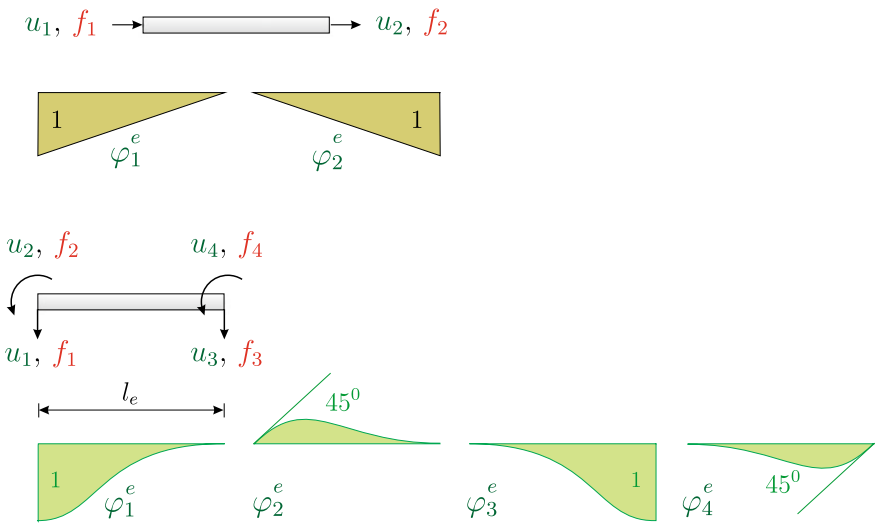

Fig. 3.7 Unit displacements of bar and a beam element

and of a beam, see Fig. 3.7,

$$\varphi_1^e(x) = 1 - \frac{3x^2}{l_e^2} + \frac{2x^3}{l_e^3} \qquad \varphi_3^e(x) = \frac{3x^2}{l_e^2} - \frac{2x^3}{l_e^3}$$
$$\varphi_2^e(x) = -x + \frac{2x^2}{l_e} - \frac{x^3}{l_e^2} \qquad \varphi_4^e(x) = \frac{x^2}{l_e} - \frac{x^3}{l_e^2} \tag{3.38}$$

are also the influence functions for the (negative) support reactions as in *actio*

$$f_i^e = \int_0^{l_e} \underset{\rightarrow}{p(x)}\, \varphi_i^e(x)\, dx \qquad i = 1, 2 \qquad \text{bar} \tag{3.39}$$

$$f_i^e = \int_0^{l_e} \underset{\downarrow}{p(x)}\, \varphi_i^e(x)\, dx \qquad i = 1, 2, 3, 4 \qquad \text{beam} \tag{3.40}$$

and *reactio*, $f_i^e \times (-1)$.

To be precise, this requires the unit displacements φ_i^e to be homogeneous solutions of the governing equations, which means EA and EI must be constant.

In a single frame element, the unit displacements of the nodes are the influence functions for the fixed end forces $\times(-1)$.

3.7 Shape Forces and the FE-load

To push a node of a plate by 1 m in horizontal or vertical direction—while all other nodes are kept fixed—requires a specific set of forces, see Fig. 3.8. We call these forces in analogy to the notion of shape functions, the ***shape forces*** $\boldsymbol{p}_i = \{p_x^{(i)}, p_y^{(i)}\}^T$, which belong to the displacement $u_i = 1$. These are active and restraining forces. The active forces push the node into the direction of the u_i and the restraining forces make that the motion stops at the neighboring nodes. The forces are always balanced, they are equilibrium forces, see Fig. 3.9.

The sum of these *shape-forces*—weighted with the nodal displacements u_i—is the FE-load

$$\boldsymbol{p}_h = \sum_i u_i \, \boldsymbol{p}_i = \sum_i u_i \begin{bmatrix} p_x^{(i)} \\ p_y^{(i)} \end{bmatrix}, \tag{3.41}$$

which produces the shape $\boldsymbol{u}$.

While in 2-D and 3-D problems some effort is necessary to detail the FE-load $\boldsymbol{p}_h$, see Fig. 3.10, in frame analysis (if EA and EI are constant) the FE-load is identical with the nodal forces f_i, which means in 1-D the f_i [Nm] are real forces/moments if you divide by 1 m (not the moments), while in 2-D and 3-D they are "mere" calculi, forces "as if".

Remark 3.3 Fig. 3.10 illustrates the flexibility (for not to say "ambiguity") of structural analysis with regard to finite elements. What holds the plate fixed at the node, is

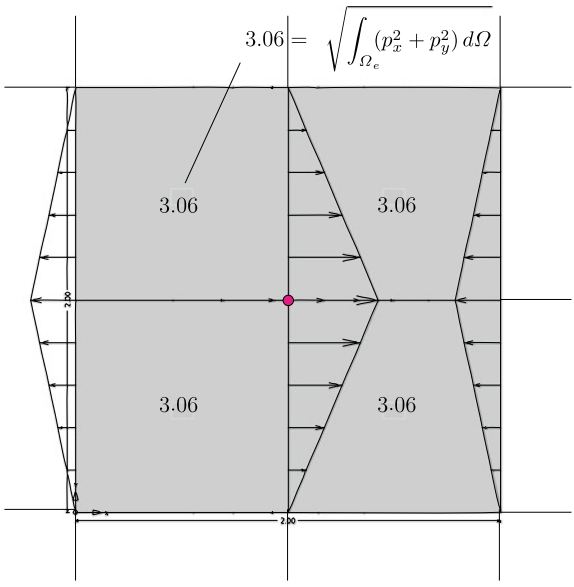

Fig. 3.8 Section of an FE-mesh: The forces, which displace the node by one unit length to the right and simultaneously stop the motion at the neighboring nodes are the *shape forces*. The distributed surface forces, p_x and p_y, are only given as integral values. The other part of the mesh (not shown) is load free, is smooth as a pond

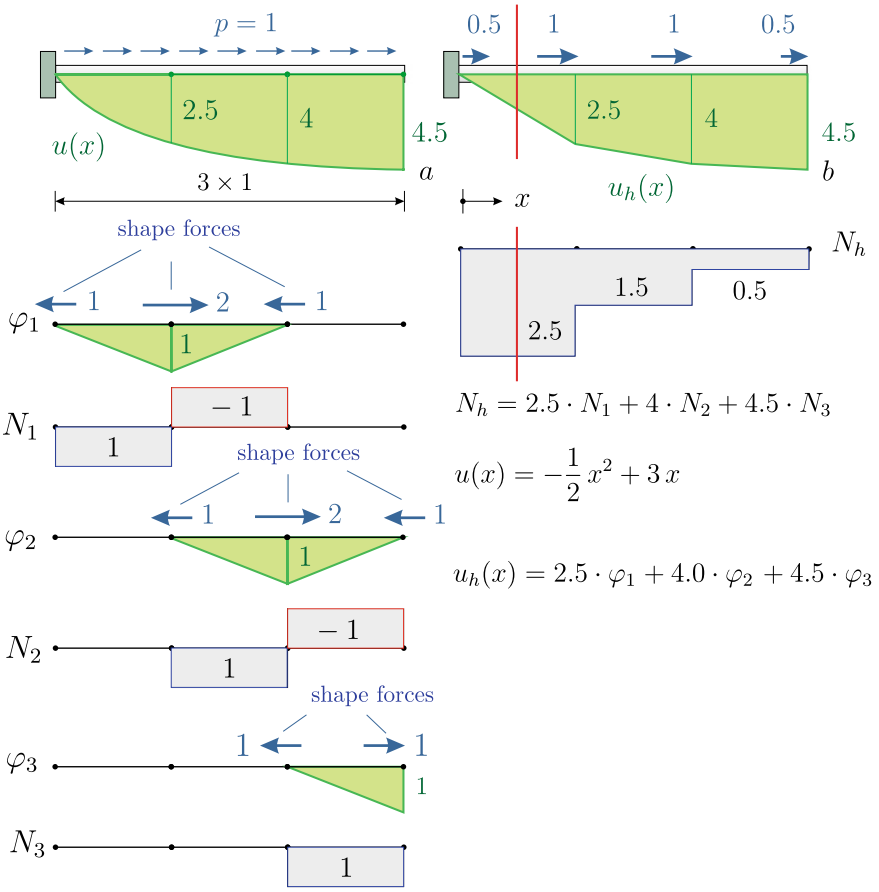

Fig. 3.9 Linear FE-solution of a stretched bar ($EA = 1$) carrying a constant load. Each shape function is plotted together with its set of *shape forces*

not a point force, but the combined action of domain loads and line loads, while in the printout you will only find an equivalent nodal force f_i. This force is the "equivalent sum" so to speak of these distributed forces, and the engineer has no problem (rightly so) to forward this force to the next floor as a real force.

The Role of the u_i

By a proper choice of the nodal displacements u_i the FE-load $\boldsymbol{p}_h$, see (3.41), is made to match the real load in the sense of the ***principle of virtual displacements***, which means the equivalent nodal forces f_i^h agree with the f_i of the applied load

$$f_i^h = \int_\Omega \boldsymbol{p}_h^T \cdot \boldsymbol{\varphi}_i \, d\Omega = \int_\Omega \boldsymbol{p}^T \cdot \boldsymbol{\varphi}_i \, d\Omega = f_i \,. \tag{3.42}$$

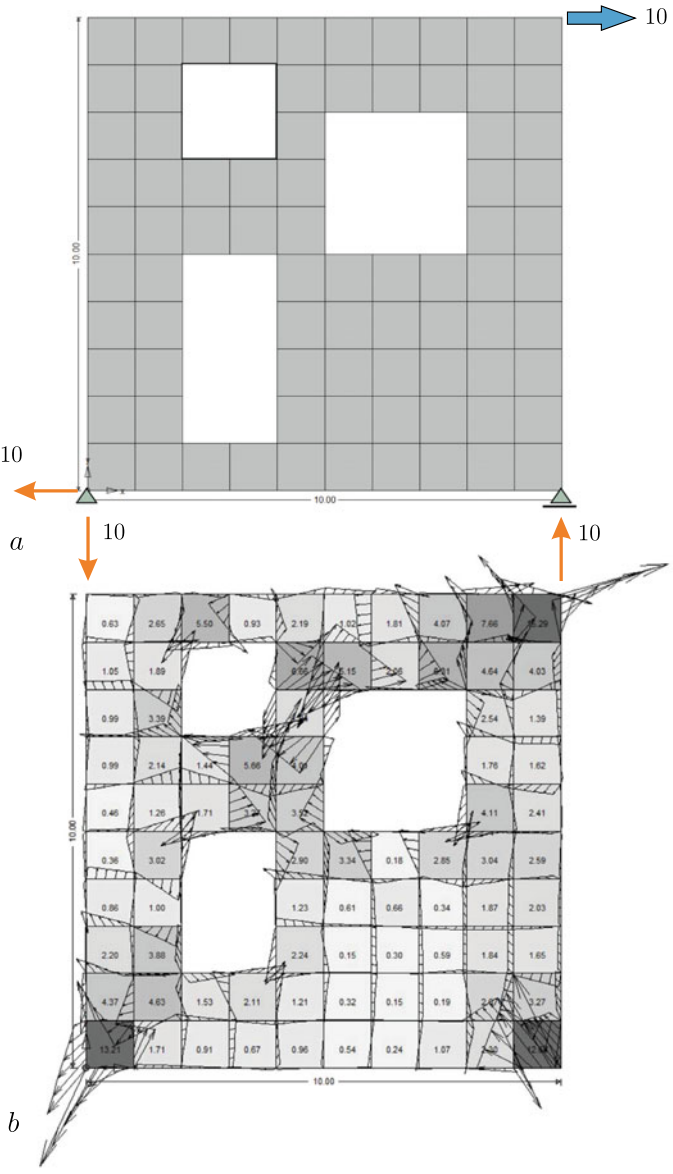

Fig. 3.10 Point load, **a** system, **b** the FE-load $\boldsymbol{p}_h$; the grey shades correspond to the strength of the element loads in the bilinear elements, as in Fig. 3.8

This is guaranteed, if $\boldsymbol{u}$ solves the system $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$, since each row (an internal energy δW_i) is—owing to ***internal = external***—identical to each row of the vector $\boldsymbol{f}_h$ (external work δW_e)

$$\boldsymbol{K}\,\boldsymbol{u} = \boldsymbol{f} \qquad \equiv \qquad \boldsymbol{f}_h = \boldsymbol{f}\,. \tag{3.43}$$

Imagine two plates side-by-side. The first carries the original load $\boldsymbol{p}$ and the second the FE-load $\boldsymbol{p}_h$. When you move from node to node and you displace the node by 1 m in horizontal and vertical direction, you will find that the work is each time the same $f_i^h = f_i$.

The FE-load is "shake-equivalent" to the original load.

Whether a structure carries the original load, or the substitute FE-load cannot be determined by testing the structure with the shape functions. Any such test is not conclusive, since each time the work comes out the same.

In Fig. 3.10 the original load and the FE-load are placed side by side. The single force at the upper right corner gets replaced by a very confusing looking system of domain loads and line loads, which constitute the FE-load $\boldsymbol{p}_h$. Because this load looks so "strange", it is normally not shown by FE-programs, since a user not familiar with the theory would be irritated. (But it is the service load, for which the user designs the structure).

That the FE-results cannot be as poor as Fig. 3.10 seems to indicate, illustrates a direct comparison of the principal stresses in Fig. 3.11. These look okay.

We would like to add a second, **indirect argument**. In the original load case, all $f_i = 0$ are zero, except $f_i = 10$ at the upper right corner, and therefore all $f_{hi} = 0$ must be zero as well, except $f_{hi} = 10$,

$$\int_\Omega \boldsymbol{p}_h^T \boldsymbol{\varphi}_i \; d\Omega = f_{hi} = f_i = 0\,. \tag{3.44}$$

The FE-program must juggle vigorously to guarantee this property, and this may explain the "chaos" in Fig. 3.10b. All the forces in this rather confusing looking load case $\boldsymbol{p}_h$ are so balanced that they do zero work when a node gets displaced by one meter in horizontal or vertical direction.

Most of the FE-load $\boldsymbol{p}_h$ is zero in terms of the energy metrics since it generates (as it must(!)) no nodal forces f_{hi}.

Remark 3.4 The *shape forces* $\boldsymbol{p}_i$ of a plate are domain forces and line forces and so the virtual work of these forces is a sum of domain integrals and line integrals. The domain integral in (3.44) is an abbreviating notation for all these integrals.

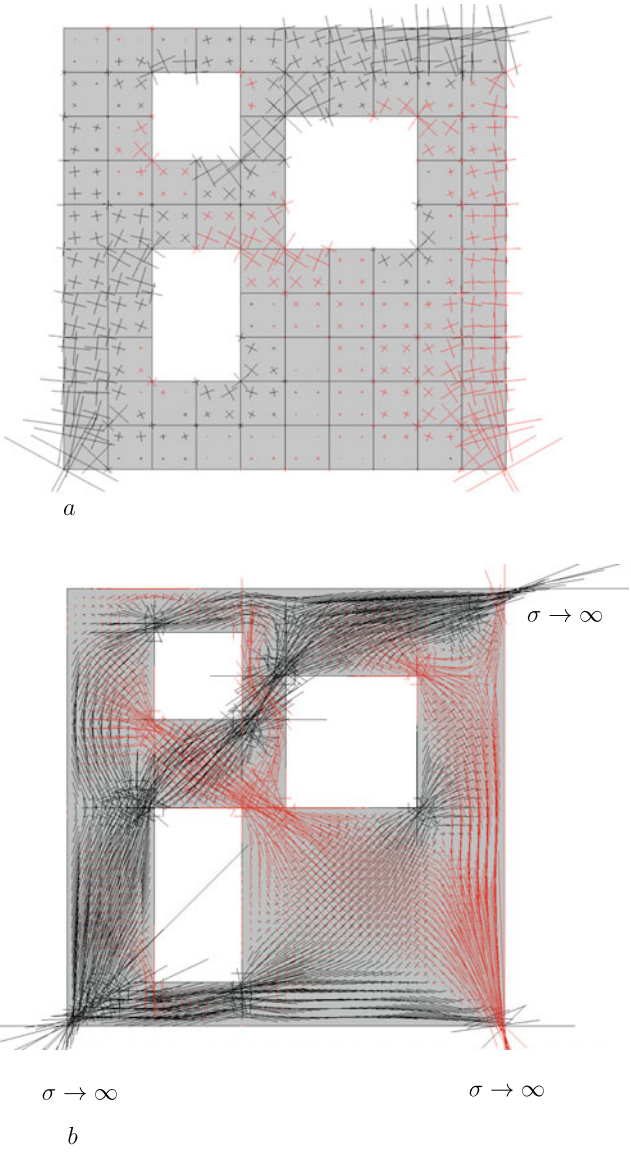

Fig. 3.11 Principal stresses, **a** coarse mesh, and **b** a very refined mesh

3.8 How the Ball Got Rolling

If we read $\boldsymbol{f}_h = \boldsymbol{f}$ as the basic equation of the finite element method, then this agrees with the approach in the **original paper** of *Turner et alteri* (1956) [1]. The authors looked at a *CST element* and formulated a matrix $\boldsymbol{S}$, which links the three (constant) stresses in an element, $\boldsymbol{\sigma} = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}^T$, to the nodal displacements[3]

$$\boldsymbol{\sigma}_{(3)} = \boldsymbol{S}_{(3\times 6)}\boldsymbol{u}_{(6)} \,. \tag{3.45}$$

The FE-load $\boldsymbol{p}_h$, which generates the shape $\boldsymbol{u}$ consists of edge forces only, since the derivatives of the constant stresses are zero—no volume forces. Next, the authors calculated the six equivalent nodal forces, vector $\boldsymbol{f}_h$, which belong to the load $\boldsymbol{p}_h$ by superposing the element edge forces (tractions) with the 3×2 unit displacements $\boldsymbol{\varphi}_i(\boldsymbol{x})$ of the three nodes. This led to a matrix $\boldsymbol{T}$

$$\boldsymbol{f}_h = \boldsymbol{T}_{(6\times 3)}\boldsymbol{\sigma}_{(3)} \,, \tag{3.46}$$

or with (3.45) to the equation

$$\boldsymbol{f}_h = \boldsymbol{T}_{(6\times 3)}\boldsymbol{S}_{(3\times 6)}\boldsymbol{u}_{(6)} \,, \tag{3.47}$$

and this is exactly the stiffness matrix $\boldsymbol{K} = \boldsymbol{T}\boldsymbol{S}$ of the *CST element*. That an analysis of virtual external work produced the stiffness matrix, whose entries we read as virtual *internal* energies, is based on the ***inside = outside***, which is guaranteed by Green's first identity. If

$$\delta W_e(\boldsymbol{p}_i, \boldsymbol{\varphi}_j) = \int_{\Gamma_e} \boldsymbol{t}(\boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_j \, ds =: f_{ij} \tag{3.48}$$

is the work done by the tractions $\boldsymbol{t}(\boldsymbol{\varphi}_i)$, the edge stresses of the field $\boldsymbol{\varphi}_i$, on acting through $\boldsymbol{\varphi}_j$, then

$$\mathscr{G}(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j) = \delta W_e(\boldsymbol{p}_i, \boldsymbol{\varphi}_j) - \delta W_i(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j) = f_{ij} - k_{ij} = 0 \,, \tag{3.49}$$

and therewith

$$f_{hi} = \delta W_e(\boldsymbol{p}_h, \boldsymbol{\varphi}_i) = \sum_j f_{ij}\, u_j = \sum_j k_{ij}\, u_j \,. \tag{3.50}$$

The elements k_{ij} of a stiffness matrix $\boldsymbol{K}$ can be read as virtual internal energy as well as virtual external work f_{ij}.

[3] We follow here the representation in [2] pp. 882–883.

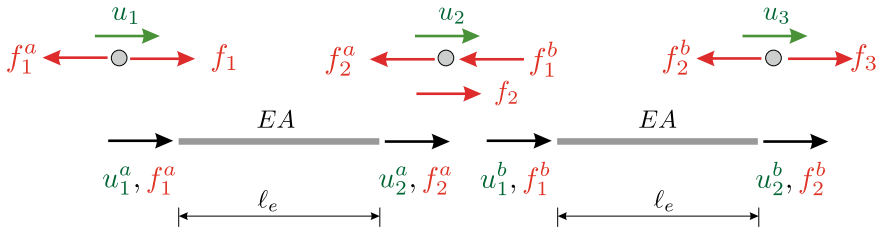

Fig. 3.12 System consisting of two bar elements

The entry $k_{11} = \delta W_i(\varphi_1, \varphi_1)$ of a beam matrix

$$k_{11} = \int_0^l \frac{M_1^2}{EI}\, dx = a(\varphi_1, \varphi_1) = \frac{12\,EI}{l^3} \cdot 1 = \delta W_e(\varphi_1, \varphi_1) = f_{11} \tag{3.51}$$

equals the work f_{11}, which the force $12\,EI/l^3$—the force, which pushes the node down by one unit length—does on acting through $\varphi_1(0) = 1$.

The "first" stiffness matrix in the history of the FEM was a table of **virtual external work**. Only the equation $\delta W_i = \delta W_e$ leads to the interpretation how we read stiffness matrices $\boldsymbol{K}$ today, $k_{ij} = a(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j)$.

What is truly remarkable is, how **simple** the finite elements began—no energy principle, no Galerkin, no higher mathematics—but an old trusted principle, the "shake equivalence"

$$\boxed{\delta W_e(\boldsymbol{p}, \boldsymbol{\varphi}_i) = \delta W_e(\boldsymbol{p}_h, \boldsymbol{\varphi}_i) \qquad \text{the start of the FEM}} \tag{3.52}$$

got the ball rolling.[4]

3.9 Assembling the Element Matrices

To prepare for the next section on the *equivalent stress transformation*, we shortly repeat, how element matrices are assembled.

At the start the two element matrices of the bar in Fig. 3.12 are placed on the diagonal of a 4×4 matrix $\boldsymbol{K}_{loc}$

[4] Only later, when the mathematicians came on board, it was recognized that the elements could be interpreted as finite functions and that the equivalence $\boldsymbol{f}_h = \boldsymbol{f}$ corresponds to the $\delta\Pi = 0$ of the potential energy.