8

The TL Bar Element

TABLE OF CONTENTS

§8.1. Introduction

In the present Chapter the key concepts of nonlinear continuum mechanics reviewed in Chapter 7 are applied to the development of governing equations of bar (truss) elements based on the Total Lagrangian (TL) kinematic description. For brevity these will be referred to as "TL bar elements."

There are two ways to construct TL elements:

1. The Standard Formulation (SF)
2. The Core Congruential Formulation (CCF).

The first method is easier to describe and will be presented in this Chapter through examples. The second one is more flexible and powerful but it is more difficult to teach because it proceeds in stages. As such it will be relegated to Chapters 10-11.

§8.2. The Two-Dimensional Bar Element

The element developed in this Chapter is a prismatic bar element that can be used to model pin-jointed *plane truss* structures of the type sketched in Figure 8.1. These structures undergo large displacements and rotations but their strains are assumed to remain small so that the material behavior stays in the linear elastic range. These assumptions allows us to consider only geometric nonlinear effects.

A two-node bar element appropriate to model members of such truss structures is shown in Figure 8.2. The element moves in the (X, Y) plane. In the reference (base) configuration the element has cross section area A_0 (constant along the element) and length L_0. In the current configuration the cross section area and length become A and L, respectively. The material has an elastic modulus E that links the axial-stress and axial-strain measures defined below.

Because this Chapter deals primarily with the formulation of an individual element, the identification superscript (e) will be omitted to reduce clutter until assemblies are considered.

The element has four node displacements and associated node forces. These quantities are collected in the vectors

$$\mathbf{u} = \begin{bmatrix} u_{X1} \\ u_{Y1} \\ u_{X2} \\ u_{Y2} \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} f_{X1} \\ f_{Y1} \\ f_{X2} \\ f_{Y2} \end{bmatrix}, \tag{8.1}$$

The loads acting on the nodes will be assumed to be conservative.

§8.2.1. Element Kinematics

In accordance with bar theory, to describe the element motion it is sufficient to consider a generic point of coordinates $\mathbf{X}$ located on the longitudinal axis of the reference configuration $\mathcal{C}_0$. That point maps to point $\mathbf{x}$ in the current configuration $\mathcal{C}$. The bar remains straight in any configuration. These coordinates can be parametrically interpolated from the end nodes as

$$\begin{aligned} \mathbf{X}(\xi) &= \begin{bmatrix} X(\xi) \\ Y(\xi) \end{bmatrix} = \begin{bmatrix} \tfrac{1}{2}(1-\xi)X_1 + \tfrac{1}{2}(1+\xi)X_2 \\ \tfrac{1}{2}(1-\xi)Y_1 + \tfrac{1}{2}(1+\xi)Y_2 \end{bmatrix} \\ \mathbf{x}(\xi) &= \begin{bmatrix} x(\xi) \\ y(\xi) \end{bmatrix} = \begin{bmatrix} \tfrac{1}{2}(1-\xi)x_1 + \tfrac{1}{2}(1+\xi)x_2 \\ \tfrac{1}{2}(1-\xi)y_1 + \tfrac{1}{2}(1+\xi)y_2 \end{bmatrix} \end{aligned} \tag{8.2}$$

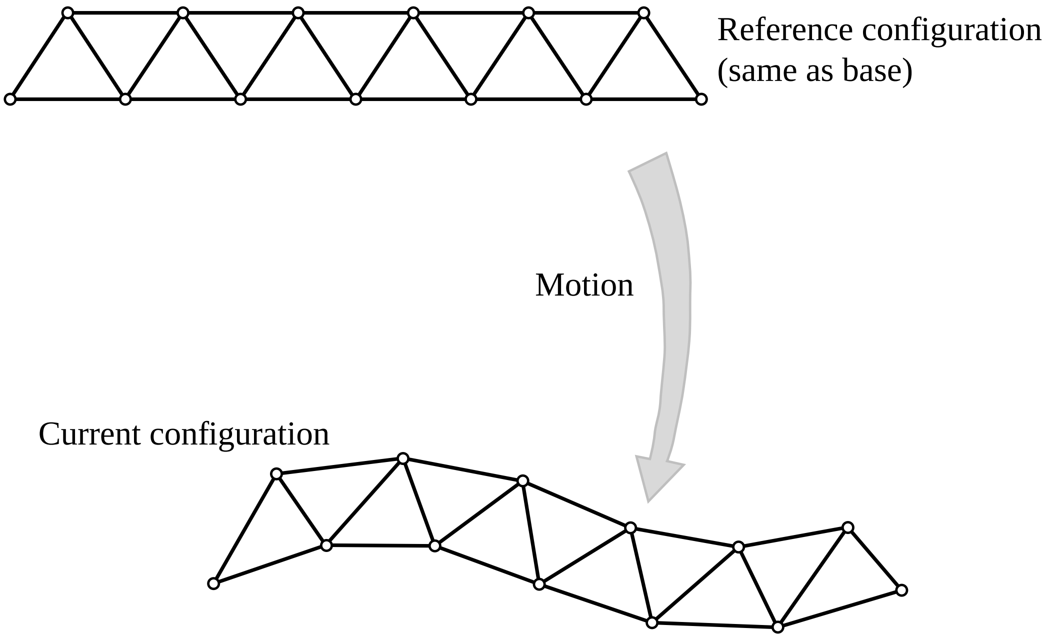


Figure 8.1. A plane truss structure undergoing large displacements while its material stays in the linear elastic range.

Here ξ is the usual isoparametric coordinate that varies from -1 at node 1 to $+1$ at node 2. The displacement field is obtained by subtracting the foregoing position vectors:

$$\mathbf{u}(\xi) = \mathbf{x}(\xi) - \mathbf{X}(\xi) = \begin{bmatrix} u_X(\xi) \\ u_Y(\xi) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1-\xi)u_{X1} + \frac{1}{2}(1+\xi)u_{X2} \\ \frac{1}{2}(1-\xi)u_{Y1} + \frac{1}{2}(1+\xi)u_{Y2} \end{bmatrix}, \tag{8.3}$$

which expressed in matrix form is

$$\boxed{\mathbf{u}(\xi) = \begin{bmatrix} u_X(\xi) \\ u_Y(\xi) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1-\xi) & 0 & \frac{1}{2}(1+\xi) & 0 \\ 0 & \frac{1}{2}(1-\xi) & 0 & \frac{1}{2}(1+\xi) \end{bmatrix} \begin{bmatrix} u_{X1} \\ u_{Y1} \\ u_{X2} \\ u_{Y2} \end{bmatrix} = \mathbf{N}(\xi)\,\mathbf{u}.} \tag{8.4}$$

The element kinematic defined by these equations is depicted in Figure 8.3.

§8.2.2. Strain Measure

As discussed in Chapter 7, in the Total Lagrangian (TL) description the Green-Lagrange (GL) strains and the second Piola-Kirchhoff (PK2) stresses are frequently used as conjugate measures in the formulation of the internal energy. The only GL strain that appears in the energy expression is

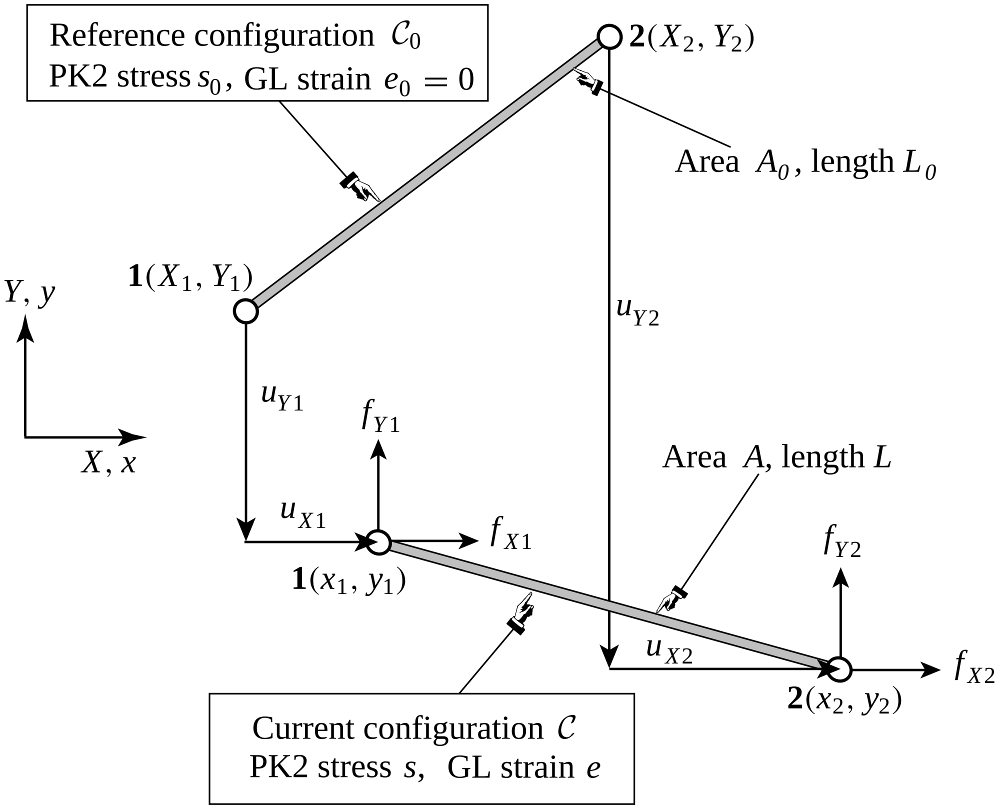


Figure 8.2. The geometrically nonlinear, two-node, two-dimensional bar element in Total Lagrangian description. This element may be used to model members of a plane truss such as that shown in Figure 8.1.

the axial strain $e_1 \equiv e$ which is most expediciously defined using the length change as

$$\boxed{e = \frac{L^2 - L_0^2}{2L_0^2},} \tag{8.5}$$

rather than through displacement gradients. Because of the linear displacement assumptions (8.4) the strain e is *constant* over the element.[1]

This expression can be maneuvered into a matrix function of the node displacements as follows. Let $X_{21} = X_2 - X_1$, $Y_{21} = Y_2 - Y_1$, $a_X = X_{21}/L_0$, $a_Y = Y_{21}/L_0$, $u_{X21} = u_{X2} - u_{X1}$, $u_{Y21} = u_{Y2} - u_{Y1}$, $u_X^m = (u_{X2} - u_{X1})/2$ and $u_Y^m = (u_{Y2} - u_{Y1})/2$. Some of these quantities can be geometrically interpreted as illustrated in Figure 8.4. Then

[1] This is in fact the only use of the displacement interpolation (8.4) in the following derivations.

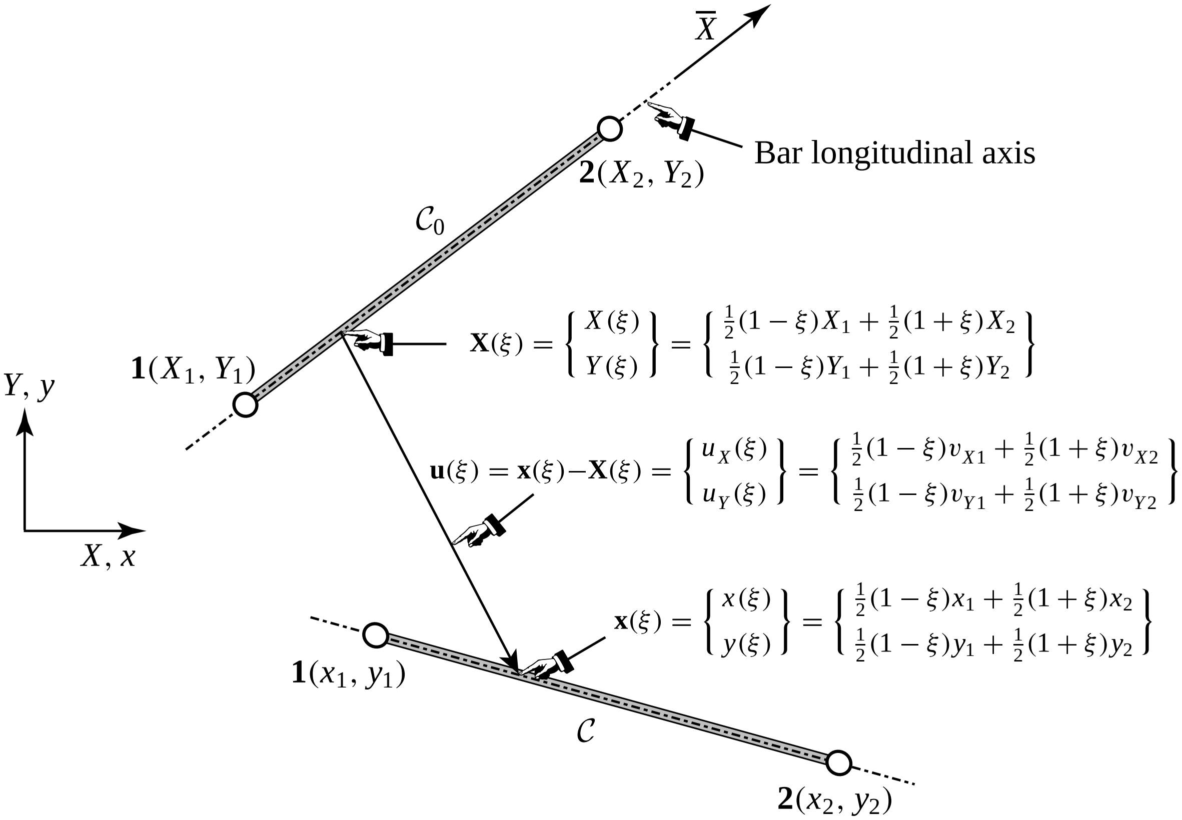

$\overline{X}$
Bar longitudinal axis
$\mathbf{2}(X_2, Y_2)$
$\mathcal{C}_0$
$\mathbf{X}(\xi) = \begin{Bmatrix} X(\xi) \\ Y(\xi) \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2}(1-\xi)X_1 + \frac{1}{2}(1+\xi)X_2 \\ \frac{1}{2}(1-\xi)Y_1 + \frac{1}{2}(1+\xi)Y_2 \end{Bmatrix}$
Y, y
$\mathbf{1}(X_1, Y_1)$
$\mathbf{u}(\xi) = \mathbf{x}(\xi) - \mathbf{X}(\xi) = \begin{Bmatrix} u_X(\xi) \\ u_Y(\xi) \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2}(1-\xi)v_{X1} + \frac{1}{2}(1+\xi)v_{X2} \\ \frac{1}{2}(1-\xi)v_{Y1} + \frac{1}{2}(1+\xi)v_{Y2} \end{Bmatrix}$
X, x
$\mathbf{x}(\xi) = \begin{Bmatrix} x(\xi) \\ y(\xi) \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2}(1-\xi)x_1 + \frac{1}{2}(1+\xi)x_2 \\ \frac{1}{2}(1-\xi)y_1 + \frac{1}{2}(1+\xi)y_2 \end{Bmatrix}$
$\mathbf{1}(x_1, y_1)$
$\mathcal{C}$
$\mathbf{2}(x_2, y_2)$

Figure 8.3. The definition of displacement field for the two-dimensional TL bar element.

$$\begin{aligned} L^2 &= (X_{21} + u_{X21})^2 + (Y_{21} + u_{Y21})^2, \\ e &= \frac{L^2 - L_0^2}{2L_0^2} = \frac{1}{L_0}(a_X u_{X21} + a_Y u_{Y21}) + \frac{1}{2L_0^2}\left(u_{X21}^2 + u_{Y21}^2\right) \\ &= \frac{1}{L_0}\,[\, -a_X \quad -a_Y \quad a_X \quad a_Y \,]\,\mathbf{u} + \frac{1}{L_0^2}\,[\, -u_X^m \quad -u_Y^m \quad u_X^m \quad u_Y^m \,]\,\mathbf{u} \\ &= \big(\mathbf{B}_l + \mathbf{B}_n(\mathbf{u})\big)\mathbf{u}. \end{aligned} \tag{8.6}$$

Observe that the GL strain e has been separated into two parts: $e = e_l + e_n$, where

1. $e_l = \mathbf{B}_l\mathbf{u}$, where $\mathbf{B}_l$ is constant, depends linearly in the node displacements $\mathbf{u}$. This is called the *linear* part of the strain.

2. $e_n = \mathbf{B}_n\mathbf{u}$, in which $\mathbf{B}_n$ is a function of the node displacements, depends quadratically on the displacements. This is called the *nonlinear* part of the strain, because $\mathbf{B}_n(\mathbf{u})$ vanishes if $\mathbf{u} \to \mathbf{0}$. Note that both matrices are constant over the element.

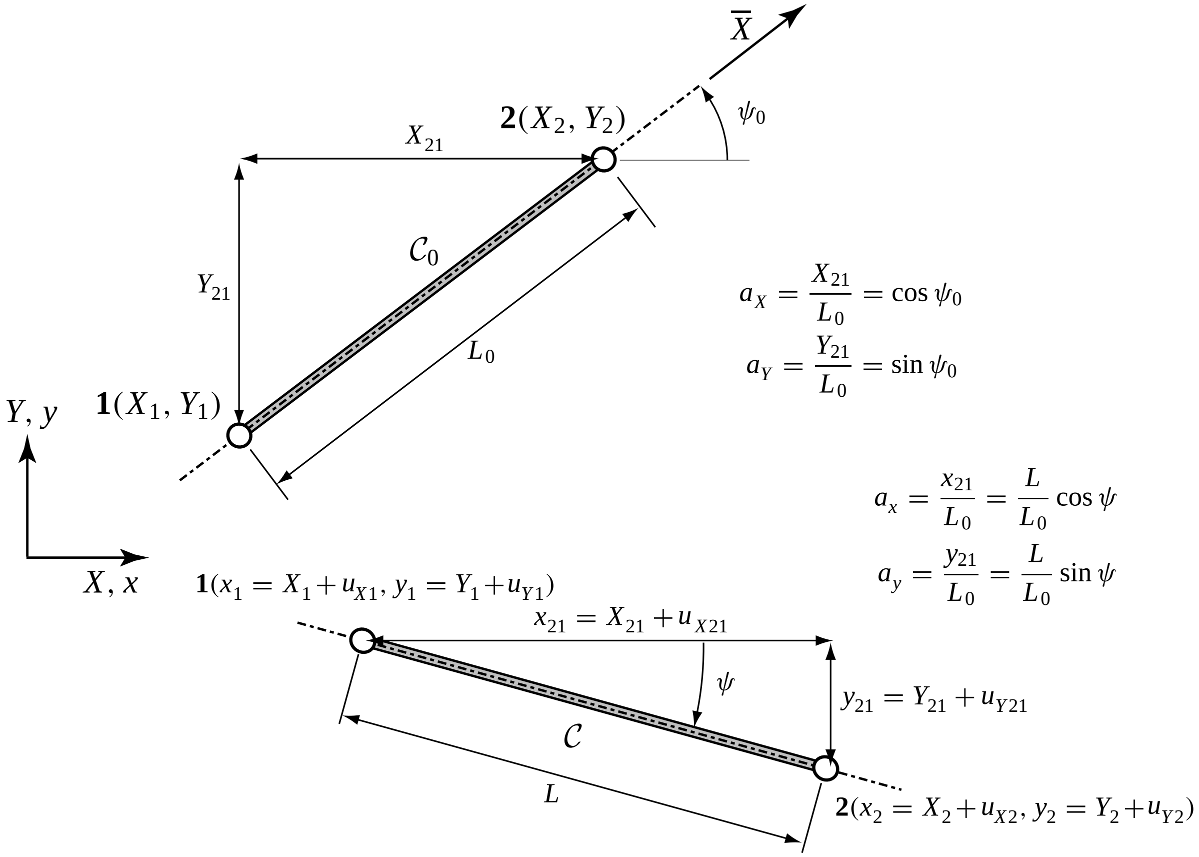

$\bar{X}$
ψ_0
X_{21}
$\mathbf{2}(X_2, Y_2)$
$\mathcal{C}_0$
Y_{21}
L_0
$a_X = \dfrac{X_{21}}{L_0} = \cos\psi_0$
$a_Y = \dfrac{Y_{21}}{L_0} = \sin\psi_0$
Y, y
$\mathbf{1}(X_1, Y_1)$
$a_x = \dfrac{x_{21}}{L_0} = \dfrac{L}{L_0}\cos\psi$
$a_y = \dfrac{y_{21}}{L_0} = \dfrac{L}{L_0}\sin\psi$
X, x
$\mathbf{1}(x_1 = X_1 + u_{X1}, y_1 = Y_1 + u_{Y1})$
$x_{21} = X_{21} + u_{X21}$
ψ
$y_{21} = Y_{21} + u_{Y21}$
$\mathcal{C}$
L
$\mathbf{2}(x_2 = X_2 + u_{X2}, y_2 = Y_2 + u_{Y2})$

Figure 8.4. Geometric interpretation of quantities used in the study of element kinematics.

The variation of e induced by variations $\delta\,\mathbf{u}$ in the node displacements is

$$\boxed{\delta e = \mathbf{B}_l\,\delta\mathbf{u} + \delta(\mathbf{B}_n\mathbf{u}) = \mathbf{B}\,\delta\mathbf{u},} \tag{8.7}$$

The matrix $\mathbf{B}$ that links δe to $\delta\mathbf{u}$ is

$$\mathbf{B} = \frac{\partial(\mathbf{B}_\ell + \mathbf{B}_n\mathbf{u})}{\partial\mathbf{u}} = \frac{1}{L_0}\,[\,-a_x \quad -a_y \quad a_x \quad a_y\,] \tag{8.8}$$

in which

$$a_x = a_X + \frac{u_{X21}}{L_0} = \frac{x_2 - x_1}{L_0} = \frac{x_{21}}{L_0}, \qquad a_y = a_Y + \frac{u_{Y21}}{L_0} = \frac{y_2 - y_1}{L_0} = \frac{y_{21}}{L_0}. \tag{8.9}$$

For a geometric interpretation of a_x and a_y see Figure 8.4.

§8.2.3. Stress Measure

The stress measure conjugate to GL strains is the second Piola-Kirchhoff (PK2) stress tensor. The only component that appears in the internal energy is the axial stress s, which is related to e through the constitutive equation

$$\boxed{s = s_0 + Ee,} \tag{8.10}$$

where s_0 is the axial stress in the reference configuration, and E is the elastic modulus.

The axial force based on this stress is

$$N = A_0 s \tag{8.11}$$

Note that this is not the true axial force in the current configuration $\mathcal{C}$, which would be

$$N_{true} = A\sigma \tag{8.12}$$

where σ is the true or Cauchy stress in $\mathcal{C}$ and A is the actual area.

§8.2.4. Total Potential Energy and Residual Force Equations

In what follows it is assumed that the element is subjected only to node forces $\mathbf{f}$ that are conservative and proportional, so that $\mathbf{f} = \lambda\mathbf{q}$. The Total Potential Energy (TPE) of the element in the current configuration is

$$\Pi = U - P = \int_{V_0} (s_0 e + \tfrac{1}{2} E e^2)\, dV_0 - \mathbf{f}^T\mathbf{u} = \int_{L_0} A_0 (s_0\, e + \tfrac{1}{2} E e^2)\, d\bar{X} - \lambda\mathbf{q}^T\mathbf{u}. \tag{8.13}$$

where $\bar{X}$ is directed along the bar longitudinal axis in $\mathcal{C}_0$, as shown in Figure 8.4. This energy expression is separable because the internal energy depends only on $\mathbf{u}$ through e and not on λ.

The finite element residual equations are obtained by differentiating this equation or, equivalently, making (8.13) stationary with respect to virtual displacement variations $\delta\,\mathbf{u}$:

$$\boxed{\delta\Pi = \delta U - \delta P = (\,\mathbf{p} - \mathbf{f})^T\,\delta\mathbf{u} = 0.} \tag{8.14}$$

The first variation of U is given by

$$\delta U = \mathbf{p}^T\,\delta\mathbf{u} = \int_{L_0} A_0 (s_0\,\delta e + E e \delta e)\, d\bar{X} = \int_{L_0} A_0\, s\, \delta e\, d\bar{X} = \int_{L_0} N_0 \mathbf{B}\,\delta\mathbf{u}\, d\bar{X} = N L_0 \mathbf{B}\,\delta\mathbf{u} \tag{8.15}$$

where $N = A_0 s$ is the axial force in the current configuration measured per area of the reference configuration.[2] It follows that the internal force vector is

$$\mathbf{p} = N_0 L_0 \mathbf{B}^T = N \begin{bmatrix} -a_x \\ -a_y \\ a_x \\ a_y \end{bmatrix}. \tag{8.16}$$

This equation admits of a simple geometric interpretation; see Figure 8.5. The relation between $N_0 = A_0 s$ and the true axial force $N = A\sigma$ can be worked out from inspection of this diagram.

The load potential variation δP simply generates $\mathbf{f} = \lambda\mathbf{q}$ as can be expected.

[2] This is the PK2 axial force; cf (8.11).

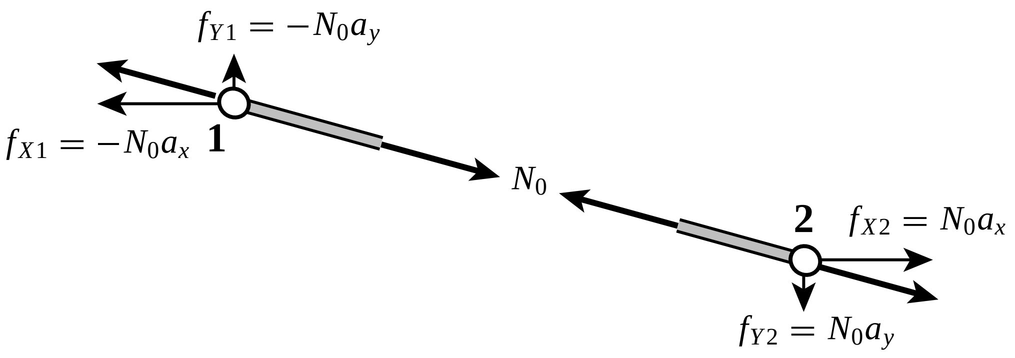


Figure 8.5. Geometrical interpretation of the internal force vector. The axial force $N_0 = A_0 s$ would be positive as shown.

§8.2.5. The Tangent Stiffness Matrix

Because the residual equations are separable the tangent stiffness matrix is obtained simply by differentiating the internal force with respect to the node displacements **u**:

$$\boxed{\mathbf{K} = \frac{\partial \mathbf{p}}{\partial \mathbf{u}} = \frac{\partial (N L_0 \mathbf{B}^T)}{\partial \mathbf{u}} = A_0 L_0 \mathbf{B}^T \frac{\partial s}{\partial \mathbf{u}} + A_0 L_0 s \frac{\partial \mathbf{B}^T}{\partial \mathbf{u}} = \mathbf{K}_M + \mathbf{K}_G.} \tag{8.17}$$

The above expression shows that **K** splits naturally into two parts: $\mathbf{K}_M$ and $\mathbf{K}_G$, which are called the *material stiffness matrix* and *geometric stiffness matrix*, respectively, in the FEM literature.

To get $\mathbf{K}_M$ note that

$$\frac{\partial s}{\partial \mathbf{u}} = \frac{\partial (s_0 + Ee)}{\partial \mathbf{u}} = E \frac{\partial e}{\partial \mathbf{u}} = E\mathbf{B}. \tag{8.18}$$

Consequently

$$\boxed{\mathbf{K}_M = E A_0 L_0 \mathbf{B}^T \mathbf{B}.} \tag{8.19}$$

Inserting the expression (8.8) for **B** yields

$$\boxed{\mathbf{K}_M = \frac{E A_0}{L_0} \begin{bmatrix} a_x^2 & a_x a_y & -a_x^2 & -a_x a_y \\ a_x a_y & a_y^2 & -a_x a_y & -a_y^2 \\ -a_x^2 & -a_x a_y & a_x^2 & a_x a_y \\ -a_x a_y & -a_y^2 & a_x a_y & a_y^2 \end{bmatrix}} \tag{8.20}$$

This component of **K** looks formally similar to the stiffness matrix of a linear bar element,[3] except that **B** now depends on **u**. The dependence of $\mathbf{K}_M$ on the material properties (here the elastic modulus E) explains the name “material stiffness” given in the FEM literature.

[3] To which it reduces if $\mathbf{u} = \mathbf{0}$. In that case a_x and a_y become the sine and cosine of the angle ψ_0 shown in Figure 8.4.

The other component can be obtained by differentiating **B** with respect to the node displacements, the result being a constant 4 × 4 matrix:

$$\frac{\partial \mathbf{B}^T}{\partial \mathbf{u}} = \frac{1}{L_0^2}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \tag{8.21}$$

Inserting this into (8.17) one gets

$$\boxed{\mathbf{K}_G = \frac{N}{L_0}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}}. \tag{8.22}$$

This component of **K** depends only on the stress state in the current configuration, because $N = A_0 s$. No material properties appear. Thus the name "geometric stiffness" applied to $\mathbf{K}_G$.[4]

Remark 8.1. Assuming that E, A_0 and L_0 are nonzero, the rank of $\mathbf{K}_M$ is obviosly one because **B** is a 1 × 4 matrix. On the other hand the rank of the numerical matrix in (8.21) is 2 (because its eigenvalues are 2, 2, 0, 0). Consequently $\mathbf{K}_G$ has rank 2 if s is nonzero and 0 otherwise. Combining these results it can be shown that the rank of $\mathbf{K} = \mathbf{K}_M + \mathbf{K}_G$ is 1 if the configuration is stressed and 2 otherwise. In other words, the rank deficiency is 3 and 2, respectively. The implications of this property in the analysis of stability are studied later.

Remark 8.2. The addition of $\mathbf{K}_G$ increases the bar stiffness if the current configuration is in tension ($s > 0$), but it reduces it if the current configuration is in compression ($s < 0$). This is in accord with physical intuition. The main effect of this stiffness is on the rotational rigid-body motions of the bar about the Z axis.

§8.3. FEM Coding Using Mathematica

This section illustrates the use of `Mathematica` as a quick way to do some simple problems with the TL bar element. The use of a computer symbolic system as prototyping language has several advantages:

(i) The code is very compact, because of higher lever array notation. Because of its compactness, it can be easily translated into other high-level languages such as, for example, `Matlab` or `Maple`.

(ii) Debugging is often quicker than with programming languages because of the interpretative nature of the language front end. This is of course platform dependent; the implementation of `Mathematica` on a Macintosh or PC is friendler than on a Unix workstation.

(iii) Graphic output is part of the language. On the Macintosh, graphics can be transported into documents such as this one via `PostScript` and `Adobe Illustrator`.

(iv) The language can manipulate both algebraic (symbolic) and numerical expressions. The later may be done exactly or in floating-point arithmetic. This flexibility is convenient for many situations, as the examples below illustrate.

[4] In the pre-1970 FEM literature, the name "initial stress stiffness" was used for $\mathbf{K}_G$ by some authors.

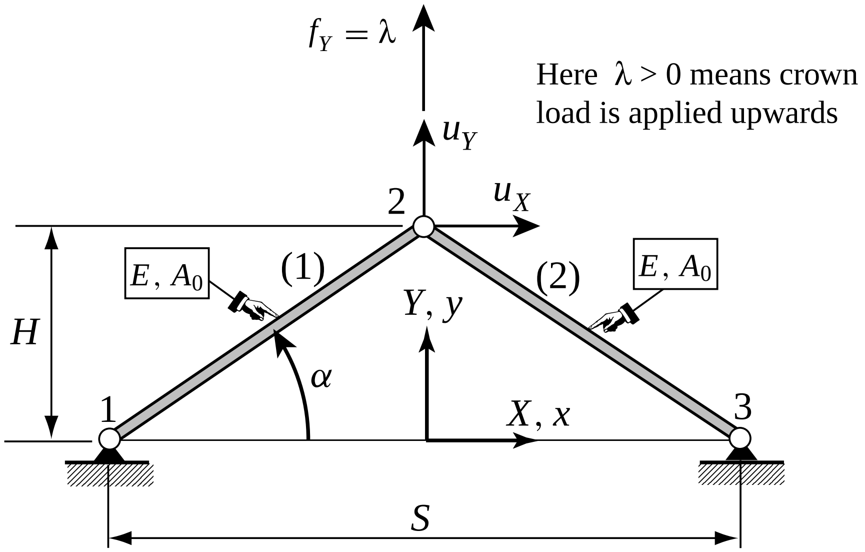


Figure 8.6. An arch as a two-bar FE model. This model has two degrees of freedom: u_X and u_Y, which are the displacements of node 2. This structure will be used as example for use of `Mathematica` in §8.3.

§8.3.1. The Example Structure

The two-spring arch studied in Exercise 6.3 will be used as a structure that illustrates the use of `Mathematica` in conjunction with finite elements. The structure is shown in Figure 8.6. It is modeled with two TL bar elements labeled (1) and (2), which join nodes 1-2 and 2-3, respectively. The modulus of elasticity and the bar areas are unity. The arch geometry is defined by the span S and height H. The reference configuration is assumed stress free.

Because nodes 1 and 3 are pinned to the ground, the only degrees of freedom are the horizontal and vertical displacements of node 2. To avoid clutter these will be denoted in the sequel as u_X and u_Y rather than u_{X2} and u_{Y2}, respectively. The vertical motion u_Y is positive upwards. The node-2 components of the applied and internal force vector will be also denoted simply by (f_X, f_Y) and (p_X, p_Y), respectively. Displacement and force components for nodes 1 and 3 will be explicitly removed from the governing equations.

The arch is loaded by a vertical force $f_Y = \lambda$ applied at node 2, positive upwards. The applied horizontal force f_X will be always zero. Recall that in Exercise 6.3 only *symmetric* motions of the arch under a vertical load were considered. That constraint effectively reduces the model to one degree of freedom: u_Y, and allows only limit point behavior. When two degrees of freedoms are considered, however, a richer set of possibilities emerges; notably lateral *bifurcation* under a vertical load.

§8.3.2. Forming the Internal Force

The `Mathematica` module `FormIntForce2DTwoNodeBar` listed in Figure 8.7 forms the internal force vector of an individual 2D bar finite element handled by the TL description. The arguments of this module are

`XY1`	Coordinates $\{X_1, Y_1\}$ of end node 1 in $\mathcal{C}_0$.
`XY1`	Coordinates $\{X_2, Y_2\}$ of end node 2 in $\mathcal{C}_0$.
`uX1`	Displacements $\{u_{X1}, u_{Y1}\}$ of end node 1.

```
FormIntForce2DTwoNodeBar[XY1_,XY2_,uXY1_,uXY2_,Em_,A0_,s0_]:=
  Module[{X1,Y1,X2,Y2,X21,Y21,uX1,uY1,uX2,uY2,uX21,uY21,L0,L,
     e,s,ax,ay,ifv},
    {X1,Y1}=XY1; {X2,Y2}=XY2; X21=X2-X1; Y21=Y2-Y1;
    {uX1,uY1}=uXY1; {uX2,uY2}=uXY2; uX21=uX2-uX1; uY21=uY2-uY1;
    L0=Sqrt[X21^2+Y21^2];     L=Sqrt[(X21+uX21)^2+(Y21+uY21)^2];
    e=(L-L0)*(L+L0)/(2*L0^2); s=s0+Em*e;
    ax=(X21+uX21)/L0;         ay=(Y21+uY21)/L0;
    ifv=A0*s*{{-ax}, {-ay}, {ax}, {ay}};
    Return[Simplify[ifv]]
        ];
pe=FormIntForce2DTwoNodeBar[{2,3},{5,7},{1,0},{1,0}, 20,12, 0];
Print["Internal elem force pe=",pe];
pe=FormIntForce2DTwoNodeBar[{2,3},{5,7},{1,0},{1,0}, 20,12, s0];
Print["Internal elem force pe=",pe];
```

```
Internal elem force pe={{0}, {0}, {0}, {0}}
                         -36 s0    -48 s0    36 s0     48 s0
Internal elem force pe={{------}, {------}, {-----}, {-----}}
                            5         5         5        5
```

Figure 8.7. A `Mathematica` module that computes the internal force vector **p** of an individual 2-node 2D bar element. Output from two test statements is shown.

`uX2`	Displacements $\{u_{X2}, u_{Y2}\}$ of end node 2.
`Em`	Elastic modulus E.
`A0`	Cross sectional area A_0.
`s0`	PK2 stress in $\mathcal{C}_0$

Two test statements, one with purely numeric arguments and the other one with one symbolic argument (`s0` for the bar prestress), are shown in the figure along with their output.

Figure 8.8 shows two modules, `AssembleMasterIntForceOfTwoBarArch` and `MergeElemIntoMasterIntForce` that, together with `FormIntForce2DTwoNodeBar`, form the internal force vector **p** for the two-bar example structure. The arguments of the former are the element internal force vector `eif`, the element freedom table `eftab` that maps element-level freedom to master-level freedoms and the master internal force vector `mif` before the element is merged. The function returns the updated internal force vector. The arguments of the second module are:

`span`	Span S
`height`	Height H.
`Em`	Elastic modulus E.
`A0`	Cross sectional area A_0.
`uX,uY`	Crown displacements u_X, u_Y.

```
MergeElemIntoMasterIntForce[pe_,eftab_,p_]:=
  Module[{i,ii,neldof,fmaster}, fmaster=p;
       neldof=Dimensions[eftab][[1]];
       For[i=1, i<=neldof, i++, ii=eftab[[i]];
           If [ii>0,fmaster[[ii,1]]+=pe[[i,1]]]
          ]; Return[fmaster]
  ];

AssembleMasterIntForceOfTwoBarArch[span_,height_,Em_,A0_,uX_,uY_]:=
  Module[{f1,f2,fmaster},
    fmaster=Table[0,{2},{1}];
    f1=FormIntForce2DTwoNodeBar[{-span/2,0},{0,height},
             {0,0},{uX,uY},Em,A0,0];
    fmaster=MergeElemIntoMasterIntForce[f1,{0,0,1,2},fmaster];
    f2=FormIntForce2DTwoNodeBar[{0,height},{span/2,0},
             {uX,uY},{0,0},Em,A0,0];
    fmaster=MergeElemIntoMasterIntForce[f2,{1,2,0,0},fmaster];
    Return[Simplify[fmaster]]
  ];

ClearAll[Em,A0,S,H,uX,uY,f];
p=AssembleMasterIntForceOfTwoBarArch[2,2.5,10,.75,-0.4,0.25];
Print["Master Int Force p= ",p];
p=AssembleMasterIntForceOfTwoBarArch[S,H,Em,A0,uX,uY];
Print["Master Int Force p= ",p];
```

```
Master Int Force p= {{-0.5336499821957073}, {1.555758744891104}}
Master Int Force p= {{(4*A0*Em*uX*
       (S^2 + 2*uX^2 + 4*H*uY + 2*uY^2))/(4*H^2 + S^2)^(3/2)},
      {(8*A0*Em*(H + uY)*(uX^2 + 2*H*uY + uY^2))/
       (4*H^2 + S^2)^(3/2)}}
```

Figure 8.8. Two `Mathematica` modules that assemble the internal force vector **p** for the two-bar arch structure of Figure 8.6. Two test statements, one numeric and the other symbolic, are shown along with output.

Two test statements are also shown in the figure. The first one is purely numeric and sets the following properties for the structure:

$$S = 2, \quad H = 2.5, \quad E = 10, \quad A_0 = 0.75, \quad u_X = 0.4, \quad u_Y = -0.25 \tag{8.23}$$

and the module `AssembleMasterIntForceOfTwoBarArch` returns

$$\mathbf{p} = \begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \begin{bmatrix} -0.53365 \\ 1.55576 \end{bmatrix} \tag{8.24}$$

The second one specifies symbolic values for all of the arguments: `S` for span S, `H` for the height H, etc. The result (displayed in `InputForm` so it can be cut and pasted easily) gives the internal force for an arbitrary structure:

$$\mathbf{p} = \begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \frac{4EA_0}{(4H^2+S^2)^{3/2}} \begin{bmatrix} u_X(S^2+2u_X^2+4Hu_Y+2u_Y^2) \\ 2(H+u_Y)(u_X^2+2Hu_Y+u_Y^2) \end{bmatrix}. \tag{8.25}$$

§8.3.3. The Equilibrium Equations

The residual equilibrium equations are of course $\mathbf{r} = \mathbf{p} - \mathbf{f} = \mathbf{0}$ or

$$\mathbf{p} = \begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \mathbf{f} = \begin{bmatrix} 0 \\ \lambda \end{bmatrix} \tag{8.26}$$

It follows that $p_X = 0$ and $p_Y = \lambda$. Expression (8.26) shows that $p_X = 0$ is a cubic equation that has the trivial root $u_X = 0$ in addition to the two roots of the quadratic $S^2 + 2u_X^2 + 4Hu_Y + 2u_Y^2 = 0$. Rather than solving these by hand, let us exemplify the use of `Mathematica` to solve a polynomial equation using the built-in `Solve` function. These roots are then substituted, using the `/.` operator, into the other equation $\lambda = p_Y$ to get the equations of the 3 equilibrium paths. The appropriate statements and output are shown in Figure 8.9.

```
ClearAll[Em,A0,S,H,uX,uY,pX,pY,lambda];
pX = (4*A0*Em*uX*(S^2+2*uX^2+4*H*uY+2*uY^2))/(4*H^2+ S^2)^(3/2);
pY = (8*A0*Em*(H+uY)*(uX^2+2*H*uY+uY^2))/(4*H^2+S^2)^(3/2);
roots=Solve[pX==0, uX];
Print["roots of pX=0 are ",roots];
Print["lambda=",Simplify[pY/.roots]];
```

```
roots of pX=0 are {{uX -> 0},
   {uX -> -((-S^2 - 4*H*uY - 2*uY^2)^(1/2)/2^(1/2))},
   {uX -> (-S^2 - 4*H*uY - 2*uY^2)^(1/2)/2^(1/2)}}
lambda={(8*A0*Em*uY*(H + uY)*(2*H + uY))/(4*H^2 + S^2)^(3/2),
   (-4*A0*Em*S^2*(H + uY))/(4*H^2 + S^2)^(3/2),
   (-4*A0*Em*S^2*(H + uY))/(4*H^2 + S^2)^(3/2)}
```

Figure 8.9. Illustration of use of `Mathematica` built-in `Solve` function to obtain the roots of $p_X = 0$ in terms of u_X, where p_X is given by (8.25). These roots are then substituted into $p_Y = \lambda$ with p_Y from the same equation, to get the expression of the 3 equilibrium paths in the form $\lambda = \lambda(u_Y)$.

The definition of `pX` in Figure 8.9 was directly set via cut-and-paste from the output shown in Figure 8.8. From the printed solution it is obvious that the three roots in terms of u_X are

$$u_X^p = 0, \quad u_X^{s1,s2} = \pm\sqrt{-S^2/2 - 2Hu_Y - u_Y^2}, \tag{8.27}$$

The solution u_X^p, where superscript p stands for "primary," corresponds to a *symmetric* deformation of the arch in its primary equilibrium path. This represents the kind of response studied in Exercise 6.3.

A nonzero u_X is possible if the roots u_X^{s1} and u_X^{s2}, where supercripts stand for "secondary," given by (8.27) are real. This can occur only if

$$\Delta = -S^2/2 - 2Hu_Y - u_Y^2 > 0. \tag{8.28}$$

Because both S and H are positive, condition (8.28) can only happen for negative u_Y, that is, *downward* displacement of the arch. Furthermore, given S and H the condition (8.28) requires (the analysis using `Mathematica` is not shown) that

$$-H + \sqrt{H^2 - \tfrac{1}{2}S^2} \geq u_Y \geq -H - \sqrt{H^2 - \tfrac{1}{2}S^2}, \tag{8.29}$$

which obviously can only happen if $H^2 \geq S^2/2$, or

$$\frac{H}{S/2} = \tan\alpha \geq \sqrt{2}. \tag{8.30}$$

If (8.29)-(8.30) are verified, the equilibrium equation $p_X = 0$ has three real roots: $u_X^p = 0$, $u_X^{s1,s2} = \pm\sqrt{\Delta}$. We shall see that the last two pertain to a *secondary equilibrium path* that intersects the primary equilibrium path $u_X = 0$ at two bifurcation points.

§8.3.4. Plotting the Equilibrium Paths

The `Mathematica` statements shown in Figure 8.10 produces the four response plots of λ versus u_Y shown in Figure 8.11. For all plots $E = A_0 = 1$ and $S = 2$. The height H is set to $\sqrt{3}/3$ ($\alpha = 30^\circ$, as in Exercise 6.3), $\sqrt{2}/2$, $\sqrt{3}$ and 3. These are labeled cases 1 through 4 respectively. All plots show the symmetric-response ($u_X = 0$) primary path with two limit points passing through $\lambda = 0$ at $u_Y = 0$, $u_X = -H$ and $u_Y = -2H$. The straight line is the projection of the *plane* on which the secondary equilibrium path lies. This path has a closed elliptical shape quite similar to that sketched in Figure 5.3 in Chapter 5; see also Figure 8.12 below. A study of the location of the secondary path with respect to the primary path leads to the following conclusions:

(a) If $H < \sqrt{2}$, or equivalently $\alpha < 54.74^\circ$, the secondary path does not interesect the primary path. The structure has only two critical points, both of which are limit points. This is exemplified by Case 1, for which $\alpha = 30^\circ$.

(b) If $\sqrt{2} \leq H \leq \sqrt{3}$, the secondary path intersects the primary path at two bifurcation points, which occur between the limit points. When the arch is loaded downward from $\lambda = 0$, the first limit point is encountered first, then the first bifurcation point. The arch will then follow the secondary path as it snaps through unsymmetrically, until it reaches the second bifurcation point, and proceeds to deform symmetrically from then on. The structure has four critical points: two limit and two bifurcation, but the load capacity is determined by the limit points. Cases 2 and 3 in Figure 8.11 illustrates the minimum and maximum heights for which this kind of behavior occur.

(c) If $H > \sqrt{3}$, which correponds to $\alpha = 60^\circ$, the structure has again four critical points: two limit and two bifurcation, but the limit points are now "inside" the bifurcation ones. Physically a gradually loaded arch will bifurcate first and snap through unsymmetrically until it reaches the second bifurcation point and continues to deform symmetrically from now on. Consequently the limit points cannot be physically reached. Case 4 in Figure 8.11, in which $H = 3$, exemplifies this kind of behavior.

The transition between (b) and (c) occurs at $\alpha = 60^\circ$, which is interesting because the limit and bifurcation points coalesce, as shown in Case 3 of Figure 8.11.

Plotting the equilibrium paths in three-dimensional state-control space (u_X, u_Y, λ) require far more programmming effort because the graphic capabilities of `Mathematica` begin to be seriously strained. Figure 8.12 illustrates the kind of complicated program needed to get decent results. The logic will not be explained here because it would not be intelligible unless the reader has substantial expertise in advanced graphics. The results of running this procedure for the interesting case $H = \sqrt{3}$ or $\alpha = 60^\circ$ are shown in Figure 8.13. This graph clearly displays the coalescence of the limit and bifurcation points.

```
lambdaP=(8*A0*Em*uY*(H + uY)*(2*H + uY))/(4*H^2 + S^2)^(3/2);
lambdaS=(-4*A0*Em*S^2*(H + uY))/(4*H^2 + S^2)^(3/2);
case1={H->Sqrt[3]/3,S->2,A0->1,Em->1};
case2={H->Sqrt[2],  S->2,A0->1,Em->1};
case3={H->Sqrt[3],  S->2,A0->1,Em->1};
case4={H->3,        S->2,A0->1,Em->1};
Plot[{lambdaP/.case1,lambdaS/.case1},{uY,0,-2*H/.case1},
     AxesLabel->{"uY","lambda"},PlotLabel->"Case 1: H=Sqrt[3]/3"];
Plot[{lambdaP/.case2,lambdaS/.case2},{uY,0,-2*H/.case2},
     AxesLabel->{"uY","lambda"},PlotLabel->"Case 2: H=Sqrt[2]"];
Plot[{lambdaP/.case3,lambdaS/.case3},{uY,0,-2*H/.case3},
     AxesLabel->{"uY","lambda"},PlotLabel->"Case 3: H=Sqrt[3]"];
Plot[{lambdaP/.case4,lambdaS/.case4},{uY,0,-2*H/.case4},
     AxesLabel->{"uY","lambda"},PlotLabel->"Case 4: H=3"];
```

Figure 8.10. Statements to generate $\lambda = \lambda(u_Y)$ response plots for four geometries. For all cases $E = A_0 = 1$ and $S = 2$. The height H is set to $\sqrt{3}/3$ ($\alpha = 30^\circ$, as in Exercise 6.3), $\sqrt{2}/2$, $\sqrt{3}$ and 3.

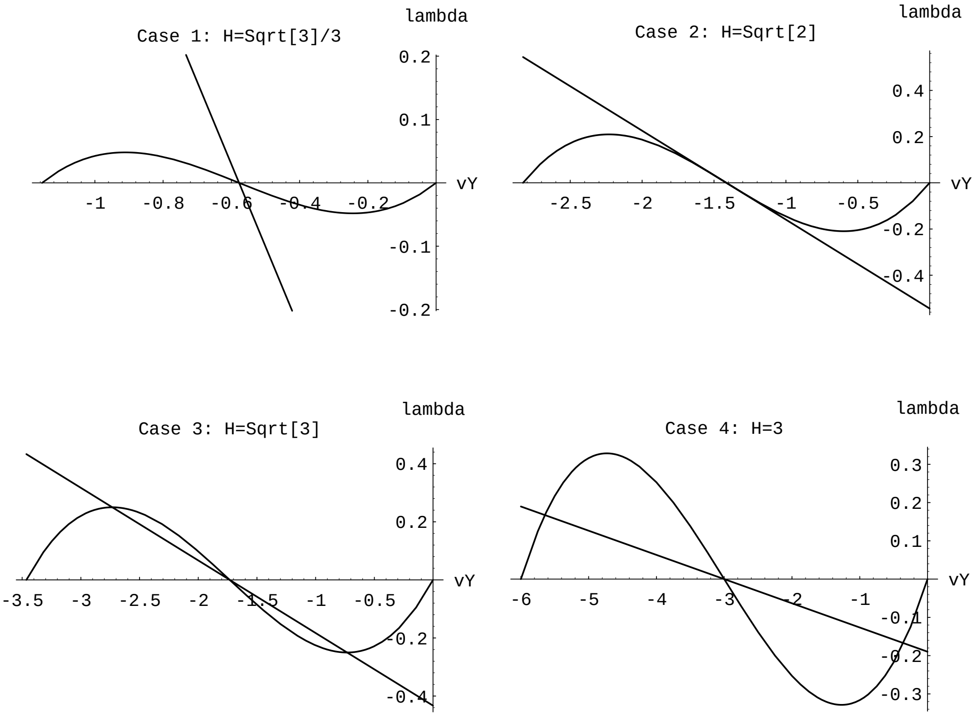


Figure 8.11. Plots of the arch equilibrium paths projected on the λ versus u_Y plane for the four cases described in Figure 8.10.

```
ClearAll[S,H,Em,A0,uX,uY]; H=Sqrt[3.];
lambdamax[S_,H_,Em_,A0_]:=3.0792*A0*Em*H^3/(4*H^2+S^2)^(3/2);
primarypath[S_,H_,Em_,A0_,uY_]:=
  {0,uY,8*A0*Em*uY*(H+uY)*(2*H+uY)/(4*H^2+S^2)^(3/2)};
secondarypath1[S_,H_,Em_,A0_,uY_]:=Module[{c,uYB1,uYB2},
  uYB1=Re[N[-H+Sqrt[H^2-S^2/2]]]; uYB2=Re[N[-H-Sqrt[H^2-S^2/2]]];
  c=4*A0*Em*S^2/(4*H^2+S^2)^(3/2);
  If [N[uY]>=uYB1, Return[{0,uYB1,-c*(H+uYB1)}]];
  If [N[uY]<=uYB2, Return[{0,uYB2,-c*(H+uYB2)}]];
  Return[{Sqrt[-S^2/2-2*H*uY-uY^2],uY,-c*(H+uY)}]
        ];
secondarypath2[S_,H_,Em_,A0_,uY_]:=Module[{c,uYB1,uYB2},
  uYB1=Re[N[-H+Sqrt[H^2-S^2/2]]]; uYB2=Re[N[-H-Sqrt[H^2-S^2/2]]];
  c=4*A0*Em*S^2/(4*H^2+S^2)^(3/2);
  If [N[uY]>=uYB1, Return[{0,uYB1,-c*(H+uYB1)}]];
  If [N[uY]<=uYB2, Return[{0,uYB2,-c*(H+uYB2)}]];
  Return[{-Sqrt[-S^2/2-2*H*uY-uY^2],uY,-c*(H+uY)}]
        ];
lambdarange=1.1*{-lambdamax[2,H,1,1],lambdamax[2,H,1,1]};
pp=ParametricPlot3D[primarypath[2,H,1,1,uY], {uY,0,-2*H},
   PlotPoints->201, PlotRange->{{-H,H},{0,-2*H},lambdarange},
   BoxRatios->{1,2,1},AxesLabel->{"uX","uY","lambda"},
   DisplayFunction->Identity];
ps1=ParametricPlot3D[secondarypath1[2,H,1,1,uY], {uY,0,-2*H},
   PlotPoints->201, PlotRange->{{-H,H},{0,-2*H},lambdarange},
   BoxRatios->{1,2,1},AxesLabel->{"uX","uY","lambda"},
   DisplayFunction->Identity];
ps2=ParametricPlot3D[secondarypath2[2,H,1,1,uY], {uY,0,-2*H},
   PlotPoints->201, PlotRange->{{-H,H},{0,-2*H},lambdarange},
   BoxRatios->{1,2,1},AxesLabel->{"uX","uY","lambda"},
   DisplayFunction->Identity];
Show[pp,ps1,ps2,ViewPoint->{3,1,2},DisplayFunction->$DisplayFunction];
```

Figure 8.12. Statements used to generate response plots in the three-dimensional state-control space (u_X, u_Y, λ), such as the one shown in the next figure.

§8.3.5. Having Some Fun: Snapshots and Animation

To illustrate what happens to the arch as it traverses its equilibrium path, configuration "snapshots" and particularly animations are invaluable and fun to do. The program shown in 8.14 does the first kind of display, showing configurations as the arch traverses the equilibrium path by incrementing u_Y. The program also depicts (very roughly) the magnitude of the applied vertical force. The result of running this program on an arch with $S = 2$, $H = 3$, $E = A_0 = 1$ is shown in Figure 8.15.

A plot such as that of Figure 8.15 is a bit confusing because all configurations are in the same cell, which appears on the screen as a flash. More instructive is the use of animations. Mathematica animations are much like movies or videos: they are a sequence of pictures that, when displayed in rapid succession, appear to move. In Mathematica versions endowed with a Notebook interface, any sequence of graphic cells can

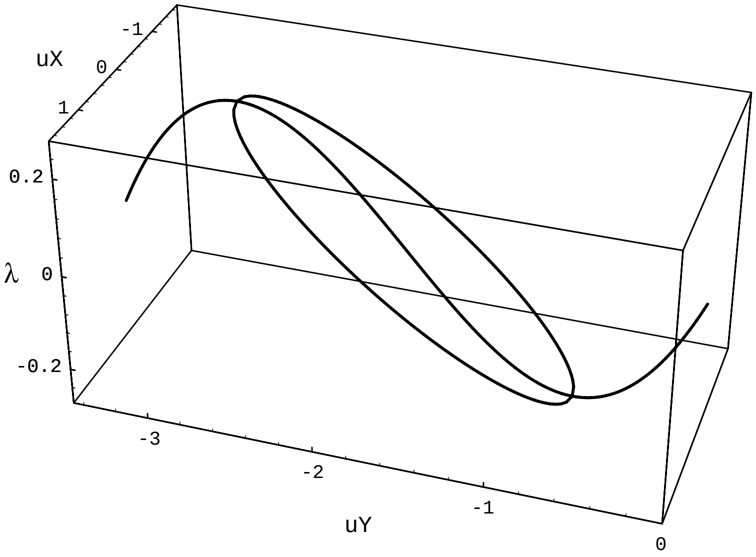


Figure 8.13. The arch equilibrium paths for $\alpha = 60^\circ$, plotted in the three-dimensional state-control space (u_X, u_Y, λ).

be animated by simply doubly clicking the mouse on one of the pictures (any one will do). The cell group containing the pictures will be selected and the animation started automatically.

The program shown in Figure 8.16 is a variant of the program of Figure 8.14. The logic is rearranged to produce a sequence of graphic cells, one per configuration, and so make the animation possible. Unfortunately the result cannot be shown on the old-fashioned medium of these Notes. But it should be possible in the next millennium.

§8.3.6. Forming the Tangent Stiffness Matrix

The tangent stiffness matrix derived in §8.2 is computed by the `Mathematica` module shown in Figure 8.17, along with some test statements. The arguments of this module, called `FormTangentStiff2DTwoNodeBar` are the same as those of the internal force subroutine shown in Figure 8.7. The module returns the 4 × 4 $\mathbf{K} = \mathbf{K}_M + \mathbf{K}_G$ matrix for the element. Figure 8.18 lists two modules: `MergeElemIntoMasterStiff` and `AssembleMasterStiffOfShallowArch`, which together assemble the 2 × 2 master stiffness matrix for the example arch structure.

The main use of the stiffness matrix in this and subsequent Chapters is the investigation of critical points, because solution argorithms based on **K** have not yet been described.

§8.3.7. Critical Point Study

A detailed symbolic analysis, not reported here, for arbitrary S, H, E, A_0, u_X and u_Y shows that under a vertical load the determinant of **K** can only vanish if $u_X = 0$. Thus it is enough to restrict consideration to symmetric motions. The stiffness matrix for $u_X = 0$ and its determinant are produced by the test statements shown in Figure 8.19 along with their output. It is seen that

```
ClearAll[S,H,Em,A0,uX,uY,a,c,d];  S=2.;H=3.;A0=1; Em=1;
nmax =20;   c= N[4*A0*Em/(4*H^2+S^2)^(3/2)];
plotelem1=Table[0,{nmax+1}];plotelem2=Table[0,{nmax+1}];
plotload =Table[0,{nmax+1}];plotarrow=Table[0,{nmax+1}];
Do [uY=-2.5*n*H/nmax;      d= N[-S^2/2-2*H*uY-uY^2];
    If[d <=0., uX=0;       lambda= N[2*c*uY*(H+uY)*(2*H+uY)],
               uX=Sqrt[d]; lambda= N[-c*S^2*(H+uY)]];
    plotelem1[[n+1]]=Graphics[Line[{{-S/2,0},{uX,H+uY}}]];
    plotelem2[[n+1]]=Graphics[Line[{{ uX,H+uY},{S/2,0}}]];
    plotload[[n+1]] =Graphics[Line[{{ uX,H+uY},{uX,H+uY-lambda}}]];
    a = Min[.5*Abs[lambda],0.2*H];
    If [lambda>0,    plotarrow[[n+1]]=Graphics[{
        Line[{{ uX,H+uY},{uX-a/2,H+uY-a}}],
        Line[{{ uX,H+uY},{uX+a/2,H+uY-a}}]} ],
                     plotarrow[[n+1]]=Graphics[{
        Line[{{ uX,H+uY},{uX-a/2,H+uY+a}}],
        Line[{{ uX,H+uY},{uX+a/2,H+uY+a}}]} ]  ],
        {n,0,nmax}];
Show[Graphics[Thickness[.004]],plotelem1,plotelem2,
      Graphics[Thickness[.005]],Graphics[RGBColor[1,0,1]],plotload,plotarrow,
      PlotRange->{{-2*H,2*H},{-2*H,2*H}},AspectRatio->1];
```

Figure 8.14. A `Mathematica` program to generate and plot configuration snapshots of the arch structure as it traverses the equilibrium paths.

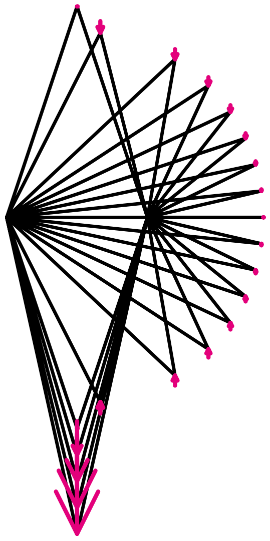


Figure 8.15. Results of running the program of Figure 8.14.

```
ClearAll[S,H,Em,A0,uX,uY,a,c,d];  S=2.;H=1.5;A0=1; Em=1;
nmax =20;   c= N[4*A0*Em/(4*H^2+S^2)^(3/2)];
Do [uY=-2.5*n*H/nmax;      d= N[-S^2/2-2*H*uY-uY^2];
    If[d <=0., uX=0;       lambda= N[2*c*uY*(H+uY)*(2*H+uY)],
               uX=Sqrt[d]; lambda= N[-c*S^2*(H+uY)]];
    plotelem1=Graphics[Line[{{-S/2,0},{uX,H+uY}}]];
    plotelem2=Graphics[Line[{{ uX,H+uY},{S/2,0}}]];
    plotload =Graphics[Line[{{ uX,H+uY},{uX,H+uY-lambda}}]];
    a = Min[.5*Abs[lambda],0.2*H];
    If [lambda>0,    plotarrow=Graphics[{
        Line[{{ uX,H+uY},{uX-a/2,H+uY-a}}],
        Line[{{ uX,H+uY},{uX+a/2,H+uY-a}}]} ],
                     plotarrow=Graphics[{
        Line[{{ uX,H+uY},{uX-a/2,H+uY+a}}],
        Line[{{ uX,H+uY},{uX+a/2,H+uY+a}}]} ]  ];
    Show[Graphics[Thickness[.004]],plotelem1,plotelem2,
      Graphics[Thickness[.005]],Graphics[RGBColor[1,0,1]],plotload,plotarrow,
      PlotRange->{{-2*H,2*H},{-2*H,2*H}},AspectRatio->1],
      {n,0,nmax}];
```

Figure 8.16. A variant of the program of Figure 8.14. This program generates a sequence of configuration plots (frames) that can be used for animation.

$$\mathbf{K}(S,H,E,A_0,u_X=0,u_Y)=\frac{8EA_0}{\sqrt{(4H^2+S^2)^3}}\begin{bmatrix} S^2/2+2Hu_Y+u_Y^2 & 0\\ 0 & 2H^2+6Hu_Y+3u_Y^2\end{bmatrix}, \tag{8.31}$$

Therefore

$$\det(\mathbf{K})=32E^2A_0^2(S^2+4Hu_Y+2u_Y^2)(2H^2+6Hu_Y+3u_Y^2)/\sqrt{(4H^2+S^2)^3}. \tag{8.32}$$

Because **K** is diagonal if $u_X=0$, $\det(\mathbf{K})$ factors out into two quadratic polynomials in u_Y. Consequently there are four critical points (C.P.s). The roots $-1\pm H/\sqrt{3}$ of $2H^2+6Hu_Y+3u_Y^2=0$ are always real and can be easily checked to correspond to limit points. The other two, which are roots of $S^2/2+2Hu_Y+u_Y^2=0$, correspond to bifurcation points and are real if $2H^2\geq S^2$, as already announced in §8.3.4.

The expression of the four roots of $\det(K)=0$ in terms of u_Y is symbolically obtained using the statements listed in Figure 8.19. These roots are substituted, again using the `/.` operator, into the equations $\lambda(u_Y)$ of the primary equilibrium path to get the four values of λ shown in the output cell of Figure 8.19:

$$\lambda_{L1}=-\lambda_{L2}=-\frac{16EA_0H^3}{3\sqrt{3}\sqrt{(4H^2+S^2)^3}},\qquad \lambda_{B1}=-\lambda_{B2}=-\frac{2\sqrt{2}\,EA_0}{\sqrt{(4H^2+S^2)^3}}\,S^2\sqrt{2H^2-S^2}. \tag{8.33}$$

The subscripts L and B stand here for limit and bifurcation point, respectively. That classification is readily done given the diagonal nature of **K** shown in (8.31).

Figure 8.20 shows statements used to produce plots of λ_{L1} and λ_{B1} versus H upon setting $E=A_0=1$ and $S=2$. Thus $H=\tan\alpha$. Negative λ (downward load) is plotted upwards for convenience. The plot

```
FormTangentStiff2DTwoNodeBar[XY1_,XY2_,uXY1_,uXY2_,Em_,A0_,s0_]:=
  Module[{X1,Y1,X2,Y2,X21,Y21,uX21,uY21,L0,L,e,s,ax,ay,Ke},
    {X1,Y1}=XY1; {X2,Y2}=XY2; X21=X2-X1; Y21=Y2-Y1;
    {uX1,uY1}=uXY1; {uX2,uY2}=uXY2; uX21=uX2-uX1; uY21=uY2-uY1;
    L0=Sqrt[X21^2+Y21^2];     L=Sqrt[(X21+uX21)^2+(Y21+uY21)^2];
    e=(L-L0)*(L+L0)/(2*L0^2); s=s0+Em*e;
    ax=(X21+uX21)/L0;         ay=(Y21+uY21)/L0;
    Ke=(Em*A0/L0)*{{ ax*ax, ax*ay,-ax*ax,-ax*ay},
                   { ax*ay, ay*ay,-ay*ax,-ay*ay},
                   {-ax*ax,-ay*ax, ax*ax, ay*ax},
                   {-ax*ay,-ay*ay, ay*ax, ay*ay}} +
      (A0*s/L0)*{{1,0,-1,0},{0,1,0,-1},{-1,0,1,0},{0,-1,0,1}};
      Return[Simplify[Ke]]
  ];

Ke=FormTangentStiff2DTwoNodeBar[{0,-S+H},{0,H},{0,0},{0,uY},Em,A0,0];
Ke=Simplify[Ke/.{Sqrt[S^2]->S,(1/Sqrt[S^2])->1/S}];
Print["Ke=",Ke//InputForm];
Ke=FormTangentStiff2DTwoNodeBar[{-4,0},{0,3},{0,0},{0,0}, Em,A0,s0];
Print["Ke=",Ke];
Print[Eigenvalues[Ke]];
```

```
Ke={{(A0*Em*uY*(2*S + uY))/(2*S^3), 0, -(A0*Em*uY*(2*S + uY))/(2*S^3),
    0}, {0, (A0*Em*(2*S^2 + 6*S*uY + 3*uY^2))/(2*S^3), 0,
    -(A0*Em*(2*S^2 + 6*S*uY + 3*uY^2))/(2*S^3)},
   {-(A0*Em*uY*(2*S + uY))/(2*S^3), 0, (A0*Em*uY*(2*S + uY))/(2*S^3),
    0}, {0, -(A0*Em*(2*S^2 + 6*S*uY + 3*uY^2))/(2*S^3), 0,
    (A0*Em*(2*S^2 + 6*S*uY + 3*uY^2))/(2*S^3)}}
       2 A0 s0  2 A0 (Em + s0)
{0, 0, -------, --------------}
          5           5
```

Figure 8.17. A module that generates the tangent stiffness matrix **K** of an individual bar element, along with test statements and output.

is shown in Figure 8.21 upon some labeling and beautification via `Adobe Illustrator`. The coalescence of the bifurcation and limit points for $H = \sqrt{3}$ or $\alpha = 60^\circ$ is clear in the plot. This figure also shows that for fixed S, E and A_0 the maximum vertical load carrying capacity for this structure is obtained for that rise angle. This optimality criterion (coalescence of critical points gives the strongest structure) is typical of those encountered in optimal design with stability constraints.

```
MergeElemIntoMasterStiff[Ke_,eftab_,Km_]:=
  Module[{i,j,ii,jj,neldof,K}, K=Km;
       neldof=Length[eftab];
       For[i=1, i<=neldof, i++, ii=eftab[[i]];
         For[j=i, j<=neldof, j++, jj=eftab[[j]];
           If [ii>0 && jj>0,
           K[[jj,ii]]=K[[ii,jj]]+=Ke[[i,j]]]
            ]
          ]; Return[K]
        ];

AssembleMasterStiffOfShallowArch[S_,H_,Em_,A0_,uX_,uY_]:=
  Module[{K1,K2,K},
    K=Table[0,{2},{2}];
    K1=FormTangentStiff2DTwoNodeBar[{-S/2,0},{0,H},
             {0,0},{uX,uY},Em,A0,0]; (*Print[K1//TableForm];*)
    K=MergeElemIntoMasterStiff[K1,{0,0,1,2},K];
    K2=FormTangentStiff2DTwoNodeBar[{0,H},{S/2,0},
             {uX,uY},{0,0},Em,A0,0]; (*Print[K2//TableForm];*)
    K=MergeElemIntoMasterStiff[K2,{1,2,0,0},K];
    Return[Simplify[K]]
        ];

ClearAll[S,H,Em,A0,uX,uY];
K=AssembleMasterStiffOfShallowArch[S,H,Em,A0,uX,uY];
Print["Master stiffness matrix = ",K//InputForm];
```

```
Master stiffness matrix = {{(4*A0*Em*(S^2 + 4*H*uY + 2*uY^2))/
     (4*H^2 + S^2)^(3/2), 0},
   {0, (8*A0*Em*(2*H^2 + 6*H*uY + 3*uY^2))/(4*H^2 + S^2)^(3/2)}}
```

Figure 8.18. Two modules that assemble the tangent stiffness matrix **K** for the arch structure of Figure 8.6.

```
ClearAll[H,S,Em,A0,uY];
detK=(32*A0^2*Em^2*(S^2 + 4*H*uY + 2*uY^2)*
     (2*H^2 + 6*H*uY + 3*uY^2))/(4*H^2 + S^2)^3;
sol=Simplify[Solve[detK==0,uY]];
Print[sol//InputForm];
lambdaCP=(8*A0*Em*uY*(H + uY)*(2*H + uY))/(4*H^2 + S^2)^(3/2)/.sol;
lambdaCP=Simplify[Expand[lambdaCP]];
Print["lambdaCP=",lambdaCP//InputForm];
```

```
{{uY -> -((3 + 3^(1/2))*H)/3}, {uY -> ((-3 + 3^(1/2))*H)/3},
  {uY -> (-4*H - (16*H^2 - 8*S^2)^(1/2))/4},
  {uY -> (-4*H + (16*H^2 - 8*S^2)^(1/2))/4}}
lambdaCP={(16*A0*Em*H^3)/(3*3^(1/2)*(4*H^2 + S^2)^(3/2)),
   (-16*A0*Em*H^3)/(3*3^(1/2)*(4*H^2 + S^2)^(3/2)),
   (2*2^(1/2)*A0*Em*S^2*(2*H^2 - S^2)^(1/2))/(4*H^2 + S^2)^(3/2),
   (-2*2^(1/2)*A0*Em*S^2*(2*H^2 - S^2)^(1/2))/(4*H^2 + S^2)^(3/2)}
```

Figure 8.19. Obtaining the critical points as roots of the tangent stiffness determinant.

```
ClearAll[H,S,Em,A0,uY];
lambdaL=(16*A0*Em*H^3)/(3*3^(1/2)*(4*H^2 + S^2)^(3/2));
lambdaB=(2*2^(1/2)*A0*Em*S^2*(2*H^2 - S^2)^(1/2))/(4*H^2 + S^2)^(3/2);
set={Em->1,A0->1,S->2};
pL=Plot[lambdaL/.set,{H,0,5},DisplayFunction->Identity,
   AxesLabel->{"H","lambda at CPs"},
   PlotLabel->"lambdaCP vs H (S=2,E=A0=1)"];
pB=Plot[lambdaB/.set,{H,Sqrt[2.00001],5},DisplayFunction->Identity];
```

Figure 8.20. Program to produce the critical loads plots of next Figure.

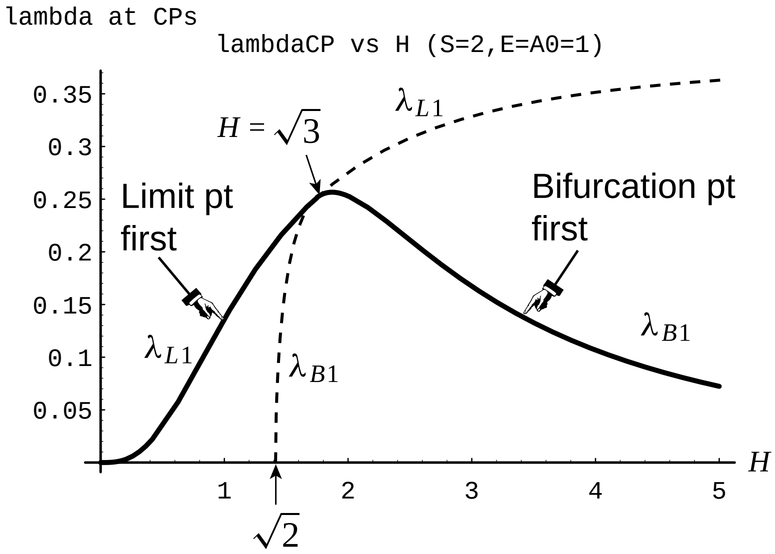


Figure 8.21. Plot of maximum load capacity as defined by λ (negative upwards) at the critical points for arches with different heights H ($S = 2$, $E = A_0 = 1$).

Homework Exercises for Chapter 8

The TL Bar Element

EXERCISE 8.1 [D:10] The $\alpha = 30^\circ$ (case 1) plot in Figure 8.11 shows a limit-point λ of about 0.048. Explain why that value could be different from that of the "exact solution" plot of Figure E6.3 for the same structure.

EXERCISE 8.2 [A/C:20] A problem in optimal design. For the 2-bar arch example structure, the plot in Figure 8.21 makes evident that, given the span S, modulus E and cross section area A_0, the largest vertical-downward-load capacity is obtained if the rise angle is $\alpha = 60^\circ$, or $H = (S/2)\tan\alpha = S\sqrt{3}/2$. Suppose that the design objective is to get the largest possible $\lambda_{max} = |\lambda|$ per unit volume of material, with respect to the total volume. Mathematically: maximize λ_{max}/V_0 with respect to V_0, where λ_{max} is the largest downward load sustained before failure by snap-through or bifurcation. What would be the best α?

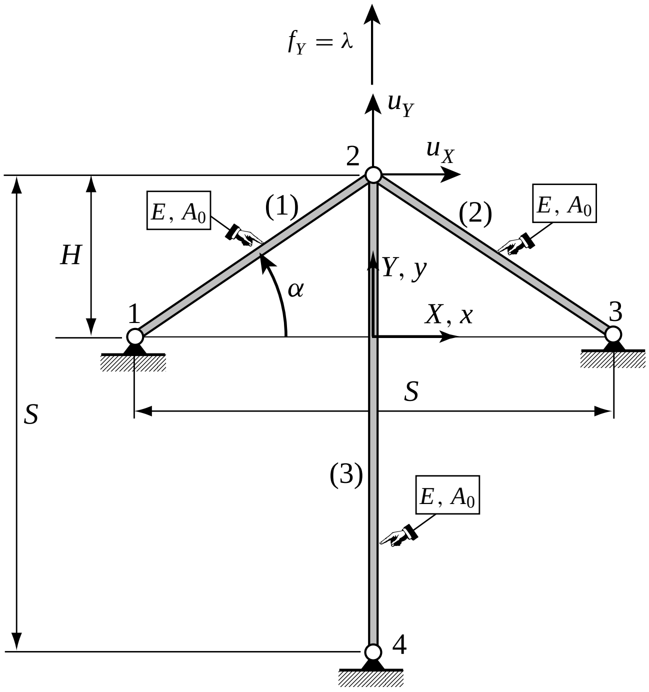
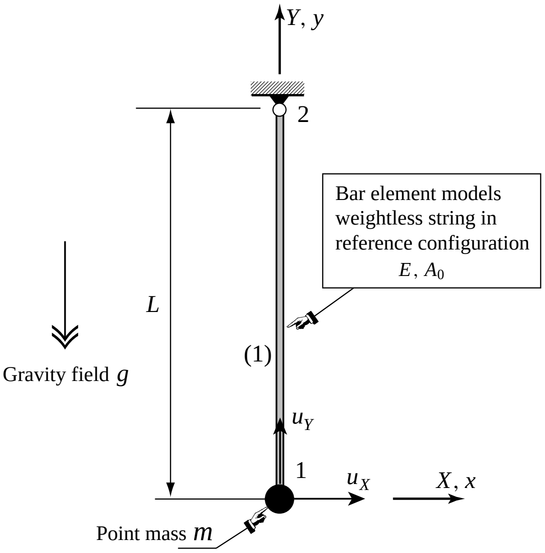


Figure E8.1. A 3-bar FEM model for Exercise 8.3.

EXERCISE 8.3 [A/C:25] You go to work as a nonlinear-FEM engineer for a car company. Your supervisor assigns you the job of designing a component of a wheel suspension system that can be modeled by the 3-bar structure depicted in Figure E8.1. The model has the dimensions and properties shown and is only subjected to vertical loads at node 2. The length S, bar section areas A_0 and elastic modulus E are known, but the rise angle $\alpha > 0$ is a design variable. Find the largest α for which bifurcation, which is bad for the wheel, cannot occur. (For the 2-bar arch example structure that maximum α was shown to be defined by $\tan\alpha \le \sqrt{2}/2$.)



Y, y
2
Bar element models weightless string in reference configuration E, A0
L
Gravity field g
(1)
u_Y
1
u_X
X, x
Point mass m

Figure E8.2. Model of a classic pendulum for Exercise 8.4.

EXERCISE 8.4 [A:C:20] Although this course focuses on statics, this exercise deals with the effect of the geometric stiffness on vibrations. Consider the pendulum configuration idealized in Figure E8.2. A lumped mass m is suspended by a weightless elastic string. The string is modeled as a 2-node bar element. This element is under a tensile prestress $s_0 = mg/A_0$, where g is the accelaration of gravity. The tangent stiffness matrix for the cable element in the reference configuration is $\mathbf{K} = \mathbf{K}_M + \mathbf{K}_G$, which is 2×2 upon removing the degrees of freedom at the fixed node 2. Because of the prestress the geometric stiffness does not vanish. The order-2 vibration eigenproblem is

$$\mathbf{K}\mathbf{z}_i = \omega_i^2 \mathbf{M}\mathbf{z}_i, \qquad i = 1, 2 \tag{E8.1}$$

where i is the mode index, ω_i is the i^{th} circular frequency in radians per second, $\mathbf{z}_i$ the associated eigenvector that include the horizontal and vertical displacements of node 1, and the mass matrix is

$$\mathbf{M} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \tag{E8.2}$$

Compute the two frequencies ω_1 and ω_2. One of them, say ω_1, describes pendulum motions while the other one pertains to a “bar mode” associated with axial motions. Discuss what happens to ω_1 and ω_2 if $E \to \infty$, which characterizes the “inextensional string” limit, and whether the classical pendulum small-oscillations frequency $\omega_P = \sqrt{g/L}$ is correct.

EXERCISE 8.5 (Requires knowledge of continuum mechanics.) [A:15] Suppose that the bar-element material is linear isotropic, with elastic modulus E and ν is Poisson’s ratio ν. Find the relation between the true (Cauchy) axial stress $\sigma = \sigma_{xx}$ in the bar and the PK2 axial stress $s = s_{XX}$. Hint: study the change in cross section areas.