2.2 Modalanalyse

Lösungen

Aufgabe 1

Das folgende GNU Octave-Skript berechnet die Eigenschwingungen unter Verwendung einer konsistenten Massenmatrix. Für eine Berechnung mit einer konzentrierten Massenmatrix muss der Wert der Variablen **mtype** geändert werden.

```
# Übungsblatt 2.2, Aufgabe 1: Balkenschwingungen
#
# ----------------------------------------------------------------

  file = mfilename();
  fid  = fopen([file, ".res"], "wt");

%  mtype = "lumped";        % konzentrierte Massenmatrix
  mtype = "consistent";     % konsistente Massenmatrix

# Daten (N, mm):

  L =      1000;
  A =       500;
  I =     10400;
  E =    210000;
  ny =      0.3;
  rho = 7.85E-9;

  nofmod = 8;               % Anzahl Eigenschwingungen
  nel    = [ 10, 15, 20];   % Anzahl Elemente

# Analytische Lösung

  f1 = pi * sqrt(E * I / (rho * A)) /(2 * L^2);
  fa = f1 * (1 : nofmod).^2;

# Lösung mit FEM

  geom.A = A;
  geom.I = I;

  mat.type = "iso";
  mat.E    = E;
  mat.ny   = ny;
  mat.rho  = rho;

  for k = 1 : length(nel)
```

```
      clear model; clear cmp;
      clear nodes; clear elem; clear hinge;

      model.type    = "solid";
      model.subtype = "2d";

      idl = 1 + nel(k);
      nodes(1).id = 1;   nodes(1).coor = [0, 0];
      nodes(2).id = idl; nodes(2).coor = [L, 0];
      [nodes, elem] = mfs_line(nodes, 1, idl, 2 : nel(k),
                               1 : nel(k), "b2", geom, mat);

      model.nodes    = nodes;
      model.elements = elem;

      hinge(1).id = 1;   hinge(1).dofs = [1, 2];
      hinge(2).id = idl; hinge(2).dofs = [1, 2];

      model.constraints.prescribed = hinge;

      cmp = mfs_new(fid, model);
      cmp = mfs_stiff(cmp);
      cmp = mfs_mass(cmp, mtype);
      cmp = mfs_freevib(cmp, nofmod);
      f(k, :) = mfs_getresp(cmp, "modes", "freq");
      mfs_print(fid, cmp, "modes", "freq");

  end

# Darstellung der Eigenschwingungen

  fgh = figure(1, "position", [100, 100, 800, 600],
               "paperposition", [0, 0, 15, 14]);
  nrow = ceil(nofmod / 2);
  for m = 1 : nofmod
      subplot(nrow, 2, m);
      mfs_plot(cmp, "deform", 2, "modes", m,
               "figure", fgh, "fontname", "Arial",
               "fontsize", 12);
  end

  print([file, ".jpg"], "-djpg");

# Ausgabe

  fprintf(fid, "Comparison of frequencies obtained");
  fprintf(fid, " with %s mass matrix:\n\n", mtype);

  fprintf(fid, "  Mode   Analytic    n = %2.0d    ",
               nel(1));
  fprintf(fid, "n = %2.0d    n = %2.0d\n",
               nel(2), nel(3));
  fprintf(fid, "  -------------------------------");
  fprintf(fid, "-----------------\n")
```

```
for m =1 : nofmod
    fprintf(fid, "     %2.0d    %8.3f   %8.3f   %8.3f   ",
             m, fa(m), f(1: 2, m));
    fprintf(fid, "%8.3f Hz\n", f(3, m));
end

fclose(fid);
```

Abbildung 1.1 zeigt die Eigenschwingungen bei einer Unterteilung des Balkens in 20 Elemente. Die ersten fünf Eigenschwingungen werden durch die Diskretisierung sehr gut erfasst.

Die Eigenfrequenzen sind in Tabelle 1.1 zusammengestellt. Bei Verwendung einer konsistenten Massenmatrix konvergieren die Eigenfrequenzen von oben, wie es aus der Eigenschaft des Rayleigh-Quotienten folgt. Bei Verwendung einer konzentrierten Massenmatrix erfolgt die Konvergenz von unten.

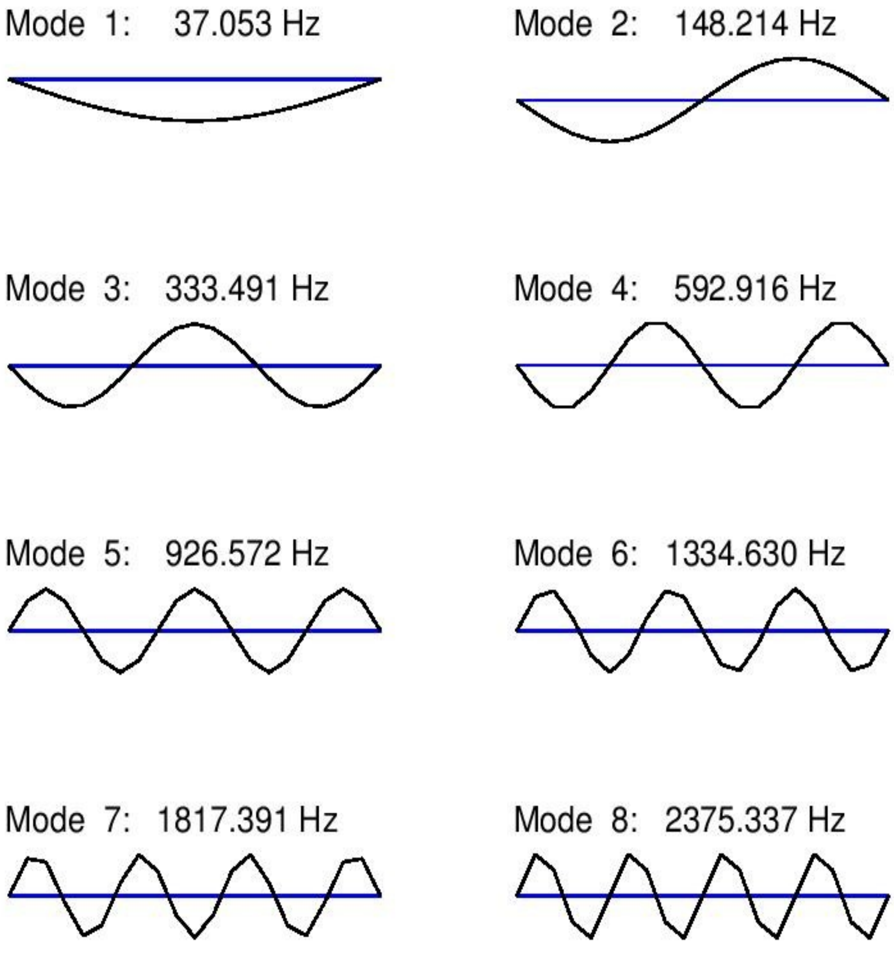

Abbildung 1.1: Eigenschwingungen

Nr.	analytisch	konzentrierte Massenmatrix			konsistente Massenmatrix		
		10	15	20	10	15	20
1	37,05	37,05	37,05	37,05	37,05	37,05	37,05
2	148,2	148,2	148,2	148,2	148,2	148,2	148,2
3	333,5	333,3	333,4	333,5	333,7	333,5	333,5
4	592,9	591,4	592,6	592,8	593,8	593,1	592,9
5	926,3	919,6	925,3	926,0	930,0	927,1	926,6
6	1334	1309	1331	1333	1345	1336	1335
7	1816	1738	1806	1813	1841	1821	1817
8	2371	2156	2348	2366	2426	2383	2375
	Hz	Hz	Hz	Hz	Hz	Hz	Hz

Tabelle 1.1: Vergleich der Eigenfrequenzen

Aufgabe 2

a) <u>Geometrie und Vernetzung</u>

Die Datei `u2_2_2.geo` beschreibt die Geometrie des Tragflügels. Die Vernetzung kann entweder interaktiv in Gmsh oder durch das Kommando

```
gmsh u2_2_2.geo -1
```

erzeugt werden.

Die Datei hat den folgenden Inhalt:

```
/* -------------------------------------------------------------------
   Übungsblatt 2.2, Aufgabe 2: Gepfeilter Flügel
                               Einheiten: mm
------------------------------------------------------------------ */

// Parameter

   L  =    30;          // Pfeilungswinkel in Grad
   b  = 10000;          // Flügellänge
   cr =  4000;          // Flügeltiefe an der Flügelwurzel
   ct =  2000;          // Flügeltiefe an der Flügelspitze
   n  =     5;          // Anzahl der Sektionen

   el =   500;          // Elementlänge
```

```
   sweep = Tan(L * Pi / 180);

// Punkte

   xr1  = -0.5 * cr;
   xr2  =  0.5 * cr;
   xtm  =  b * sweep;
   xt1 =  xtm - 0.5 * ct;
   xt2  = xtm + 0.5 * ct;

   m1 = (xt1 - xr1) / b;
   m2 = (xt2 - xr2) / b;

   dy   = b / n;

   For k In {0 : n}
       y  = k * dy;
       x1 = xr1 + m1 * y;
       x2 = xr2 + m2 * y;
       Point(newp) = {x1, y, 0, el};
       Point(newp) = {x2, y, 0, el};
   EndFor

// Einspannung: Punkte 1 und 2

   Physical Point("Constraints") = {1, 2};

// Vorderer Holm

   p1 = 1;
   For k In {1 : n}
       p2 = p1 + 2;
       Line(newl) = {p1, p2};
       p1 = p2;
   EndFor

   Physical Line("Front_Spar") = {1 : newl-1};

// Hinterer Holm

   l1 = newl;
   p1 = 2;
   For k In {1 : n}
       p2 = p1 + 2;
       Line(newl) = {p1, p2};
       p1 = p2;
   EndFor

   Physical Line("Rear_Spar") = {l1 : newl-1};

// Rippen

   l1 = newl;
   p1 = 1;
```

```
   p2 = 2;
   For k In {0 : n}
       Line(newl) = {p1, p2};
       p1 += 2;
       p2 += 2;
   EndFor

   Physical Line("Ribs") = {l1 : newl-1};
```

b) Eigenschwingungen

Das folgende GNU Octave-Skript berechnet die Eigenschwingungen:

```
# Übungsblatt 2.2, Aufgabe 2: Schwingungen eines gepfeilten
#                             Flügels
#
# ------------------------------------------------------------------

  file = mfilename();
  fid  = fopen([file, ".res"], "wt");

# Anzahl der Eigenschwingungen

  nmodes = 5;

# Definition des Modells
# ---------------------

# Materialdaten

  mat.type =   "iso";
  mat.E    =    70E3;
  mat.ny   =    0.34;
  mat.rho  = 2.70E-9;

# Modelltyp

  data.type    = "solid";
  data.subtype = "3d";

# Vorderer Holm

  b = 200;
  h = 200;
  t =   2;
  s =   2;

  spar.type    = "elements";
  spar.name    = "b2";
  spar.geom    = mfs_beamsection("I", b, h, t, s);
  spar.geom.v = [0, 0, 1];
  spar.mat     = mat;

  data.Front_Spar = spar;
```

```
# Hinterer Holm

  data.Rear_Spar = spar;

# Rippen

  b = 100;
  h = 200;
  t =   2;
  s =   2;

  spar.geom   = mfs_beamsection("I", b, h, t, s);
  spar.geom.v = [0, 0, 1];

  data.Ribs = spar;

# Einspannung

  data.Constraints.type = "constraints";
  data.Constraints.name = "prescribed";
  data.Constraints.dofs = 1 : 6;

# Rechnung
# --------

# Modell importieren und übersetzen

  model = mfs_import(fid, [file, ".msh"], "msh", data);

# Komponente erzeugen und exportieren

  wing = mfs_new(fid, model);
  mfs_export("axes.msh", "msh", wing, "mesh", "axes");

# Steifigkeits- und Massenmatrix

  wing = mfs_stiff(wing);
  wing = mfs_mass(wing);

# Massenkennwerte

  mfs_massproperties(fid, wing);

# Eigenschwingungen

  wing = mfs_freevib(wing, nmodes);
  mfs_print(fid, wing, "modes", "freq");
  mfs_export([file, ".pos"], "msh", wing, "modes", "disp");

  fclose(fid);
```

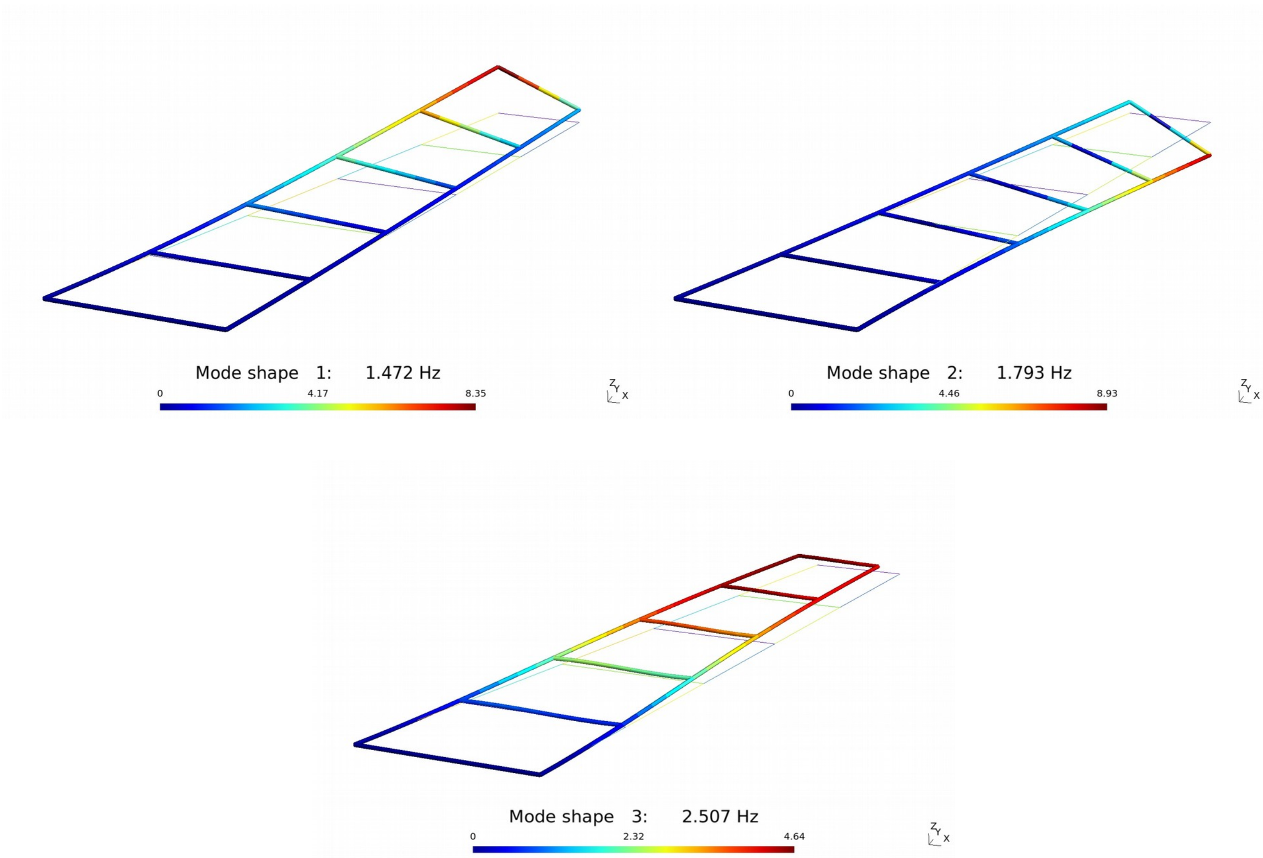

Abbildung 2.1: Darstellung einiger Eigenschwingungen

Die Ausgabedatei enthält die folgenden Eigenfrequenzen:

```
Natural frequencies:

  Mode   Circ. Frequency    Frequency
  ---------------------------------------
     1          9.24790       1.47185 Hz
     2         11.26559       1.79297 Hz
     3         15.75270       2.50712 Hz
     4         38.31474       6.09798 Hz
     5         42.56799       6.77491 Hz
```

Die ersten drei Eigenschwingungen sind in Abbildung 2.1 dargestellt. Die erste Schwingung ist die 1. vertikale Biegeschwingung, der eine Torsion überlagert ist. Die zweite Schwingung ist die 1. Torsionsschwingung. Bei der dritten Schwingung handelt es sich um die 1. horizontale Biegeschwingung.