MATLAB Toolbox for Two-Dimensional Finite Element Analysis of Truss Structures

Gerald Recktenwald

November 29, 2001

Introduction

The MATLAB codes described here are computational toys. They are intended to be used as educational devices for undergraduate courses in numerical analysis/numerical methods, stress analysis, and design of structures.

Theory

To do:

1. define geometry of one element
2. stiffness matrix for one element
3. global stiffness matrix (sparse)
4. constraints
5. loads

Toolbox m-files

The functions in the truss FEA toolbox are listed in Table 1. In a typical application, only the `frame2d`, `meshFrame`, and `spyFrame` functions are invoked by the user. The remaining functions are modules called by these main programs. Note that several of these m-files contain multiple (private) functions. This is a feature of MATLAB 5.x. I do not expect these files to work with MATLAB 4.x.

The toolbox also contains several `*.msh` files that define sample meshes with constraints and loads. Use these as demonstrations. Try, for example,

```
>> frame2d('tower.msh')
>> spyFrame('tower.msh')
>> frame2d('longFrame.msh')
>> frame2d('longFrame2.msh')
```

functions	Description
`frame2d`	Main program for computing displacement of the truss.
`constrain`	Apply constraints to global (unconstrained) stiffness matrix.
`drawMesh2d`	Draws the nodes and elements defining the mesh.
`meshFrame`	Main program to draw the nodes and elements of a mesh.
`parseMesh`	Reads a plain text file defining the mesh, and creates necessary data structures for the analysis.
`spyFrame`	Main program for creating `spy` plots of the unconstrained and constrained stiffness matrices. A `spy` plot depicts the structure of a sparse matrix. Non-zero entries are indicated by dots. Zero entries are not shown. The `spyFrame` function also verifies that the constrained stiffness matrix is symmetric.
`stress`	Given original and displaced node locations, compute the stress in each element.
`stiffness`	Set up the unconstrained global stiffness matrix

Table 1: MATLAB functions in the truss toolbox.

Input File Format

A truss mesh is defined by data contained in a plain text file that consists of `*` commands ("star" commands), comment statements, and numerical data. The sample mesh files provided with the truss toolbox have a three character `msh` extension. Any extension (including none) can be used. In the following discussion the text file containing the mesh definition is refered to as a `msh` file.

Table 2 summarizes the purpose of the `*` commands. Figure 2 is the mesh definition file for the simple truss in Figure 1.

Command	Required	Purpose
`%...`	no	Provide comment statment in the `msh` file
`*description`	no	Supply a text string to describe the analysis
`*nodelist`	yes	Define global nodes with three columns of numbers. First column is global node number: a sequential list of numbers must be supplied. Second and third columns are x and y positions of the node
`*ellist`	yes	Identify the global nodes that define each element, and indicate the material used in the element. The material is specified as an ID number. The `*materials` command defines a database of material properties.
`*constraints`	yes	Apply constraints at the nodes
`*loads`	yes	Apply loads at the nodes
`*materials`	yes	Define types of materials used in element definitions. At least one material must be defined.

Table 2: `*` commands used to define a two-dimensional truss mesh.

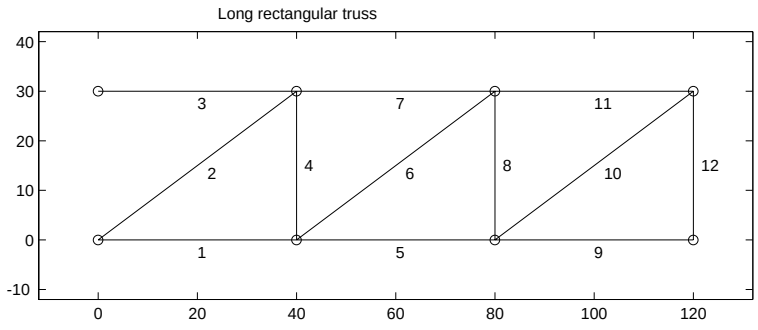

Figure 1: A simple two-dimensional truss.

```
% File:  longTruss.mesh
%
% Define a two-dimensional truss for simple Finite Element Analysis
% Truss consists of three repeating frames
*description
Long rectangular truss
*nodelist
 1     0   0
 2     0  30
 3    40   0
 4    40  30
 5    80   0
 6    80  30
 7   120   0
 8   120  30
*ellist
 1     1   3   1
 2     1   4   1
 3     2   4   1
 4     3   4   1
 5     3   5   1
 6     3   6   1
 7     4   6   1
 8     5   6   1
 9     5   7   1
10     5   8   1
11     6   8   1
12     7   8   1
*constraints
 1     1      0
 1     2      0
 2     1      0
 2     2      0
*loads
7       1        25000
8       1        25000
*materials
1  29.5e6  1.0   1.0
```

Figure 2: Listing of the `longFrame.msh` file used to define a simple two-dimensional truss mesh.

```
% File:  tower.mesh
%
% 2D finite element truss model of an electrical transmission tower
%
*description
Electrical Transmission Tower
%  The three node list arguments are
%    node number, x-position, and y-position
*nodelist
 1    10   0
 2    50   0
 3    18  20
 4    42  20
 5     0  40
 6    20  40
 7    40  40
 8    60  40
 9    20  55
10    40  55
%  The four element list arguments are
%    element number, global node 1,  global node 2, and material ID number
*ellist
 1      1   2   1
 2      1   3   1
 3      1   4   1
 4      2   4   1
 5      3   4   1
 6      3   6   1
 7      4   6   1
 8      4   7   1
 9      5   6   1
10      5   9   1
11      6   9   1
12      6   7   1
13      6  10   1
14      9  10   1
15      7  10   1
16      7   8   1
17      8  10   1
%  The three *constraint arguments are
%    node number, local DOF for constraint,  constraint value
*constraints
 1     1      0
 1     2      0
 2     2      0
%  The three *load arguments are
%    node number, local DOF for load,  load value
*loads
5       1        3713.9
5       2       -9284.8
8       2      -10000
%  The four *material arguments are
%    material ID number, Young's modulus, cross-sectional area,  density
*materials
 1    29.5E6  8   1
```

Figure 3: Listing of the `tower.msh` file used to define a two-dimensional model of an electrical transmission tower.

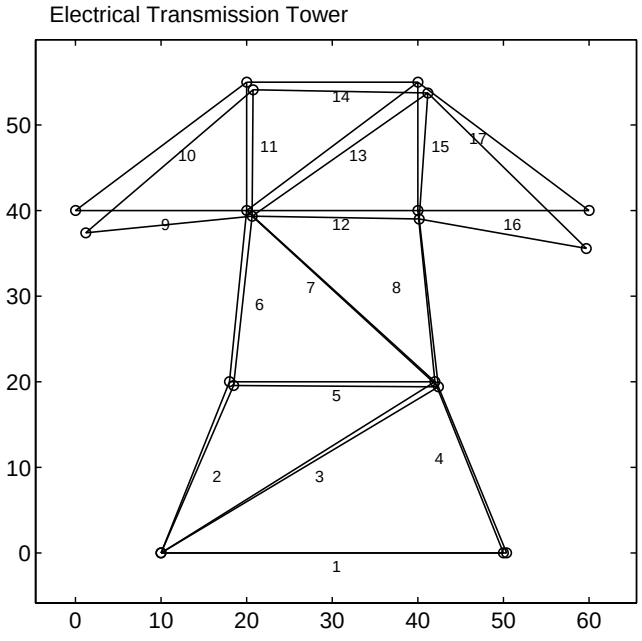

Figure 4: Deflection of a two dimensional model of an electrical transmission tower.