User Guide for Tensegrity Static Analysis Functions

@ Department of Mechanical, Industrial and Aerospace Engineering

(MIAE), Concordia University, Montreal, Canada

October, 2025

1. Overview

This MATLAB package performs static equilibrium analysis and visualization of tensegrity structures.

It allows users to:

- Compute member forces (bars in compression, strings in tension).
- Check pre-tensionability and determine feasible tension distributions.
- Compute reaction forces at fixed nodes.
- Visualize the tensegrity structure with applied loads and reactions.

The workflow consists of two main functions:

- tensegrity_statics.m – performs the equilibrium analysis.
- tensegrity_plot.m – visualizes the geometry, bars, strings, and forces.

2. Function Descriptions

2.1. **tensegrity_statics**

Purpose: Performs static equilibrium analysis of a tensegrity structure with given node geometry, connectivity, and external forces.
The function automatically detects if the system is statically determinate or under determined, and checks if it is pre-tensionable.

[c_bars, t_strings, V] = tensegrity_statics(b, s, q, p, dim, Q, P, C, U, extra_constraints)

Inputs Variable	**Description**
b	Number of bars (compression members)
s	Number of strings (tension members).
q	Number of free nodes.
p	Number of fixed nodes.
dim	System dimension (2 or 3).
Q	(dim×q) matrix of free node coordinates
P	(dim×p) matrix of fixed node coordinates.
C	(m×n) connectivity matrix defining member-node relationships. Each row corresponds to one member, with 1 for start node and -1 for end node.
U	(dim×q) applied external forces at free nodes.
extra_constraints	(**optional**) Symbolic expressions for additional constraints (e.g., pulley). Default = "".
Outputs Variable	**Description**
c_bars	Vector of bar compression forces (negative if under tension).
t_strings	Vector of string tension forces (negative if under compression).
V	(dim × p) matrix of reaction forces at fixed nodes

Key Features

Automatically distinguishes determinate vs. Under determined systems using SVD.

For under determined systems:

- Checks pre-tensionability (all strings can sustain minimum tension tau_min).

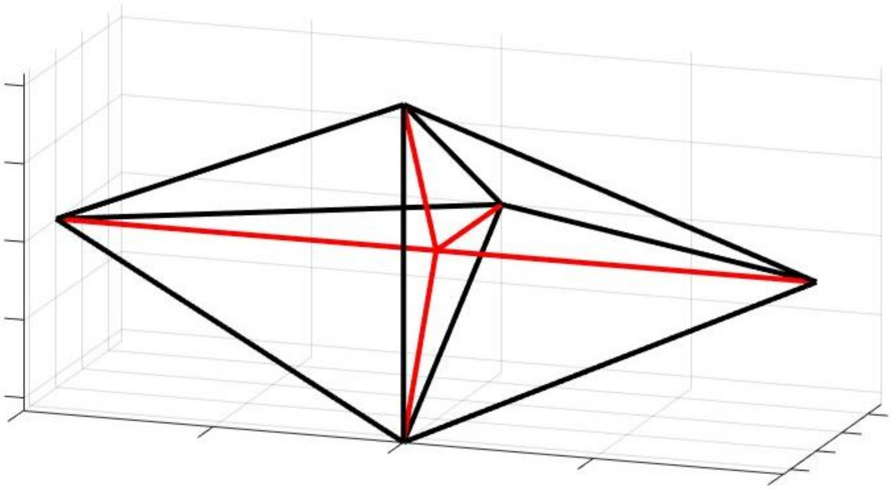
- Solves linear programming problem to minimize or maximize tension distribution.

Computes reaction forces at supports.

2.2. tensegrity_plot

Visualizes the tensegrity structure in 2D or 3D with:

- **Red lines = bars (compression)**
- **Blue/black lines = strings (tension)**
- **Magenta arrows = applied loads**
- **Blue arrows = reaction forces**



tensegrity_plot(Q, P, C, b, s, U, V, outer, fac, fac1)

Inputs Variable	Description
b	Number of bars (compression members)
s	Number of strings (tension members).
V	Reaction forces at fixed nodes.
outer	(optional) true to draw external arrows pointing outward.
fac	(optional) Force arrow scaling factor.
fac1	(optional) Arrow tip size factor.
Q	(dim×q) matrix of free node coordinates
P	(dim×p) matrix of fixed node coordinates.
C	(m×n) connectivity matrix defining member-node relationships. Each row corresponds to one member, with 1 for start node and -1 for end node.
U	(dim×q) applied external forces at free nodes.

Notes

- Works for both 2D and 3D tensegrity systems.
- Requires the mArrow3 function for 3D arrow visualization.

3. Example: Simple 3D Hexagonal Tensegrity

Input

```
clear; clf; figure(1);

% Node geometry (Q = free nodes, P = fixed nodes)
Q(:,1)=[0; 0; 0];
Q(:,2)=[1; 0; 0];
Q(:,3)=[2; 0; 0];
r=0.5;
Q(:,4)=[1; r*cos(0);         r*sin(0)];
Q(:,5)=[1; r*cos(2*pi/3); r*sin(2*pi/3)];
Q(:,6)=[1; r*cos(4*pi/3); r*sin(4*pi/3)];
P=[]; % No fixed nodes

[dim,q]=size(Q); p=size(P,2);

% Connectivity matrix
C(1,1)=1; C(1,2)=-1; % bars
C(2,2)=1; C(2,3)=-1;
C(3,2)=1; C(3,4)=-1;
C(4,2)=1; C(4,5)=-1;
C(5,2)=1; C(5,6)=-1; b=5;
C(b+1,1)=1; C(b+1,4)=-1; % strings
C(b+2,1)=1; C(b+2,5)=-1;
C(b+3,1)=1; C(b+3,6)=-1;
C(b+4,3)=1; C(b+4,4)=-1;
C(b+5,3)=1; C(b+5,5)=-1;
C(b+6,3)=1; C(b+6,6)=-1;
C(b+7,4)=1; C(b+7,5)=-1;
C(b+8,5)=1; C(b+8,6)=-1;
C(b+9,6)=1; C(b+9,4)=-1; s=9;

% Applied forces
U=zeros(dim,q);
U(1,1)=1;   % force on node 1 (positive x)
U(1,3)=-1; % force on node 3 (negative x)

% Static analysis
[c_bars,t_strings,V]=tensegrity_statics(b,s,q,p,dim,Q,P,C,U);

% Visualization
tensegrity_plot(Q,P,C,b,s,U,V,true,1,0.08);
grid on;
```

Output

```
mhat =     18
nhat =     14
r =     12

Warning: Ase is potentially inconsistent, implying the presence of soft modes,or instability!   More strings or fixed points should fix the problem.

Bar compressions and string tensions with loads as specified,
least squares solution (i.e., NO pretensioning):
u in column space of Ase, so at least one solution exists, with residual 8.0414e-16.

c_bars =   0.3143      0.3143     -0.0571     -0.0571     -0.0571
Note: some bars not under compression.   Maybe replace them with strings?

t_strings =    -0.2556      -0.2556      -0.2556      -0.2556      -0.2556
-0.2556      0.0990      0.0990      0.0990
Some strings not under tension. Needs different tensioning or external loads.

Ase is underdetermined with 2 DOF. Checking now to see if system is pretensionable,
with tension >= 0.1 in all tethers for zero applied load.

Result with external load ZERO, pretensioned with given tau_min
while minimizing the L1 norm of the tensions:

c_bars =     0.2683      0.2683      0.2626      0.2626      0.2626
No bars under tension.   Good.

t_strings =    0.1000      0.1000      0.1000      0.1000      0.1000
0.1000      0.1000      0.1000      0.1000
The 9 strings are all under tension with tau_min=0.1. Good.

Pretensionable!
Results with external forces u as specified and tensioned with given tau_min
while minimizing the L1 norm of the tensions.

c_bars =   1.2683      1.2683      0.2626      0.2626      0.2626
No bars under tension.   Good.
```

```
t_strings =     0.1000      0.1000      0.1000      0.1000      0.1000
0.1000      0.1000      0.1000      0.1000
The 9 strings are all under tension with tau_min=0.1. Good.
```

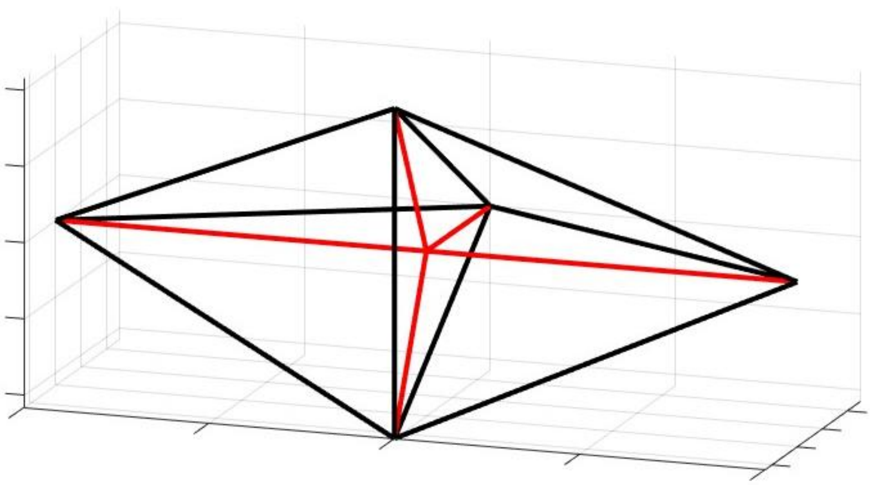
4. Output Interpretation

Command Window Messages: The solver prints information about:

- Rank and degrees of freedom of the system.
- Whether the system is pretensionable.
- Minimum tension in strings (tau_min).
- Any members under the wrong stress type (e.g., bars in tension).

Plot Display:

- Red members → compression bars
- Blue/black members → tension strings
- Magenta arrows → external forces
- Blue arrows → reactions at fixed nodes



5. Something you need to be careful

- Ensure that your connectivity matrix C is consistent with node numbering.
- For under determined systems, MATLAB's linprog function requires the Optimization Toolbox.
- The system may not be pretensionable if the geometry or loading is ill-conditioned, such as those structures from Class 2 to Class K.
- Always check printed warnings for "potential inconsistency" or "soft modes."