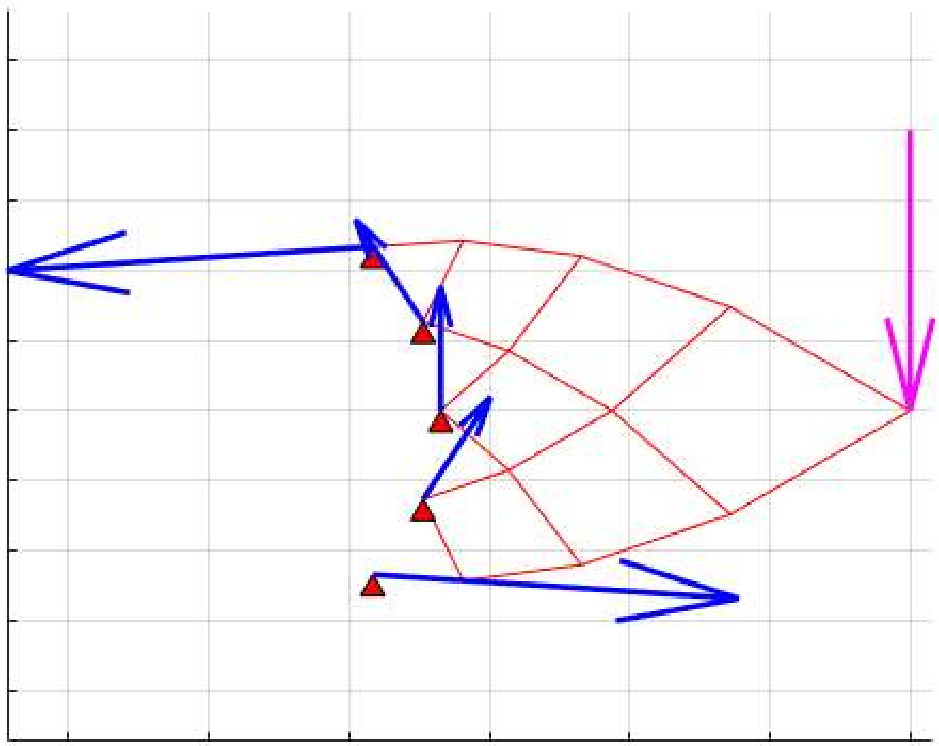
```
% Static force analysis of 2D MitchelTruss (Fig 3.3 of Skelton & de Oliviera 2009)
% By Thomas Bewley, UC San Diego (+ faculty fellow at JPL)
clear; clf; figure(1);

beta = pi/6;
phi = pi/16;

r0 = 1;
r1 = r0*sin(beta)/(sin(beta+phi));
r2 = r1*sin(beta)/(sin(beta+phi));
r3 = r2*sin(beta)/(sin(beta+phi));
r4 = r3*sin(beta)/(sin(beta+phi));

% Free [Q=Q_(dim x q)] and fixed [Q=Q_(dim x q)] node locations
Q(:,1)=[r0; 0];
Q(:,2)=[r1*cos(phi); r1*sin(phi)];
Q(:,3)=[r1*cos(phi); -r1*sin(phi)];
Q(:,4)=[r2*cos(phi*2); r2*sin(phi*2)];
Q(:,5)=[r2,0];
Q(:,6)=[r2*cos(phi*2); -r2*sin(phi*2)];
Q(:,7)=[r3*cos(phi*3); r3*sin(phi*3)];
Q(:,8)=[r3*cos(phi); r3*sin(phi)];
Q(:,9)=[r3*cos(phi); -r3*sin(phi)];
Q(:,10)=[r3*cos(phi*3); -r3*sin(phi*3)];
P(:,1)=[r4*cos(phi*4); r4*sin(phi*4)];
P(:,2)=[r4*cos(phi*2); r4*sin(phi*2)];
P(:,3)=[r4; 0];
P(:,4)=[r4*cos(phi*2); -r4*sin(phi*2)];
P(:,5)=[r4*cos(phi*4); -r4*sin(phi*4)];
[dim,q]=size(Q), p=size(P,2), n=q+p;


% Connectivity matrix
b = 20;
C(:,1)= conVect(1,2,15);
C(:,2)= conVect(1,3,15);
C(:,3)= conVect(2,5,15);
C(:,4)= conVect(3,5,15);
C(:,5)= conVect(2,4,15);
C(:,6)= conVect(3,6,15);
C(:,7)= conVect(4,7,15);
C(:,8)= conVect(4,8,15);
C(:,9)= conVect(5,8,15);
C(:,10)= conVect(5,9,15);
C(:,11)= conVect(6,9,15);
C(:,12)= conVect(6,10,15);
C(:,13)= conVect(7,11,15);
C(:,14)= conVect(7,12,15);
C(:,15)= conVect(8,12,15);
C(:,16)= conVect(8,13,15);
C(:,17)= conVect(9,13,15);
C(:,18)= conVect(9,14,15);
C(:,19)= conVect(10,14,15);
C(:,20)= conVect(10,15,15); s = 0; m=b+s;
C=C';
```

```
% Applied external force U=U_(dim x q)
U(1:dim,1:q)=0; U(2,1)=-10;

% Solve for the forces at equilibrium, and plot
[c_bars,t_strings,V]=tensegrity_statics(b,s,q,p,dim,Q,P,C,U);
tensegrity_plot(Q,P,C,b,s,U,V,true,2.0); grid on;

function C_vect = conVect(start,finish,total_nodes)
C_vect = zeros(1,total_nodes);
C_vect(1,start) = -1;
C_vect(1,finish) = 1;
return
end
% end script MitchelTruss4
```

```
dim =

     2


q =

    10


p =

     5


mhat =

    20


nhat =

    20


r =

    20

Ase is not potentially inconsistent. Good.

Bar compressions and string tensions with loads as specified,
least squares solution (i.e., NO pretensioning):

c_bars =

  Columns 1 through 7

  -10.0000   10.0000    2.2527   -2.2527  -10.9342   10.9342  -11.9557
```

```
  Columns 8 through 14

    2.4632   -2.9706    2.9706   -2.4632   11.9557  -13.0726    2.6933

  Columns 15 through 20

   -3.8030    3.3625   -3.3625    3.8030   -2.6933   13.0726

Note: some bars not under compression.  Maybe replace them with strings?

t_strings =

  1×0 empty double row vector

Some strings not under tension. Needs different tensioning or external loads.

Ase is not underdetermined (thus, it is not tensionable). The above solution is unique.
```



Published with MATLAB® R2019b