function as5truss2d
%=================================================================================================
% W. Schleter
% EF 102 Section NC
% 2/27/03
% Program Description: Build upon the given program which analyzes any stable,
% statically determinant, two-dimensional truss.
% % Input: the joint numbers and their x and y coordinate values,
% the member numbers and which joints they connect,
% the joint locations and angle of application of the three reaction forces,
% the magnitude, angle, and joint number location of all applied loads
% Output: calculated values for the member and reaction forces
% %
% Key Variables (in order of appearance):
% w: weight of person
% jts: joint data, 1st col = joint number, 2nd col = x location, 3rd col = y location
% mem: member data, 1st col = member number, 2nd col = 1st joint it connects,
% 3rd col = 2nd joint it connects
% loads: load data, 1st col = joint number, 2nd col = magnitude, 3rd col = angle (deg)
% react: reaction data, 1st col = joint number, 2nd col = angle (deg)
% A: the coefficient matrix
% B: the load vector
% U: the calculated member and reaction forces
%=================================================================================================
% Clear all memory and the command window.
clear all;
close all;
clear;
% define weight
w = 135/2;
% Get the joint (jts), member (mem), reaction (react), and load (loads) data.
[jts,mem,react,loads] = as5getdata(w);
% Initialize the coefficient matrix (A) and the load vector (B).
% NOTE: changed the as5initialize function to create multicolumn B matrix
[A,B] = as5initialize(mem,loads);
% Enter all the non-zero terms of the coefficient matrix due to the member forces.
A = as5addmembers(A,jts,mem);
% Enter all the non-zero terms of the coefficient matrix due to the reaction forces.
A = as5addreactions(A,mem,react);
% Enter all the non-zero terms of the load vector due to the applied loads.
% NOTE: the original as5addloads was modified to add each load to different
% column of the B matrix
B = as5addloads(B,loads);
% Solve for all the unknown member and reaction forces.
U = A \ (-1.0.*B);
% calculate and output results
writeout(U,w,mem,jts);
% extra credit
plottruss(jts,mem);
% extra extra
showwebpage;
% End of main program.
%=================================================================================================

function [jts,mem,react,loads] = as5getdata(w)
% Purpose: To get the joint, member, reaction, and load data.
% Input: None.
% Output:
% jts: joint data, 1st col = joint number, 2nd col = x location, 3rd col = y location
% mem: member data, 1st col = member number, 2nd col = 1st joint it connects,
% 3rd col = 2nd joint it connects
% loads: load data, 1st col = joint number, 2nd col = magnitude, 3rd col = angle (deg)
% react: reaction data, 1st col = joint number, 2nd col = angle (deg)
% jts = [1 0 0
2 1 0
3 2 0
4 3 0
5 4 0
6 5 0
7 6 0
8 7 0
9 6 1.25
10 5 2.5
11 4 2.5
12 3 2.5
13 2 2.5
14 1 2.5
15 0 2.5 ];
mem = [1 1 2
2 2 3
3 3 4
4 4 5
5 5 6
6 6 7
7 7 8
8 8 9
9 9 10
10 10 11
11 11 12
12 12 13
13 13 14
14 14 15
15 15 1
16 2 15
17 2 14
18 2 13
19 3 13];
react = [1 0 ; 1 90 ; 8 90];
loads = [2 w 270
4 w 270
5 w 270
6 w 270
7 w 270];

function [A,B] = as5initialize(mem,loads)
% Purpose: To initialize the coefficient matrix and the load vector.
% Input:
%   mem: the matrix of member data
%   loads: the matrix of load data
% Output:
%   A: the initialized coefficient matrix (i.e., all terms = 0)
%   B:   the initialized load vector (i.e., all terms = 0)
neqn = size(mem,1) + 3;  % Determine the number of equations = number of unknown
%   member forces + the 3 unknown reaction forces.

nloads = size(loads,1);  % number of loads
A = zeros(neqn,neqn);    % Initialize the coefficient matrix.
B = zeros(neqn,nloads);  % Initialize the load vector.

function A = as5addmembers(A,jts,mem)
% Purpose: To enter all the non-zero member force terms into the coefficient matrix.
% Input:
%   A:     the initialized coefficient matrix
%   jts:   joint data matrix
%   mem:   member data matrix
% Output:
%   A:  the updated coefficient matrix including all non-zero member force terms
for ii=1:1:size(mem,1) % Loop over all the members.
    mn = mem(ii,1); % Get the current member number.
    j1 = mem(ii,2); % Get the 1st joint number that the member connects.
    j2 = mem(ii,3); % Get the 2nd joint number that the member connects.
    ang = as5calang(jts,j1,j2); % Calculate the angle pointing from the 1st joint to
%   the 2nd joint.
    A = as5setcoef(A,mn,j1,ang); % Set the x & y coefficients of A associated with the
%   1st joint of the current member.
    ang2 = as5calang(jts,j2,j1); % Calculate the angle pointing from the 2nd joint to
%   the 1st joint of the current member.
    A = as5setcoef(A,mn,j2,ang2); % Set the x & y coefficients of A associated with the
%   2nd joint of the current member.
end

function ang = as5calang(jts,j1,j2)
% Purpose: Calculate the 4 quadrant angle (radians) of a line connecting two joints.
% Input:
%   jts:  joint data matrix
% Output:
%   ang:  the 4 quadrant angle (radians)

delta_x = jts(j2,2) - jts(j1,2);  % Calculate the difference in the x coordinates.
delta_y = jts(j2,3) - jts(j1,3);  % Calculate the difference in the y coordinates.
ang = atan2(delta_y,delta_x);     % Calculate the 4 quadrant angle (radians).

function A = as5setcoef(A,mn,jt,ang)
% Purpose: Set the x and y terms of the coefficient matrix for the current joint of the%   current member. This function will be called twice for each member.
% Input:
%   A:     the current copy of coefficient matrix
%   mn:    the member number
%   jt:    joint number
%   ang:   the 4 quadrant angle of the current member force at the current joint
% Output:
%   A:  the updated coefficient matrix

A(2.*jt-1,mn) = cos(ang);         % Set x term. Row location = (Current Joint No.)*2 - 1
A(2.*jt,mn)   = sin(ang);         % Set y term. Row location = (Current Joint No.)*2

function A = as5addreactions(A,mem,react)
% Purpose: To enter all the non-zero reaction force terms into the coefficient matrix.
% Input:
%   A:     the coefficient matrix
%   mem:   member data matrix
%   react: reaction data matrix
% Output:
%   A:  the updated coefficient matrix including all non-zero reaction force terms
for ii = 1:1:3 % Loop over all 3 reactions (there must be exactly 3).
    jnum = react(ii,1); % Get the joint number of the current reaction.
    ang = react(ii,2); % Get the angle of the current reaction.
    coln = size(mem,1) + ii; % Set the column number of the coefficient matrix
    A(:,coln) = zeros(size(A,1),1); % Set a column to zero.
    ang2 = as5calang(jts,j1,j2); % Calculate the angle pointing from the 1st joint to
%   the 2nd joint.
    A = as5setcoef(A,mn,j1,ang2); % Set the x & y coefficients of A associated with the
%   1st joint of the current member.
    ang2 = as5calang(jts,j2,j1); % Calculate the angle pointing from the 2nd joint to
%   the 1st joint of the current member.
    A = as5setcoef(A,mn,j2,ang2); % Set the x & y coefficients of A associated with the
%   2nd joint of the current member.
end
% always be in the last three columns.

A(2.*jnum-1,coln) = cos(ang.*pi./180); % Enter the x component of the reaction
% into the x equation row of the joint.
A(2.*jnum,coln)   = sin(ang.*pi./180); % Enter the y component of the reaction
% into the y equation row of the joint.
end
return

%========================================================================================
function B = as5addloads(B,loads)
% Purpose: To enter all the non-zero applied force terms into the load vector
% WRS MOD: enter each load in a separate column.
% Input:  B:     the initialized load vector (i.e., all terms = 0)
% loads: load data, 1st col = joint number, 2nd col = mag, 3rd col = angle (deg)
% Output: B:     the updated load vector including all non-zero applied load terms

for ii = 1:size(loads,1)                        % Loop over all the applied loads.
    jn  = loads(ii,1);                      % Set the joint number of the applied load.
    mag = loads(ii,2);                      % Set the magnitude of the applied load.
    ang = loads(ii,3);                      % Set the angle (deg) of the applied load.
    B(jn.*2-1,ii) = mag.*cos(ang.*pi./180);  % Enter the x component of the applied load
% into the x equation row of the joint.
    B(jn.*2,ii)   = mag.*sin(ang.*pi./180);  % Enter the y component of the applied load
% into the y equation row of the joint.
end
return
%========================================================================================

function writeout(U,w,mem,jts)
% Purpose: calculate and output results% Input:   U:     results matrix%          w:     weight applied at each joint%          ... details)%          jts:   joint coordinates (see getdata for details)% Output:  results to the screen and to diary file
% set up to log cmd window contents to a file as well as displaying them
delete('as5sol_cmdwin.txt');
diary('as5sol_cmdwin.txt');

num_m = size(U,1) - 3;  % don't look at reactions in last 3 rows
num_lc = size(U,2);
% round off values to one decimal place
% doing this avoids printing -0 and allows exact comparisons
i.e.  95.0 == 95.0 not 95.000 == 95.001
U = round(U,.01);/.01;

% output the weight used
fprintf('The weight used for all load cases was %.1f lbs.\n\n',w);

% output member forces table header
fprintf('Member forces in lbs - negative indicates compression\n\n');
head = 'M#   Jnt#s';
dashes = '----------';
fmt = '%2u  %2u-%2u';

for lc=1:num lc
    fprintf([fmt '  %8.1f'],mem(mn,2),mem(mn,3),U(mn,lc));    if U(mn,lc)>80
        fprintf('Member %2u-%2u failed under tension force of %.1f pounds\n',...mem(mn,2),mem(mn,3),U(mn,lc));            fail = 1;
    end
end

for lc=1:num lc
    fprintf([fmt '  %8.1f'],mem(mn,2),mem(mn,3),abs(U(mn,lc)));    if U(mn,lc)<-60
        fprintf('Member %2u-%2u failed under compression force of %.1f pounds\n',...mem(mn,2),mem(mn,3),abs(U(mn,lc)));
        fail = 1;
    end
end

if fail == 1
    fprintf('\nThe truss did not fail on this step.\n\n');
    end
end

% output overall max tension and max compression
max_t = max(max(U));
max_c = min(min(U));

% check all places to find locations of max and min
max_t_loc = '1';
max_c_loc = '1';

% output member forces table data
for nn=1:1:num_m
    fprintf([fmt '\n'],num,mem(nn,2),mem(nn,3),U(nn,lc));    if fail == 0
        fprintf('\n\nThe Matlab Man made it across!\n\n');
    end
    end

% delete('as5sol_cmdwin.txt');
diary('as5sol_cmdwin.txt');

if U(mn,lc) == max_t
    max_t_loc = [max_t_loc num2str(mn) ', ' num2str(lc+1) '   '];
eend
if U(mn,lc) == max_c
    max_c_loc = [max_c_loc num2str(mn) ', ' num2str(lc+1) '   '];
eend
fprintf('

');fprintf('Overall maximum tensile force of %.1f lbs',max_t);printf(' occurred at the following member,joint combinations:
');fprintf('   %s

',max_t_loc);printf('Overall maximum compressive force of %.1f lbs',abs(max_c));fprintf(' occurred at the following member,joint combinations:
');fprintf('   %s

',max_c_loc);
diary off;return
%========================================================================================
function plottruss(jts,mem)
% Purpose: calculate and output results
% Input:   U: results matrix
%          w: weight applied at each joint
%         ... (see getdata for details)
%          jts: joint coordinates (see getdata for details)
% Output:  results to the screen

num_jts = size(jts,1);
num_m = size(mem,1);

figure();clf reset;
axis equal;
axis([min(jts(:,2))-1,max(jts(:,2))+1,min(jts(:,3))-1,max(jts(:,3))+1]);hold on;
title('AS 5 Truss Geometry and Connectivity (WRS 2/27/03)');xlabel('X (ft)');ylabel('Y (ft)');
set(gca,'DefaultTextBackgroundColor','w','DefaultTextHorizontalAlignment','center','DefaultTextFontSize',8);
mmimg = imread('as5sol_mm.png');image([2 4],[0 5],flipdim(mmimg,1));

for mn=1:1:num_m
    x = [jts(mem(mn,2),2) jts(mem(mn,3),2)];
    y = [jts(mem(mn,2),3) jts(mem(mn,3),3)];
    plot(x,y,'b-');text(mean(x),mean(y),num2str(mn),'EdgeColor','b');end

% plot joint labels
for jn=1:1:num_jts
    x = jts(jn,2);
    y = jts(jn,3);
    text(x,y,num2str(jn));plot(x,y,'ro','MarkerSize',12);end
return
%========================================================================================