```
%=============================
%   PROGRAM topo.m
%=============================

close all;

%     ---------------------------
%     Mesh, NewNum and PSI DATA
%     ---------------------------
      load MESHo   -ASCII
      load NODES   -ASCII
      load NP      -ASCII
      load PSI     -ASCII

%     -------------------------------
%     Transfer data to variable names
%     -------------------------------
      NUMNP = MESHo(1);
      NUMEL = MESHo(2);
      NNPE  = MESHo(3);
      XMAX  = NODES(1,1);
      XMIN  = XMAX;
      YMAX  = NODES(1,2);
      YMIN  = YMAX;
      for I=1:NUMNP
          XORD(I)=NODES(I,1);
          YORD(I)=NODES(I,2);
          NPBC(I)=NODES(I,3);
      if XORD(I) > XMAX
         XMAX = XORD(I);
          elseif XORD(I) < XMIN
         XMIN = XORD(I);
          end
      if YORD(I) > YMAX
         YMAX = YORD(I);
          elseif YORD(I) < YMIN
         YMIN = YORD(I);
          end
      end

      clear NODES
      clear MESHo
      clear NPA

%     ---------------------------------------
%     PREPARE ARRAYS TO MATCH TYPE OF ELEMENT
%     ---------------------------------------
      if NNPE == 3
        NSIDES=3;
            NTRIAG=1;
            NPA=[ 1  2  3  1] ;
      elseif NNPE == 6
        NSIDES=3;
            NTRIAG=4;
            NPA = [  1 2 6 1 2 3 4 2 4 5 6 4 2 4 6 2 ]  ;
      elseif NNPE == 4
        NSIDES=3;
            NTRIAG=4;
            NPA = [ 4  1  0  4  1  2  0  1  2  3  0  2  3  4  0  3 ];
      elseif NNPE == 8
        NSIDES=3;
            NTRIAG=8;
            NPA= [ 0  1  2  0    0  2  3  0   0  3  4  0   0  4  5  0 ...
                   0  5  6  0    0  6  7  0   0  7  8  0   0  8  1  0 ];
      end


%     -----------------------
%     Ask User What is wanted
%     -----------------------
%        disp(' ')
%        disp(' ENTER:')
%        disp(' ---------------------------------')
%        disp(' y if you wish mesh to be plotted ')
%        disp(' ---------------------------------')
%        imsh = input(' < ','s');
%
%        disp(' ')
%        disp(' ENTER:')
%        disp(' --------------------------------------')
%        disp(' Do you wish contours to be plotted ')
%        disp('   0 if you do not ')
%        disp('   n for n-number of contours ')
%        disp(' --------------------------------------')
%        icntr = input(' < ');

    icntr = 0;

      if icntr > 0
%        ------------------------------------------------
%        Determine contour parameters:
%          Min and Max PSI values
%          Scaling factor for PSI values
%          Contour interval for scaled values
%        ------------------------------------------------
         pMax=PSI(1);
         pMin=pMax;
         for i=2:NUMNP
        if PSI(i) > pMax
             pMax=PSI(i);
           end
        if PSI(i) < pMin
             pMin=PSI(i);
           end
         end

         nc   = icntr;              % set number of contours
         dc   = (pMax-pMin)/nc;     % unrounded contour interval
         n    = fix(log10(dc));     % determine order of magnitude of dc
         mfc  = 10^(1-n);           % scaling factor for PSI values
         dc   = fix(dc*mfc);        % rounded scaled-contour interval
     dca  = dc/mfc;             % actual contour interval
%        ----------------------------------
%        Report to user and allow changes:
%        ----------------------------------
%           fprintf(1,'\n  ')
%           fprintf(1,'\n  ----------------------------------------')
%           fprintf(1,'\n  For your contour plot:  ')
%           fprintf(1,'\n  ----------------------------------------')
%           fprintf(1,'\n       Maximum PSI = %3i',pMax )
%           fprintf(1,'\n       Minimum PSI = %3i',pMin )
%           fprintf(1,'\n  Contour interval = %3i',dca)
%           fprintf(1,'\n  ----------------------------------------')
%           fprintf(1,'\n  Enter 1 if you would like any of ' )
%           fprintf(1,'\n  these values to be changed.')
%           fprintf(1,'\n  ')
%           a = input(' < ');


         if a == 1
            fprintf(1,'\n  ')
            fprintf(1,'\n  ----------------------------------------')
            fprintf(1,'\n  Enter Maximum PSI')
        pMax = input(' < ');
            fprintf(1,'\n  Enter Minimum PSI')
        pMin = input(' < ');
            fprintf(1,'\n  Enter Contour interval')
        dca = input(' < ');
        dc   = dca*mfc;
            fprintf(1,'\n  ----------------------------------------')
         end
      end


%        disp(' ')
%        disp(' ENTER:')
%        disp(' --------------------------------------')
%        disp(' y if you wish a color mapping ')
%        disp(' --------------------------------------')
%        colr = input(' < ','s');
%
%        disp(' ')
%        disp(' ENTER:')
%        disp(' --------------------------------------')
%        disp('    0  for no SYMMETRY')
%        disp('    1  for SYMMETRY about X axis ')
%        disp('    2  for SYMMETRY about Y axis ')
%        disp('    3  for SYMMETRY about both axes')
%        disp(' --------------------------------------')
%        isym = input(' < ');

    isym = 1;

      if isym < 0
         isym == 0;
      elseif isym > 3
         isym == 0;
      end


    TITLE = 'Velocity Potential Field';

%     ------------------------
%     Put hold on all graphics
%     ------------------------
      hold on
      axis equal

%     ------------------------
%     Add space for boarder
%     ------------------------
      if isym == 1 | isym == 3
         YMIN = -YMAX;
      end
      if isym == 2 | isym == 3
         XMIN = -XMAX;
      end
      xmin = XMIN - 0.01*(XMAX-XMIN);
      xmax = XMAX + 0.01*(XMAX-XMIN);
      ymin = YMIN - 0.01*(YMAX-YMIN);
      ymax = YMAX + 0.01*(YMAX-YMIN);
      PropertyName={'Color'};
      PropertyValue={'w'};
      H = line([xmin xmax xmax xmin],[ymin ymin ymax ymax]);
      set(H,PropertyName,PropertyValue)

    colr = 'y';
      if colr == 'y'
%        -----------------------------
%        Plot color map of PSI values
%        -----------------------------
         for J=1:NUMEL
        xave = 0;
        yave = 0;
        pave = 0;
        for K=1:NNPE
           xave = xave + XORD(NP(J,K));
           yave = yave + YORD(NP(J,K));
           pave = pave + PSI(NP(J,K));
            end
        xave = xave/NNPE;
        yave = yave/NNPE;
        pave = pave/NNPE;

            for  K=1:NTRIAG
               for L=1:4
              NL = NPA((K-1)*4+L);
          if NL == 0
             xp(L) = xave;
             yp(L) = yave;
             pp(L) = pave;
                  else
             NLP=NP(J,NL);
                 xp(L)=XORD(NLP);
                 yp(L)=YORD(NLP);
                 pp(L)= PSI(NLP);
                  end
               end
           clear gg
           if colr == 'y'
              gg = pp;
               else
              gg = [ 0.7 1 0.7 ];
               end

               fill(xp,yp,gg)
           if isym == 1 | isym == 3
              fill(xp,-yp,gg)
           end
           if isym == 2 | isym == 3
              fill(-xp,yp,gg)
           end
           if isym == 3
              fill(-xp,-yp,gg)
           end
            end
         end
     if colr == 'y'
            colorbar('vert')
         end
         shading interp
      end

      if icntr > 0
%        ----------------
%        Plot contours
%        ----------------
         %clear H
     nL = 0;
         for J=1:NUMNP
        PSIp(J) = PSI(J)*mfc;   % scale PSI values
         end

         for J=1:NUMEL
%           -----------------------------
%           Determine average values in
%           current element to assign to
%           interior points when necessary.
%           -----------------------------
        xave = 0;
        yave = 0;
        pave = 0;
        for K=1:NNPE
           xave = xave + XORD(NP(J,K));
           yave = yave + YORD(NP(J,K));
           pave = pave + PSIp(NP(J,K));
            end
        xave = xave/NNPE;
        yave = yave/NNPE;
        pave = pave/NNPE;

%           -----------------------------
%           Determine max and min PSI
%           values in current element
%           -----------------------------
            JNP=NP(J,1);
            cmin=PSIp(JNP);
            cmax=cmin;
            for K=1:NNPE
               KNP=NP(J,K);
               if PSIp(KNP) <  cmin
               cmin=PSIp(KNP);
               end
               if PSIp(KNP) > cmax
               cmax=PSIp(KNP);
               end
            end

%------------------------------------
        if cmin < pMin*mfc;
           cmin = pMin*mfc;;
        end
        if cmax > pMax*mfc;
           cmax = pMax*mfc;;
        end
%------------------------------------

%           ---------------------------
%           Begin plotting each contour
%           in current element
%           ---------------------------
            n=floor(cmin/dc)-1;
            C=n*dc;  % lowest possible contour

            while C <=  cmax
               clear x y
               for  K=1:NTRIAG
%                 ---------------------------
%                 Search for current contour
%                 in each sub-element
%                 ---------------------------
              J3=0;
                  for  L=1:NSIDES
                 L0=(K-1)*(NSIDES+1)+L;
             L1=NPA(L0);
                     L2=NPA(L0+1);

                 if L1 ~= 0
                        L1 =NP(J,L1);
                        XL1=XORD(L1);
                        YL1=YORD(L1);
                PL1=PSIp(L1);
                 else
                XL1=xave;
                YL1=yave;
                PL1=pave;
                     end
                 if L2 ~= 0
                        L2 =NP(J,L2);
                        XL2=XORD(L2);
                        YL2=YORD(L2);
                PL2=PSIp(L2);
                 else
                XL2=xave;
                YL2=yave;
                PL2=pave;
                     end
                     SLOPE=PL2-PL1;
             PT=-100;
             if abs(SLOPE) ~= 0
                        PT=(C-PL1)/SLOPE;
             end

%                    -------------------------------------------
%                    Determine if contour intersects current side.
%                    If so, record intersection
%                    -------------------------------------------
                     if PT >= 0 & PT < 1
                        J3=J3+1;
                        x(J3)=XL1+PT*(XL2-XL1);
                        y(J3)=YL1+PT*(YL2-YL1);
                     elseif SLOPE == 0
                if PL1 == C
                           J3=J3+1;
                           x(J3)=XL1;
                           y(J3)=YL1;
                           J3=J3+1;
                           x(J3)=XL2;
                           y(J3)=YL2;
                end
                     end

                  end % Finished with current side

%                 --------------------------------------
%                 Plot contour if in current sub-element
%                 --------------------------------------
                  if J3 >= 2
             nL=nL+1;
                 H(nL) = line(x,y);

                 if isym == 1 | isym == 3
                nL=nL+1;
                    H(nL) = line(x,-y);
                 end

                 if isym == 2 | isym == 3
                nL=nL+1;
                    H(nL) = line(-x,y);
                 end

                 if  isym == 3
                nL=nL+1;
                    H(nL) = line(-x,-y);
                 end
              end

               end    % Finished with current triangle

               C=C+dc;
            end  % Finished with contours (while loop)
         end  % Finished with all elements

%        ----------------------------
%        plotting properties and plot
%        ----------------------------
         if colr == 'y'
       PropertyName={'Color'};
       PropertyValue={'w'};
         else
       PropertyName={'Color'};
       PropertyValue={'r'};
         end
         set(H,PropertyName,PropertyValue)

      end  % Finished with contour plotting


%     ------------------
%     Plot mesh boundary
%     ------------------
      clear H
      clear rot
      nL=0;
      if NNPE == 3;
         nS=3;  % number of sides
     pS=2;  % points per side
     rot=[1 2 3 1 2];
      end
      if NNPE == 4
         nS=4;
     pS=2;
     rot=[1 2 3 4 1 2];
      end
      if NNPE == 6
         nS=3;
     pS=3;
     rot=[1 2 3 4 5 6 1 2];
      end
      if NNPE == 8
         nS=4;
     pS=3;
     rot=[1 2 3 4 5 6 7 8 1 2];
      end

      for I=1:NUMNP
    sA(I)=0.0;
    LpN(I)=0;
      end

      for I=1:NUMEL
         for J=2:NNPE+1
         no =NP(I,rot(J  ));
         na =NP(I,rot(J-1));
         nb =NP(I,rot(J+1));

             LpN(no)=LpN(no)+1;
         a(1)=XORD(na)-XORD(no);
         a(2)=YORD(na)-YORD(no);
         b(1)=XORD(nb)-XORD(no);
         b(2)=YORD(nb)-YORD(no);

         aa=a(1)*a(1)+a(2)*a(2);
```

```
   bb=b(1)*b(1)+b(2)*b(2);
   ab=a(1)*b(1)+a(2)*b(2);

   ang=acos(ab/sqrt(aa*bb));
   sA(no)=sA(no)+ang;
   end
end
Atest=2*pi-1.0e-06;
clear H
nL=0;
for I=1:NUMEL
   for J=1:NNPE
      r1=rot(J);
  r2=rot(J+1);
  JP1=NP(I,r1);
  JP2=NP(I,r2);
  if sA(JP1) < Atest
     if sA(JP2) < Atest
        nL=nL+1;
    Hx(nL,1) = XORD(JP1);
    Hx(nL,2) = XORD(JP2);
    Hy(nL,1) = YORD(JP1);
    Hy(nL,2) = YORD(JP2);
     end
  end
   end
end

for I=1:nL-1
    x11=Hx(I,1);
    x12=Hx(I,2);
    y11=Hy(I,1);
    y12=Hy(I,2);
for J=I+1:nL
       x21=Hx(J,1);
       x22=Hx(J,2);
       y21=Hy(J,1);
       y22=Hy(J,2);

   if x11 == x22 & x12 == x21 ...
    & y11 == y22 & y12 == y21
          Hx(I,1) = XMIN;
          Hx(I,2) = XMIN;
          Hy(I,1) = YMIN;
          Hy(I,2) = YMIN;
          Hx(J,1) = XMIN;
          Hx(J,2) = XMIN;
          Hy(J,1) = YMIN;
          Hy(J,2) = YMIN;
   end
end
end

for I=1:nL
    x1=Hx(I,1);
    x2=Hx(I,2);
    y1=Hy(I,1);
    y2=Hy(I,2);
```

```
       if isym == 1 | isym == 3
          if y1 == 0 & y2 == 0
             Hx(I,1) = XMIN;
             Hx(I,2) = XMIN;
             Hy(I,1) = 0;
             Hy(I,2) = 0;
          end
       end
       if isym == 2 | isym == 3
          if x1 == 0 & x2 == 0
             Hx(I,1) = 0;
             Hx(I,2) = 0;
             Hy(I,1) = YMIN;
             Hy(I,2) = YMIN;
          end
       end
      end

       for I=1:nL
         line(Hx(I,:),Hy(I,:),'Color','k','LineWidth',2)
       end

       if isym == 1 | isym == 3
          for I=1:nL
            line(Hx(I,:),-Hy(I,:),'Color','k','LineWidth',2)
          end
       end
       if isym == 2 | isym == 3
          for I=1:nL
            line(-Hx(I,:),Hy(I,:),'Color','k','LineWidth',2)
          end
       end
       if isym == 3
          for I=1:nL
            line(-Hx(I,:),-Hy(I,:),'Color','k','LineWidth',2)
          end
       end

    imsh = 'n';

      if imsh == 'y'
%        ---------
%        Plot mesh
%        ---------
         clear H
         nL=0;
         for I=1:NUMEL
            for J=1:NNPE
           r1=rot(J);
           r2=rot(J+1);
           JP1=NP(I,r1);
           JP2=NP(I,r2);
           nL=nL+1;
           H(nL)=line([XORD(JP1) XORD(JP2)],[YORD(JP1) YORD(JP2)]);
            end
         end
         set(H,'Color','k','LineWidth',1)
      end
      title(TITLE)
%     ------------------------
%     Remove hold on graphics
%     ------------------------
      hold off
```