```
% Filename: convertBathymetryXyzFile.m
% Author:   Don Brutzman
% Created:  26 June 2000
% Revised:  26 May 2001
% Description:  Read gridded depth data, build elevation grids
% Metadata: Fort Lauderdale Florida bathymetry, see metadata.txt and
%       University of Miami mission report
% Invocation:
%   cd C:\www.web3D.org\TaskGroups\x3d\translation\examples\NpsMilitaryModels\Locations\FortLauderdale
Florida
%   convertBathymetryXyzFile;
%
% Results:
%   http://web.nps.navy.mil/~brutzman/vrml/examples/NpsMilitaryModels/Locations/FortLauderdaleFlorida/
chapter.html
%
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

% depthMatrix data structure:
%
%
%   y=vertical=depth=grid values
%   |
%   |
%   |      x=latitude=north, northings=rows
%   |     /
%   |    +-----+
%   |   /     /
%   |  /     /
%   | /     /
%   |/     /
%   *-----+-> z=longitude=east, eastings=columns

% helpful mnemonic for this particular dataset:  "columns are long,"
```

```
% i.e. columns are longitudes, and also because
% these columns are twice as long as these rows

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%
%    05m dataset   Spacing          Number
%   Latitude              5            1340
%   Longitude             5             646
%
%   10m dataset    Spacing          Number
%   Latitude             10             670
%   Longitude            10             323
%
%   20m dataset    Spacing          Number
%   Latitude             20             335
%   Longitude            20             161
%
%   40m dataset    Spacing          Number
%   Latitude             40             167
%   Longitude            40              80
%
%   80m dataset    Spacing          Number
%   Latitude             80              83
%   Longitude            80              40
%
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
%
% VRML ElevationGrid references:
%
% http://www.web3D.org/technicalinfo/specifications/vrml97/part1/nodesRef.html#ElevationGrid
% Figure 6.5:
% http://www.web3D.org/technicalinfo/specifications/vrml97/Images/ElevationGrid.gif
%
% VRML ElevationGrid arrangement from Figure 6.5, showing order of data points:
```

```
%
%    0  1  2  3  4 ----> xDimension=5
%    5  6  7  8  9
%   10 11 12 13 14
%   15 16 17 18 19
%    |
%   zDimension=4
%
% However the above diagram is slightly misleading here, because
% it is oriented 90 degrees clockwise compared to what we want.
%
% Initialization note:  VRML indices begin at 0, MatLab indices begin at 1.
%
% Matlab layout for this dataset table is x vertical north, z horizontal east:
%
%   xDimension=maxLat
%    |
%   (maxLat, 1) ... (maxLat, maxLong)
%    |               |
%    |               |
%    |               |
%    |               |
%   (1, 1)      ... (1, maxLong)    -> zDimension=maxLong
%
% thus ElevationGrid xDimension maps to row    latitudes,
%  and ElevationGrid zDimension maps to column longitudes.
%
% Therefore, the order for filling in the ElevationGrid values is
% first from (1, 1) to (maxLat, 1)  [meaning inner loop is    rows = latitudes ]
%  then from (1, 1) to (1, maxLong) [meaning outer loop is columns = longitudes]
%
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

true  = 1;
```

```
false = 0;

%cd 'D:\Large Projects\MCM 2000 evaluation of search tactics\fortLauderdaleBathymetry';
 cd 'C:\www.web3D.org\TaskGroups\x3d\translation\examples\NpsMilitaryModels\Locations\FortLauderdaleFl
orida';
pwd;

if ~exist('shallow')
   disp('loading shallow.xyz.txt ...');
   load shallow.xyz.txt% loads data into matrix 'shallow'
else whos shallow;
end;

nDataPoints = length(shallow);
disp (['Process ' num2str(nDataPoints)' evenly spaced data points']);
disp (['   in UTM xyz format (easting/northing/depth) into grids ...']);
fprintf ('\n');

missingDatumDepth = -99;% meters
disp (['missingDatumDepth  =   'num2str(missingDatumDepth)' m']);
fprintf ('\n');

% easting/northing/depth ordering in data file = long/lat/depth
minLong =  min(shallow(:,1));
maxLong =  max(shallow(:,1));
minLat  =  min(shallow(:,2));
maxLat  =  max(shallow(:,2));
minDepth= -min(shallow(:,3));
maxDepth= -max(shallow(:,3));

% format short g; % 5 character places per variable
disp ( '             Minimum          Maximum');
disp (['Latitude    '       num2str(minLat, '%+10.1f') '       '      num2str(maxLat, '%+10.1f') '
m northing']);
```

```
disp (['Longitude    '       num2str(minLong,'%+10.1f') '         '     num2str(maxLong,'%+10.1f') '
m easting']);
disp (['Depth             '  num2str(minDepth,'%+10.1f') '            ' num2str(maxDepth,'%+10.1f') '
m']);

latSpacing  = 5; % meters, by visual inspection and mission report
longSpacing = 5; % meters, by visual inspection and mission report

nLats05m  = (maxLat  - minLat)  / latSpacing + 1;
nLongs05m = (maxLong - minLong) / latSpacing + 1;

disp('creating   depthMatrix05m ...');
depthMatrix05m = missingDatumDepth * ones(nLats05m, nLongs05m);

disp('filling in depthMatrix05m ...');
for i = 1:1:nDataPoints,
   long =  shallow(i,1);%  easting values are positive
   lat  =  shallow(i,2);% northing values are positive
   depth= -shallow(i,3);% depth values need to be negative, zero at surface
   row  = (lat  - minLat )/latSpacing  + 1;
   col  = (long - minLong)/longSpacing + 1;
   depthMatrix05m(row,col) = depth;
end;

disp('smoothing  depthMatrix05m ...');
% push unmeasured shallow depths within 50m of shoreline to 1m depth,
% to eliminate shoreline ambiguity and polygonal aliasing effects
for    row = 1:1:nLats05m,
   for col = 1:1:10,
      if (depthMatrix05m(row,col) == missingDatumDepth) depthMatrix05m(row,col) = -1% meter
      end;
   end;
end;
```

```
whos depthMatrix05m;

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

disp('commencing  FortLauderdaleDepths05m.xml ...');
fprintf ('\n');

disp ( ' 05m dataset   Spacing          Number');
disp (['Latitude             ' num2str(latSpacing)  '            ' num2str(nLats05m)]);
disp (['Longitude            ' num2str(longSpacing) '             '  num2str(nLongs05m)]);
fprintf ('\n');

nLats05mHalf = floor (nLats05m/2);

file05m = fopen('FortLauderdaleDepths05m.xml', 'w');
printX3dHeader (file05m, 'FortLauderdaleDepths05m.xml', missingDatumDepth);
fprintf (file05m, ['xDimension="' num2str(nLats05mHalf)   '" ']);
fprintf (file05m, ['zDimension="' num2str(nLongs05m)   '" ']);
fprintf (file05m, ['xSpacing="'   num2str(latSpacing)  '" ']);
fprintf (file05m, ['zSpacing="'   num2str(longSpacing) '"\n']);
fprintf (file05m,  'height="\n');

for    col = 1:1:nLongs05m,
   for row = 1:1:nLats05mHalf,
      fprintf(file05m, '%4.1f', depthMatrix05m(row,col));
      if (row < nLats05m)
           fprintf(file05m, '\t');
      else fprintf(file05m, '\n');
      end;
   end;
end;
fprintf (file05m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file05m, '                <ColorNode color="\n');
for    col = 1:1:nLongs05m,
```

```
   for row = 1:1:nLats05mHalf,
      fprintf(file05m, colorValue(depthMatrix05m(row,col)));
      if (row < nLats05m)
           fprintf(file05m,'\t');
      else fprintf(file05m, '\n');
      end;
   end;
end;
fprintf (file05m, '"/>\n'); % end ColorNode tag

fprintf (file05m, '            </ElevationGrid>\n');
fprintf (file05m, '        </Shape>\n');

fprintf (file05m, '        <Transform translation="');
fprintf (file05m, '%3.1f 0 0', (nLats05mHalf+1) * latSpacing);
fprintf (file05m, '">\n');

fprintf (file05m, '            <Shape>\n');
fprintf (file05m, '                <ElevationGrid colorPerVertex="true" solid="false"');

fprintf (file05m, ['xDimension="' num2str(nLats05m - nLats05mHalf)   '" ']);
fprintf (file05m, ['zDimension="' num2str(nLongs05m)  '" ']);
fprintf (file05m, ['xSpacing="'   num2str(latSpacing) '" ']);
fprintf (file05m, ['zSpacing="'   num2str(longSpacing)'"\n']);
fprintf (file05m,  'height="\n');

for    col = 1:1:nLongs05m,
   for row = nLats05mHalf+1:1:nLats05m,
      fprintf(file05m,'%4.1f', depthMatrix05m(row,col));
      if (row < nLats05m)
           fprintf(file05m,'\t');
      else fprintf(file05m, '\n');
      end;
   end;
```

```
end;
fprintf (file05m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file05m, '                    <ColorNode color="\n');
for    col = 1:1:nLongs05m,
   for row = nLats05mHalf+1:1:nLats05m,
      fprintf(file05m, colorValue(depthMatrix05m(row,col)));
      if (row < nLats05m)
           fprintf(file05m,'\t');
      else fprintf(file05m, '\n');
      end;
   end;
end;
fprintf (file05m, '"/>\n'); % end ColorNode tag

fprintf (file05m, '                </ElevationGrid>\n');
fprintf (file05m, '            </Shape>\n');
fprintf (file05m, '        </Transform>\n');

printX3dFooter (file05m, false);
fclose(file05m);
disp('FortLauderdaleDepths05m.xml complete.');
fprintf ('\n');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

disp('commencing FortLauderdaleDepths10m.xml ...');
nLats10m =floor( nLats05m/2);
nLongs10m=floor(nLongs05m/2);
depthMatrix10m = missingDatumDepth * ones(nLats10m, nLongs10m);

disp ( '10m dataset    Spacing          Number');
disp (['Latitude           ' num2str(latSpacing*2) '            ' num2str(nLats10m)]);
disp (['Longitude          ' num2str(longSpacing*2) '            ' num2str(nLongs10m)]);
```

```
file10m = fopen('FortLauderdaleDepths10m.xml', 'w');
printX3dHeader (file10m,'FortLauderdaleDepths10m.xml', missingDatumDepth);
fprintf (file10m, ['xDimension="' num2str(nLats10m)    '" ']);
fprintf (file10m, ['zDimension="' num2str(nLongs10m)     '" ']);
fprintf (file10m, ['xSpacing="'   num2str(latSpacing*2)'" ']);
fprintf (file10m, ['zSpacing="'   num2str(longSpacing*2) '"\n']);
fprintf (file10m,  'height="\n');

for    col = 1:1:nLongs10m,
   for row = 1:1:nLats10m,
      value = (depthMatrix05m(2*row,  2*col) + depthMatrix05m(2*row,  2*col-1) +...
               depthMatrix05m(2*row-1,2*col) + depthMatrix05m(2*row-1,2*col-1))/4;
      fprintf(file10m,'%4.1f', value);
      depthMatrix10m(row, col) = value;
      if (row < nLats10m)
           fprintf(file10m,'\t');
      else fprintf(file10m, '\n');
      end;
   end;
end;
fprintf (file10m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file10m, '                <ColorNode color="\n');
for    col = 1:1:nLongs10m,
   for row = 1:1:nLats10m,
      fprintf(file10m, colorValue(depthMatrix10m(row,col)));
      if (row < nLats10m)
           fprintf(file10m,'\t');
      else fprintf(file10m, '\n');
      end;
   end;
end;
fprintf (file10m, '"/>\n'); % end ColorNode tag

printX3dFooter (file10m, true);
```

```
fclose(file10m);
disp('FortLauderdaleDepths10m.xml complete.');
fprintf ('\n');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

disp('commencing FortLauderdaleDepths20m.xml ...');
nLats20m =floor( nLats10m/2);
nLongs20m=floor(nLongs10m/2);
depthMatrix20m = missingDatumDepth * ones(nLats20m, nLongs20m);

disp ('20m dataset    Spacing          Number');
disp (['Latitude           ' num2str(latSpacing*4) '            ' num2str(nLats20m)]);
disp (['Longitude          ' num2str(longSpacing*4) '            ' num2str(nLongs20m)]);

file20m = fopen('FortLauderdaleDepths20m.xml', 'w');
printX3dHeader (file20m, 'FortLauderdaleDepths20m.xml', missingDatumDepth);
fprintf (file20m, ['xDimension="' num2str(nLats20m)    '" ']);
fprintf (file20m, ['zDimension="' num2str(nLongs20m)     '" ']);
fprintf (file20m, ['xSpacing="'   num2str(latSpacing*4) '" ']);
fprintf (file20m, ['zSpacing="'   num2str(longSpacing*4) '"\n']);
fprintf (file20m,  'height="\n');

for    col = 1:1:nLongs20m,
   for row = 1:1:nLats20m,
      value = (depthMatrix10m(2*row,  2*col) + depthMatrix10m(2*row,  2*col-1) +...
               depthMatrix10m(2*row-1,2*col) + depthMatrix10m(2*row-1,2*col-1))/4;
      fprintf(file20m,'%4.1f', value);
      depthMatrix20m(row, col) = value;
      if (row < nLats20m)
           fprintf(file20m,'\t');
      else fprintf(file20m, '\n');
      end;
   end;
```

```
end;
fprintf (file20m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file20m, '                <ColorNode color="\n');
for    col = 1:1:nLongs20m,
   for row = 1:1:nLats20m,
      fprintf(file20m, colorValue(depthMatrix20m(row,col)));
      if (row < nLats20m)
           fprintf(file20m,'\t');
      else fprintf(file20m, '\n');
      end;
   end;
end;
fprintf (file20m, '"/>\n'); % end ColorNode tag

printX3dFooter (file20m, true);
fclose(file20m);
disp('FortLauderdaleDepths20m.xml complete.');
fprintf ('\n');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

disp('commencing FortLauderdaleDepths40m.xml ...');
nLats40m =floor( nLats20m/2);
nLongs40m=floor(nLongs20m/2);
depthMatrix40m = missingDatumDepth * ones(nLats40m, nLongs40m);

disp ('40m dataset    Spacing          Number');
disp (['Latitude           ' num2str(latSpacing*8) '            ' num2str(nLats40m)]);
disp (['Longitude          ' num2str(longSpacing*8) '             '  num2str(nLongs40m)]);

file40m = fopen('FortLauderdaleDepths40m.xml', 'w');
printX3dHeader (file40m,'FortLauderdaleDepths40m.xml', missingDatumDepth);
fprintf (file40m, ['xDimension="' num2str(nLats40m)    '" ']);
fprintf (file40m, ['zDimension="' num2str(nLongs40m)     '" ']);
```

```
fprintf (file40m, ['xSpacing="'   num2str(latSpacing*8)'" ']);
fprintf (file40m, ['zSpacing="'   num2str(longSpacing*8) '"\n']);
fprintf (file40m,  'height="\n');

for    col = 1:1:nLongs40m,
   for row = 1:1:nLats40m,
      value = (depthMatrix20m(2*row,  2*col) + depthMatrix20m(2*row,  2*col-1) +...
               depthMatrix20m(2*row-1,2*col) + depthMatrix20m(2*row-1,2*col-1))/4;
      fprintf(file40m,'%4.1f', value);
      depthMatrix40m(row, col) = value;
      if (row < nLats40m)
           fprintf(file40m,'\t');
      else fprintf(file40m, '\n');
      end;
   end;
end;
fprintf (file40m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file40m, '                <ColorNode color="\n');
for    col = 1:1:nLongs40m,
   for row = 1:1:nLats40m,
      fprintf(file40m, colorValue(depthMatrix40m(row,col)));
      if (row < nLats40m)
           fprintf(file40m,'\t');
      else fprintf(file40m, '\n');
      end;
   end;
end;
fprintf (file40m, '"/>\n'); % end ColorNode tag

printX3dFooter (file40m, true);
fclose(file40m);
disp('FortLauderdaleDepths40m.xml complete.');
fprintf ('\n');
```

```
% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

disp('commencing FortLauderdaleDepths80m.xml ...');
nLats80m =floor( nLats40m/2);
nLongs80m=floor(nLongs40m/2);
depthMatrix80m = missingDatumDepth * ones(nLats80m, nLongs80m);

disp ('80m dataset    Spacing          Number');
disp (['Latitude           ' num2str(latSpacing*16) '             ' num2str(nLats80m)]);
disp (['Longitude          ' num2str(longSpacing*16)'             ' num2str(nLongs80m)]);

file80m = fopen('FortLauderdaleDepths80m.xml', 'w');
printX3dHeader (file80m,'FortLauderdaleDepths80m.xml', missingDatumDepth);
fprintf (file80m, ['xDimension="' num2str(nLats80m)     '" ']);
fprintf (file80m, ['zDimension="' num2str(nLongs80m)      '" ']);
fprintf (file80m, ['xSpacing="'   num2str(latSpacing*16) '" ']);
fprintf (file80m, ['zSpacing="'   num2str(longSpacing*16)'"\n']);
fprintf (file80m,  'height="\n');

for    col = 1:1:nLongs80m,
   for row = 1:1:nLats80m,
      value = (depthMatrix40m(2*row,  2*col) + depthMatrix40m(2*row,  2*col-1) +...
               depthMatrix40m(2*row-1,2*col) + depthMatrix40m(2*row-1,2*col-1))/4;
      fprintf(file80m,'%4.1f', value);
      depthMatrix80m(row, col) = value;
      if (row < nLats80m) fprintf(file80m,'\t');
      end;
   end;
   fprintf(file80m, '\n');
end;
fprintf (file80m, '">\n'); % end height table, end ElevationGrid tag
fprintf (file80m, '                <ColorNode color="\n');
for    col = 1:1:nLongs80m,
   for row = 1:1:nLats80m,
```

```
      fprintf(file80m, colorValue(depthMatrix80m(row,col)));
      if (row < nLats80m)
           fprintf(file80m,'\t');
      else fprintf(file80m, '\n');
      end;
   end;
end;
fprintf (file80m, '"/>\n'); % end ColorNode tag
printX3dFooter (file80m, true);
fclose(file80m);
disp('FortLauderdaleDepths80m.xml complete.');
fprintf ('\n');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

% mesh(depthMatrix);

disp ('XML syntax checks:  c:\xml\microsoft\xmlint *.xml');
! c:\xml\microsoft\xmlint *.xml

mesh(depthMatrix80m);


disp ('XML to VRML translations');

% instant saxon not yet able to handle such large attributes
% can convert 10..80 using X3D-Edit (which uses xalan)
% keep the following commented until autotranslation capacity corrected

! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepthSelection

%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat ElevationGridExample
%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepths05m
%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepths10m
```

```
%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepths20m
%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepths40m
%   ! c:\www.web3D.org\TaskGroups\x3d\translation\X3dToVrml97.bat FortLauderdaleDepths80m

! mv FortLauderdaleDepthSelectionTranslated.wrl FortLauderdaleDepthSelection.wrl

%   ! mv ElevationGridExampleTranslated.wrl         ElevationGridExample.wrl
%   ! mv FortLauderdaleDepths05mTranslated.wrl     FortLauderdaleDepths05m.wrl
%   ! mv FortLauderdaleDepths10mTranslated.wrl     FortLauderdaleDepths10m.wrl
%   ! mv FortLauderdaleDepths20mTranslated.wrl     FortLauderdaleDepths20m.wrl
%   ! mv FortLauderdaleDepths40mTranslated.wrl     FortLauderdaleDepths40m.wrl
%   ! mv FortLauderdaleDepths80mTranslated.wrl     FortLauderdaleDepths80m.wrl

! c:\vrml\vorlon\vorlon *.wrl
```

```
% Filename: colorValue.m
% Author:   Don Brutzman
% Created:  26 June 2000
% Revised:  18 March 2001
% Description:  Map depth values to MEDAL color map.

function rgbValue = colorValue (depthValueMeters);

    depthValueFeet = -depthValueMeters * (39.3 / 12.0);% 1m = 39.3"

    if     (depthValueFeet < 5) rgbValue ='1 1 1';     % white
    elseif (depthValueFeet < 10)    rgbValue ='1 0 0';     % red
    elseif (depthValueFeet < 20)    rgbValue ='1 .529 0';      % orange
    elseif (depthValueFeet < 30)    rgbValue ='1 1 0';     % yellow
    elseif (depthValueFeet < 40)    rgbValue ='0 1 0';     % green
    elseif (depthValueFeet < 50)    rgbValue ='0 1 1';     % cyan
    elseif (depthValueFeet < 60)    rgbValue ='0 0 1';     % blue
    elseif (depthValueFeet < 70)    rgbValue ='1 0 1';     % magenta
    elseif (depthValueFeet < 80)    rgbValue ='.561 0 .322';   % maroon
    elseif (depthValueFeet < 90)    rgbValue ='.871 .721 .529';    % tan
    elseif (depthValueFeet < 100)   rgbValue ='.322 .584 .517';    % sea green
    elseif (depthValueFeet < 150)   rgbValue ='.494 .533 .671';    % slate blue
    elseif (depthValueFeet < 200)   rgbValue ='.137 .137 .459';    % navy blue
    elseif (depthValueFeet < 250)   rgbValue ='.5 .5 .5';      % grey
    else                            rgbValue ='.439 .502 .565';    % slate grey
    end;
return;

% some helpful color-value definitions available at
% http://www.w3.org/TR/1998/REC-html40-19980424/types.html#idx-color

%   From:    "Heytler, Susan L." <SUSAN.L.HEYTLER@saic.com>
%   Subject: FW: Bathymetry colors
%
```

```
%   -----Original Message-----
%   From: Craig Pell [mailto:cpell@uranus.isd.saic.com]
%   Sent: Friday, June 23, 2000 14:55
%   To: Susan Heytler
%   Subject: Bathymetry colors
%
%   white           255 255 255
%   red             255   0   0
%   orange          255 135   0
%   yellow          255 255   0
%   green             0 255   0
%   blue              0   0 255
%   cyan              0 255 255
%   magenta         255   0 255
%   black             0   0   0
%   maroon          143   0  82
%   tan             222 184 135
%   seagreen         82 149 132
%   olivedrab       107 142  35
%   skyblue         114 159 255
%   slateblue       126 136 171
%   navyblue         35  35 117
%   grey            126 126 126
%   slategrey       112 128 144
```

```
% Filename:     printX3dHeader.m
% Author:       Don Brutzman
% Created:      25 June 2000
% Revised:      4 July 2001

function printX3dHeader (fileID, fileName, missingDatumDepth);
fprintf (fileID, '<?xml version="1.0" encoding="UTF-8"?>\n');
fprintf (fileID, '<!DOCTYPE X3D PUBLIC "http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"\n');
fprintf (fileID, '                      "file:///C:/www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd">\n');
fprintf (fileID, '<X3D>\n');
fprintf (fileID, '  <head>\n');
fprintf (fileID, '    <meta name="filename" content="');
fprintf (fileID, fileName);
fprintf (fileID, '"/>\n');
fprintf (fileID, '    <meta name="description" content="Bathymetry for Fort Lauderdale, Hollywood Beach, Florida."/>\n');
fprintf (fileID, '    <meta name="author" content="Don Brutzman"/>\n');
fprintf (fileID, '    <meta name="revised" content="%s"/>\n', date);
fprintf (fileID, '    <meta name="url" content="http://web.nps.navy.mil/~brutzman/Savage/Locations/FortLauderdaleFlorida/chapter.html"/>\n');
fprintf (fileID, '    <meta name="reference" content="matlabPerspectivePlot.png"/>\n');
fprintf (fileID, '    <meta name="reference" content="FortLauderdaleDepths80m.fig"/>\n');
fprintf (fileID, '    <meta name="reference" content="metadata.txt"/>\n');
fprintf (fileID, '    <meta name="reference" content="shallow.xyz.txt"/>\n');
fprintf (fileID, '    <meta name="generator" content="convertBathymetryXyzFile.m"/>\n');
fprintf (fileID, '    <meta name="generator" content="colorValue.m"/>\n');
fprintf (fileID, '    <meta name="generator" content="printX3dHeader.m"/>\n');
fprintf (fileID, '    <meta name="generator" content="printX3dFooter.m"/>\n');
fprintf (fileID, '  </head>\n');
fprintf (fileID, '  <Scene>\n');
fprintf (fileID, '    <!--NavigationInfo: increased speed for responsiveness, increased collision boundary to eliminate aliasing-->\n');
```

```
fprintf (fileID, '    <NavigationInfo speed="100" type='''"EXAMINE" "WALK" "ANY"''' avatarSize="4 1.6 0.
75"/>\n'); % (double '' to print ' inside format string)
fprintf (fileID, '    <Viewpoint description="North side, looking southward, 1500m up" position=" 8100
 1500  1600"    orientation="-.2460949718952179, .9325416088104248, .2642034888267517, 1.612414479255
763"/>\n');
fprintf (fileID, '    <Viewpoint description="NW corner,  looking southeast,  500m up" position=" 7200
  500  -400"    orientation="-0.07347918301820755, .9866307377815247, .1454669684171676, 2.276195287
044678"/>\n');
fprintf (fileID, '    <Viewpoint description="SW corner,  looking northeast,  500m up" position=" -500
  500  -400"    orientation="0.0634222999215126, .9905853867530823, .12131864577531815, 3.992870569229
126"/>\n');
fprintf (fileID, '    <Viewpoint description="South side, looking northward, 1500m up" position="-1400
 1500  1600"    orientation=".24548138678073883, .9383738040924072, .24329698085784912, 4.644650936126
709"/>\n');
fprintf (fileID, '    <Viewpoint description="SE corner,  looking northwest,  500m up" position=" -500
  500  3600"    orientation="-.3247299194335937, -.9263483285903931, -.19086489081382751, .860675692
582886"/>\n');
fprintf (fileID, '    <Viewpoint description="NE corner,  looking southwest,  500m up" position=" 7200
  500  3600"    orientation="-.26649707555770874, .9542668461799622, .13547702133655548, .880036950111
3892"/>\n');
fprintf (fileID, '    <Transform>\n');
fprintf (fileID, '        <!--semi-transparent flat plate at surface-->\n');
fprintf (fileID, '        <Shape>\n');

fprintf (fileID, '            <!--IndexedFaceSet subdivided to enable view-frustrum culling for perfor
mance improvement-->\n');
fprintf (fileID, '            <!--northing:  2879500.0 - 2886195.0 = 6695 m;  easting:  592680.0 - 589
455.0 = 3225 m-->\n');

%   x
%   |
%   6695    0   1
%   4500    2   3
%   3000    4   5
```

```
%   1500    6   7
%   0   8   9
%
%       0   3225 -> z

fprintf (fileID, '           <IndexedFaceSet coordIndex="1 0 2 3 -1, 3 2 4 5 -1, 5 4 6 7 -1, 7 6 8 9 -1" solid="false">\n');
fprintf (fileID, '               <Coordinate point=');
fprintf (fileID, '6695 0 0, 6695 0 3225, ');
fprintf (fileID, '4500 0 0, 4500 0 3225, ');
fprintf (fileID, '3000 0 0, 3000 0 3225, ');
fprintf (fileID, '1500 0 0, 1500 0 3225, ');
fprintf (fileID, '   0 0 0,    0 0 3225');
fprintf (fileID, '"/>\n');
fprintf (fileID, '           </IndexedFaceSet>\n');
fprintf (fileID, '           <Appearance>\n');
fprintf (fileID, '               <Material diffuseColor="0 .1 .5" transparency=".75"/>\n');
fprintf (fileID, '           </Appearance>\n');
fprintf (fileID, '       </Shape>\n');
fprintf (fileID, '       <!--solid flat plate at (missingDatumDepth+1)=');
fprintf (fileID, '%3.1f', missingDatumDepth+1);
fprintf (fileID, 'm depth, for perspective reference-->\n');
fprintf (fileID, '       <Shape>\n');
fprintf (fileID, '           <IndexedFaceSet coordIndex="1 0 2 3 -1, 3 2 4 5 -1, 5 4 6 7 -1, 7 6 8 9 -1" colorPerVertex="false" solid="false">\n');
fprintf (fileID, '               <Color color="0 .5 .5, 0 .5 .5, 0 .5 .5, 0 .5 .5"/>\n');
fprintf (fileID, '               <Coordinate point=');
fprintf (fileID, '6695 -100 0, 6695 -100 3225, ');
fprintf (fileID, '4500 -100 0, 4500 -100 3225, ');
fprintf (fileID, '3000 -100 0, 3000 -100 3225, ');
fprintf (fileID, '1500 -100 0, 1500 -100 3225, ');
fprintf (fileID, '   0 -100 0,    0 -100 3225');
fprintf (fileID, '"/>\n');
fprintf (fileID, '           </IndexedFaceSet>\n');
```

```
fprintf (fileID, '        </Shape>\n');
fprintf (fileID, '        <!--terrain grid-->\n');
fprintf (fileID, '        <Shape>\n');
fprintf (fileID, '            <ElevationGrid colorPerVertex="true" solid="false"');
return;
```

```
% Filename: printX3dFooter.m
% Author:   Don Brutzman
% Created:  25 June 2000
% Revised:  18 March 2001

function printX3dFooter (fileID, finishElevationGrid);

if (finishElevationGrid)
    fprintf (fileID, '            </ElevationGrid>\n');
    fprintf (fileID, '        </Shape>\n');
end;
fprintf (fileID, '    </Transform>\n');
fprintf (fileID, '  </Scene>\n');
fprintf (fileID, '</X3D>\n');
return;
```