```
function [xx,yy,zz] = earth_sphere(varargin)
%EARTH_SPHERE Generate an earth-sized sphere.
%   [X,Y,Z] = EARTH_SPHERE(N) generates three (N+1)-by-(N+1)
%   matrices so that SURFACE(X,Y,Z) produces a sphere equal to
%   the radius of the earth in kilometers. The continents will be
%   displayed.
%
%   [X,Y,Z] = EARTH_SPHERE uses N = 50.
%
%   EARTH_SPHERE(N) and just EARTH_SPHERE graph the earth as a
%   SURFACE and do not return anything.
%
%   EARTH_SPHERE(N,'mile') graphs the earth with miles as the unit rather
%   than kilometers. Other valid inputs are 'ft' 'm' 'nm' 'miles' and 'AU'
%   for feet, meters, nautical miles, miles, and astronomical units
%   respectively.
%
%   EARTH_SPHERE(AX,...) plots into AX instead of GCA.
%
%  Examples:
%    earth_sphere('nm') produces an earth-sized sphere in nautical miles
%
%    earth_sphere(10,'AU') produces 10 point mesh of the Earth in
%    astronomical units
%
%    h1 = gca;
%    earth_sphere(h1,'mile')
%    hold on
%    plot3(x,y,z)
%      produces the Earth in miles on axis h1 and plots a trajectory from
%      variables x, y, and z

%   Clay M. Thompson 4-24-1991, CBM 8-21-92.
%   Will Campbell, 3-30-2010
%   Copyright 1984-2010 The MathWorks, Inc.

%% Input Handling
[cax,args,nargs] = axescheck(varargin{:}); % Parse possible Axes input
error(nargchk(0,2,nargs)); % Ensure there are a valid number of inputs

% Handle remaining inputs.
% Should have 0 or 1 string input, 0 or 1 numeric input
j = 0;
k = 0;
n = 50; % default value
units = 'km'; % default value
for i = 1:nargs
    if ischar(args{i})
        units = args{i};
        j = j+1;
    elseif isnumeric(args{i})
```

```
        n = args{i};
        k = k+1;
    end
end

if j > 1 || k > 1
    error('Invalid input types')
end

%% Calculations

% Scale factors
Scale = {'km' 'm'  'mile'            'miles'           'nm'              'au'
'ft';
         1    1000 0.621371192237334 0.621371192237334 0.539956803455724 6.6845871226706
e-009 3280.839895};

% Identify which scale to use
try
    myscale = 6378.1363*Scale{2,strcmpi(Scale(1,:),units)};
catch %#ok<*CTCH>
    error('Invalid units requested. Please use m, km, ft, mile, miles, nm, or AU')
end

% -pi <= theta <= pi is a row vector.
% -pi/2 <= phi <= pi/2 is a column vector.
theta = (-n:2:n)/n*pi;
phi = (-n:2:n)'/n*pi/2;
cosphi = cos(phi); cosphi(1) = 0; cosphi(n+1) = 0;
sintheta = sin(theta); sintheta(1) = 0; sintheta(n+1) = 0;

x = myscale*cosphi*cos(theta);
y = myscale*cosphi*sintheta;
z = myscale*sin(phi)*ones(1,n+1);

%% Plotting

if nargout == 0
    cax = newplot(cax);

    % Load and define topographic data
    load('topo.mat','topo','topomap1');

    % Rotate data to be consistent with the Earth-Centered-Earth-Fixed
    % coordinate conventions. X axis goes through the prime meridian.
    % http://en.wikipedia.org/wiki/Geodetic_system#Earth_Centred_Earth_Fixed_.
28ECEF_or_ECF.29_coordinates
    %
    % Note that if you plot orbit trajectories in the Earth-Centered-
    % Inertial, the orientation of the contintents will be misleading.
    topo2 = [topo(:,181:360) topo(:,1:180)];
```

```
    % Define surface settings
    props.FaceColor= 'texture';
    props.EdgeColor = 'none';
    props.FaceLighting = 'phong';
    props.Cdata = topo2;

    % Create the sphere with Earth topography and adjust colormap
    surface(x,y,z,props,'parent',cax,'HandleVisibility','off')
    colormap(topomap1)

% Replace the calls to surface and colormap with these lines if you do
% not want the Earth's topography displayed.
%     surf(x,y,z,'parent',cax)
%     shading flat
%     colormap gray

    % Refine figure
    axis equal
    xlabel(['X [' units ']'])
    ylabel(['Y [' units ']'])
    zlabel(['Z [' units ']'])
else
    xx = x; yy = y; zz = z;
end
```