

```

%% Kartesische X,Y,Z -> Ellipsoidische B,L,h
%
% @param <Double>X ..... X-Komponente
% @param <Double>Y ..... Y-Komponente
% @param <Double>Z ..... Z-Komponente
% @param <String>ell ... Name des Referenzellipsoid
%
% @return <Double>B .... Breite
% @return <Double>L .... Länge
% @return <Double>h .... Höhe
%
% @version 1.0 vom 30.10.2007 [XYZ2BLh.m]
% @author Michael Loesler - http://derletztekick.com

function [B L h] = XYZ2BLh(X, Y, Z, ell)
    if nargin==3
        ell = 'WGS84';
    elseif nargin<3
        error('Zu wenig Eingangsargumente!');
    end

    if strcmp(upper(ell), 'WGS84') || strcmp(upper(ell), 'GRS80')
        a = 6378137.0;
        alpha = 1.0/298.257222101;
    elseif strcmp(upper(ell), 'BESSEL')
        a = 6377397.155;
        alpha = 1.0/299.15281285;
    elseif strcmp(upper(ell), 'KRASSOWSKI')
        a = 6378245.0;
        alpha = 1.0/298.3;
    elseif strcmp(upper(ell), 'HAYFORD')
        a = 6378388.0;
        alpha = 1.0/297.0;
    else
        error('Ellipsoid unbekannt!')
    end

    roh = 180/pi;

    b = a*(1-alpha);
    c = a^2/b;

    e1 = sqrt((a^2-b^2)/a^2);
    e2 = sqrt((a^2-b^2)/b^2);

    p = sqrt(X^2+Y^2);

    Theta = atan((Z*a)/(p*b));

    L = atan(Y/X)*roh;
    B = atan((Z+ e2^2*b*(sin(Theta))^3) / (p-e1^2*a*(cos(Theta))^3))

    eta2 = e2^2*(cos(B))^2;
    V = sqrt(1 + eta2);
    N = c/V;

    h = p/cos(B) - N;
    B = B*roh;

```

```
return;
```

und

```
%% Kartesische X,Y,Z -> Ellipsoidische B,L,h
%
% @param <Double>B ..... Breite
% @param <Double>L ..... Länge
% @param <Double>h ..... Höhe
% @param <String>ell ... Name des Referenzellipsoid
%
% @return <Double>X .... X-Komponente
% @return <Double>Y .... Y-Komponente
% @return <Double>Z .... Z-Komponente
%
% @notice Die Eingabe der ellipsoidischen Breite und Länge kann sowohl
%          dezimaler Wert als auch als [°, ', "]-Matrix erfolgen.
%
% @version 1.0 vom 30.10.2007 [BLh2XYZ.m]
% @author Michael Loesler - http://derletztekick.com

function [X Y Z] = BLh2XYZ(B, L, h, ell)
    if nargin==3
        ell = 'WGS84';
    elseif nargin<3
        error('Zu wenig Eingangsargumente!');
    end

    if length(B)==3
        B = B(1) + B(2)/60 + B(3)/3600;
    end
    if length(L)==3
        L = L(1) + L(2)/60 + L(3)/3600;
    end

    if strcmp(upper(ell), 'WGS84') || strcmp(upper(ell), 'GRS80')
        a = 6378137.0;
        alpha = 1.0/298.257222101;
    elseif strcmp(upper(ell), 'BESSEL')
        a = 6377397.155;
        alpha = 1.0/299.15281285;
    elseif strcmp(upper(ell), 'KRASSOWSKI')
        a = 6378245.0;
        alpha = 1.0/298.3;
    elseif strcmp(upper(ell), 'HAYFORD')
        a = 6378388.0;
        alpha = 1.0/297.0;
    else
        error('Ellipsoid unbekannt!')
    end

    roh = pi/180;

    b = a*(1-alpha);
    c = a^2/b;

    e2 = sqrt((a^2-b^2)/b^2);

    B = B*roh;
```

```
L = L*roh;  
  
eta2 = e2^2*(cos(B))^2;  
V = sqrt(1 + eta2);  
N = c/V;  
  
X = (N+h)*cos(B)*cos(L);  
Y = (N+h)*cos(B)*sin(L);  
Z = ((b/a)^2*N+h)*sin(B);  
  
return;
```