Subject: Re: UTM conversion
Posted by Pavel A. Romashkin on Wed, 08 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

I think I have it. Is it it, below?
Cheers,
Pavel

> Luis Alonso wrote:
>
>> Hello!
>>
>> I need to change coordinates from Latitude/Longitude to UTM.
>> I know that with MAP one can proyect on either one, but i don't know how to
>> transform a vector from one into the other.
>>
>> Thanks
>> Luis Alonso
>
> Hello,
>
> The book by Snyder (referenced below) is very much worth the $40US. I have
> used the following code
> to convert Lon/Lat to UTM in meters.
>
> Ben
>
> --------SNIP-------
> ;+
> ; NAME:
> ; ll_to_utm
> ;
> ; PURPOSE:
> ; This function converts latitude/longitude measurements to UTM coordinates.
> ; Two map data are are available: NAD27 and WGS84.
> ;
> ; The formulae come from: J.P. Snyder, "Map projections - A working
> manual',1987,
> ; U.S.G.S. Professional Paper 1395, Supt. of Docs No: I 19.16:1395,
> ; U.S. Govt Prinitng Office,Washington, DC 20402.
> ;
> ; Constants used in formulae come from this source also unless otherwise
> noted.
> ;
> ;
> ; CATEGORY:
> ; Mapping.
> ;

```
>  ; CALLING SEQUENCE:
>  ; Result = ll_to_utm(lon,lat)
>  ;;
>  ; INPUTS:
>  ; LAT/LON   Identically sized vectors containning the decimal degrees of
>  latitude and longitude.
>  ; Degrees west and south are negative.
>  ;
>  ; OPTIONAL INPUTS:
>  ;  None.
>  ;
>  ; KEYWORD PARAMETERS:
>  ; DATUM Set this keyword to 1 for NAD27 (default) and to 2 for WGS84.
>  ; NOFALSE Set this keyword to prevent the return of False Easting. Default is
>  to
>  ;  return a False Easting (NoFalse = 0).
>  ;
>  ; OUTPUTS:
>  ; A 2,n elements double precision array where [0,*] = Easting
>  ;    and [1,*] = False Northing.
>  ;
>  ; OPTIONAL OUTPUTS:
>  ; None.
>  ;
>  ; COMMON BLOCKS:
>  ; None.
>  ;
>  ; SIDE EFFECTS:
>  ; None known.
>  ;
>  ; RESTRICTIONS:
>  ; None known.
>  ;
>  ; EXAMPLE:
>  ; Lat =double( [44.0551666000000000, 44.0551999,44.0552166])
>  ; Lon =double( [-69.5231832999999990, -69.5231999,-69.5232166])
>  ; UTM = ll_to_utm(Lon, Lat, Datum = 2)
>  ;  For i = 0,2 Do Print, Lon[i],UTM[0,i], Lat[i],UTM[1,i]
>  ;   -69.523186       458092.92       44.055168       4878133.2
>  ;       -69.523201       458091.72       44.055199       4878136.6
>  ;       -69.523216       458090.51       44.055218       4878138.8
>
>  ;
>  ; MODIFICATION HISTORY:
>  ;  Written by: Ben Tupper  Spring 1998
>  ; Pemaquid River Company
>  ; email: pemaquidriver@tidewater.net
>  ; tel:  (207) 563 - 1048
```

```
>  ;       248 Lower Round Pond Road
>  ; POB 106
>  ; Bristol, ME 04539-0106
>  ;
>  ;  SEP 1999 Cleaned up documentation and added NoFalse keyword.
>  ;  DEC 2, 1999  Reversed order of input and output arguments to fit familiar
>  x,y ordered pair sense.
>  ;  4 FEB 2000 Documentation change on 2 DEC 1999.
>  ;-
>
>  FUNCTION ll_to_utm, lonV,latV,datum=datum, NoFalse = NoFalse
>
>  On_Error,2
>
>  Num = n_elements(LatV)  & Utm = DblArr(2,Num)
>  Lon=Double(LonV)   & Lat=Double(LatV)
>
>  If n_elements(NoFalse) Eq 0 Then NoFalse = 0
>  If n_elements(Datum) EQ 0 Then Datum = 1
>  Data = ['Blank','NAD27','WGS84']
>
>  ;A   emimajor axis of the of ellipsoid earth
>  ;eccsq    eccentricity of ellipsoid (squared)
>  ;k0    relative scale factor along parallel of latitude
>  ;lon   longitude in +/-ddd.dddddd
>  ;lat  latitude in +/-dd.ddddddd
>  ;X  the rectangualar coordinate output...measured in meters from central
>  meridian
>  ;   this is converted to a false easting by adding half the distance between
>  ;   the bounding meridians
>  ;Y  the rectangular coordinate output.... measured in meters from the equator
>
>  Case Data[datum] of
>
>  'NAD27': BEGIN
>   ; based on Clarke 1866 (aka NAD27) or
>   ; assign constant values for NAD 27 to UTM
>   A = 6378206.4d
>   eccsq = 0.00676866d
>   k0 = 0.9996d
>
>      END
>
>  'WGS84':BEGIN
>   ; WGS84 (Aka NAD83)
>   ; from Chuck Gantz via http://gpsy.com/gpsinfo/geotoutm.htm
>   ; assign constant values for WGS 84 datum for lat/lon to UTM
>   A = 6378137.d
```

```
>   eccsq = 0.00669439d
>   k0 = 0.9996d
>
>      END
>
>  EndCase ;which datum
>
>   ;determine central meridian (lambda0 in Snyder)
>  central_meridian = double(long(lon/6)*6+3*long(lon/6.)/abs(long(lon/6.)))
>
>   ;determine lon origin in radians
>  lon_orig = central_meridian*!dtor
>
>   ; convert to radians
>  lon = lon*!dtor  &  lat = lat*!dtor
>
>   ;
>  eccsq_prime =eccsq/(1.-eccsq)
>
>  N = a/(1.-eccsq*sin(lat)*sin(lat))^.5
>
>  T = tan(lat)*tan(lat)
>
>  C = eccsq_prime * cos(lat)*cos(lat)
>
>  AA = cos(lat)*(lon-lon_orig)
>
>  M = a*((1-eccsq/4.-3.*eccsq^2/64.-5.*eccsq^3/256.)*lat - $
>   (3.*eccsq/8. + 3.*eccsq^2/32.+45.*eccsq^3/1024)*sin(2.*lat)+$
>   (15.*eccsq^2/256.+45.*eccsq^3/1024.)*sin(4.*lat)- $
>   (35.*eccsq^3/3072.)*sin(6.*lat))
>
>   ; k is unused for UTM... default to k0
>  ;k = k0*(1.d+ (1.d+C)*A^2/2.d + $
>  ; (5.d - 4.d*T + 42.d*C + 13.d*C^2 -28.d*eccsq_prime)*A^4/24.d + $
>  ; (61.d - 148.d*T + 16.d*T^2)*A^6/720.d)
>
>   ;  Easting or X in Snyder
>  UTM[0,*]=  k0*N*(AA+(1.-T-C)*AA^3/6.d $
>   + (5.d - 18.d*T + T^2 + 72.d*C - 58.d*eccsq_prime)*AA^5/120.d)
>
>   ; make False Easting if desired (default)
>  If NoFalse EQ 0 Then UTM[0,*] = UTM[0,*] + 500000.0d
>
>   ;Northing or Y in Snyder
>  UTM[1,*] =  k0*(M+N*tan(lat)*( AA^2 / 2.d + ( 5.d - T + 9.d *C + 4.d * C^2 ) *
>  $
>   $ AA^4/24.d + (61d - 58d*T + T^2 + 600d*C - 330d*eccsq_prime)*
```

> AA^6 / 720.d))
>
> Return,UTM
>
> end
>
> ------SNIP------
>
> --
> Ben Tupper
>
> Bigelow Laboratory for Ocean Science
> tupper@seadas.bigelow.org
>
> pemaquidriver@tidewater.net
