



Corso Luigi Einaudi, 55 - Torino

Appunti universitari

Tesi di laurea

Cartoleria e cancelleria

Stampa file e fotocopie

Print on demand

Rilegature

NUMERO : 287

DATA : 08/05/2012

A P P U N T I

STUDENTE : Sannipoli

MATERIA : Topografia I, esercitaz. svolte

Prof. Cina

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.

Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

**ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTIE NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.**

ESERCIZI DI GEODESIA

ESERCIZIO N. 1 - relazioni tra parametri geometrici ellissoidi

Svolgimento effettuato a mano.

Nota che:

$$\alpha = \frac{a-c}{a} \quad e^2 = \frac{a^2 - c^2}{a^2} \quad e'^2 = \frac{a^2 - c^2}{c^2}$$

Dopo alcuni semplici passaggi si ricavano le relazioni richieste.

$$e'^2 = \frac{e^2}{1 - e^2}$$

$$e^2 = \frac{e'^2}{1 + e'^2}$$

$$\alpha = 1 - \sqrt{1 - e^2}$$

$$e^2 = 2\alpha - \alpha^2$$

ESERCIZIO N. 2 - calcolo parametri geometrici ellissoidi

Svolgimento effettuato su excel.

ELLISSOIDE	a (m)	1/α	c (m)	e ²	e' ²
Delambre	6376985	308,6	6356320,758	0,006470381	0,006512519
Everest	6377276	300,8	6356074,949	0,006637884	0,00668224
Bessel	6377397	299,1528128	6356078,808	0,006674372	0,006719219
Fisher	6378338	288,5	6356229,376	0,006920394	0,00696862
Clarke	6378249	293,5	6356517,317	0,006802701	0,006849295
Helmert	6378140	298,3	6356758,371	0,006693422	0,006738525
Hayford	6378388	297	6356911,946	0,00672267	0,00676817
Krassovsky	6378245	298,3	6356863,019	0,006693422	0,006738525
WGS84	6378137	298,2572236	6356752,314	0,00669438	0,006739497

Dove:

$$c = a(1 - \alpha)$$

$$e'^2 = \frac{e^2}{1 - e^2}$$

$$e^2 = 2\alpha - \alpha^2$$

	<pre> %MAIN clc clear %parametri di volta in volta da cambiare %ellissoide=[6378137,0.006694379990]; %ellissoide WGS84 (semiasse, eccentricità) %ellissoide=[637397,0,006674372]; %ellissoide BESSELL (semiasse, eccentricità) ellissoide=[6378388,0.006722670022]; %ellissoide HAYFORD (semiasse, eccentricità) format long; %per aumentare il numero di cifre significative lat=[44,43,48]; % latitudine del punto in esame in gradi SESSAGESIMALI lat=(lat(1))+(lat(2)/60)+(lat(3)/3600); %trasformo in gradi SESSADECIMALI lat=lat/180*pi; %conversione in radianti effettuata nel main così da poterla usare in seguito %calcolo raggi principali e raggio sfera locale [raggi]= sfeloc(ellissoide,lat); disp(raggi); %calcolo differenza relativa diff=(raggi(2)-raggi(1))/raggi(2); disp(diff); %raggio curvatura del parallelo r=raggi(2)*cos(lat); disp(r); %raggio curvatura sezione normale di azimut 45° az=45; az=az/180*pi; %conversione in radianti ra=raggi(1)*raggi(2)/((raggi(2)*cos(az)^2)+(raggi(1)*sin(az)^2)); disp(ra); %costante di Clairaut di geodetica passante per P con azimut 45° c=r*sin(az); disp(c); </pre>
--	--

Di volta in volta si seleziona un ellissoide diverso, sfruttando i “commenti” (%).

ESERCIZIO N. 4 – da coordinate ECEF e geografiche

Svolgimento effettuato su Matlab.

Usiamo la funzione già vista a lezione.

xyz2flh (Da coordinate geocentriche ECEF a geografiche)

<p>Analisi dei dati di input</p>	<ul style="list-style-type: none"> • Vettore ellissoide 1x2 (semiasse a, eccentricità e^2) • X, Y, Z (ECEF)
<p>Analisi dei dati di output</p>	<ul style="list-style-type: none"> • Coordinate latitudine e longitudine (sessadecimali) e altezza ellissoidica (φ, λ, h)
<p>Algoritmo e diagramma di flusso</p>	$\lambda = \arctg \frac{Y}{X} \quad r = \sqrt{X^2 + Y^2}$ <div style="text-align: center;"> $\varphi = \arctg \frac{Z}{r} \text{ (valore approx)}$ $N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}}$ $h = \frac{X}{\cos \varphi \cos \lambda} - N$ $\varphi = \arctg \frac{Z}{r \left(1 - \frac{e^2 N}{N+h} \right)}$ </div> <div style="text-align: center;"> <p>end ← si $\left\{ \begin{array}{l} \varphi_n - \varphi_{n-1} < \varepsilon_\varphi \\ h_n - h_{n-1} < \varepsilon_h \end{array} \right.$ no</p> </div>
<p>Listato Matlab</p>	<pre>function [fi,lam,h]=xyz2flh(ellissoide,x,y,z) % trasformazione da geocentriche a geografiche % input: vettore ellissoide[a,e2] % coord x y z % output: fi lam h sessadecimali % sintassi: [fi,lam,h]=xyz2flh(ellissoide,x,y,z) % e2=ellissoide(2); a= ellissoide(1); lam=atan2(y,x); %restituisce l'arco tangente! ATTENZIONE occorre stare attenti se lam è negativo!! if lam<0 lam=lam+2*pi; %se lam è negativo aggiungo 2pi... end raggio=sqrt(x^2+y^2); fi=atan2(z,raggio); if fi<0 fi=fi+2*pi; end [raggi]=sfeloc(ellissoide, fi); %all'interno di una funzione posso usare un'altra funzione gran_n=raggi(2); for i=1:1:10 %impongo di fare il ciclo 10 volte perchè dall'esperienza so che dopo 10 iterazioni i valori calcolati si "stabilizzano"... h=x/(cos(fi)*cos(lam))-gran_n; fi=atan2(z,(raggio*(1-e2*gran_n/(gran_n+h)))); if fi<0 fi=fi+2*pi; end end</pre>

ESERCIZIO N. 5 – da coordinate geografiche e ECEF

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> • Vettore ellissoide 1x2 (semiasse a, eccentricità e^2) • Coordinate latitudine e longitudine (sessagesimali) e altezza ellissoidica (metri) (φ, λ, h)
Analisi dei dati di output	<ul style="list-style-type: none"> • Coordinate geocentriche (ECEF) (X,Y,Z).
Algoritmo e diagramma di flusso	$N = a/W$ $\begin{cases} X = (N+h) \cdot \cos \varphi \cdot \cos \lambda \\ Y = (N+h) \cdot \cos \varphi \cdot \sin \lambda \\ Z = [N \cdot (1-e^2) + h] \cdot \sin \varphi \end{cases}$
Listato Matlab	<pre>function [x,y,z]=flh2xyz(ellissoide,lat,lon,h) % trasformazione da geografiche a geocentriche % input: vettore ellissoide[a,e2] % coord lat lon h (sessagesimali) % output: x y z % sintassi: [x,y,z]=flh2xyz(ellissoide,lat,lon,h) % e2=ellissoide(2); a=ellissoide(1); w=sqrt(1-e2*sin(lat)^2); x=(a/w+h)*cos(lat)*cos(lon); y=(a/w+h)*cos(lat)*sin(lon); z=(a/w*(1-e2)+h)*sin(lat); %MAIN clc clear ellissoide=[6378137,0.006694379990]; %ellissoide WGS84 (semiasse, eccentricità) a=xlsread('esercizio 5.xlsx'); format short; for i=1:1:size(a,1) %ciclo che parte da 1, incrementa di 1, fino al numero di righe della nostra matrice nome=a(i,1); lat=(a(i,2)+a(i,3)/60+a(i,4)/3600)/180*pi; lon=(a(i,5)+a(i,6)/60+a(i,7)/3600)/180*pi; h=a(i,8); [x,y,z]=flh2xyz(ellissoide,lat,lon,h); %effettuo il cambio di coordinate xyz(i,1)=nome; %e salvo in una nuova matrice xyz xyz(i,2)=x; xyz(i,3)=y; xyz(i,4)=z; %la matrice xyz è una matrice di comodo!!! end %chiude ciclo for xlswrite('risultato_5.xlsx',xyz); save('risultato_5.txt','xyz','-ascii'); %in questo modo ho salvato sia in excel che in txt! disp(xyz)</pre>

ESERCIZIO N. 6 – trasformazione piana conforme

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> Coordinate punti nel sistema iniziale (X',Y') ed in quello finale (X,Y)
Analisi dei dati di output	<ul style="list-style-type: none"> Parametri della trasformazione piana conforme (Tx,Ty,α,λ) (la variazione di scala è isotropa)
Algoritmo e diagramma di flusso	Usare calcolo matriciale come indicato sul listato Matlab
Listato Matlab	<pre> %MAIN clc clear a=xlsread('input_6.xlsx'); A=[1 0 a(1,2) a(1,3) 0 1 a(1,3) -a(1,2) 1 0 a(2,2) a(2,3) 0 1 a(2,3) -a(2,2)]; l=[a(1,4) a(1,5) a(2,4) a(2,5)]; B=inv(A); X=B*l; disp(X(1,1)); disp(X(2,1)); alfa=atan(X(4,1)/X(3,1)); disp(alfa*180/pi); lambda=sqrt(X(4,1)^2+X(3,1)^2); disp(lambda); %seconda parte esercizio b=xlsread('i_6_bis.xlsx'); %punto 808 Y=[X(1,1) X(2,1)]; %Y è il vettore traslazione Z=[X(3,1) X(4,1) -X(4,1) X(3,1)]; %Z è la matrice di rotazione già moltiplicata per il fattore di scala isotropo %infatti X(3,1)=lambda*cos(alfa) e X(4,1)=lambda*sin(alfa) Y=Y+Z*[b(1,2) b(1,3)]; disp(Y); %punto 809 Y=[X(1,1) X(2,1)]; %Y è il vettore traslazione Z=[X(3,1) X(4,1) -X(4,1) X(3,1)]; %Z è la matrice di rotazione già moltiplicata per il fattore di scala isotropo %infatti X(3,1)=lambda*cos(alfa) e X(4,1)=lambda*sin(alfa) </pre>

ESERCIZIO N. 7 – trasformazione piana affine

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> • Coordinate punti nel sistema iniziale (X',Y') ed in quello finale (X,Y)
Analisi dei dati di output	<ul style="list-style-type: none"> • Parametri della trasformazione piana affine a 6 parametri (Xo,Yo,a,b,c,d)
Algoritmo e diagramma di flusso	Usare calcolo matriciale come indicato sul listato Matlab
Listato Matlab	<pre> %MAIN clc clear a=xlsread('input_7.xlsx'); A=[1 0 a(1,2) a(1,3) 0 0 0 1 0 0 a(1,2) a(1,3) 1 0 a(2,2) a(2,3) 0 0 0 1 0 0 a(2,2) a(2,3) 1 0 a(3,2) a(3,3) 0 0 0 1 0 0 a(3,2) a(3,3)]; l=[a(1,4) a(1,5) a(2,4) a(2,5) a(3,4) a(3,5)]; B=inv(A); X=B*l; disp(X); %seconda parte esercizio b=xlsread('i_7_bis.xlsx'); %punto 4 Y=[X(1,1) X(2,1)]; Z=[X(3,1) X(4,1) X(5,1) X(6,1)]; Y=Y+Z*[b(1,2) b(1,3)]; disp(Y); %punto 5 Y=[X(1,1) X(2,1)]; Z=[X(3,1) X(4,1) X(5,1) X(6,1)]; </pre>

ESERCIZIO N. 8 – trasformazione piana omografica

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> Coordinate punti nel sistema iniziale (X',Y') ed in quello finale (X,Y)
Analisi dei dati di output	<ul style="list-style-type: none"> Parametri della trasformazione piana omografica a 8 parametri (a,b,c,d,e,f,g,h)
Algoritmo e diagramma a di flusso	Usare calcolo matriciale come indicato sul listato Matlab
Listato Matlab	<pre> %MAIN clc clear a=xlsread('input_8.xlsx'); A=[a(1,2) a(1,3) 1 0 0 0 (a(1,4)*a(1,2)) (a(1,3)*a(1,4)) 0 0 0 a(1,2) a(1,3) 1 (a(1,5)*a(1,2)) (a(1,5)*a(1,3)) a(2,2) a(2,3) 1 0 0 0 (a(2,4)*a(2,2)) (a(2,3)*a(2,4)) 0 0 0 a(2,2) a(2,3) 1 (a(2,5)*a(2,2)) (a(2,5)*a(2,3)) a(3,2) a(3,3) 1 0 0 0 (a(3,4)*a(3,2)) (a(3,3)*a(3,4)) 0 0 0 a(3,2) a(3,3) 1 (a(3,5)*a(3,2)) (a(3,5)*a(3,3)) a(4,2) a(4,3) 1 0 0 0 (a(4,4)*a(4,2)) (a(4,3)*a(4,4)) 0 0 0 a(4,2) a(4,3) 1 (a(4,5)*a(4,2)) (a(4,5)*a(4,3))]; l=[a(1,4) a(1,5) a(2,4) a(2,5) a(3,4) a(3,5) a(4,4) a(4,5)]; B=inv(A); X=B*l; disp(X); %seconda parte esercizio b=xlsread('i_8_bis.xlsx'); %punto 5 X1=(X(1,1)*b(1,2)+X(2,1)*b(1,3)+X(3,1))/(X(7,1)*b(1,2)+X(8,1)*b(1,3)+1); Y1=(X(4,1)*b(1,2)+X(5,1)*b(1,3)+X(6,1))/(X(7,1)*b(1,2)+X(8,1)*b(1,3)+1); disp(X1); disp(Y1); %punto 6 X2=(X(1,1)*b(2,2)+X(2,1)*b(2,3)+X(3,1))/(X(7,1)*b(2,2)+X(8,1)*b(2,3)+1); Y2=(X(4,1)*b(2,2)+X(5,1)*b(2,3)+X(6,1))/(X(7,1)*b(2,2)+X(8,1)*b(2,3)+1); </pre>

ESERCIZIO N. 9 – coordinate euleriane e geodetiche rettangolari, da geodetiche polari

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> • Coordinate geodetiche polari (s,α) • Latitudine e longitudine origine della terna euleriana • Ellissoide Hayford e WGS84
Analisi dei dati di output	<ul style="list-style-type: none"> • Coordinate euleriane (x,y,z) • Coordinate geodetiche rettangolari (X,Y) e eccesso sferico (3ε)
Algoritmo e diagramma di flusso	<p>Da geodetiche polari ad euleriane Sviluppi di Puiseaux-Weingarten</p> <p>Da geodetiche polari a geodetiche rettangolari $X=s*\sin(\alpha-\epsilon)$ $Y=s*\cos(\alpha-2\epsilon)$ $3\epsilon=(s^2*\sin\alpha*\cos\alpha)/(2\rho N)$</p> <p>Trasformazioni inverse $s=[(X+\epsilon Y)^2+(Y-2\epsilon X)^2]^{1/2}$ $\alpha=\arctan[(X+\epsilon Y)/(Y-2\epsilon X)]$</p>
Listato Matlab	<pre>function [raggi]=sfeloc(ellissoide, lat) % sfeloc: calcola i raggi di curvatura di sez normali ellissoide e % sfera locale noti i parametri ellissoide e lat sessadecimali % input: vettore ellissoide[a,e2], numero latitudine (in radianti) % output: vettore raggi[ro, gran_n, r_sfera] % sintassi: [raggi]=sfeloc(ellissoide, latitudine) e2=ellissoide(2); a= ellissoide(1); w=sqrt(1-e2*sin(lat)^2); ro=a*(1-e2)/w^3; gran_n=a/w; r_sfera=sqrt(ro*gran_n); raggi=[ro,gran_n,r_sfera]; %funzione degli sviluppi di Puiseaux-Weingarten function [x,y,z]=saz2xyz(ellissoide,s,az,lat_origine,Ra,raggi); % saz2xyz: trasforma coordinate geodetiche polari in euleriane % input: vettore ellissoide[a,e2], s, azimut, latitudine origine, % longitudine origine % output: coordinate euleriane xyz % sintassi: [x,y,z]=saz2xyz(ellissoide,s,az,lat_origine,Ra); x=s*sin(az)*(1-((s^2)/(6*raggi(1)*raggi(2)))*(1- (ellissoide(2)*(sin(az)^2)*(cos(lat_origine)^2))/(1- ellissoide(2)*(sin(lat_origine)^2)))); y=s*cos(az)*(1- ((s^2)/(6*raggi(1)*raggi(2)))*(1+(ellissoide(2)*(cos(az)^2)*(cos(lat_o rigine)^2))/(1-ellissoide(2)))); z=-s*(s/(2*Ra)- (3*s^2*ellissoide(2)*(sin(lat_origine)^2)*cos(az))/(6*Ra^2*(1-</pre>

```

%seconda parte: geodetiche rettangolari
for i=1:1:size(a,1) %ciclo che parte da 1, incrementa di 1, fino al
numero di righe della nostra matrice
    nome=a(i,1);
    s=a(i,2);
    az=(a(i,3)+(a(i,4))/60+(a(i,5))/3600)/180*pi; %trasformo azimut in
radianti
    [x,y,e]=saz2xye(s,az,raggi); %effettuo il cambio di coordinate: da
geodetiche polari a geodetiche rettangolari
    xye(i,1)=nome; %e salvo in una nuova matrice xye
    xye(i,2)=x;
    xye(i,3)=y;
    xye(i,4)=e; %la matrice xye è una matrice di comodo che verrà
stampata!!!
end %chiude ciclo for
xlswrite('risultato_9_seconda parte.xlsx',xye);
save('risultato_9_seconda parte.txt','xye','-ascii'); %in questo modo
ho salvato sia in excel che in txt!
disp(xye)

%seconda parte bis: verifica risultati
b=xlsread('risultato_9_seconda parte.xlsx');
for i=1:1:size(b,1) %ciclo che parte da 1, incrementa di 1, fino al
numero di righe della nostra matrice
    nome=b(i,1);
    x=b(i,2);
    y=b(i,3);
    e=b(i,4);
    [s,az]=xye2saz(x,y,e); %effettuo il cambio di coordinate: da
geodetiche rettangolari a geodetiche polari
    saz(i,1)=nome; %e salvo in una nuova matrice saz
    saz(i,2)=s;
    saz(i,3)=az*180/pi; %la matrice saz è una matrice di comodo che
verrà stampata!!!
end %chiude ciclo for

xlswrite('risultato_9_seconda parte_bis.xlsx',saz);
save('risultato_9_seconda parte_bis.txt','saz','-ascii'); %in questo
modo ho salvato sia in excel che in txt!
disp(saz)
    
```

Risultato

Prima parte esercizio

WGS84			
PUNTO	x(m)	y(m)	z(m)
VIGONE	9652,593	7034,352	-11,178
MORETTA	12443,834	-1435,965	-12,281
SAVIGLIANO	22719,806	-14793,625	-57,585
SALUZZO	9107,760	-15228,412	-24,703
MONTE BRACCO	-2755,954	-11057,916	-10,197

ESERCIZIO N. 10 – coordinate euleriane da coordinate ECEF

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> • Coordinate geocentriche X,Y,Z • Coordinate geografiche ellissoidiche dell'origine P • Ellissoide WGS84
Analisi dei dati di output	<ul style="list-style-type: none"> • Coordinate euleriane x,y,z
Algoritmo e diagramma di flusso	ROTOTRASLAZIONE 3D: formule con matrice di rotazione
Listato Matlab	<pre>function Coord_euleriane=XYZ2xyz(lat_P,lon_P,X,Y,Z,Xo,Yo,Zo); % XYZ2xyz: trasforma coordinate ECEF in euleriane avente origine nel punto P % input: latitudine e longitudine di P, XYZ (ECEF) del punto da % convertire,coordinate ECEF del punto P origine % output: coordinate euleriane xyz % sintassi: Coord_euleriane=XYZ2xyz(lat_P,lon_P,X,Y,Z,Xo,Yo,Zo); %matrice di rotazione R=[-sin(lon_P) cos(lon_P) 0 -sin(lat_P)*cos(lon_P) -sin(lat_P)*sin(lon_P) cos(lat_P) cos(lon_P)*cos(lat_P) cos(lat_P)*sin(lon_P) sin(lon_P)]; traslazione=[X-Xo;Y-Yo;Z-Zo]; Coord_euleriane=R*traslazione; function Coord_ecef=xyz2XYZ(lat_P,lon_P,x,y,z,Xo,Yo,Zo); % xyz2XYZ: trasforma coordinate euleriane in ECEF % input: latitudine e longitudine di P, xyz (euleriane),coordinate ECEF del punto P origine % output: coordinate ECEF XYZ % sintassi: Coord_ecef=xyz2XYZ(lat_P,lon_P,x,y,z,Xo,Yo,Zo); %matrice di rotazione R=[-sin(lon_P) cos(lon_P) 0 -sin(lat_P)*cos(lon_P) -sin(lat_P)*sin(lon_P) cos(lat_P) cos(lon_P)*cos(lat_P) cos(lat_P)*sin(lon_P) sin(lon_P)]; %faccio la trasposta T=R'; origine_P=[Xo;Yo;Zo]; Coord_ecef=(T*[x;y;z]); Coord_ecef=Coord_ecef+origine_P;</pre>

```
comodo che verrà stampata!!!
end %chiude ciclo for
xlswrite('risultato_10_seconda parte.xlsx',XYZ);
save('risultato_10_seconda parte.txt','XYZ','-ascii'); %in
questo modo ho salvato sia in excel che in txt!
disp(XYZ)
```

Risultato

Prima parte esercizio

PUNTO	x(m)	y(m)	z(m)
1	0,001	-0,0001	191,924
2	7855,739	4013,768	-1460,949
3	-7326,501	-4189,853	1978,771
4	-21853,899	6135,772	-2085,985

Seconda parte esercizio (verifica)

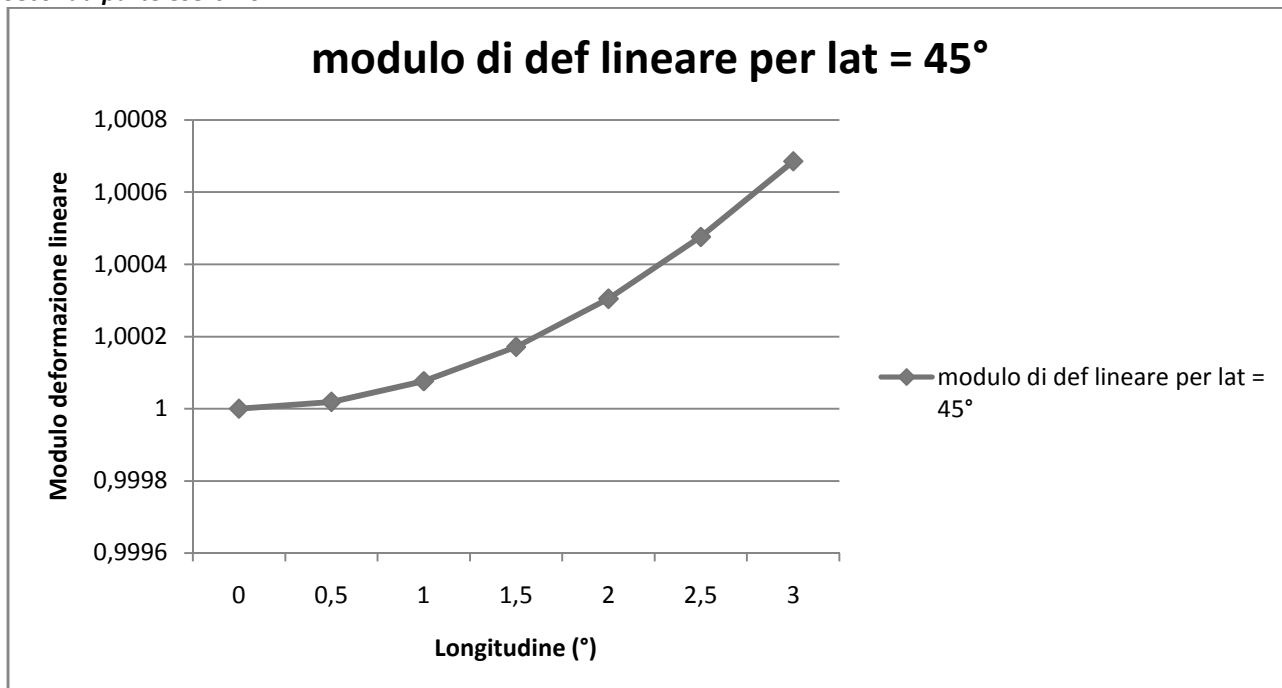
PUNTO	X (m)	Y (m)	Z (m)
1	4499525,427	585034,129	4467910,359
2	4495694,269	592457,860	4470744,778
3	4503484,717	578160,750	4465024,300
4	4498329,371	562840,765	4472537,612

Risultato

Prima parte esercizio

		LONGITUDINE IN GRADI SESSADECIMALI						
		0	0,5	1	1,5	2	2,5	3
LATTUDINE IN GRADI SESSADECIMALI	37	1	1,000024286	1,000097145	1,000218577	1,000388581	1,000607158	1,000874308
	38	1	1,000023644	1,000094578	1,0002128	1,000378311	1,000591111	1,0008512
	40	1	1,000022345	1,000089378	1,000201101	1,000357514	1,000558615	1,000804406
	41	1	1,000021688	1,000086753	1,000195194	1,000347012	1,000542206	1,000780777
	42	1	1,000021029	1,000084115	1,000189258	1,000336459	1,000525717	1,000757032
	43	1	1,000020367	1,000081467	1,0001833	1,000325866	1,000509166	1,0007332
	44	1	1,000019703	1,000078812	1,000177327	1,000315248	1,000492576	1,000709309
	45	1	1,000019039	1,000076154	1,000171347	1,000304617	1,000475965	1,000685389
	46	1	1,000018374	1,000073497	1,000165367	1,000293986	1,000459354	1,000661469
	47	1	1,000017711	1,000070842	1,000159395	1,000283368	1,000442763	1,000637579

Seconda parte esercizio



Risultato

		LONGITUDINE IN GRADI SESSADECIMALI							
		0	0,5	1	1,5	2	2,5	3	
LATITUDINE IN GRADI	SESSADECIMALI	37	1,00023028	1,000007162	0,999832596	0,999706585	0,999629126	0,999600221	0,999619869
		38	1,000213621	0,9999964	0,999826449	0,999703768	0,999628356	0,999600215	0,999619343
		40	1,000179888	0,999974608	0,999814	0,999698063	0,999626797	0,999600203	0,99961828
		41	1,000162854	0,999963604	0,999807714	0,999695182	0,99962601	0,999600197	0,999617743
		42	1,000145737	0,999952547	0,999801397	0,999692288	0,999625219	0,999600191	0,999617203
		43	1,000128556	0,999941448	0,999795057	0,999689382	0,999624425	0,999600185	0,999616662
		44	1,000111334	0,999930322	0,999788701	0,99968647	0,999623629	0,999600179	0,999616119
		45	1,00009409	0,999919183	0,999782337	0,999683554	0,999622833	0,999600173	0,999615575
		46	1,000076847	0,999908044	0,999775974	0,999680638	0,999622036	0,999600167	0,999615032
47	1,000059624	0,999896918	0,999769618	0,999677726	0,99962124	0,999600161	0,999614489		

```

%MAIN

clc
clear

ellissoide=[6378388,0.006722670022]; %ellissoide HAYFORD
(semiasse, eccentricità)
a=xlsread('input_13.xlsx');
lon_mc_e=15/180*pi;%longitudine del meridiano centrale del fuso
est
lon_mc_o=9/180*pi; %longitudine del meridiano centrale del fuso
ovest
lon_mm=(12+27/60+8.40/3600)/180*pi; %longitudine di Roma Monte
Mario

for i=1:1:size(a,1) %ciclo che parte da 1, incrementa di 1, fino
al numero di righe della nostra matrice

    nome=a(i,1);
    lat=(a(i,2)+a(i,3)/60+a(i,4)/3600)/180*pi;
    lon=(a(i,5)+a(i,6)/60+a(i,7)/3600)/180*pi;

    if(lon<0)
        %il primo punto ha lon < 0 quindi appartiene al fuso ovest
        lon_mc=lon_mc_o;
        falsa_origine=1500000;
    else
        %il secondo punto ha lon > 0 quindi appartiene al fuso est
        lon_mc=lon_mc_e;
        falsa_origine=2520000;
    end

    lon=lon+lon_mm-lon_mc; %longitudine RISPETTO al meridiano
centrale del fuso (che sul libro è lambda.mc)

    [x,y]=hirvonen(ellissoide,lat,lon);
    ESTNORD(i,1)=x+falsa_origine;
    ESTNORD(i,2)=y;

end %chiude il ciclo for esterno
xlswrite('risultato_13.xlsx',ESTNORD);
save('risultato_13.txt','ESTNORD','-ascii'); %in questo modo ho
salvato sia in excel che in txt!
disp(ESTNORD)

```

Risultato

Prima parte esercizio

PUNTO	EST (m) (x+falsa origine)	NORD (m) (y)
A	1500122,968	4814288,175
B	2535819,051	4070624,42


```

%
Alberto CINA ultima modifica 2 giugno 06

e2=ellissoide(2);
a= ellissoide(1);
mc=0.9996;
if est>1000000 % sistema Gauss Boaga - Roma40
  if est>2000000 % fuso est
    estzero=2520000;
    lamzero=2+32/60+51.60/3600;
  else % fuso ovest
    estzero=1500000;
    lamzero=-(3+27/60+8.40/3600);
  end
e2=0.006722670022; %ell. Hayford
nordzero=0;
a=6378388;
else % sistema WGS84 o UTM
  estzero=500000;
  nordzero=0; % emisfero nord
  if nord<0 % emisfero sud
    nordzero=10000;
  end
  lamzero=fuso*6+177;
  if lamzero>360
    lamzero=lamzero-360;
  end
end
x=(est-estzero)/mc;
y=(nord-nordzero)/mc;
c=a*sqrt(1-e2);
d=a^2/c;
eps2=e2/(1-e2);
a1=(1-e2/4-3*e2^2/64-5*e2^3/256);
teta=y/(a*a1);
e1=(1-sqrt(1-e2))/(1+sqrt(1-e2));
b1=3*e1/2-27*e1^3/32;
b2=21*e1^2/16-55*e1^4/32;
b3=151*e1^3/96;
b4=1097*e1^4/512;
csi=teta+b1*sin(2*teta)+b2*sin(4*teta)+b3*sin(6*teta)+b4*sin(8*teta);
v1=sqrt(1+eps2*(cos(csi))^2);
lam=atan(v1*sinh(x/d)/cos(csi));
fi=atan(tan(csi)*cos(v1*lam));
lam=lam/pi*180+lamzero;
fi=fi/pi*180;
    
```

Risultato

Seconda parte esercizio (verifica)

PUNTO	Latitudine (sessadecimali)	Longitudine (sessadecimali)
A	43,48072538	-4,549186111
B	36,78068608	2,724940798

USO LA FUNZIONE DEL PROF

```

function [fi,lam]=ne2ge(ellissoide,fuso,nord,est);
% trasformazione da cartografiche a geografiche - formule Hirvonen
% input: vettore ellissoide[a,e2]
% est,nord : coordinate cartografiche
%
% output: [fi, lam]: coordinate geografiche in sessadecimali
% N.B.1) è considerato il modulo di contrazione 0.9996
% N.B.2) gestisce le origini convenzionali UTM nord-sud e GaussBoaga
% N.B.3) nel caso Gauss Boaga l'ellissoide considerato è hayford,
anche se diversamente
% specificato in input e il fuso è automaticamente determinato (dare
un
% valore numerico arbitrario nella chiamata)
% sintassi: [fi,lam]=ne2ge(ellissoide,fuso,nord,est)
%
%           Alberto CINA ultima modifica 2 giugno 06

e2=ellissoide(2);
a= ellissoide(1);
mc=0.9996;
if est>1000000    %sistema Gauss Boaga - Roma40
    if est>2000000    %fuso est
        estzero=2520000;
        lamzero=2+32/60+51.60/3600;
    else            %fuso ovest
        estzero=1500000;
        lamzero=-(3+27/60+8.40/3600);
    end
    e2=0.006722670022;    %ell. Hayford
    nordzero=0;
    a=6378388;
else                %sistema WGS84 o UTM
    estzero=500000;
    nordzero=0;    %emisfero nord
    if nord<0    %emisfero sud
        nordzero=10000;
    end
    lamzero=fuso*6+177;
    if lamzero>360
        lamzero=lamzero-360;
    end
end
x=(est-estzero)/mc;
y=(nord-nordzero)/mc;
c=a*sqrt(1-e2);
d=a^2/c;
eps2=e2/(1-e2);
a1=(1-e2/4-3*e2^2/64-5*e2^3/256);
teta=y/(a*a1);
e1=(1-sqrt(1-e2))/(1+sqrt(1-e2));
b1=3*e1/2-27*e1^3/32;
b2=21*e1^2/16-55*e1^4/32;
b3=151*e1^3/96;
b4=1097*e1^4/512;
csi=teta+b1*sin(2*teta)+b2*sin(4*teta)+b3*sin(6*teta)+b4*sin(8*teta)
;
v1=sqrt(1+eps2*(cos(csi))^2);
lam=atan(v1*sinh(x/d)/cos(csi));
fi=atan(tan(csi)*cos(v1*lam));
lam=lam/pi*180+lamzero;
fi=fi/pi*180;

```

Risultato

Prima parte esercizio

PUNTO	φ (sessadecimali)	λ (sessadecimali)
NOVARA	45,43610046	8,61822802
BARDONECCHIA	45,07583953	6,709093291

Seconda parte esercizio (verifica)

PUNTO	EST(m)	NORD(m)
NOVARA	470127,7111	5033481,763
BARDONECCHIA	319607,7633	4995926,778

ESERCIZIO N. 16 – coordinate UTM da coordinate Gauss-Boaga

Svolgimento effettuato su Matlab.

Analisi dei dati di input	<ul style="list-style-type: none"> • Coordinate GAUSS-BOAGA • Foglio n. 140
Analisi dei dati di output	<ul style="list-style-type: none"> • Coordinate UTM • Fuso di appartenenza
Algoritmo e diagramma di flusso	Trasformazione con tabelle IGM
Listato Matlab	<pre> %MAIN clc clear a=xlsread('input_16.xlsx'); %coordinate nel sistema Gauss-Boaga est_GB=a(1,1); nord_GB=a(1,2); %calcolo delle coordinate UTM tramite tabelle di conversione IGM: 938 e 182 sono letti proprio da tali tabelle est_UTM = est_GB + 938; nord_UTM = nord_GB + 182; coordinate_UTM(1,1)=est_UTM; coordinate_UTM(1,2)=nord_UTM; xlswrite('risultato_16.xlsx',coordinate_UTM); save('risultato_16.txt','coordinate_UTM','-ascii'); %in questo modo ho salvato sia in excel che in txt! disp(coordinate_UTM); disp('IL FUSO è IL 33'); </pre>

Risultato

COORDINATE UTM	
EST (m)	NORD(m)
2415310	4717833

IL FUSO è IL 33

```

%MAIN

clc
clear

ellissoide=[6378388,0.006722670022]; %ellissoide HAYFORD (semiasse,
%eccentricità)
a=xlsread('input_17.xlsx');

for i=1:1:size(a,1)

    est(i)=a(i,2);
    x(i)=est(i)-1500000;
    nord(i)=a(i,3);
    quota(i)=a(i,4);
    lat(i)=(a(i,5)+a(i,6)/60+a(i,7)/3600)/180*pi;
    lon(i)=(a(i,8)+a(i,9)/60+a(i,10)/3600)/180*pi;

end %chiude il ciclo for esterno

%calcolo raggi del punto di latitudine media
raggim=sfeloc(ellissoide,(lat(1)+lat(2)/2));

%calcolo dist cartografica
dist_car=sqrt((est(1)-est(2))^2+(nord(1)-nord(2))^2);
disp(dist_car);
%calcolo modulo di def lineare
ml=0.9996*(1+(x(1)^2+x(1)*x(2)+x(2)^2)/(6*raggim(1)*raggim(2)*0.9996^2));
%calcolo dist sul geoide
dist_ge=dist_car/ml;
disp(dist_ge);

%calcolo dist orizzontale
dist_or1=dist_ge/(1-quota(1)/raggim(3)); %con piano passante per
primo punto
disp(dist_or1);
%calcolo dist orizzontale
dist_or2=dist_ge/(1-quota(2)/raggim(3)); %con piano passante per
secondo punto
disp(dist_or2);

%calcolo angoli di direzioni reciproci

%calcolo l'angolo di direzione 1-2
ang_dir_12=atan2((x(2)-x(1)),(nord(2)-nord(1)));
%discuto il valore restituito dalla funzione atan2
if ang_dir_12<0
    ang_dir_12=ang_dir_12+2*pi;
end

%calcolo l'angolo di direzione 2-1
if ang_dir_12<pi
    ang_dir_21=ang_dir_12+pi;
else
    ang_dir_21=ang_dir_12-pi;
end

```