HOMEWORK 2 Capobianco Salvatore 0124000974

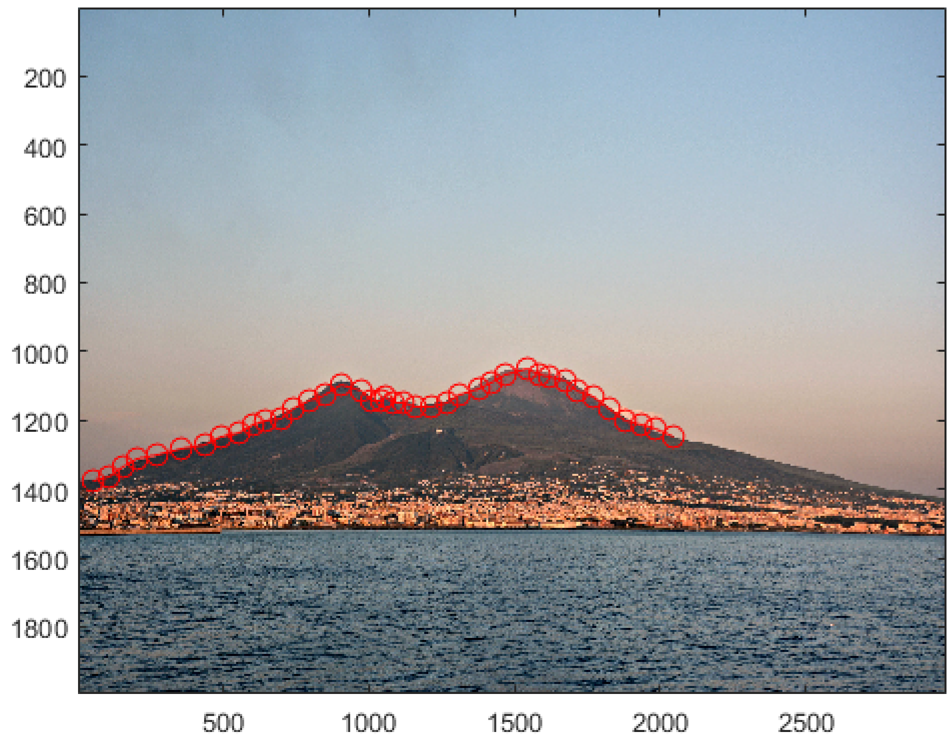
Table of Contents

PUNTO 1

Visualizzare la foto del Vesuvio e poi visualizzare la foto con i 40 punti evidenziati in rosso sul contorno. Per comodità, le componenti dei vettori x e y possono essere scalate, per esempio dividendole per 1000.

```
load 'mieidati';
image(A)
```

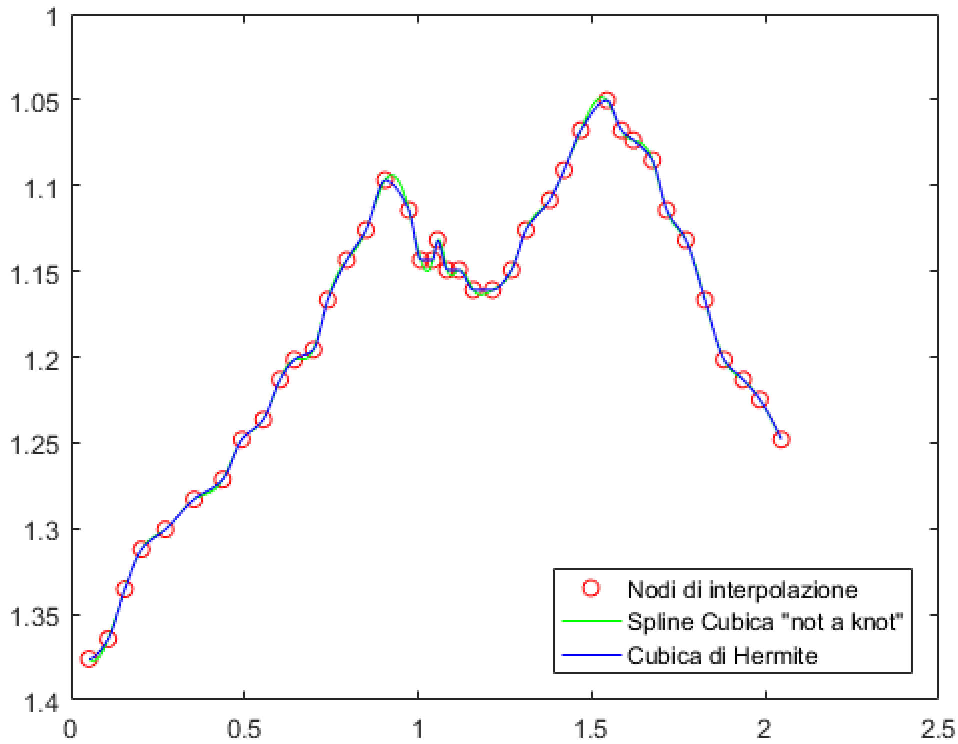
```
image(A)
hold on;
plot(x,y,'or',x,y,'r','MarkerSize', 8);
hold off;
x=x./1e3;
y=y./1e3;
```



PUNTO 2

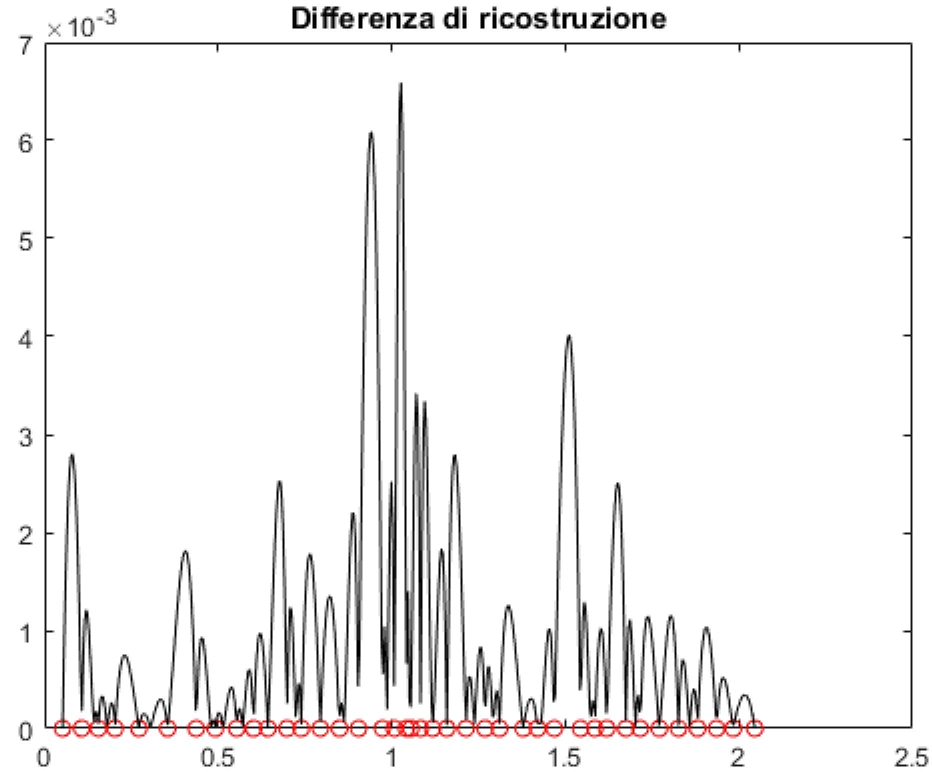
Interpolare i 40 punti sul contorno con una spline cubica e con una cubica di Hermite, visualizzando (su una griglia fitta di 500 punti) il Vesuviospline e il VesuviocubHerm. Commentare i risultati.

```
xx=linspace(x(1),x(end),500);
VesuvioSpline=spline(x,y,xx);
VesuvioCubHerm=pchip(x,y,xx);
plot(x,y,'ro',xx,VesuvioSpline,'g',xx,VesuvioCubHerm,'b')
set(gca, 'ydir' , 'reverse')
legend('Nodi di interpolazione', 'Spline Cubica "not a knot"', 'Cubica
 di Hermite','Location','southeast')
```



Le due funzioni interpolanti sono quasi identiche (i loro grafici si sovrappongono quasi per tutta la griglia); tuttavia ci sono delle differenze sostanziali che è possibile apprezzare in specifici sottointervalli, in particolare quelli in cui i **nodi cambiano valori più rapidamente** o, in altri termini, laddove ci sono dei **bruschi cambiamenti della pendenza**, ovvero delle **derivate**, delle funzioni interpolanti. Queste differenze sono dovute proprio alla natura delle Spline Cubiche e delle Cubiche di Hermite: le prime con derivate prima e seconda continue e derivata terza eventualmente discontinua, ma solo nei nodi, le seconde con derivata prima continua e derivata seconda eventualmente discontinua solo nei nodi. Quindi le Spline Cubiche, in generale, interpolano con maggiore accuratezza delle Cubiche di Hermite, poiché hanno un andamento "più liscio", seppure entrambe abbiano delle criticità intorno ai "picchi". Tuttavia le cubiche di Hermite trovano la loro ragione di esistere e l'impiego nella grafica computazionale, poichè interpolano *"a conservazione di forma"* **(shape preserving)**, cioè vengono costruite in modo tale (pendenze nei nodi determinate automaticamente) che non oscillino intorno ai nodi, evitando, cosi, l'overshooting, o, in altri termini, **conservano la monotonicità locale dei dati**.

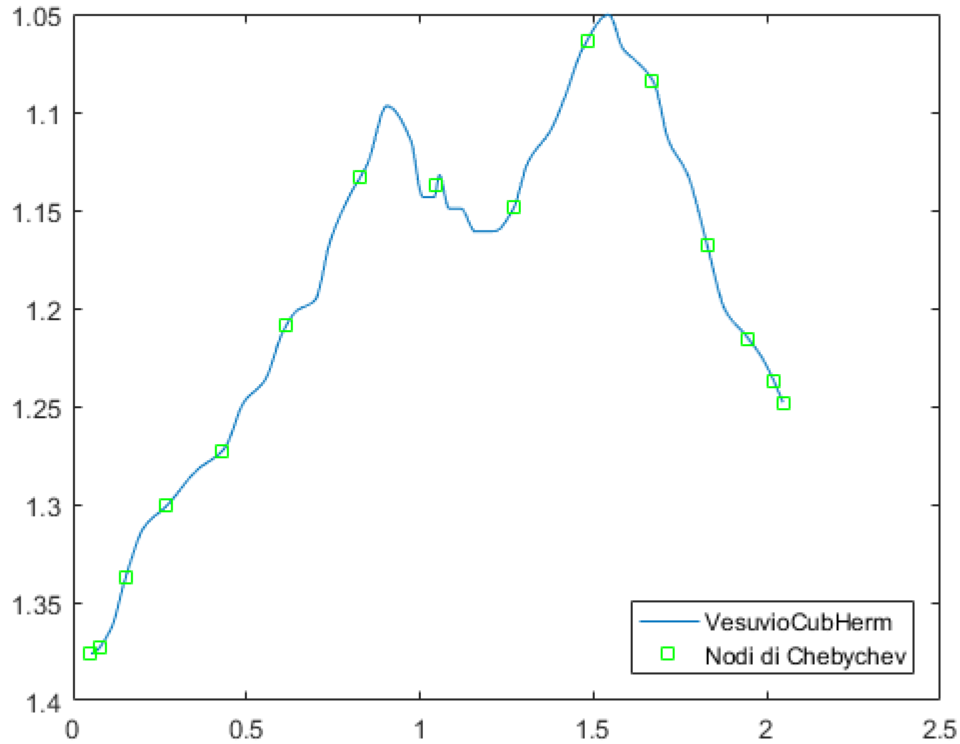
```
VesuvioDiff=abs(VesuvioCubHerm - VesuvioSpline);
plot(x,zeros(size(x)),'ro',xx,VesuvioDiff,'k')
title('Differenza di ricostruzione')
```



PUNTO 3

Utilizzando la cubica di Hermite del punto 2, generare le ordinate di punti sul contorno del VesuviocubHerm in corrispondenza di 15 nodi di Chebychev e visualizzare in verde tali punti.

```
for k=1:15
    xk(k)=-cos((k-1)*pi/14);
end
a=x(1);
b=x(end);
xchebab=(a+b)/2+xk.*((b-a)/2);
ychebab=pchip(x,y,xchebab);
plot(xx,VesuvioCubHerm,xchebab,ychebab,'sg')
set(gca, 'ydir' , 'reverse')
legend('VesuvioCubHerm','Nodi di Chebychev','Location','southeast')
```

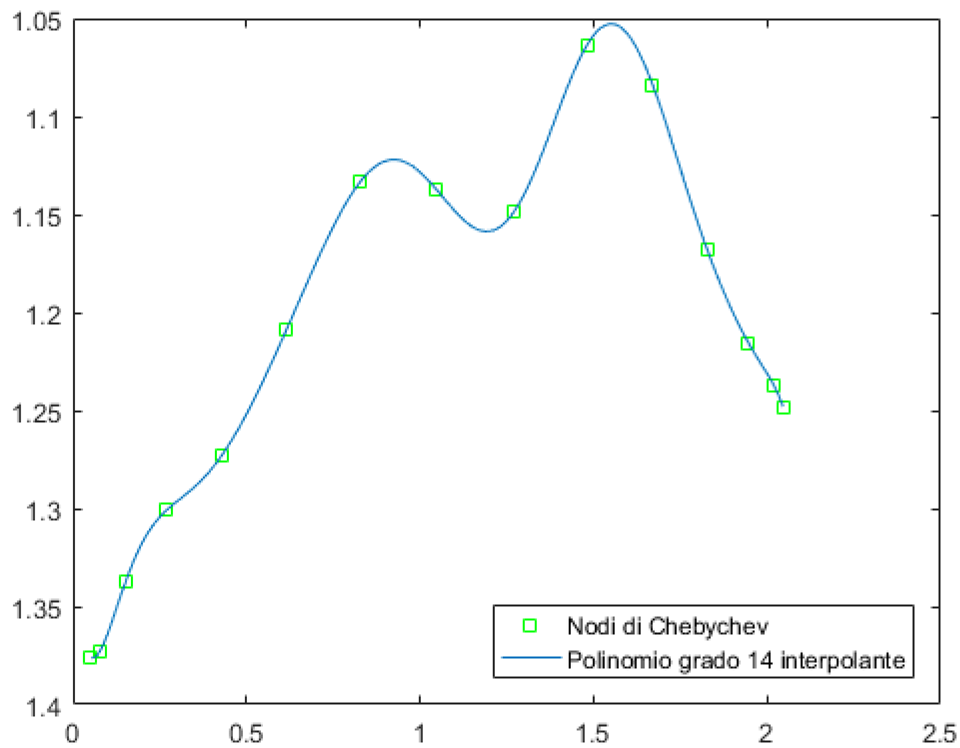


PUNTO 4

Interpolare i 15 punti ottenuti in 3 con un polinomio e visualizzare (su una griglia fitta di 500 punti) il suo grafico. Commentare il risultato.

```
P14=polyfit(xchebab, ychebab, 14);
y14=polyval(P14,xx);
plot(xchebab,ychebab,'sg',xx,y14);
set(gca, 'ydir' , 'reverse');
legend('Nodi di Chebychev','Polinomio grado 14
 interpolante','Location','southeast')
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X
 values,
reduce the degree of the polynomial, or try centering and scaling as
 described
in HELP POLYFIT.
```



Poichè su n punti, interpola (se esiste, cioè se la matrice dei coefficienti è non singolare) un unico polinomio di grado n-1, il polinomio di grado 14 interpolante sui 15 nodi di Chebychev **esiste ed è unico**. Si noti, come ci si attendeva, che i nodi di Chebychev sono più fitti, più densi in prossimità degli estremi dell'intervallo, al fine di evitare le **oscillazioni spurie** (localizzate agli estremi dell'intervallo e che non ricostruisco informazioni dei dati) che caratterizzano proprio l'interpolazione con polinomi su un numero relativamente elevato di nodi **equispaziati**, poichè per questo insieme di funzioni, al crescere del numero di nodi cresce anche il grado del polinomio.

PUNTO 5

Costruire esplicitamente la matrice B che descrive le condizioni di interpolazione sui 15 nodi di Chebychev e calcolare il suo indice di condizionamento. Commentare il valore di tale indice.

```
for j=1:15
    B(:,j)=xchebab.^(15-j);
end
B
cond(B)
```

```
B =

   1.0e+04 *

  Columns 1 through 3
```

```
 0.000000000000000   0.000000000000000   0.000000000000000
 0.000000000000000   0.000000000000000   0.000000000000000
 0.000000000000000   0.000000000000002   0.000000000000014
 0.000000000001070   0.000000000003970   0.000000000014726
 0.000000000679417   0.000000001589534   0.000000003718800
 0.000000114467180   0.000000185684306   0.000000301210020
 0.000007021165330   0.000008488061988   0.000010261429965
 0.000195435628161   0.000186302103163   0.000177595426021
 0.002866905252834   0.002255861266921   0.001775053448509
 0.024557843541125   0.016575336530861   0.011187536912645
 0.131911655329468   0.078959756603473   0.047263777770871
 0.466938160583169   0.255369727635411   0.139662386366403
 1.126980174294446   0.578753572610973   0.297215253160657
 1.896236079418395   0.938271797048382   0.464263903999980
 2.252324081362399   1.100851337746401   0.538054748801956

Columns 4 through 6

 0.000000000000000   0.000000000000000   0.000000000000000
 0.000000000000000   0.000000000000001   0.000000000000010
 0.000000000000092   0.000000000000608   0.000000000004032
 0.000000000054628   0.000000000202649   0.000000000751744
 0.000000008700334   0.000000020354903   0.000000047621396
 0.000000488611440   0.000000792606895   0.000001285736762
 0.000012405298770   0.000014997075270   0.000018130338561
 0.000169295648349   0.000161383754030   0.000153841615651
 0.001396723633349   0.001099029952926   0.000864785852110
 0.007551037165295   0.005096578694388   0.003439939947252
 0.028291180029753   0.016934551261562   0.010136693702025
 0.076381732267837   0.041773373461697   0.022845969560515
 0.152633021879818   0.078383727350534   0.040253469646959
 0.229721252664049   0.113667794267163   0.056243674904809
 0.262980933738964   0.128535193982040   0.062823170703317

Columns 7 through 9

 0.000000000000005   0.000000000000104   0.000000000001992
 0.000000000000124   0.000000000001614   0.000000000020944
 0.000000000026738   0.000000000177314   0.000000001175850
 0.000000002788659   0.000000010344774   0.000000038374841
 0.000000111412835   0.000000260656357   0.000000609819655
 0.000002085673278   0.000003383299872   0.000005488260383
 0.000021918218747   0.000026497481635   0.000032033466821
 0.000146651952971   0.000139798293323   0.000133264933880
 0.000680467868977   0.000535434893599   0.000421313831782
 0.002321790273490   0.001567094239066   0.001057711534996
 0.006067628106681   0.003631964417908   0.002174023407668
 0.012494521794811   0.006833287353706   0.003737143111608
 0.020671915885965   0.010615932244954   0.005451745162429
 0.027829791078401   0.013770388808666   0.006813691393070
 0.030705604083576   0.015007744938406   0.007335221529047

Columns 10 through 12
```

```
  0.000000000038265   0.000000000734927   0.000000014115074
  0.000000000271781   0.000000003526759   0.000000045764830
  0.000000007797582   0.000000051709215   0.000000342906691
  0.000000142354815   0.000000528077580   0.000001958949757
  0.000001426706089   0.000003337856116   0.000007809095048
  0.000008902847271   0.000014441860261   0.000023427036481
  0.000038726057467   0.000046816897321   0.000056598115534
  0.000127036905673   0.000121099939294   0.000115440432207
  0.000331516206682   0.000260857790564   0.000205259307166
  0.000713903263362   0.000481849589965   0.000325224773810
  0.001301328216153   0.000778949812676   0.000466264239211
  0.002043853553014   0.001117788969118   0.000611321773832
  0.002799709401894   0.001437773135302   0.000738359340865
  0.003371465471678   0.001668226335915   0.000825450870317
  0.003585180525190   0.001752301460468   0.000856459078360

 Columns 13 through 15

  0.000000271095377   0.000005206682028   0.000100000000000
  0.000000593865233   0.000007706265198   0.000100000000000
  0.000002273966032   0.000015079675170   0.000100000000000
  0.000007266894662   0.000026957178379   0.000100000000000
  0.000018269800537   0.000042743187219   0.000100000000000
  0.000038002447633   0.000061646125290   0.000100000000000
  0.000068422874334   0.000082718120345   0.000100000000000
  0.000110045417576   0.000104902534562   0.000100000000000
  0.000161510925502   0.000127086948780   0.000100000000000
  0.000219510726381   0.000148158943834   0.000100000000000
  0.000279096723857   0.000167061881905   0.000100000000000
  0.000334333511500   0.000182847890745   0.000100000000000
  0.000379179790509   0.000194725393955   0.000100000000000
  0.000408439265487   0.000202098803927   0.000100000000000
  0.000418605000026   0.000204598387097   0.000100000000000


ans =

     3.292452130001256e+11
```

Come già segnalato dal Matlab, con il corrispondente "warning" nella polyfit del punto 4, il problema di interpolazione con il polinomio di grado 14 sui 15 nodi di Chebychev è **mal condizionato**. Un problema risulta essere ben condizionato quando il suo indice di condizionamento è uguale o "vicino" a 1, poichè, in tal caso l'errore assoluto è maggiorato e minorato dal residuo:

```
r = y - B*x

r/cond(B) <= Ea <= r*cond(B)
```

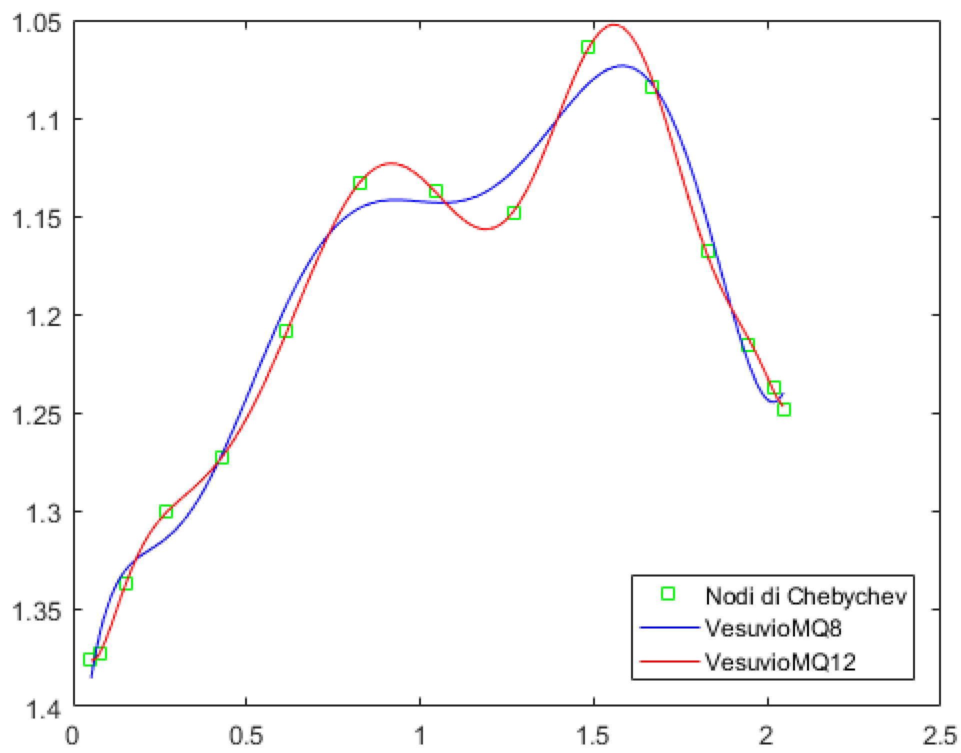
Al crescere dell'indice di condizionamento, la matrice dei coefficienti si avvicina all'essere singolare, come nel nostro caso, in cui `cond(B)` è molto maggiore di 1, pertanto si allarga la forbice di incertezza sulla soluzione.

PUNTO 6

Usando i 15 punti ottenuti in 3, determinare il polinomio di grado 8 e di grado 12 dei minimi quadrati e visualizzare (su una griglia fitta di 500 punti) il VesuvioMQ8 e il VesuvioMQ12. Commentare i risultati.

```
VesuvioMQ8=polyval(polyfit(xchebab,ychebab,8),xx);
VesuvioMQ12=polyval(polyfit(xchebab,ychebab,12),xx);
plot(xchebab,ychebab,'sg',xx,VesuvioMQ8,'b',xx,VesuvioMQ12,'r')
legend('Nodi di
 Chebychev','VesuvioMQ8','VesuvioMQ12','Location','southeast')
set(gca, 'ydir' , 'reverse');
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X
 values,
reduce the degree of the polynomial, or try centering and scaling as
 described
in HELP POLYFIT.
```



Il polinomio dei minimi quadrati di grado 12 è un migliore approssimante di quello di grado 8. In controtendenza all'interpolazione mediante polinomi in cui al crescere del numero dei nodi, cresce anche il grado del polinomio, e peggiora la qualità della ricostruzione del fenomeno continuo, l'approssimazione dei minimi quadrati migliora al tendere all'infinito del grado del polinomio, fino a diventare essa stessa una funzione interpolante quando il numero dei coefficienti raggiunge ed è uguale al numero dei nodi **(l'interpolazione è un caso limite di approssimazione)**.

PUNTO 7

Usando opportunamente la cubica di Hermite interpolante per ottenere i valori del contorno di VesuviocubHerm, determinare un'approssimazione dell'integrale definito del Vesuvio mediante il metodo Monte Carlo (con 1000 punti) e mediante trapezoidale composita con 100 punti. Commentare i risultati.

```
format long
xr=a+(b-a)*rand(1,1000);
QfMC=mean(pchip(x,y,xr))*b-a


QfMC =

   2.359948888912620


xtrap=linspace(a,b,100);
ytrap=pchip(x,y,xtrap);
h=(b-a)/99;
QfTC=h*(0.5*(ytrap(1)+ytrap(end))+sum(ytrap(2:end-1)))

DQf=abs(QfMC-QfTC)


QfTC =

   2.355967496508626


DQf =

   0.003981392403994
```

Come atteso, la differenza tra le due approssimazioni dell'integrale definito è relativamente grande nonostante **una venga applicata su 1000 punti mentre l'altra su 100 punti**. Questo perchè, per quanto riguarda il metodo Monte Carlo, **per ridurre l'errore di un fattore 10**, cioè per guadagnare una cifra significativa in base 10 corretta, **è necessario moltiplicare il numero dei punti per un fattore 100**. Ciò significa che le formule di quadratura Monte Carlo, benchè siano sempre convergenti, hanno una bassa velocità di convergenza; tuttavia risultano particolarmente convenienti nel caso di integrali definiti multi-dimensionali poichè l'errore di quadratura è indipendente dal numero delle dimensioni:

```
abs(If-QfMC) = alpha*(b-a)/sqrt(n)
```

dove n è il numero di punti

Mentre, per quanto riguarda la trapezoidale composita, **al raddoppiare del numero dei punti, l'errore di quadratura diminuisce di un fattore 4, e si annulla se f è lineare** (polinomio di primo grado quindi con derivata seconda nulla).

Th sull'errore di quadratura della formula composita trapezoidale su n nodi:

```
abs(If-QfTC) <= ((K*(b-a))/12)*h^2 = (K*(b-a)^3)/(12*n^2)
```

dove K è un maggiorante della derivata seconda.

PUNTO 8

Considerare i 500 valori di VesuviocubHerm sulla griglia fitta e determinare la media, la mediana, il percentile 25, il percentile 90, la deviazione standard e la varianza di tali dati. Visualizzare il box and wiskers plot di tali dati (comando boxplot).

La media è un indice di posizione della distribuzione dei dati, è compresa tra il più grande e il più piccolo dei dati del campione.

```
media=mean([xx,VesuvioCubHerm])
```

```
media =

   1.115431332931760
```

La mediana è un numero che è maggiore o uguale del 50% dei dati del campione e minore o uguale del restante 50%. Infatti è il parametro ordinale centrale se n è dispari, la media dei due parametri ordinali centrali se n è pari, dove n è il numero dei campioni.

```
mediana=median([xx,VesuvioCubHerm])
```

```
mediana =

   1.151050589326581
```

Il quantile-p con p in [0,1], è un numero che è maggiore o uguale del (100*p)% dei dati del campione e minore o uguale del restante (100*(1-p))%. Quindi **la mediana è il quantile-0.50**; Il quantile-p è anche detto (100*p) percentile. Il quantile-0.25 è detto **primo quartile** del campione:

```
perc25=prctile([xx,VesuvioCubHerm],25)
```

```
perc25 =

   1.048693091295683
```

```
perc90=prctile([xx,VesuvioCubHerm],90)
%calcola il quantile-0.90 (90-percentile), cioè relativo al 90% del
 campione.
```

```
perc90 =

   1.648399208555359
```

La deviazione standard è un indice delle distanze dei dati dalla media, che valuta la distanza nella stessa unità di misura dei dati; essa è uguale alla radice quadrata dello scarto quadratico medio (root mean square).

```
devstd=std([xx,VesuvioCubHerm])
```

```
devstd =

   0.417609384406821
```

La varianza è un indice delle distanze dei dati dalla media, che valuta la distanza come il quadrato dello scarto. E' interpretabile come una media delle distanze tra tutte le possibili coppie di dati. La varianza è il quadrato della deviazione standard.

```
varianza=var([xx,VesuvioCubHerm])
ctrlvarianza=devstd^2
```

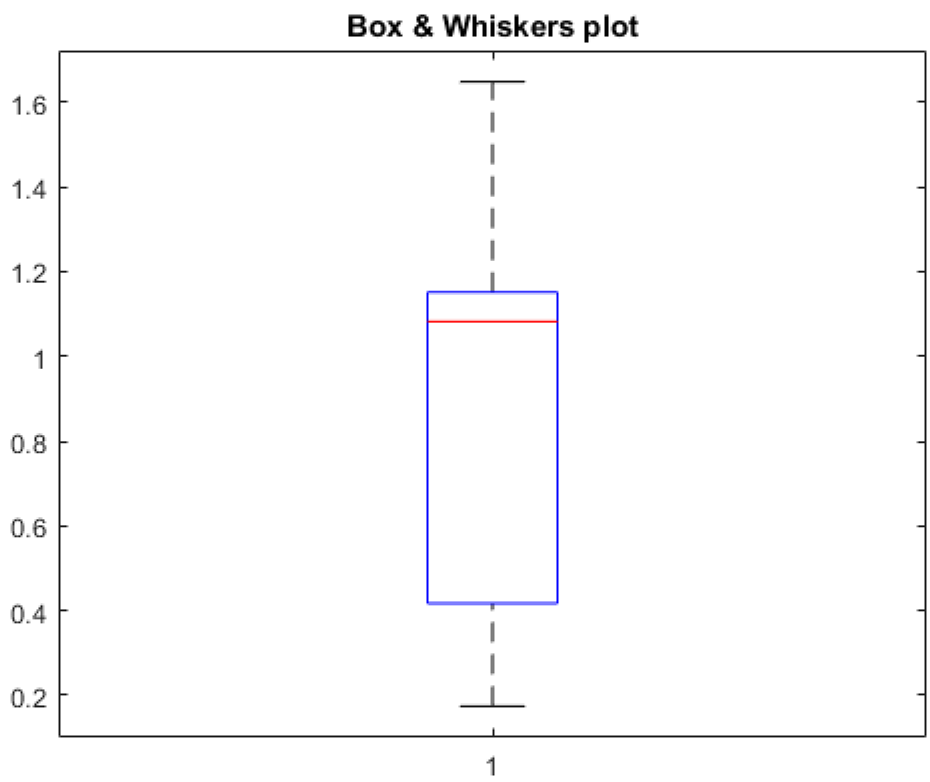
```
varianza =

   0.174397597944644


ctrlvarianza =

   0.174397597944644
```

```
boxplot([media,mediana,perc25,perc90,devstd,varianza])
title('Box & Whiskers plot')
```



Si noti che:

- Non ci sono outliers (dati anomali, perchè distanti dalle altre osservazioni, ragionevolmente ritenuti errati);
- La mediana è molto vicina al terzo quartile;
- L'area del sottobox Q25_mediana è molto più grande del sottobox mediana_Q75, ciò significa che la distribuzione dei dati è molto più concentrata tra il primo quartile e la mediana.

Published with MATLAB® R2016a